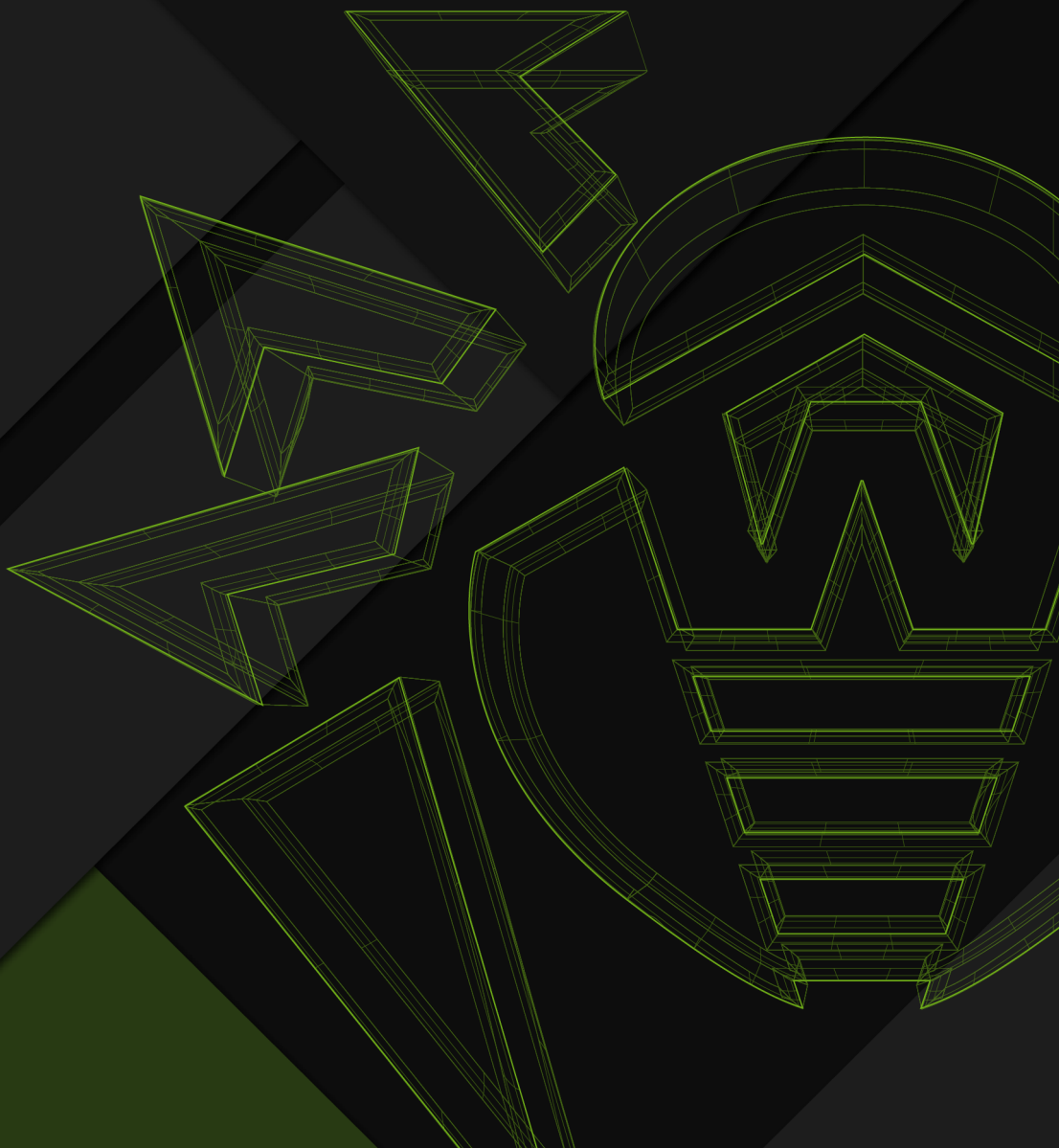
Dr.WEB

Исследование целевой атаки на российского оператора грузовых железнодорожных перевозок



«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.

Исследование целевой атаки на российского оператора грузовых железнодорожных перевозок
03.09.2024

ООО «Доктор Веб», Центральный офис в России
Адрес: 125124, Москва, ул. 3-я Ямского Поля, д. 2, корп. 12А

