АНТИВИРУСНАЯ ЗАЩИТА БИЗНЕС-КЛАССА

Quasar, Sobaken и Vermin: раскрываем детали действующей кибершпионской кампании

25 июля 2018 года

С помощью инструментов удаленного доступа Quasar, Sobaken и Vermin киберпреступники следят за украинскими правительственными учреждениями и крадут данные из их систем. Эта кибергруппа была впервые упомянута в отчете в январе 2018 года, привлекла внимание ESET в середине 2017 года и сегодня продолжает разработку своего ПО.

В данном отчете мы раскрываем детали нынешней кампании, предоставим информацию о вредоносных программах и опишем методы, которые атакующие используют для распространения, таргетирования и ухода от обнаружения.

Профиль атакующих

Похоже, что данная группа не обладает выдающимися техническими познаниями или доступом к уязвимостям нулевого дня. Тем не менее, она успешно применяет социальную инженерию для распространения малвари и скрытной работы на протяжении длительного времени.

Нам удалось отследить ее работу вплоть до октября 2015 года, но не исключено, что группа начала деятельность значительно раньше.

Атакующие используют три модификации .NET-малвари: Quasar RAT (Remote Administration Tool), Sobaken (производный RAT от Quasar) и кастомный RAT Vermin. Инструменты одновременно использовались на одних и тех же целевых объектах, они частично делят инфраструктуру и подключаются к одним и тем же C&C серверам. Возможным объяснением использования трех параллельных модификаций является тот факт, что они разрабатывались независимо друг от друга.

Жертвы

Вредоносное ПО используется в атаках на украинские правительственные учреждения. По данным телеметрии ESET, зафиксированы сотни жертв в различных организациях и несколько сотен исполняемых файлов, относящихся к данной кампании.

Хронология

На рисунке 1 представлены основные события кампании в хронологическом порядке.

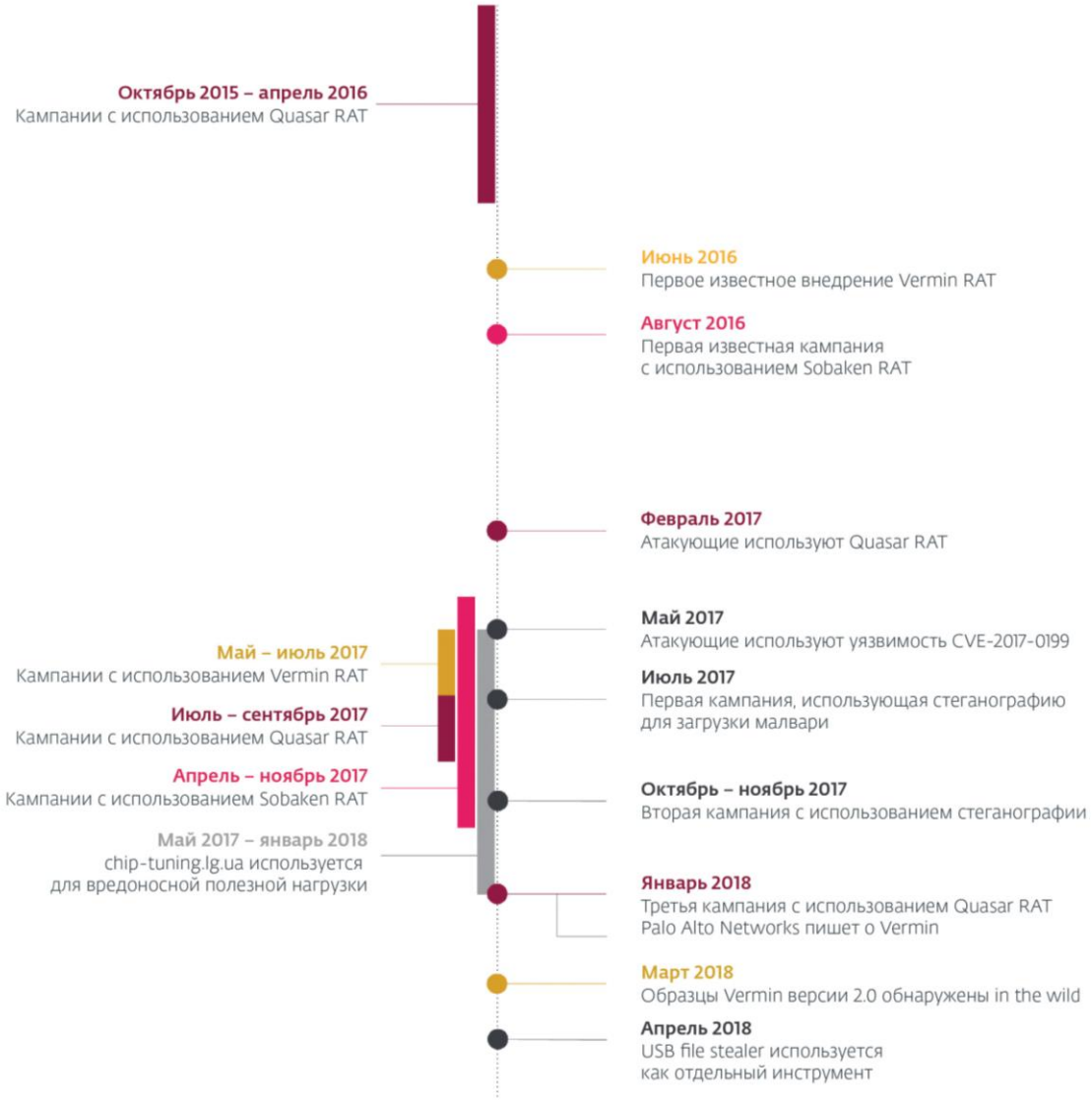

Рисунок 1. Хронология кампании

Распространение

По данным нашей телеметрии, в качестве первичного канала распространения трех RAT злоумышленники используют электронную почту. Они применяют социальную инженерию, чтобы убедить жертв скачать и запустить малварь.

В большинстве случаев имена файлов написаны на украинском языке и имеют отношения к работе жертв. Вот примеры таких файлов:

- «ІНСТРУКЦІЯ з організації забезпечення військовослужбовців Збройних Сил України та членів їх сімей» («Приказ об обеспечении безопасности военнослужащих украинской армии и членов их семей»)
- «новий проекту наказу, призначення перевірки вилучення» («Новый проект приказа проверки изъятия»)
- «Відділення забезпечення Дон ОВК Збільшення ліміту» («Отдел снабжения Дон ОВК. Увеличение кредитного лимита»)

Помимо базовых приемов социальной инженерии (привлечение внимания к вложению), атакующие используют три технических метода. Вероятно, это еще больше повышает эффективность кампаний.

