Статья Файлы MS Office снова вовлечены в недавнюю троянскую кампанию Emotet — часть II


xss.is/threads/67809

В этом посте вы узнаете, как выглядят данные в ответных пакетах с вредоносными модулями, какие модули были получены от C2-сервера для текущей кампании Emotet и как они развернуты на устройстве жертвы. Вы также узнаете, какие конфиденциальные данные эти модули крадут с устройства жертвы.

Затрагиваемые платформы: Microsoft Windows
Затронутые стороны: пользователи 64-разрядной версии Windows
Воздействие: контролирует устройство жертвы и собирает конфиденциальную информацию.
Уровень серьезности: критический

Когда X.dll получает ответ от модулем

Как только C2-сервер обработает и обнаружит первый отправленный пакет, содержащий важные данные, такие как версия системы устройства жертвы, архитектура Windows и т. д., он ответит вредоносными модулями, которые Emotet запустит на устройстве жертвы. Все полученные модули безфайловые. То есть они существуют только в памяти и обрабатываются X.dll (ядром Emotet), работающим в Rundll32.exe.

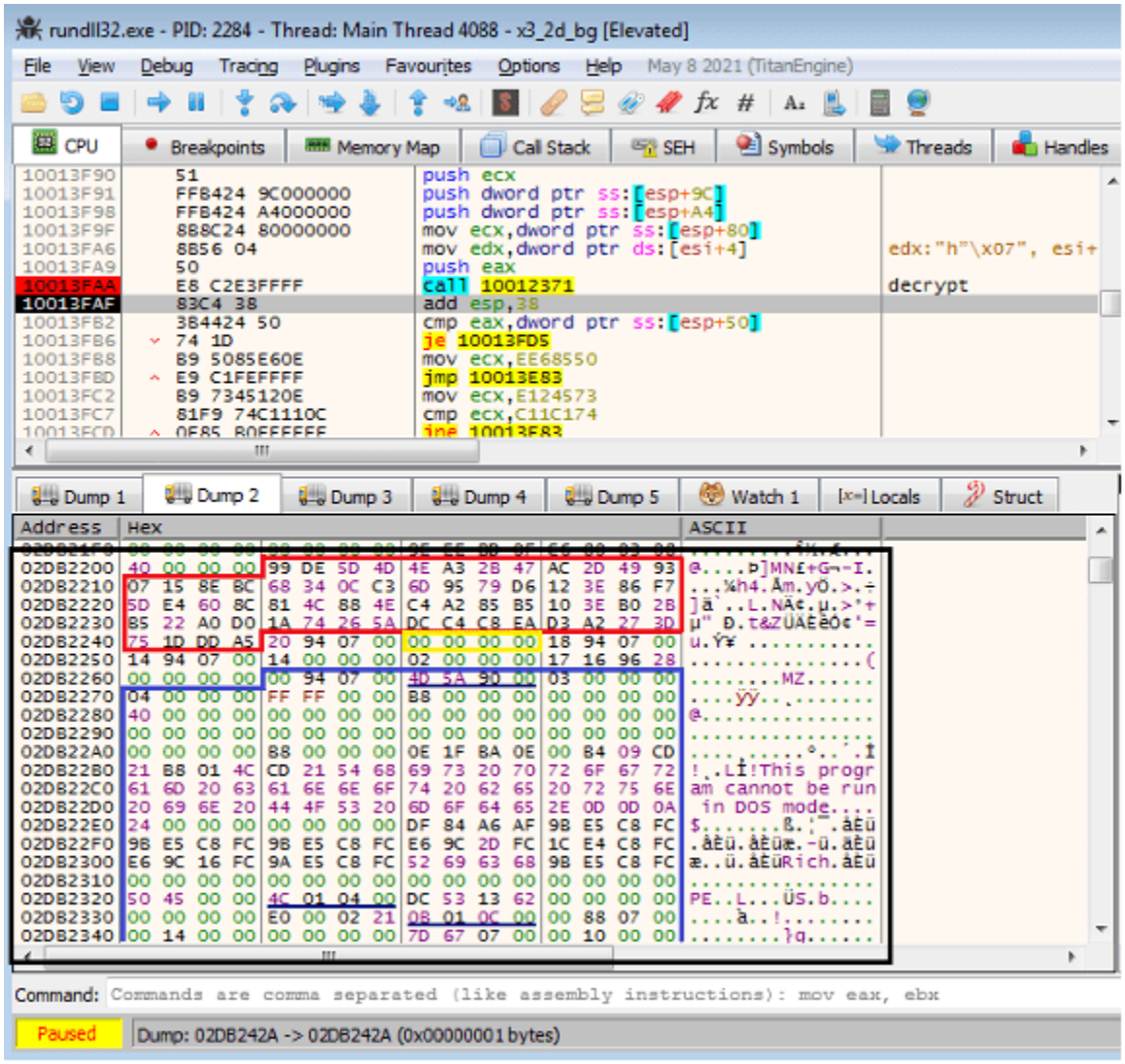

Рисунок 1.1 – Расшифрованный модуль в пакете

Рисунок 1.1 — это скриншот кода и памяти X.dll. Внизу показан ответный пакет C2, только что расшифрованный в памяти вызовом функции 10012371. Обращение к рисунку 5.3 в части I этой серии поможет вам понять структуру пакета.
Поле, отмеченное красным, — это данные проверки (99 DE … DD A5), подписанный хэш остальных данных пакета. Следующее двойное слово **0x00000000,** отмеченное желтым цветом, является флагом, который сообщает Emotet, как запустить ответный модуль. 0x00 говорит ему выполнить модуль во вновь созданном потоке. Бинарный блок синего цвета — это модуль. Он начинается с размера модуля, в данном примере 0x79400, а остальная часть — это двоичные данные модуля (4D 5A 90 00 …).
Emotet должен проверить расшифрованные данные, как показано на рисунке 1.1, используя данные проверки 40H.
Затем он развертывает полученный модуль в памяти и готовится к его выполнению. Затем он вызывает свою точку входа во вновь созданном потоке. В этом посте этот

модуль будет называться «потоковым модулем». Его основной целью является извлечение и выполнение последнего функционального модуля, который крадет конфиденциальные данные с устройства жертвы, и отправка украденных данных на его C2-сервер, который будет обсуждаться позже в этом анализе. На рис. 1.2 показано, где ASM-код функции потока вызывает точку входа развернутого модуля потока.

```