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

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

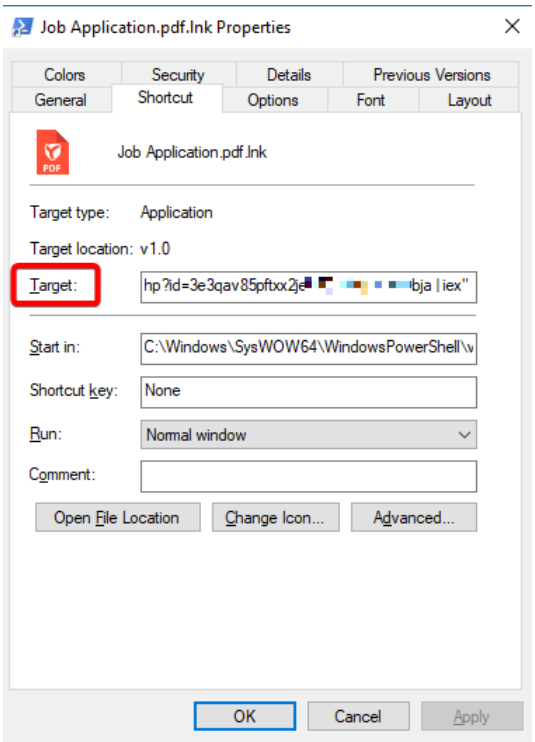
Введение

Целевой фишинг — популярный метод доставки вредоносного ПО на компьютеры сотрудников крупных компаний. От обычного фишинга он отличается тем, что злоумышленники заранее собирают информацию и персонализируют свое сообщение, побуждая жертву выполнить какое-то действие, которое приведёт к компрометации. Основными целями преступники выбирают или высокопоставленных сотрудников, обладающих доступом к ценной информации, или сотрудников тех отделов, которые по долгу службы контактируют с множеством адресатов. В частности, это касается работников отдела кадров: они получают массу писем от ранее незнакомых лиц с вложениями в самых разных форматах. Такой вектор атаки и был избран мошенниками при атаке на одну крупную транспортную компанию в марте 2024 года.

Общие сведения об атаке и используемые инструменты

В марте 2024 года в компанию «Доктор Веб» обратились представители крупного российского предприятия, работающего в отрасли грузовых железнодорожных перевозок. Внимание сотрудников отдела информационной безопасности привлекло подозрительное письмо с прикрепленным к нему вложением. После попытки самостоятельно определить, какую опасность несет приложенный файл, они обратились к нашим специалистам. Ознакомившись с полученным запросом, наши аналитики пришли к выводу, что компания чуть не стала жертвой целевой атаки. Задачами, которые ставили перед собой злоумышленники, был сбор информации о системе и запуск модульного вредоносного ПО на скомпрометированном ПК.

Для реализации атаки киберпреступники отправили на электронный адрес компании фишинговое письмо, замаскированное под резюме соискателя вакансии. К письму был приложен архив, якобы содержащий PDF-файл с анкетой. Этот файл имел «двойное» расширение `.pdf.lnk`. Сокрытие вредоносных объектов при использовании двойных расширений — довольно частая тактика злоумышленников, которой они пользуются для того, чтобы вводить своих жертв в заблуждение. По умолчанию ОС Windows скрывает расширения файлов для удобства пользователя. А если файл имеет «двойное» расширение, то система скрывает только последнее из них. Таким образом, в данном случае жертва могла видеть первое расширение — `.pdf`, а расширение `.lnk` было скрыто. Отметим, что даже при включенном отображении полных имен файлов, расширение `.lnk` всегда скрывается ОС.



Метаданные, хранящиеся в lnk файле

Истинное расширение `.lnk` является расширением ярлыков в ОС Windows. В поле «Объект» (Target) можно указать путь до любого объекта ОС — например, исполняемого файла — и запустить его с требуемыми параметрами. В рамках этой атаки скрытно происходил запуск интерпретатора команд PowerShell, скачивавший с сайта злоумышленников два вредоносных скрипта, каждый из которых запускал свою полезную нагрузку.