Метод №1: приложения электронной почты используют символ Unicode right-to-left override, меняющий направление чтения символов, для сокрытия настоящего расширения. На самом деле это исполняемые файлы, использующие иконки Word, Excel, PowerPoint или Acrobat Reader, чтобы не вызывать подозрения.

Пример имени файла: как показано на рисунке 2 «Перевезення твердого палива (дров) для забезпечення опалювання_<>xcod scr» («Транспортировка дров для обеспечения отоплением») будет выглядеть для невнимательных пользователей как файл .DOCX.

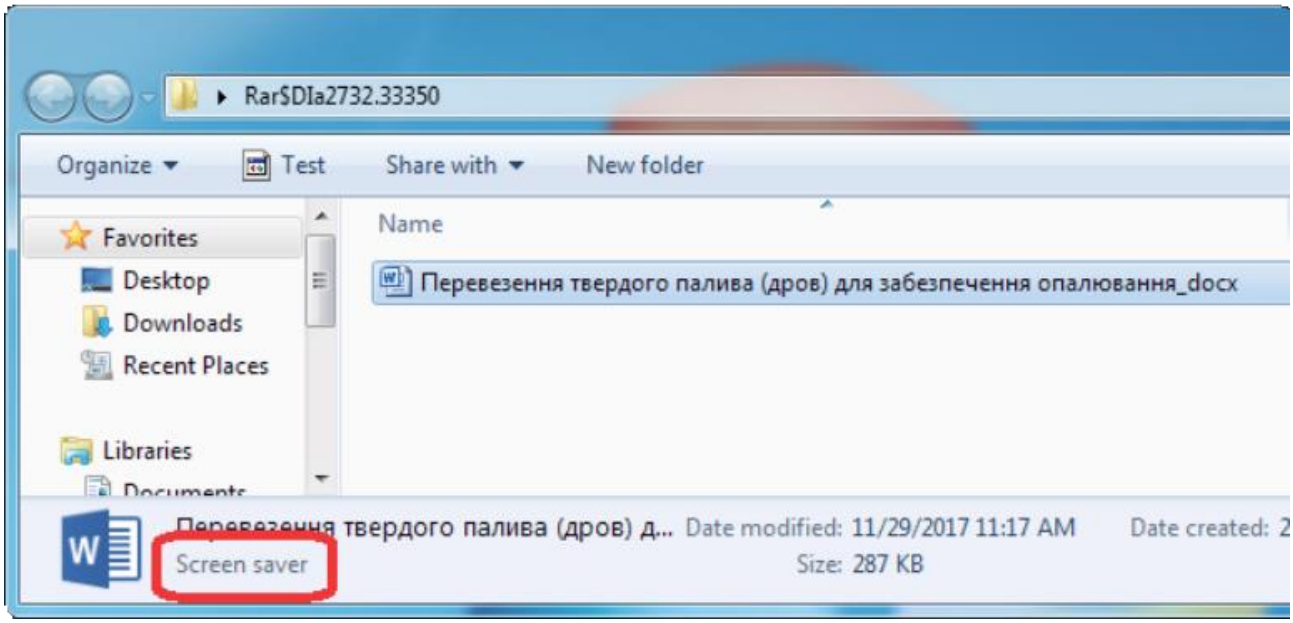

Рисунок 2. Исполняемый файл, маскирующийся под документ Word

Метод №2: приложения электронной почты, маскирующиеся под самораспаковывающиеся архивы RAR.

Пример: электронное письмо с приложением «Наказ_МОУ_Додатки_до_Інструкції_440_ост.rar»

(«Приказ Министерства Обороны, Приложения к инструкции №440»), как показано на рисунке 3. Внутри архива RAR есть исполняемый файл с названием «Наказ_МОУ_Додатки_до_Інструкції_440_ост.exe», использующий иконку RAR SFX.

Предположительно, жертвы запускают этот файл, ожидая дальнейшей распаковки содержимого самораспаковывающегося архива, но тем самым невольно запускают вредоносный исполняемый файл.

Метод №3: документ Word + эксплойт CVE-2017-0199. Эта уязвимость применяется, когда жертва открывает специально созданный документ Word. Процесс работы Word передает HTTP-запрос файла HTA, содержащего вредоносный скрипт, расположенный на удаленном сервере. Затем вредоносный скрипт запускается `mshta.exe`. Первая публичная информация об этой уязвимости появилась в апреле 2017 года, и Microsoft закрыл ее, выпустив обновление безопасности для всех версий Windows и Office.

Согласно телеметрии ESET, эти злоумышленники начали использовать данный метод в мае 2017 года. Атакующие использовали hxxp://chip-tuning lg[ ]ua/ для доставки файлов HTA и финальной полезной нагрузки.