10009137 thread_fun   proc near               ; DATA XREF: sub_1001C0DC+3B1↓o
10009137
10009137 var_20       = dword ptr -20h
10009137 var_1C       = dword ptr -1Ch
10009137 var_18       = dword ptr -18h
10009137 var_14       = dword ptr -14h
10009137 var_10       = dword ptr -10h
10009137 var_C        = dword ptr -0Ch
10009137 var_8        = dword ptr -8
10009137 var_4        = dword ptr -4
10009137 arg_0        = dword ptr  8
10009137
10009137              push    ebp
10009138              mov     ebp, esp
1000913A              sub     esp, 20h
                      ●●●
1000917A              push    0
1000917C              push    1
1000917E              push    dword ptr [esi+28h]
10009181              call    dword ptr [esi+1Ch]
10009184              test    eax, eax
10009186              jz      short loc_100091C2
10009188              mov     eax, [ebp+var_8]
1000918B              mov     eax, [ebp+var_4]
1000918E              call    get_victim_id
10009193              mov     [ebp+var_20], eax
10009196              mov     eax, [esi+0Ch]  ;
10009199              mov     [ebp+var_14], eax
1000919C              mov     eax, [esi+18h]  ;
1000919F              mov     [ebp+var_10], eax
100091A2              mov     eax, [esi+10h]
100091A5              mov     [ebp+var_C], eax
100091A8              lea     eax, [ebp+var_20]
100091AB              push    eax
100091AC              push    10h
100091AE              push    dword ptr [esi+28h] ;; The thread-module's base address.
100091B1              mov     [ebp+var_1C], offset dword_10001060 ; ;;  L"ECK1 "
100091B8              mov     [ebp+var_18], offset dword_10001000 ; L"ECS1 "
100091BF              call    dword ptr [esi+1Ch] ; The thread-module's entrypoint.
                      ●●●
100091C5              mov     esp, ebp
100091C7              pop     ebp
100091C8              retn    4
100091C8 thread_fun   endp
```

Рисунок 1.2 – Функция потока Emotet для вызова точки входа модуля потока

Thread-Module — выполняет очистку процесса

Модуль-поток приступает к расшифровке PE-файла, последнего функционального модуля, из своего раздела .text в память. Чтобы выполнить этот модуль, он выполняет очистку процесса. Это делается путем копирования файла Windows «certutil.exe» из «%Windir%\SysWOW64\certutil.exe» или «%Windir%\system32\certutil.exe» в папку «%temp%». Затем он переименовывает его в случайное имя файла, например «uvbubqj.exe». Далее поток-модуль создает с этим файлом приостановленный процесс.