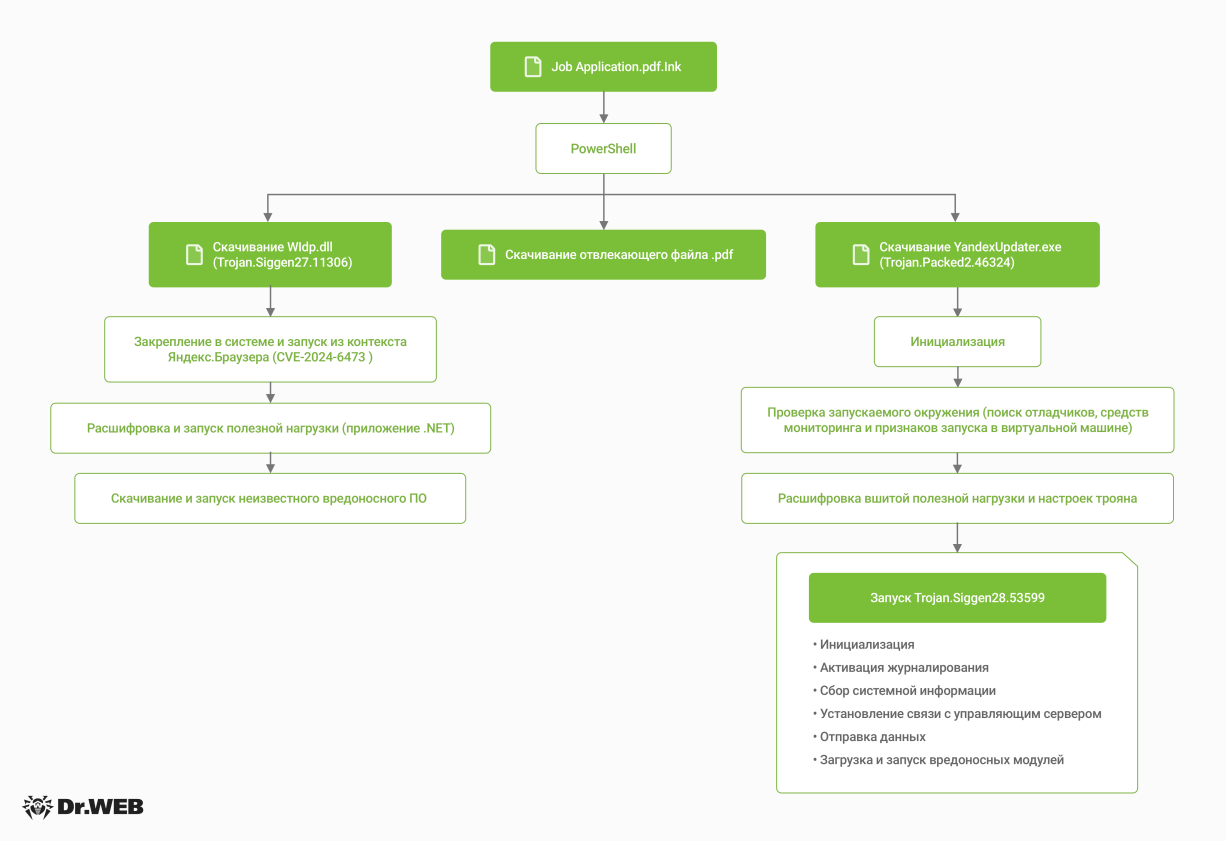
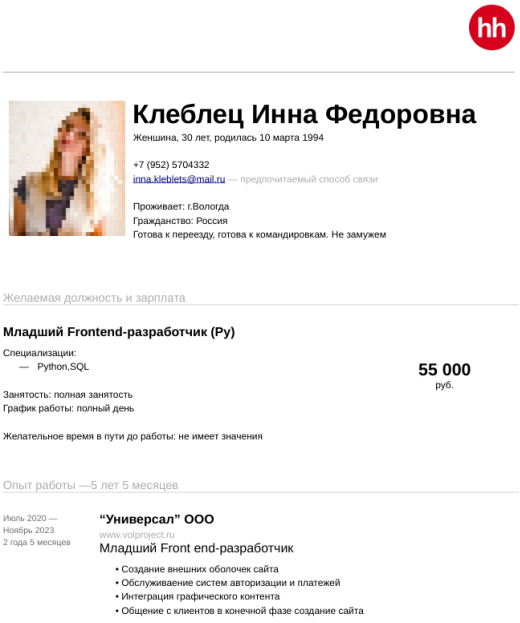


Схема атаки

Первая из них представляла собой отвлекающий PDF и исполняемый файл с названием `YandexUpdater.exe`, маскирующийся под компонент для обновления Яндекс Браузера (название реального компонента — `service_update.exe`). Данный исполняемый файл представляет собой дроппер трояна **Trojan.Packed2.46324**, который после ряда проверок, направленных на выявление факта запуска в эмулируемом окружении и наличия ПО для отладки, распаковывал в скомпрометированной системе трояна **Trojan.Siggen28.53599**. Последний имеет возможность удаленного управления, выполняет сбор системной информации и скачивание различных вредоносных модулей. Помимо данных функций троян также обладает возможностями по противодействию отладке. При выявлении процессов антивирусов, виртуальных машин и отладчиков троян перезаписывает свой файл нулями и удаляет его вместе с папкой, в которой он хранился.



Клеблец Инна Федоровна

Женщина, 30 лет, родилась 10 марта 1994

+7 (952) 5704332
inna.kleblets@mail.ru — предпочитаемый способ связи

Проживает: г.Вологда
Гражданство: Россия
Готова к переезду, готова к командировкам. Не замужем

Желаемая должность и зарплата

Младший Frontend-разработчик (Ру)

Специализации:
— Python,SQL

55 000
руб.

Занятость: полная занятость
График работы: полный день

Желательное время в пути до работы: не имеет значения

Опыт работы —5 лет 5 месяцев

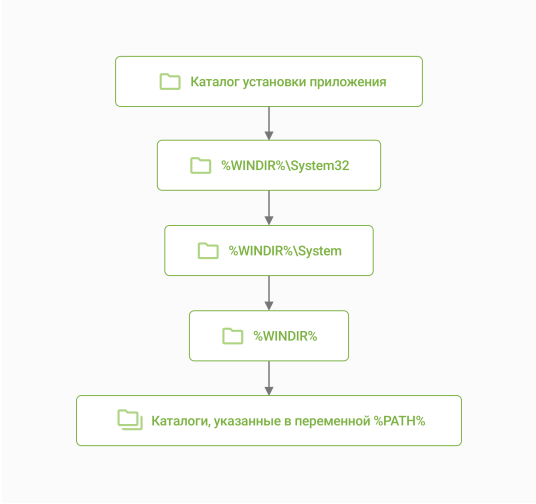
Июль 2020 —
Ноябрь 2023
2 года 5 месяцев

“Универсал” ООО
www.volproject.ru
Младший Front end-разработчик

- Создание внешних оболочек сайта
- Обслуживание систем авторизации и платежей
- Интеграция графического контента
- Общение с клиентов в конечной фазе создание сайта

Отвлекающий PDF-файл

Вторая полезная нагрузка состояла из отвлекающего PDF-файла и трояна **Trojan.Siggen27.11306**. Данный троян представляет собой динамическую библиотеку (DLL) с зашифрованной полезной нагрузкой. Особенность данного трояна заключается в том, что он эксплуатирует уязвимость Яндекс Браузера к перехвату порядка поиска DLL (DLL Search Order Hijacking). В ОС Windows DLL-файлы представляют собой библиотеки, которые используются приложениями для хранения функций, переменных и элементов интерфейса. В момент своего запуска приложения выполняют поиск библиотек в различных хранилищах данных в определенном порядке, поэтому злоумышленники могут попытаться «пролезть без очереди» и поместить вредоносную библиотеку в ту папку, где поиск DLL происходит с наибольшим приоритетом.