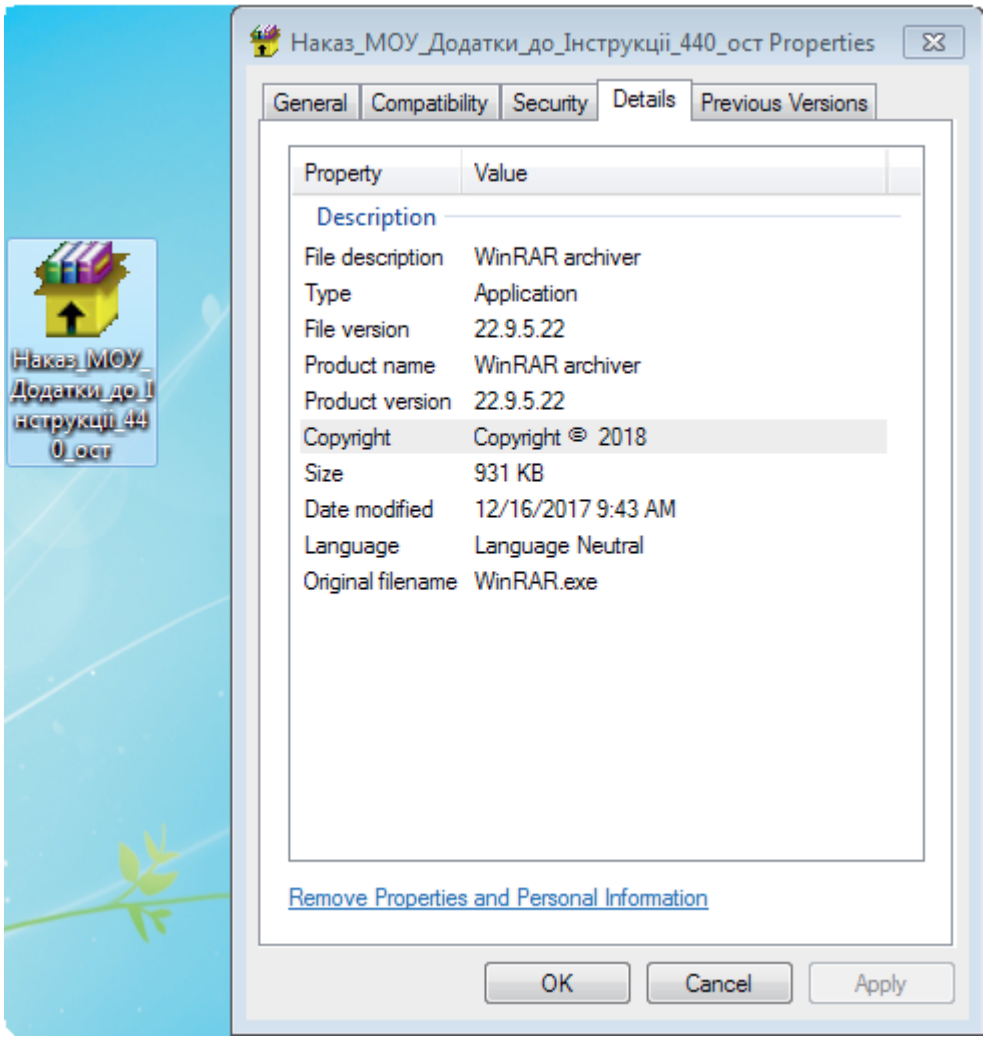

Рисунок 3. Файл, замаскированный под самораспаковывающийся архив RAR. Данные о версии и копирайт подсказывают, что это фейк

Установка и персистентность

Процедура установки одинакова для трех модификаций малвари, используемых данной группой. Дроппер сбрасывает файлы полезной нагрузки (Vermin, Quasar или Sobaken) в папку `%APPDATA%`, в подпапку с названием легитимной компании (обычно это Adobe, Intel или Microsoft). Затем, как показано на рисунке 4, он создает задание в планировщике для запуска компонента каждые 10 минут для обеспечения персистентности. Некоторые версии также применяют метод, состоящий в использовании функционала быстрого вызова панели управления Windows Control Panel, чтобы сделать свои папки недоступными из Windows Explorer. Эти папки при нажатии в Windows Explorer не будут открыты, вместо этого открывается страница «Все задачи».

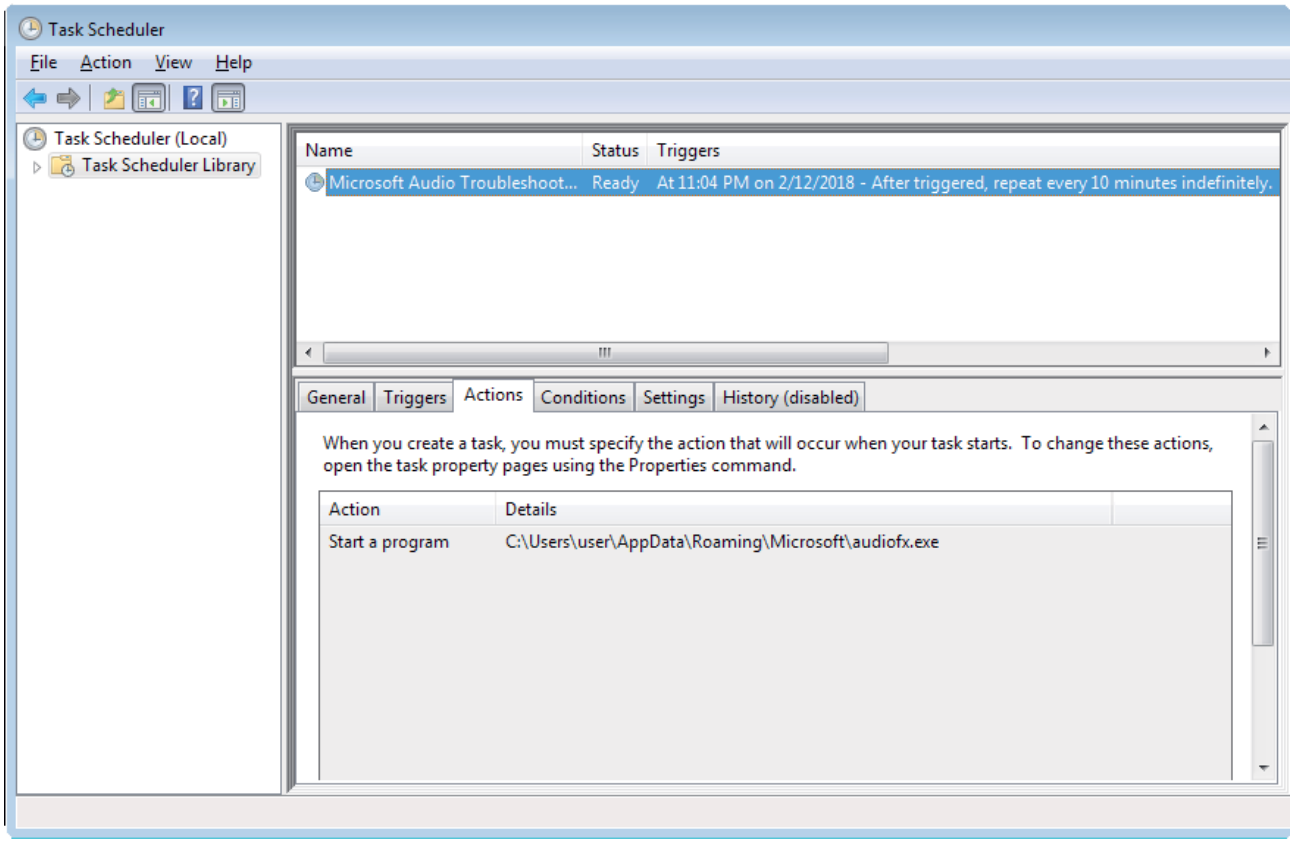

Рисунок 4. Задача, запускающая вредоносный компонент каждые 10 минут

Примеры:

```
C:\Users\Admin\AppData\Roaming\Microsoft\Proof\Settings.{ED7BA470-8E54-465E-825C- 99712043E01C}\TransactionBroker32.exe
C:\Users\Admin\AppData\Roaming\Adobe\SLStore\Setting.{ED7BA470-8E54-465E-825C- 99712043E01C}\AdobeSLService.exe
```

Таргетирование

Атакующие используют достаточно много приемов, чтобы малварь работала только на целевых машинах. Особенно тщательно они стараются избежать автоматизированных систем анализа и песочниц.

Метод №1: проверка раскладки клавиатуры в Windows
Малварь проверяет, установлена ли русская или украинская раскладка клавиатуры. Если нет, ее работа немедленно прекращается.