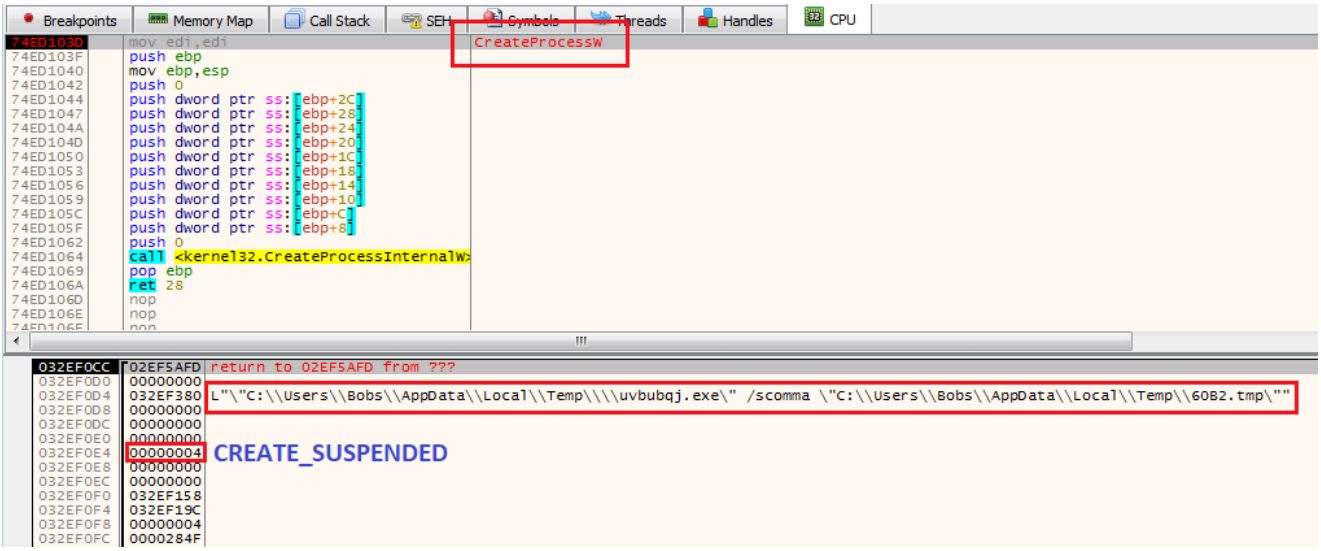

Рисунок 2.1. Вызов API CreateProcessW() для создания приостановленного процесса

Как видно из строки командной строки на рис. 2.1, «uvbubqj.exe» — это скопированный «certutil.exe», «/scomma» и последующий временный файл — «C:\Users\Bobs\AppData\Local\Temp». \60B2.tmp» — параметры процесса. Имя временного файла создается путем вызова API GetTempFileNameW(). Путь временного файла «60B2.tmp» считывается функциональным модулем и используется для сохранения украденной информации. Шестой аргумент CreateProcessW() — 0x00000004, который является флагом создания, указывающим «CREATE_SUSPENDED», с помощью которого CreateProcessW() создает процесс и переходит в состояние приостановки. Затем он вызывает группу API, таких как GetThreadContext(), VirtualAllocEx(), ReadProcessMemory(), WriteProcessMemory() и т. д., чтобы внедрить окончательный функциональный модуль в память нового процесса. API SetThreadContext() вызывается позже, чтобы установить новый регистр EIP процесса, указывающий на точку входа функционального модуля, который вызывается после вызова API ResumeThread(). После этого модуль-поток начинает отслеживать временный файл в цикле, пока он не будет создан с украденной информацией с устройства жертвы.

Глядя на функциональные модули

В приведенном выше анализе я объяснил, как модуль C2 загружается и выполняется на устройстве жертвы.
Сервер C2 может возвращать множество модулей, каждый из которых проходит тот же процесс, что и описанный выше. У них будет модуль потока, они будут работать в своем потоке и выполнять собственный процесс.
Я получил три модуля C2. Я подробно расскажу о том, как они работают на устройстве жертвы, в следующих разделах.

Module1 — Кража учетных данных из браузеров жертвы

Этот модуль защищает самораспаковывающийся пакер. Он расшифровывает PE-файл при запуске, переопределяет существующий код «certutil.exe», а затем запускает его. Распакованный PE-файл представляет собой бесплатное программное обеспечение под названием «WebBrowserPassView» , разработанное NirSoft. Он был разработан как инструмент для восстановления пароля, но злоумышленники использовали его для кражи учетных данных жертвы. Пользовательский интерфейс отображает сохраненные учетные данные, хранящиеся в нескольких веб-браузерах.