Упрощенная схема приоритета поиска библиотек

Данный троян сохраняется в скрытую папку `%LOCALAPPDATA%\Yandex\YandexBrowser\Application` под именем `Wldp.dll`. Именно в этот каталог устанавливается Яндекс Браузер и там же браузер будет искать необходимые ему библиотеки при запуске. В свою очередь, легитимная библиотека `Wldp.dll`, функция которой заключается в обеспечении безопасности запуска приложений, является системной библиотекой ОС и находится в папке `%WINDIR%\System32`. А так как вредоносная библиотека располагается в папке установки Яндекс Браузера, то первой будет загружаться именно она. При этом она получает все разрешения основного приложения: может выполнять команды и создавать процессы от имени браузера, а также наследовать правила брандмауэра для доступа в интернет.

После запуска браузера вредоносная библиотека `Wldp.dll` расшифровывает зашитую в нее полезную нагрузку. Следует отметить, что расшифровка выполняется дважды. В первый раз она производится с помощью ключа, создаваемого на основе хеша пути, по которому расположена вредоносная DLL, а затем с помощью глобального ключа, зашитого в тело трояна. Результатом расшифровки является шелл-код, выполнение которого позволяет злоумышленникам запустить на скомпрометированной системе приложение, написанное на языке .NET. В свою очередь этот стейджер загружал из сети вредоносное ПО. К сожалению, на момент нашего расследования на сервере, с которым связывался загрузчик, искомый файл не был доступен и нам не удалось узнать, какой конкретно троян скачивался в данном случае. Однако, учитывая, под каким именем должно было бы сохранено данное ПО, можно предположить, что этим трояном являлся тот же **Trojan.Packed2.46324**.

После обнаружения эксплуатации уязвимости в Яндекс Браузере мы передали информацию о ней в компанию Яндекс. Разработчики оперативно отреагировали на наше сообщение, в результате чего была выпущена версия Яндекс Браузера 24.7.1.380 с исправлением, а найденной уязвимости был присвоен идентификатор CVE-2024-6473.

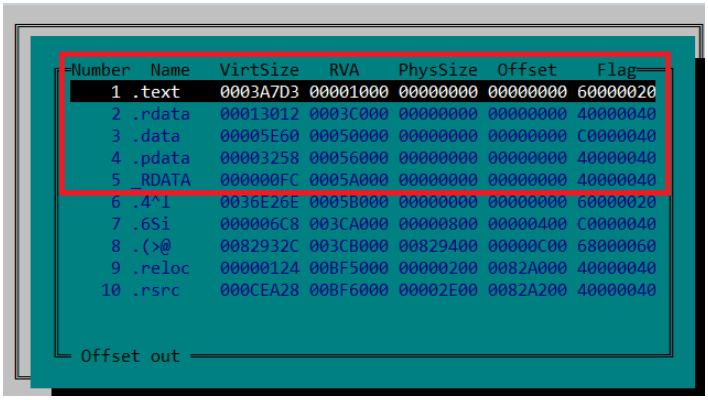
Принцип действия найденных образцов вредоносных программ

Trojan.Packed2.46324

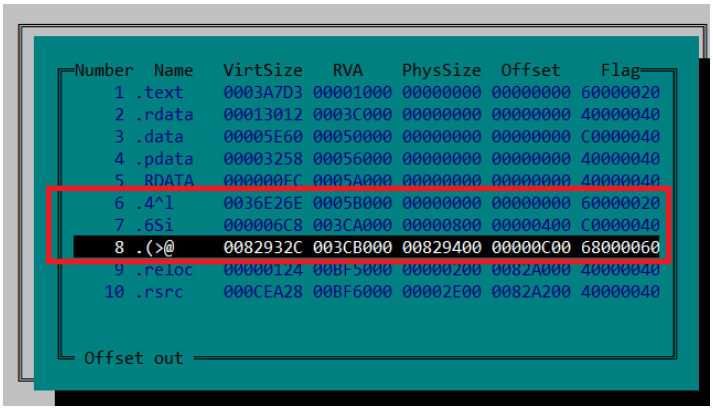
Вредоносная программа-дроппер для ОС Windows, написанная на языке C++. Исполняемый файл обфусцирован шифром XOR и запакован самописным упаковщиком. Применяется для доставки на скомпрометированные ПК трояна **Trojan.Siggen28.53599**.

Структура упаковщика и распаковка

Для упаковки трояна используется неизвестный упаковщик, характерной особенностью которого являются нулевые физические размеры секций с «традиционными» названиями `.text`, `.rdata` и др.