Метод №2: Проверка IP-адреса
Малварь получает IP-адрес зараженного компьютера через запрос к легитимному сервису Ipinfo.io/json. Работа будет завершена, если IP-адрес не обнаруживается на территории Украины или России, либо если он зарегистрирован на одного из нескольких выбранных поставщиков облачных услуг или разработчиков антивирусного ПО. Код, выполняющий проверку, показан в результатах дизассемблирования на рисунках 5 и 6.

```
// Token: 0x06000005 RID: 5 RVA: 0x00003B98 File Offset: 0x00001D98
private static bool CheckForGeoLocation()
{
    bool flag = false;
    try
    {
        IpSite ipSite = Program.CheckGeoInfo();
        if (ipSite != null)
        {
            if (ipSite.country.ToLowerInvariant() == "ru" || ipSite.country.ToLowerInvariant() == "ua")
            {
                flag = true;
            }
            if (flag)
            {
                flag = Program.CheckForAntiVendors(ipSite.org);
            }
        }
    }
    catch (Exception)
    {
        flag = false;
    }
    return flag;
}
```

Рисунок 5. Код, выполняющий проверку геолокации IP-адреса зараженной машины

```
// Token: 0x06000004 RID: 4 RVA: 0x00003A80 File Offset: 0x00001C80
private static bool CheckForAntiVendors(string name)
{
    bool result = true;
    string[] source = new string[]
    {
        "amazon",
        "anonymous",
        "blue coat systems",
        "cisco systems",
        "cloud",
        "data center",
        "dedicated",
        "eset, spol",
        "fireeye",
        "forcepoint",
        "hetzner",
        "hosted",
        "hosting",
        "leaseweb",
        "microsoft",
        "nforce",
        "ovh sas",
        "security",
        "server",
        "strong technologies",
        "trend micro",
        "blackoakcomputers",
        "kaspersky"
    };
    try
    {
        if (source.Any((string t) => name.ToLowerInvariant().Contains(t.ToLowerInvariant())))
        {
            result = false;
        }
    }
    catch (Exception)
    {
        result = false;
    }
    return result;
}
```

Рисунок 6. Код, сверяющий IP-адрес со списком поставщиков облачных услуг и антивирусных вендоров

Метод №3: проверка эмуляции сетевой среды

Системы автоматизированного анализа часто используют такие инструменты как Fakenet-NG, где коммуникация DNS/HTTP завершается успешно и выдает какой-либо результат. Авторы малвари пытаются определить такие системы через генерацию случайного имени/URL сайта и проверяют отсутствие подключения к данному URL (на рисунке 7), как это было бы на реальной системе.

```
// Token: 0x06000006 RID: 6 RVA: 0x00003C08 File Offset: 0x00001E08
private static bool CheckForFakeAddress()
{
    bool result = false;
    try
    {
        ((HttpWebRequest)WebRequest.Create(string.Concat(new string[]
        {
            "http://",
            FakesHelper.RandomizeText(5, 50),
            ".org/",
            FakesHelper.RandomizeText(10, 32),
            ".html"
        }))).GetResponse();
    }
    catch (WebException)
    {
        result = true;
    }
    return result;
}
```

Рисунок 7. Код, генерирующий случайный URL и запускающий загрузку

Метод №4: проверка определенного имени пользователя
Малварь не запускается под аккаунтом с именем пользователя, типичным для автоматизированных систем анализа вредоносного ПО, как показано на рисунке 8.

```
// Token: 0x06000003 RID: 3 RVA: 0x000039C8 File Offset: 0x00001BC8
public static bool CheckForSandboxUserNames()
{
    bool result = true;
    string[] source = new string[]
    {
        "ANDY",
        "COMPUTERNAME",
        "CUCKOO",
        "SANDBOX",
        "NMSDBOX",
        "XXXX-OX",
        "CWSX",
        "WILBERT-SC",
        "XPAMAST-SC",
        "ANTONY",
        "JOHN"
    };
    try
    {
        string strName = PcAccountHelperEx.GetUserName().ToUpperInvariant();
        if (source.Any((string t) => string.Equals(strName, t, StringComparison.InvariantCultureIgno
        {
            result = false;
        }
    }
    catch (Exception)
    {
        result = false;
    }
    return result;
}
```

Рисунок 8. Сверка текущего имени пользователя со списком известных систем анализа вредоносного ПО

Применение стеганографии

С середины 2017 года атакующие используют стеганографию, скрывая вредоносные компоненты в изображениях на бесплатных хостингах saveshot.net и ibb.co.

Стеганография – это наука, позволяющая скрывать данные «у всех на виду», внутри другой, несекретной информации. В нашем случае вредоносный файл EXE был зашифрован и спрятан в файл JPEG, как на рисунке 9. Малварь скачивает и декодирует JPEG, извлекает скрытые данные, расшифровывает из этих данных файл EXE и запускает его.

Рисунок 9. Пример изображения JPEG, используемого для сокрытия доставляемого вредоносного компонента (размер изменен, а компонент удален)

Процесс расшифровки достаточно сложен и может быть описан следующим образом:

1. Скачивание файла JPEG с адреса URL, жестко прописанного в бинарнике.

2. Брутфорс пароля из восьми цифр через вычисление его хеша и сверка с хешем, жестко закодированным в скачанном бинарном файле. Этот шаг интенсивно использует ЦП, для его завершения на обычном компьютере требуется больше 10 минут. Скорее всего, это еще одна мера для противодействия автоматизированным системам анализа вредоносного ПО.

3. Обработка файла JPEG и извлечение скрытых в нем данных, как видно из дизассемблированного кода на рисунках 10 и 11. Алгоритм, используемый малварью, очень похож на тот, что применяется в JSteg, одном из старейших и простейших алгоритмов стеганографии для файлов JPEG. Он прячет данные в LSB (самый младший двоичный разряд) коэффициентов дискретного косинусного преобразования файла JPEG. Такие скрытые данные обычно не влияют на изображение видимым для невооруженного глаза образом, но их наличие легко обнаруживают специализированные алгоритмы. Однако данный алгоритм стеганографии очень легко внедрить, что, вероятно, стало причиной его выбора авторами малвари.

4. Извлечение данных и их распаковка с помощью GZip.

5. Расшифровка распакованных данных по AES с паролем, полученным в шаге №2.

6. Декодирование дешифрованных данных по Base64.

7. Запись файла EXE на диск и исполнение.