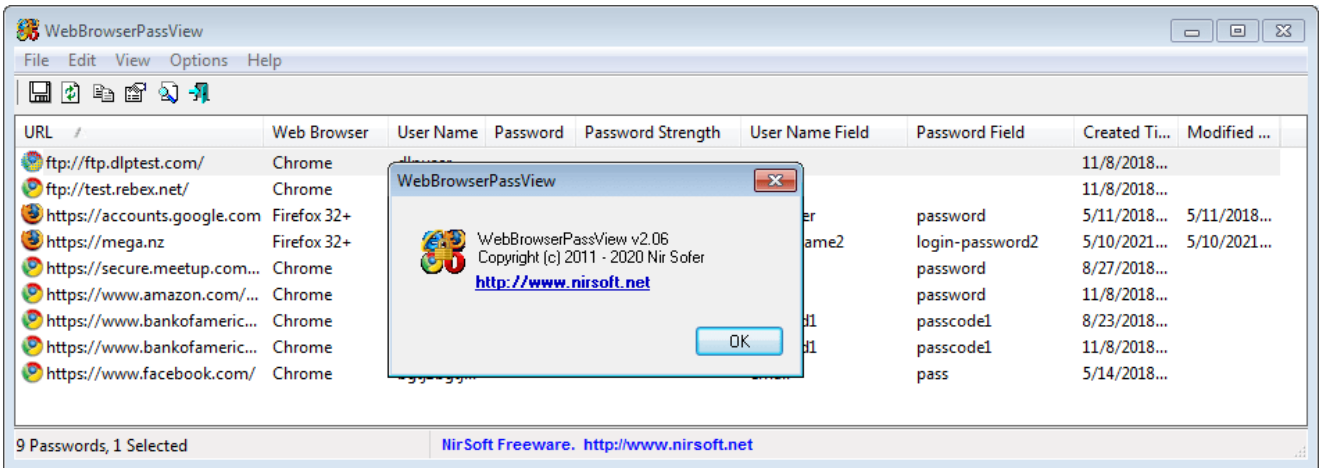

Рисунок 3.1 – Открытие модуля WebBrowserPassView

На рис. 3.1 показано, как выглядит этот модуль, когда я открываю его в своей тестовой среде. Этот вариант Emotet использует WebBrowserPassView v2.06.
Его поток-модуль передает параметры командной строки, такие как «/scomma C:\Users\Bobs\AppData\Local\Temp\7B3C.tmp», в процесс, который может переключить WebBrowserPassView в режим без окна и сохранить полученные учетные данные в данный временный файл.
Из его кода я узнал, что он может собирать учетные данные из различных веб-браузеров:
Microsoft IE, Microsoft Edge, Google Chrome, Mozilla Firefox, Opera, Apple

Safari, SeaMonkey, Yandex, Vivaldi, Waterfox и все другие браузеры на базе Chromium.

Украденные учетные данные содержат следующую информацию:

• URL-адрес: URL-адреса, для которых сохраняются учетные данные.

• Веб-браузер: имя браузера, в котором хранятся учетные данные.

• Имя пользователя, пароль: учетные данные

• Надежность пароля: надежный или слабый

• Поле имени пользователя: введите имя элемента управления в поле имени пользователя.

• Поле пароля: строка, введенная в поле пароля.

• Время создания: когда оно было сохранено

• Время изменения: время обновления учетных данных.

• Имя файла: из какого файла были украдены учетные данные.

Все учетные данные сохраняются во временном файле.

Module2 - Кража контактной информации электронной почты

Этот модуль крадет контакты электронной почты своей жертвы из их папок электронной почты в Microsoft Outlook, просматривая электронные письма жертвы одно за другим. Он хранит собранную контактную информацию в двусвязной цепочке. На рис. 4.1 показан один контакт электронной почты, полученный из сообщения электронной почты в моей тестовой учетной записи Outlook, который затем был добавлен в двусвязную цепочку, как показано внизу. Собранные данные показывают имя человека и адрес электронной почты отправителя электронной почты. В этом примере он собрал «Outlook» и «outlook@email2.office.com» из отображаемого сообщения электронной почты.