При этом, внутри одной из следующих секций с названием `.(>@` содержится код. Там же находится точка входа.



Исходный код с оригинальной точкой входа распаковывается в пустые секции.

Принцип действия

Инициализация

Троян читает структуру `KUSER_SHARED_DATA`, содержащую различную системную информацию. В самом начале в этой структуре проверяется значение поля `NtMajorVersion`, которое должно быть равно `10`. Далее загружаются библиотеки `ntdll.dll`, `kernel32.dll`, `user32.dll`, ole32.dll, `wevtapi.dll` и инициализируется структура с указателями на них. В будущем данную структуру трояном использует для вызова WinAPI функций, необходимых ему для работы. После происходит инициализация 3 потоков, отвечающих за антиотладку.

Работа с WinAPI

Троян опосредованно работает с системными функциями WinAPI через структуру-обертку, содержащую таблицу с функциями, указатели на библиотеки, адреса загрузки библиотек и флаг, отвечающий за противодействие отладке.

Таблица функций содержит в себе:

- функции для работы с WinAPI — нахождение указателя на функцию и ее вызова
- вспомогательные функции — собственная реализация `LoadLibrary` и `GetProcAddress`
- конфигурацию входных параметров для ряда функций

В самом начале выполнения троян инициализирует основную структуру приложения. Для этого он с помощью модифицированного алгоритма CRC32 находит в структуре `PEB_LDR_DATA` адреса загрузки библиотек. Для получения функций из библиотек троян использует два способа:

- Собственная реализация вызовов `LoadLibrary` и `GetProcAddress`

Троян имеет две функции, которые повторяют реализацию `LoadLibrary` и `GetProcAddress`. Данный способ применяется в тех случаях, когда требуется доступ к API, содержащемуся в ещё незагруженной в память процесса библиотеке.

- Поиск библиотек по хешам в `PEB_LDR_DATA`

Троян ищет нужную библиотеку в структуре `PEB_LDR_DATA` посредством списка `InMemoryOrderModuleList`, который содержит в том числе указатели на все загруженные в память процесса библиотеки и их имена. При этом совпадение имени библиотеки вычисляется по совпадению значения хеша модифицированного алгоритма CRC32 с искомым. Далее через таблицу экспортируемых функций библиотеки находится нужная, при этом имена функций хешируются аналогичным способом. Имена библиотеки и функции считываются с помощью модифицированного алгоритма CRC32.

Защита от отладки

Проверка отладочных регистров

Троян получает контекст родительского потока и проверяет, чтобы значение отладочных регистров `Dr0–Dr7` было равно `0`.

Проверка подключенного отладчика

В структуре `KUSER_SHARED_DATA` троян проверяет первые два бита в поле `KdDebuggerEnabled`, их значение должно быть равно `0`.

С помощью функции `NtQueryInformationProcess` проверяет наличие отладчика, запрашивая следующие параметры из структуры `PROCESSINFOCLASS`: `ProcessDebugFlags`, `ProcessDebugPort`, `ProcessDebugObjectHandle`, `ProcessTlsInformation`.

Поиск драйверов отладчиков

Выполняет в директории `%WINDIR%\System32\drivers` поиск на предмет выявления файлов отладочного ПО. Считает хеши имен файлов с помощью модифицированного алгоритма CRC32 и сравнивает результат с хешами из черного списка.

Каждая проверка осуществляется по таймеру, при неуспешной проверке в глобальной переменной выставляется флаг, который проверяется на различных этапах исполнения. При неудачной проверке флага троян завершает свою работу.

Проверка окружения

Для защиты от запуска в виртуальной среде троян просматривает ряд журналов ОС, используя библиотеку `wevtapi.dll`.

Поиск следующих строк в журналах `Microsoft-Windows-Shell-Core/Operational`, `System`, `Application`:

- `\npcap.sys`
- `Wireshark`
- `API_Monitor`
- `apimonitor`
- `API Monitor`
- `rohitab.com`
- `hex-rays.com`
- `processhacker.sys`
- `ProcessHacker`

- PROCMON2
- ida64.exe

В журналах Microsoft-Windows-Storage-Storport/Operational, System, Microsoft-Windows-Storsvc/Diagnostic, Microsoft-Windows-StorageSpaces-Driver/Operational, Microsoft-Windows-Partition/Diagnostic, Microsoft-Windows-Kernel-PnP/Configuration, Application производится поиск следующих строк:

- VMTools
- VMUpgradeHelper
- VirtualBox Guest
- VBoxService.exe
- VBOX HARDDISK
- _FLOPPY_
- \VMWVM
- _VBOX&
- NECVMWar
- prl_
- VMware

Дополнительно в журнале Application троян ищет следующие строки:

- VMware Player
- VMware NAT Service
- \Device\VBoxNet
- Oracle VM VirtualBox

После описанных проверок выполняется распаковка содержащейся в ресурсах полезной нагрузки — **Trojan.Siggen28.53599**. Для этого используется модифицированный алгоритм RC4, ключ для шифра имеет длину 8 байт. Затем происходит расшифровка конфигурации, зашифрованной шифром XOR. Полученная конфигурация сохраняется в строку, помеченную символами DANTEMARKER, которые перезаписываются.