В итоге авторы данной угрозы оставили идею использования стеганографии и начали применять hxxp://chip-tuning lg[ ]ua для передачи незашифрованных исполняемых файлов.

```
int num7 = 0;
int num8 = (int)(this._sHeader.Ns - 1);
for (int j = num7; j <= num8; j++)
{
    int num9 = 0;
    int num10 = (int)(this._fHeader.Csp[j].V - 1);
    for (int k = num9; k <= num10; k++)
    {
        int num11 = 0;
        int num12 = (int)(this._fHeader.Csp[j].H - 1);
        for (int l = num11; l <= num12; l++)
        {
            this.DecodeDctDataUnit(ref this._fHeader.Csp[j]);
            JpegFile.TotalMax++;
            this._dataUCounter++;
            if (this._fieldWriter != null)
            {
                this.WriteEmbedData();
                this.EncodeDctDataUnit(ref this._fHeader.Csp[j]);
            }
            else
            {
                this.ReadEmbedData();
            }
        }
    }
}
```

Рисунок 10. Код стеганографии внутри декодера JPEG

```
private void ReadEmbedData()
{
    if (this._dataEnded)
    {
        return;
    }
    checked
    {
        JpegFile.Modified++;
        int num = 1;
        do
        {
            if (this._block[num] != 0)
            {
                this._byteProcd = (unchecked((byte)(this._byteProcd << 1)) | (byte)
                  (this._bitCode[32767 + this._block[num]].Value & 1));
                this._bitProcd++;
                if (this._bitProcd == 8)
                {
                    if (this._indOfByteProcd == 4)
                    {
                        this._dataLengthWithCheckVal = 0;
                        int num2 = 0;
                        do
                        {
                            byte value = this.EmbedData[num2];
                            if (this._rotChosen == this._passStore.Count)
                            {
                                this._rotChosen = 0;
                            }
                            int num3 = 1;
                            int num4 = this._passStore[this._rotChosen];
                            for (int i = num3; i <= num4; i++)
                            {
                                this.RotateLeft(ref value);
                            }
                            this._rotChosen++;
                            this.EmbedData[num2] = value;
                            this._dataLengthWithCheckVal <<= 8;
                            this._dataLengthWithCheckVal |= (int)this.EmbedData[num2];
                            num2++;
                        }
                        while (num2 <= 3);
                    }
                    this.EmbedData.Add(this._byteProcd);
                    this._indOfByteProcd++;
                    if (this._indOfByteProcd >= this._dataLengthWithCheckVal + 4)
                    {
                        goto IL_187;
                    }
                    this._bitProcd = 0;
                    this._byteProcd = 0;
                }
            }
            num++;
        }
        while (num <= 63);
        return;
        IL_187:
        this._dataEnded = true;
    }
}
```

Рисунок 11. Код стеганографии внутри декодера JPEG

Модификации малвари

Эти злоумышленники используют в атаках три модификации малвари. Ниже мы приводим краткий обзор каждой и сконцентрируемся на описании характерных особенностей.

Quasar

Quasar – RAT с открытым исходным кодом, доступным на GitHub. Мы видели несколько кампаний, в которых данная кибергруппа применяла бинарные файлы Quasar RAT. Первая известная нам кампания длилась с октября 2015 по апрель 2016 года. Следующая прошла в феврале 2017 года. Артефакты компиляции показывают путь к PDB `n:\projects\Viral\baybak_files_only\QRClient\QuasarRAT-master\Library\obj\ Release\Library.pdb`

Другая кампания с Quasar RAT, использующая командные серверы этих злоумышленников (mailukr.net), реализована в июле – сентябре 2017 года. Атакующие применили старую версию Quasar RAT под названием “xRAT 2.0 RELEASE3”. Артефакты компиляции в дроппере показывают путь к PDB `N:\shtorm\WinRARArchive\ obj\Release\WinRAR.pdb`

Sobaken

Sobaken – значительно модифицированная версия Quasar RAT. Если сравнить структуры программы Quasar и Sobaken, можно наблюдать много общего – см. рисунок 12.

Авторы Sobaken сократили функционал малвари, поэтому исполняемый файл стал меньше и его проще скрыть. Они также добавили приемы для обхода песочницы и другие, описанные выше.

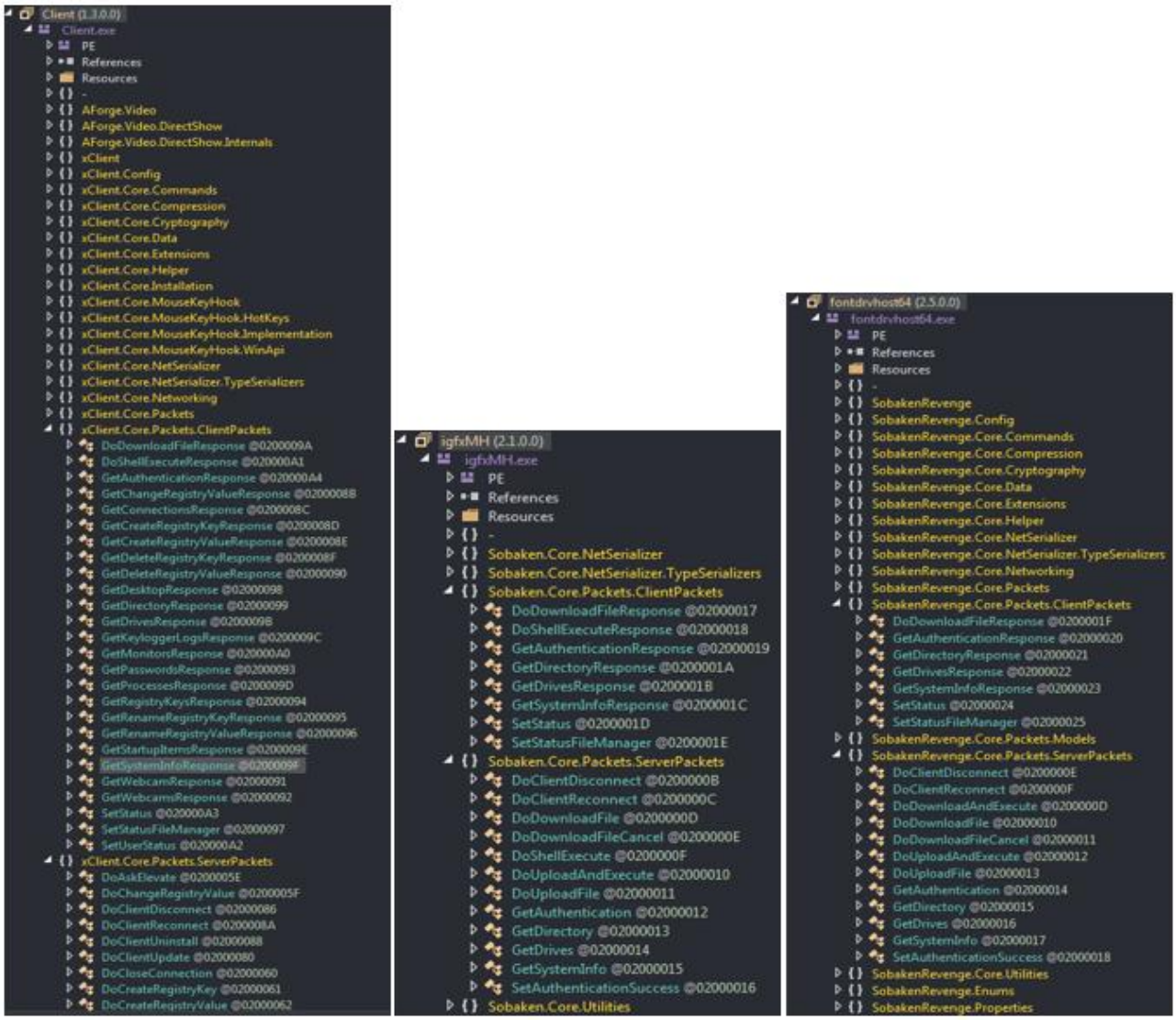

Рисунок 12. Эволюция Sobaken. Слева Quasar RAT v1.3, посередине и справа – две версии Sobaken