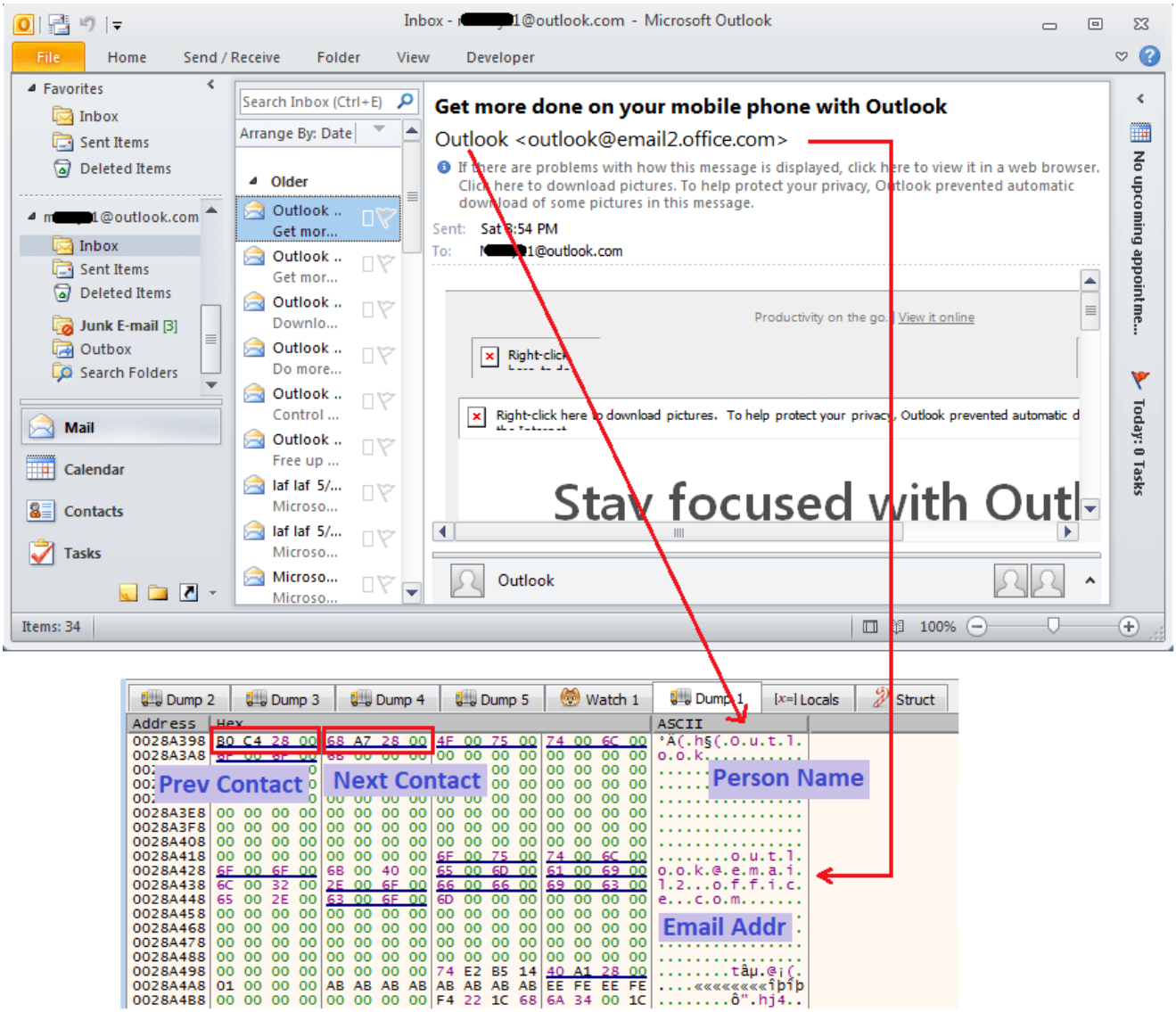

Рисунок 4.1 – Один украденный контакт в двусвязной цепочке

Этот модуль перечисляет все собранные электронные письма и помещает уникальную контактную информацию электронной почты в двусвязную цепочку. Для сбора данных Outlook необходимо вызвать несколько API, включая MAPIInitialize(), MAPILogonEx() и MAPIFreeBuffer(), а также создать некоторые COM-объекты, вызвав API CoCreateInstance(), например OlkAccountManager и OlkMail.
Наконец, он извлекает эти контакты электронной почты из связанной цепочки один за другим и сохраняет их во временный файл, полученный из параметра командной строки. На рис. 4.2 показан снимок экрана временного файла, в данном примере «%temp%\6827.tmp», вместе с собранными контактами электронной почты.