Завершив все подготовительные операции, дроппер загружает в память полезную нагрузку и передает ей управление.

Trojan.Siggen28.53599

Вредоносная программа для ОС Windows, написанная на C++. Основная функциональность трояна заключается в загрузке и управлении модулями, получаемыми от C2-сервера.

Принцип действия

Троян имеет ряд основных и вспомогательных структур, которые инициализируются при запуске, и сохраняются как указатели в глобальных переменных.

Основные структуры:

Работа с WinAPI

Троян опосредованно работает с системными функциями WinAPI через структуру-обертку, содержащую таблицу с функциями, указатели на библиотеки, адреса загрузки библиотек и флаг, отвечающий за противодействие отладке.

Таблица функций содержит в себе:

- функции для работы с WinAPI — нахождение указателя на функцию и ее вызова,
- вспомогательные функции — собственная реализация `LoadLibrary` и `GetProcAddress`,
- конфигурацию входных параметров для ряда функций.

В самом начале выполнения троян инициализирует основную структуру приложения. Для этого он с помощью модифицированного алгоритма CRC32 находит в структуре `PEB_LDR_DATA` адреса загрузки библиотек. Для получения функций из библиотек троян использует два способа:

- Собственная реализация вызовов `LoadLibrary` и `GetProcAddress`

Троян имеет две функции, которые повторяют реализацию `LoadLibrary` и `GetProcAddress`. Данный способ применяется в тех случаях, когда требуется доступ к API, содержащемуся в ещё незагруженной в память процесса библиотеке.

- Поиск библиотек по хешам в `PEB_LDR_DATA`

Троян ищет нужную библиотеку в структуре `PEB_LDR_DATA` посредством списка `InMemoryOrderModuleList`, который содержит в том числе указатели на все загруженные в память процесса библиотеки и их имена. При этом совпадение имени библиотеки вычисляется методом сравнения значения хеша модифицированного алгоритма CRC32 с искомым. Далее через таблицу экспортируемых функций библиотеки находится нужная, при этом имена функций хешируются аналогичным способом.

Имена библиотеки и функции считываются с помощью модифицированного алгоритма CRC32.

Структура для журналирования событий

Представляет собой структуру, основное назначение которой — формирование журнала приложения. Журнал содержит как сведения об ошибках, так и сведения о текущем выполняемом этапе.

Структура для сбора системной информации

Основное назначение данной структуры — сбор системной информации для последующей отправки на C2-сервер.

Структура для общения с C2-сервером

Данная структура обеспечивает канал связи для взаимодействия с управляющим сервером. Содержит структуру для работы с библиотекой `winhttp.dll` и сведения об управляющем сервере: порт, IP-адрес и таблицу маршрутизации.

Структура для работы с модулями и конфигурациями

Основная функция данной структуры заключается в управлении работой модулей и их конфигураций. Содержит векторы, описывающие модули, их конфигурацию, а также вспомогательную системную информацию.

Структура для управления приложением

Основное назначение данной структуры — управление работой программы и централизация остальных структур. Содержит в себе указатели на структуры для работы с WinAPI, журналированием, взаимодействием с управляющим сервером, а также работы с модулями и конфигурациями.

Вспомогательные структуры

Структуры для работы с криптографией: SHA-1, SHA-256

Структуры для работы со вспомогательными библиотеками: `bcrypt.dll`, `winhttp.dll`

Структуры, хранящие в себе различные флаги

Защита от отладки

Также в самом начале инициализируются 3 потока, отвечающие за противодействие отладке:

Проверка отладочных регистров

Троян получает контекст родительского потока и проверяет, чтобы значения отладочных регистров `Dr0–Dr7` были равны 0.

Проверка подключенного отладчика

В структуре `KUSER_SHARED_DATA` троян проверяет первые два бита в поле `KdDebuggerEnabled`, значение которых должно быть равно `0`.

С помощью функции `NtQueryInformationProcess` троян проверяет наличие отладчика, запрашивая разные параметры структуры `PROCESSINFOCLASS`: `ProcessDebugFlags`, `ProcessDebugPort`, `ProcessDebugObjectHandle`, `ProcessTlsInformation`.

Поиск драйверов отладчиков

Выполняет поиск в директории `%WINDIR%\System32\drivers` на предмет выявления файлов отладочного ПО. Считает хеши имен файлов с помощью модифицированного алгоритма CRC32 и сравнивает результат с хешами из черного списка.

Проверка уникальности трояна в системе

После инициализации троян пытается создать мьютекс, которым является закодированный с помощью Base64 SHA-1 хеш значения строки `MachineGuid`. При неудачной попытке захватить мьютекс в журнал пишется строчка “`Found another agent running. Exiting...`” и выполнение программы завершается.