Vermin

Vermin – кастомный бэкдор, который использует только эта кибергруппа. Впервые малварь была задокументирована в отчете Palo Alto Networks за январь 2018 года. Бэкдор появился в середине 2016 года и все еще используется. Как и Quasar с Sobaken, он написан на .NET. Для затруднения анализа код защищен с помощью коммерческой системы защиты кода .NET Reactor или опенсорсным протектором ConfuserEx.

Кроме того, как и Sobaken, он использует Vitevic Assembly Embedder, бесплатное ПО для встраивания нужных DLL в основной исполняемый файл, которое есть в магазине Visual Studio Marketplace.

Функционал

Vermin – полнофункциональный бэкдор с несколькими опциональными компонентами. Последняя известная на момент написания версия (Vermin 2.0) поддерживает следующие команды, суть которых можно понять из названий:

— StartCaptureScreen
— StopCaptureScreen
— ReadDirectory

— UploadFile
— DownloadFile
— CancelUploadFile
— CancelDownloadFile
— GetMonitors
— DeleteFiles
— ShellExec
— GetProcesses
— KillProcess
— CheckIfProcessIsRunning
— CheckIfTaskIsRunning
— RunKeyLogger
— CreateFolder
— RenameFolder
— DeleteFolder
— UpdateBot
— RenameFile
— ArchiveAndSplit
— StartAudioCapture
— StopAudioCapture
— SetMicVolume

Большинство команд реализовано в основной полезной нагрузке. Лишь несколько команд и дополнительных функций – через опциональные компоненты, загружаемые атакующими на машину жертвы. В числе опциональных компонентов:
— Запись аудио
— Кейлоггер
— Кража пароля
— Кража файла с USB (USB file stealer)

Инструмент записи аудио (AudioManager)

Полноценный компонент Vermin, способный записывать звук с микрофона компьютера жертвы. Он принимает три команды Vermin: StartAudioCapture, StopAudioCapture и SetMicVolume. Полученные данные сжимаются с помощью кодеков Speex и загружаются в формате SOAP на C&C серверы Vermin.

Кейлоггер (KeyboardHookLib)

Кейлоггер в Vermin – простой отдельный исполняемый файл, который перехватывает все нажатия клавиатуры и пишет их в файл в зашифрованном виде. Также он записывает содержимое буфера и названия активных окон. Сам по себе кейлоггер не может связываться с C&C серверами Vermin; для передачи собранной информации используется основной бэкдор.

Путь к PDB в кейлоггере подтверждает связь с малварью Vermin:
`Z:\Projects\Vermin\KeyboardHookLib\obj\Release\AdobePrintLib.pdb`

Инструмент кражи пароля (PwdFetcher)

Отдельный компонент Vermin для кражи паролей используется для извлечения сохраненных паролей из браузеров (Chrome, Opera). Основная часть кода, похоже, скопирована из статьи с Хабра. Некоторые образцы также содержат код для извлечения информации из браузера Firefox, но, кажется, он не используется. Как показано на рисунке 13, этот компонент тоже содержит пути к PDB,

схожие с компонентом кейлоггера, что подтверждает связь с Vermin.

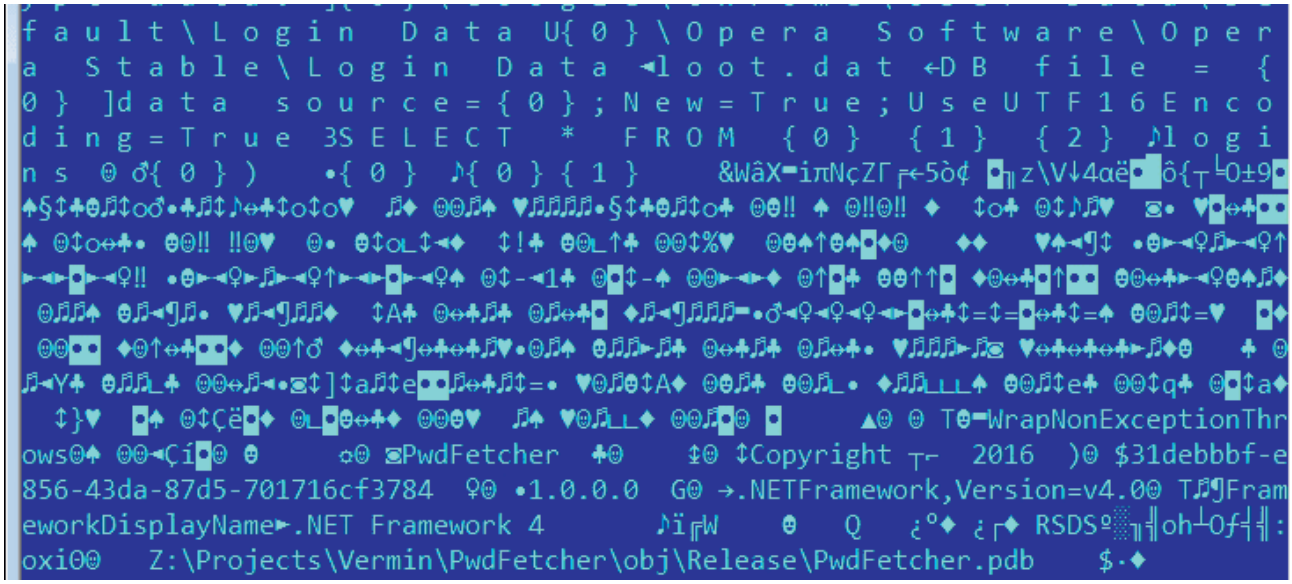

Рисунок 13. Артефакты компиляции, позволяющие связать компонент для кражи пароля с Vermin

Инструмент кражи файлов с USB (UsbGuard)

`UsbGuard.exe` – опциональный компонент, используемый и Sobaken, и Vermin. Это небольшая отдельная программа, которая отслеживает подключенные к компьютеру USB-носители и копирует все файлы, которые соответствуют фильтру, настроенному атакующими. Краденые файлы затем передаются с помощью модуля основного бэкдора. В образце данного компонента обнаружено множество различных путей к PDB, что явно связывает его с Vermin.