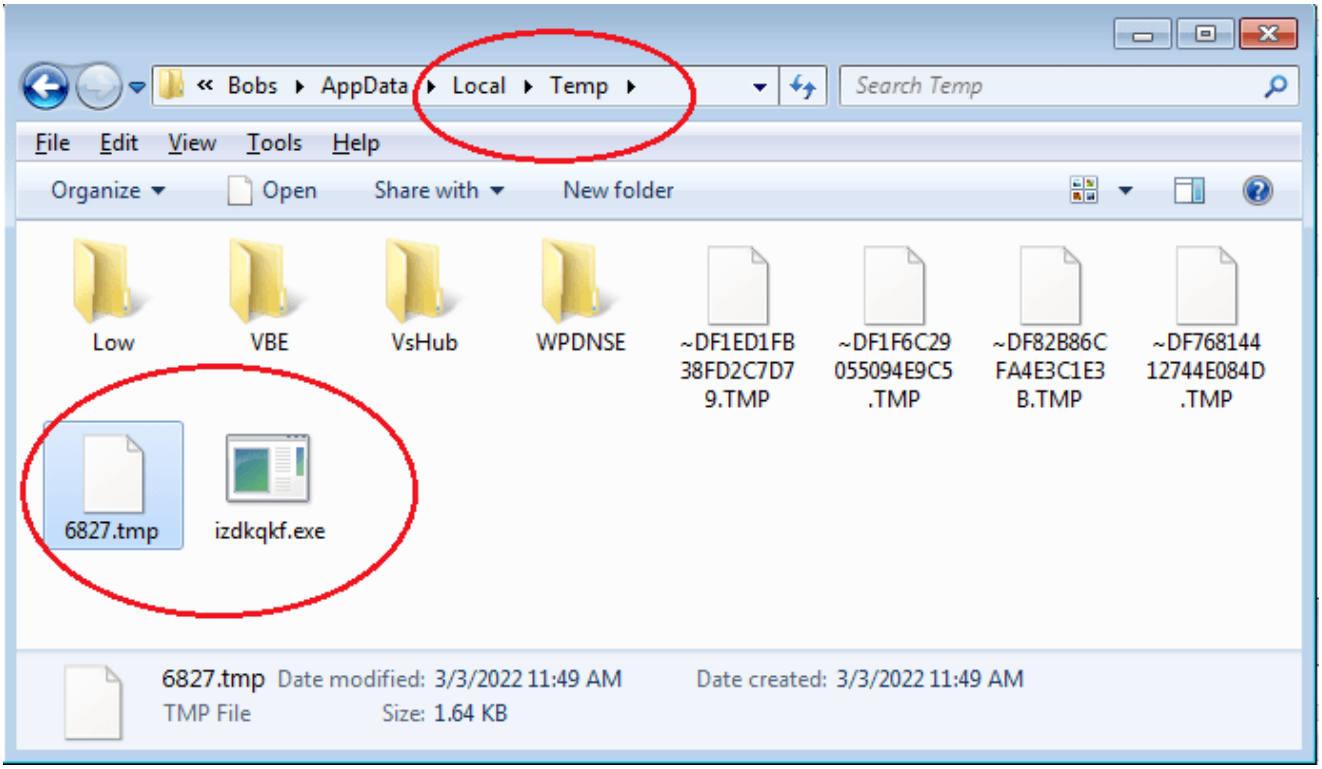

Рисунок 4.2 – Временный файл с украденной контактной информацией электронной почты

Module3 - Кража настроек учетной записи почтовых клиентов жертвы

Этот функциональный модуль фокусируется на краже настроек учетной записи электронной почты жертвы и учетных данных из их почтовых клиентов. Это также модуль, защищенный упаковщиком, поэтому он делает то же самое, что и Module1, когда вызывается его точка входа.
Согласно моему анализу, распакованный PE-файл представляет собой EXE-файл, который является еще одним бесплатным программным обеспечением от NirSoft под названием « Mail PassView ». Первоначально он был разработан как небольшой инструмент для восстановления пароля для почтовых клиентов. Emotet использует последнюю версию — v1.92. На рис. 5.1 показан снимок экрана этого программного обеспечения, работающего в моей тестовой среде.