Проверка ключей и создание хэндшейка

Троян выполняет проверку наличия уже созданного хэндшейка. Она осуществляется путем получения доступа к ключу с помощью функции `NCryptOpenKey`, при этом поставщиком хранилища ключей CNG является "`Microsoft Software Key Storage Provider`", а именем ключа — SHA-256 хеш от значения `MachineGuid`. Если такой ключ отсутствует, то проверяется наличие интернет-соединения: если оно установлено, то начинается создание хэндшейка:

- Создается копия внутреннего ключа RSA из хранилища ключей CNG "`Microsoft Software Key Storage Provider`",
- Происходит получение ключа от сервера,
- Ключ сохраняется в хранилище с именем, соответствующим SHA-256 хешу значения `MachineGuid`.

Из пришедшего пакета троян получает новые адрес сервера и номер порта. После создания хендшейка собирается системная информация: архитектура процессора,

название ОС, тип пользовательского интерфейса, идентификаторы установленных приложений, сведения о диске, имена пользователей и региональные настройки.

Основные функции

Троян выполняет следующие основные функции:

- загрузка и выгрузка модулей
- формирование сообщений для C2-сервера о результатах работы или ошибках
- изменение конфигурации и настройка модулей
- обновление тела трояна при необходимости

Структура модулей и конфигурация

Модуль представляет собой динамическую библиотеку, которая проецируется в память и обладает следующими экспортируемыми функциями:

- `Start`
- `Stop`
- `Configure`
- `GetID`
- `GetStatus`
- `SetStatus`
- `GetStarted`
- `GetHandler`
- `Destroy`
- `PushErrorCMR`

Идентификатор модуля обозначает его функцию — то есть, зная идентификатор, можно определить приходящие от сервера задачи.

Идентификатор	**Функция**
238	Inject
27	Назначение неизвестно
44	Назначение неизвестно

JSON с конфигурацией модуля

```
{
  "triggers": [
    {
      "schedule": "<str_value>",
      "process": "<str_value>",
      "repetitions": "<int_value>",
      "sendCmr": {
        "name": "<str_value>",
        "interval": "<int_value>"
      }
    }
  ]
}
```

Результат выполнения модуля

После выполнения задачи модуля формируется ответ

```
{
  "CommandModuleResponse": "<str_value>",
  "requestId": "<int_value>",
  "moduleId": "<int_value>",
  "exitCode": "<int_value>",
  "info": "Error" //если в работе модуля произошла ошибка, иначе данное поле
отсутствует
}
```

Обновление трояна

Во время работы троян проверяет наличие флага обновления своего основного тела. Если данный флаг установлен, троян выполняет ряд системных проверок по результатам которых выбирается одна из двух стратегий обновления. При выявлении установленного на скомпрометированном ПК антивирусного ПО обновление происходит через загрузку шелл-кодов, в противном случае — с помощью модуля Inject.

Проверка наличия антивирусного ПО

В теле трояна зашит список хешей имен антивирусных программ. В рамках проверки троян получает список процессов и считает хеши запущенных приложений, которые сравниваются со следующим встроенным списком:

- msmpeng
- mssense
- avastsvc
- dwservice
- avp
- nortonsecurity
- coreserviceshell
- avguard

- fshoster32
- vsserv
- mbam
- adawareservice
- avgsvc
- wrsa

Шелл-код для очистки директории

Входные аргументы:

- имя директории.

Основные выполняемые действия:

- поиск в структуре PEB_LDR_DATA адреса библиотеки kernel32.dll,
- получение из экспорта библиотеки функций для работы шелл-кода, при этом имя библиотеки и имена функций определяются по хешу,
- формирование пути до директории %LOCALAPPDATA%\EROCS\,
- перезаписывание нулями и удаление всех файлов в указанной директории,
- удаление самой директории.

Шелл-код для перезапуска трояна

Основные выполняемые действия:

- получение списка процессов посредством функции NtQuerySystemInformation (параметр SystemProcessInformation), проверка совпадения значения поля UniqueProcessId значению 0x434F5245,
- если данное значение не найдено, то будет создан процесс, перезапускающий трояна,
- удаление ключа HKEY_CURRENT_USER\Software\Uninstall.

Самоудаление

Также троян имеет функцию самоудаления, запускаемую при определенном значении ключа реестра "deadline", который обновляется при получении новых ответов от C2-сервера и отвечает за время жизни трояна.

Также, если в ходе описанных выше проверок были обнаружены ошибки, троян запускает процедуру самоудаления посредством WinAPI. При этом выполняются следующие действия:

- удаляется директория хранения трояна,
- удаляется созданный хендшейк,
- удаляются ключи реестра, созданные во время работы трояна.

Отправка сообщений на C2-сервер

Для отправки сообщений троян использует следующие `User-Agent`:

```
Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:88.0) Gecko/20100101
Firefox/88.0
Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:79.0) Gecko/20100101
Firefox/79.0
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/90.0.4430.93 Safari/537.3
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/80.0.3987.149 Safari/53
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/79.0.3945.88 Safari/537.3
```

Все исходящие и входящие сообщения шифруются с помощью RSA.

Trojan.Siggen27.11306

Троян для ОС Windows, написанный на языке C. Представляет собой DLL с зашифрованной полезной нагрузкой.

Принцип действия

При инициализации троян последовательно создает два потока: один используется для расшифровки данных, а другой для запуска полезной нагрузки.

Изначально полезная нагрузка зашифрована ключом, соответствующим пути до исполняемого файла. Во время первого запуска троян пересоздает исполняемый файл, накладывая на него еще один этап шифрования, за счет чего полезная нагрузка привязывается к зараженному ПК.