С апреля 2018 года компонент для кражи файлов используется как отдельный инструмент. Он копирует файлы и незамедлительно загружает их на подконтрольный атакующим сервер.

В анализируемых образцах атакующие искали файлы со следующими расширениями:
— doc
— docx
— xls
— xlsx
— zip
— rar
— 7z
— docm
— txt
— rtf
— xlsm
— pdf
— jpg
— jpeg
— tif
— odt
— ods

Заключение

Среди множества атак с применением вредоносного ПО, нацеленных на высокоранговые объекты на территории Украины, данная кампания не получила особого внимания. Возможно, это связано с использованием кода из открытых источников. Тем не менее, группа уже перешла к разработке собственного инструментария.

Использование нескольких семейств вредоносного ПО и механизмов заражения, включая социальную инженерию и стеганографию, на протяжении последних трех лет может быть связано с тем, что атакующие экспериментируют с методами, либо работают несколькими группами.

Успешное применение тривиальных приемов (например, отправка RAR и EXE по электронной почте) подчеркивает важность защиты от человеческого фактора.

Индикаторы компрометации (IoC)

C&C-серверы

Sobaken C&C

```
akamaicdn.ru
akamainet021.info
cdnakamai.ru
windowsupdate.kiev.ua
akamainet022.info
akamainet066.info
akamainet067.info
notifymail.ru
mailukr.net
188.227.16.73
212.116.121.46
206.54.179.160
```

Quasar C&C

```
188.227.75.189
mailukr.net
cdnakamai.ru
notifymail.ru
```

Vermin C&C

```
185.158.153.222
188.227.17.68
195.78.105.23
tech-adobe.dyndns.biz
notifymail.ru
akamainet023.info
mailukr.net
185.125.46.24
akamainet024.info
206.54.179.196
```

Передача компонентов, эксфильтрация данных

```
chip-tuning.lg.ua
www.chip-tuning.lg.ua
olx.website
news24ua.info
rst.website
1ua.eu
novaposhta.website
```

SHA1

Vermin

```
028EBDBEBAC7239B41A1F4CE8D2CC61B1E09983B
07E1AF6D3F7B42D2E26DF12A217DEBACEDB8B1B9
09457ACB28C754AA419AB6A7E8B1940201EF3FFE
0EEE92EC2723ED9623F84082DAD962778F4CF776
10128AB8770FBDECD81B8894208A760A3C266D78
131F99A2E18A358B60F09FD61EE312E74B02C07C
14F69C7BFAF1DF16E755CCF754017089238B0E7B
1509F85DE302BE83A47D5AFAD9BEE2542BA317FC
170CEE6523B6620124F52201D943D7D9CA7B95E5
191159F855A0E580290871C945245E3597A5F25C
1F12C32A41D82E978DE333CD4E93FDAA1396BE94
22B17966B597568DB46B94B253CD37CBCF561321
2C7332D8247376842BD1B1BD5298844307649C99
2E08BA5DF30C0718C1733A7836B5F4D98D84905E
2EDF808F8252A4CBCB92F47A0AEDC1AAAE79A777
360F54B33AC960EE29CA0557A28F6BB8417EF409
431FCE6A47D0A48A57F699AA084C9FF175A9D15F
45438834FDC5C690DA3BC1F60722BE86B871280D
4A8A8188E3A7A137651B24780DF37CB6F610CC19
4C1E4E136B7922F9E28D1B38E9760E28929E4F0B
5B6EA57FFC09593C3B65D903368EA5F7FAA2EB68
61D366939FE36861B2FECB38A4DFF6D86C925A00
6A72366D8AE09F72F0466FB59E8ED372F8B460D7
6FECA622B0FB282064F7DE42BA472A8EC908D0D6
70A772485C5ED330C6876FA901BA722CD44CA05E
70D97367A3DBD5D45482B6AF8C78C58B64D3F3B3
7803FD9753930522705F2B6B4E73622887892C28
7B11A84B18DC4B5F1F2826E7925F0B2DC1B936AE
889FD0BEB3197DDD6C88F5C40D6B8E4D74A892CE
9B6FBABFA2A77FA633F7A2EB352979D5C68CEBC6
A451291F17489E3A59F440A1B693D691B053C531
A53D77E55A06CF131D670339BACEC5AC0F0C6D66
A925D0AFB5D4F5FAC65543C993BE4172F1DBF329
B5F81C804E47B76C74C38DF03A5CBE8A4FE69A9A
B99DE55043099E9506B304660B8E1374787AB195
C00C104FC3E9F5977D11C67EF0C8C671D4DFC412
CA0296FA9F48E83EA3F26988401B3F4C4E655F7A
D4C6540E789BD3839D65E7EDA5CCA8832493649E
D5EDE1BBB9A12757E24BE283AFC8D746ADC4A0D4
DEFBFD98C74BEFF839EEB189F0F6C385AD6BA19B
ECF152EB6417A069573F2C7D9A35B9CC31EC8F56
EE2D40825C77C8DFEF67999F0C521919E6672A10
EF09AC6BA08A116F2C4080CBEE8CEF9523E21265
F414C49CF502D1B6CC46E08F3AC97D7846B30732
```

Sobaken

```
087F77998004207BCCFFBF3030B6789648930FA5
0A4A2BCB3EF4E19973D5C4BE4E141B665CC0BFE0
1CEEF0813C0F096E6DA5461DC4B3BF901C500C56
293DBFF0230DAB3C4C21428F90C8EF06E9F35608
37E2947BFB5FC0839087C5BCE194EC193F824C85
39525CBCA591F2A10946BA62A56E4C3382CD4FC0
3CE0A18E9A8A2B95827008DBFF16364B6FEDF361
3E869038080DAE006FF6B20DF9B0CD9CB3A5E1A1
400830AB6DD46789B00D081ADF0F82623472FB13
43F382A330A454FF83F4F35FB571ECF587A4694A
4449FBE2B28A81B760B284880ADBED43462C2030
4712AF28168FD728A13EFD520E0665FFD076B6FB
4F504D7B35660943B206D6034752C686365EA58D
53239A62E09BB0B4E49B7954D533258FEF3342C4
540292753FA0CC4ACB49E5F11FEDEA4B7DEF11D8
5589E8018DC7F934A8FDAB62670C9140AF31CAB6
57BBA7D8786D3B0C5F93BC20AB505DF3F69C72D4
630FE59D60F6882A0B9E35ED606BF06AD4BA048C
63EA7C844D86882F491812813AAAD746738A6BE9
64121FA2FD2E38AC85A911A9F7ADD8CA1E1A9820
64DBA711FDD52FECF534CAC0C6FE8848FE36F196
650AB5E674FEF431EBC8CF98141506DDC80C5E64
6EF13E9D5B0B6FCB5EB2A7439AAD7B21EA7FB7AC
7177F64362A504F3DF8AA815CEF7136D5A819C04
9B91EC03A09C4CF6DBEC637B3551BDCA11F04A9B
A26764AFB1DAC34CAA2123F7BF3543D385147024
A55319D3DBD7B9A587F5156CF201C327C803FBC9
A841FF1EE379269F00261337A043448D3D72E6FD
AAB5BAAAE8A2577E1036769F0D349F553E4D129B
ACB989B3401780999474C5B1D7F9198ECA11549A
B65372E41E7761A68AEF87001BBB698D8D8D5EC6
BDB5E0B6CA0AA03E0BECA23B46A8420473091DFF
C4421084C19423D311A94D7BB6CB0169C44CBECD
C7E76993BB419DC755BD0C04255AB88E6C77B294
CF5238C467EBE2704528EED18AB4259BFDC604E3
D2334E161A1720E2DF048E4366150729B9395144
D35FB6E031720876482E728A40532703EF02A305
D82DF2903AA4BC5FD4274B5D1BFAF9E081771628
E4B3CBCA9A53B7B93177A270C2A76F981D157C34
E585AA2C5BFB9D42D2E58DB3833330D056713B9A
F4A485696FC871307C22906701CBBB3FA522499B
F5C75450108440D0BC9E7B210F072EF25A196D20
```