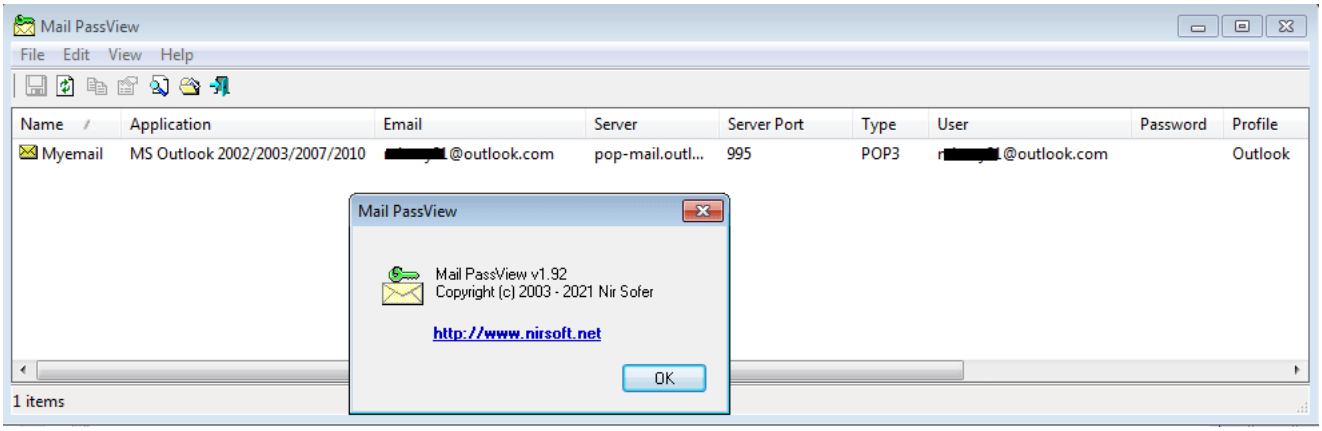

Рисунок 5.1. Открытие Mail PassView в моей тестовой среде

Изучив его код и постоянные строки, мы узнали, что он может получать настройки учетной записи электронной почты и учетные данные от следующих почтовых клиентов или других клиентов, которые могут сохранять учетные данные электронной почты:

Mozilla Thunderbird, Eudora, Microsoft Outlook, Microsoft Outlook Express, Почта Windows, MSNMessenger, Почта Windows Live, Групповая почта, IncrediMail, Yahoo! Почта Yahoo! Messenger, Hotmail, Google Desktop и Google Talk.

Он собирает параметры и учетные данные как из системного реестра, так и из локальных файлов конфигурации этих почтовых клиентов. На рис. 5.2 показан фрагмент кода ASM из общей функции, в которой предопределено множество имен значений.

Программное обеспечение неоднократно считывает имя пользователя, адрес сервера, порт сервера и аналогичную информацию из системного реестра через эти имена значений в подразделах « HKCU\Software\Microsoft\Internet Account Manager\Accounts » и « HKCU\Software\Microsoft\Office \Outlook\OMI Account Manager\Accounts », где хранятся настройки и учетные данные для Microsoft Outlook и Microsoft Outlook Express.

```
0040244A
0040244A          push    ebp
0040244B          lea     ebp, [esp-6Ch]
0040244F          sub     esp, 480h
00402455          push    ebx
00402456          push    esi
00402457          push    edi
00402458          mov     [ebp+6Ch+var_20], offset aPop3UserName ; "POP3 User Name"
0040245F          mov     [ebp+6Ch+var_1C], offset aImapUserName ; "IMAP User Name"
00402466          mov     [ebp+6Ch+var_18], offset aHttpmailUserNa ; "HTTPMail User Name"
0040246D          mov     [ebp+6Ch+var_14], offset aSmtpUserName ; "SMTP USer Name"
00402474          mov     [ebp+6Ch+var_50], offset aPop3Server ; "POP3 Server"
0040247B          mov     [ebp+6Ch+var_4C], offset aImapServer ; "IMAP Server"
00402482          mov     [ebp+6Ch+var_48], offset aHttpmailServer ; "HTTPMail Server"
00402489          mov     [ebp+6Ch+var_44], offset aSmtpServer ; "SMTP Server"
00402490          mov     [ebp+6Ch+var_30], offset aPop3 ; "POP3"
00402497          mov     [ebp+6Ch+var_2C], offset aImap ; "IMAP"
0040249E          mov     [ebp+6Ch+var_28], offset aHttpmail ; "HTTPMail"
004024A5          mov     [ebp+6Ch+var_24], offset aSmtp ; "SMTP"
004024AC          mov     [ebp+6Ch+lpValueName], offset aPop3Port ; "POP3 Port"
004024B3          mov     [ebp+6Ch+var_3C], offset aImapPort ; "IMAP Port"
004024BA          mov     [ebp+6Ch+var_38], offset aHttpmailPort ; "HTTPMail Port"
004024C1          mov     [ebp+6Ch+var_34], offset aSmtpPort ; "SMTP Port"
004024C8          mov     [ebp+6Ch+var_10], offset aPop3SecureConn ; "POP3 Secure Connection"
004024CF          mov     [ebp+6Ch+var_C], offset aImapSecureConn ; "IMAP Secure Connection"
004024D6          mov     [ebp+6Ch+var_8], offset aHttpmailSecure ; "HTTPMail Secure Connection"
004024DD          mov     [ebp+6Ch+var_4], offset aSmtpSecureConn ; "SMTP Secure Connection"
004024E4          xor     ebx, ebx
004024E6
004024E6 loc_4024E6:      ; CODE XREF: sub_40244A+1CA↓j
004024E6          push    88h             ; Size
```

Рисунок 5.2 – Определенные значения-имена для чтения из системного реестра

На этот раз строка параметра командной строки для этого программного обеспечения имеет вид «/scomma C:\Users\Bobs\AppData\Local\Temp\8042.tmp», где «/scomma» позволяет процессу запускаться без окна и сохранять полученные данные информацию во временный файл.