Подготовительный этап состоит из следующих шагов:

- Генерируется случайная соль, которая сохраняется в новое тело трояна по определенному смещению
- Получается системная информация о BIOS
- Данная информация хешируется с использованием соли, созданной на 1 этапе, полученный хеш является ключом для шифрования полезной нагрузки
- Полезная нагрузка шифруется «пользовательским» ключом

После данного преобразования троян имеет два этапа расшифровки:

Этап 1. Расшифровка с помощью констант с зараженного ПК

- По определенному смещению берется соль, сохраненная в тело трояна
- С использованием соли получается хеш системной информации о BIOS
- Происходит расшифровка полезной нагрузки

Этап 2. Расшифровка полезной нагрузки, зашифрованной ключом по умолчанию

- Из структуры `RTL_USER_PROCESS_PARAMETERS` извлекается значение `ImagePathName` — данное поле является Unicode строкой, ее длина должна быть больше `0x76` байт (в нашем случае именем файла было `%LOCALAPPDATA%\Yandex\YandexBrowser\Application\Wldp.dll`)
- Из данного пути отбираются `0x76` последних байт
- Вычисляется хеш данного пути, который является ключом для симметричного алгоритма
- Происходит расшифровка полезной нагрузки

Алгоритм шифрования

В качестве симметричного алгоритма шифрования используется модифицированный алгоритм ChaCha20. Модификация заключается в добавочном слое для инициализации ключа: входной ключ проходит один раунд алгоритма, после чего становится уже ключом для нормального алгоритма.

Алгоритм хеширования

В качестве функции хеширования используется модифицированный алгоритм BLAKE2. Модификация заключается в использовании многократного хеширования входных данных.

Полезная нагрузка

Представляет собой шелл-код, сгенерированный с помощью https://github.com/TheWover/donut/tree/master. Данный шелл-код раскодирует и загружает MZPE-файл, написанный на .NET, основной задачей которого является запуск вредоносного ПО, скачиваемого из интернета. Основное тело шелл-кода находится в https://github.com/TheWover/donut/blob/master/loader_exe_x64.h.

Выполняемые действия шелл-кода:

- Проверка флага, отвечающего за выполнение загрузки в отдельном или основном потоке
- Расшифровка MZPE-файла в новую выделенную область памяти
- Загрузка библиотек `ole32.dll`, `oleaut32.dll`, `wininet.dll`, `mscoree.dll`, и `shell32.dll` с помощью функции `LoadLibraryA`
- Получение адресов функций `WldpQueryDynamicCodeTrust`, `WldpIsClassInApprovedList`, `EtwEventWrite` и `EtwEventUnregister` с помощью функции `GetProcAddress`
- Инициализация работы с интерфейсом AMSI
  - загрузка библиотеки `amsi.dll`
  - получение адресов функций `AmsiInitialize`, `AmsiScanBuffer` и `AmsiScanString`
- Проверка флага, включающего обход AMSI; в данном семпле этот флаг не установлен
- Загрузка .NET приложения

Функции .NET стейджера заключаются в скачивании другого вредоносного ПО, сохранении его под именем «`YandexUpdater.exe`» и последующем запуске. К моменту нашего расследования на сервере, с которого должно было скачиваться данное вредоносное ПО, файл был уже недоступен, вследствие чего нам не удалось его

однозначно идентифицировать. Однако, можно предположить, что данный файл мог являться тем же **Trojan.Packed2.46324**.

Заключение

Таким образом, мы видим многовекторную и многоступенчатую схему инфицирования одновременно двумя разными троянами, которые доставляются в скомпрометированную систему при открытии файла из фишингового письма. Несмотря на запутанную реализацию, методы профилактики и защиты от таких атак довольно просты. Они изложены ниже:

- Повышение осведомленности сотрудников в вопросах информационной безопасности (внимательно проверять ссылки и имена файлов, не открывать подозрительные объекты)
- Использование программных продуктов, выполняющих фильтрацию писем, чтобы предотвратить доставку вредоносных писем и вложений, например, Dr.Web Mail Security Suite
- Установка на всех узлах сети антивирусного ПО, которое не пропустит опасный файл при работе в интернете или заблокирует подозрительную активность на компьютерах пользователей, если файл был доставлен на USB носителе — например, Dr.Web Desktop Security Suite и Dr.Web Server Security Suite
- Своевременное обновление ПО, в рамках которого устраняются программные ошибки.

Приложение №1. Индикаторы компрометации

SHA1-хеши

Trojan.Packed2.46324

34a4c5f28c7df23662962c3eaa0a15b7ae48b488: YandexUpdater.exe

Trojan.Siggen27.11306

60eaa4fd53b78227760864e6cf27b08bc4bdde72: Wldp.dll

Trojan.Siggen28.53599

853d6a17f0a1a4035b52699a447eeb4ad1ca6cf7

Файловые артефакты

Job Application_202402523.rar (пароль: Инна)

Job Application.pdf.lnk

102fa066-cc9d-4a80-b3aa-12d5df196b42.pdf

Домены

infosecteam[.]info

IP-адреса

109.248.147[.]132