Quasar

```
0A4915B81D9A9ACF4E19181DEEEBBE244430C16B
323160C88A254127D9ADB2848AE044AFFF376A4D
395166835495B418773C9690227779D592F94F71
3EE410DD50FC64F39DFF0C4EE8CC676F0F7D5A74
5B665152F6596D4412267F9C490878455BA235F9
5FE8558EB8A3C244BE2DA8BE750221B9A9EE8539
61CB5E535F0AC90A1F904EC9937298F50E2B4974
6A1CD05F07B1024287CEA400237E1EA9D2FE1678
7676AFF05A3550E5BBFF78CF4D10C9E094447D72
86165F464EC1912A43445D80559D65C165E2CF76
AB3CD05BE6B0BA8567B84D10EDE28ABF87E115AC
```

BFD7158E1C2F6BA525E24F85ED8CCF8EF40FD370
CFEBEFC92DCDF1687FD0BC1B50457EBDEA8672A2
D21B8514990B0CEAC5EAE687DEAA60B447139B9D

Стеганография

04DA3E81684E4963ABEC4C0F6D56DF9F00D2EF26
3C618A0C4BF4D3D24C9F2A84D191FC296ED22FA4
746155881D5AB2635566399ACC89E43F6F3DA91A
CADBC40A4EFB10F4E9BD8F4EC3742FA8C37F4231
E22CE72406B14EF32A469569FBE77839B56F2D69

HTA файлы

39F5B17471FD839CC6108266826A4AD8F6ECD6A3
751FBD034D63A5E0A3CA64F55045AE24E575384A
76433D1D13DF60EC0461ED6D8007A95C7A163FF9
89DF6A7551B00969E22DC1CAE7147447ACA10988
D6D148050F03F5B14681A1BBF457572B9401B664

Инструмент записи аудио

1F49946CA2CE51DC51615000BAA63F6C5A9961F1
98F62C2E6045D5A15D33C8383ADACF9232E5FBE3
E7C4A69EBD7B41A6AF914DD3D3F64E1AA1ABE9B4
F233A0F2997BB554D4F1A4B7AC77DAE4180850FA

Кейлоггер

21921864D2F1AB2761C36031A2E1D2C00C9B304A
3C2D0615BEF6F88FED6E308D4F45B6133080C74F
91E8346910E0E6783ACFC4F2B9A745C81BD7573A

Инструмент кражи пароля

2A5C9D4DAE5E53B2962FBE2B7FA8798A127BC9A6
9B1586766AF9885EF960F05F8606D1230B36AC15
A2F0D5AF81D93752CFF1CF1E8BB9E6CAEE6D1B5E
CE18467B33161E39C36FC6C5B52F68D49ABCFC2A

Инструмент кражи файлов с USB

050EB7D20EE8EF1E1DAEE2F421E5BF648FB645DF
069A919B3BC8070BB2D71D3E1AD9F7642D8ECF0F
0D265E0BDA9DF83815759ABCA64938EC0FF65733
0D7DF910D0FB7B100F084BFB8DFA0A9F2371171A
2FF3F5DA2960BE95E50B751680F450896AD1ED67
3200ECC7503F184F72AB9DA1DC3E1F8D43DDFD48
46D256EF277328E803D2B15CA7C188267059949D
524EE1B7269D02F725E55254A015200BB472463A
53A0EFD3D448DA8E32CFDDA5848312D3CF802B06
6FC150A9CAFA75813E7473C687935E7E4A5DCE24
70559245303F99630A27CB47B328C20C9666F0BB
7D8044A5CBEFE3B016F2132A5750C30BB647E599
8FD919D531A7A80615517E1AC13C2D0F050AF20D
9D22421DA9696B535C708178C72323F64D31FC80
BFD2DFA3D6AF31DF4B9CC2F6B31B239ADF1CECA1
C08A6222B59A187F3CF27A7BAE4CACFACC97DDEE

```
C2F6A65E14605828880927B9BA3C386507BD8161
C562006D2FA53B15052A4B80C94B86355CCA7427
CB43058D9EBB517832DF7058641AEDF6B303E736
CC8A9C28E884FDA0E1B3F6CEAB12805FEA17D3C1
D3CC27CA772E30C6260C5A3B6309D27F08A295CD
E7A2DE3776BA7D939711E620C7D6AB25946C9881
EE6EFA7A6A85A1B2FA6351787A1612F060086320
EF0ABB3A0CD1E65B33C0F109DD18F156FC0F0CDE
F63BE193C8A0FBB430F3B88CC8194D755BAD9CD1
```

Детектирование продуктами ESET

Большинство файлов было автоматически распознано ESET по принципу схожести с вредоносной программой. Обнаружения напрямую связаны с большинством файлов в кампании:

```
MSIL/Agent AWB
MSIL/Agent AZG
MSIL/Agent AZJ
MSIL/Agent AZX
MSIL/Agent BCH
MSIL/Agent BCV
MSIL/Agent BCY
MSIL/Agent BFT
MSIL/Agent BGB
MSIL/Agent BGC
MSIL/Agent BGE
MSIL/Agent BGM
MSIL/Agent BJU
MSIL/Agent SCM
MSIL/Spy Agent BBB
MSIL/Spy Agent BIF
MSIL/TrojanDownloader Agent DYV
MSIL/TrojanDownloader Small BBM
MSIL/TrojanDropper Agent DBE
MSIL/TrojanDropper Agent DJQ
MSIL/TrojanDropper Agent DJR
```