Thread-Module — отправка украденных данных

Когда функциональные модули работают для кражи конфиденциальных данных, модуль потока продолжает отслеживать временный файл до тех пор, пока он не будет создан с украденной информацией. Затем он загружает украденные данные из временного файла в память, а затем удаляет файл. Перед отправкой украденных данных на сервер C2 он сжимает данные и шифрует их.

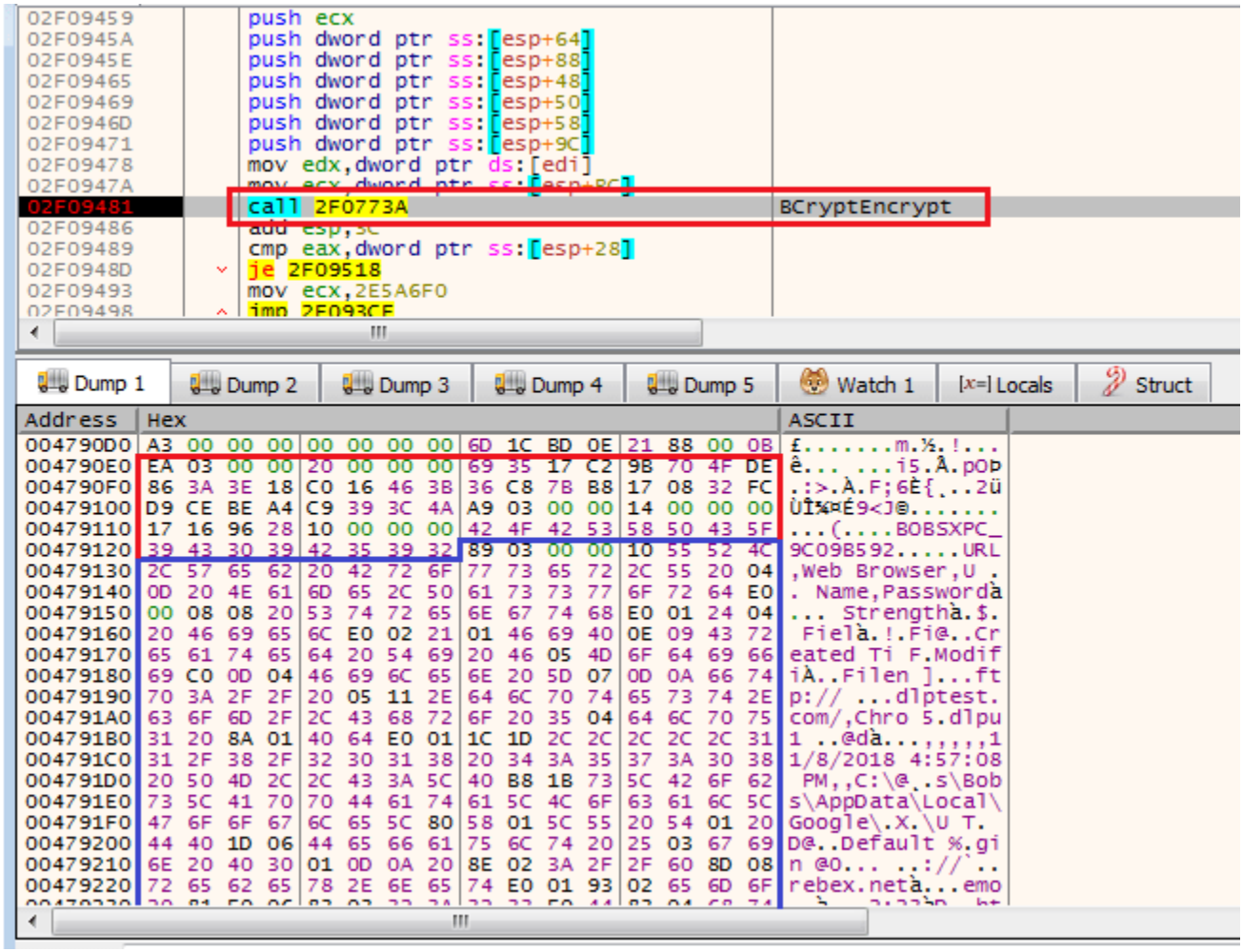

Рисунок 6.1. Вызов BCryptEncrypt() для шифрования украденных данных

В этом примере, показанном на рис. 6.1, он собирался вызвать API BCryptEncrypt() для шифрования пакета, который начинается с 4790E0. Раздел, обведенный красным, подобен заголовку пакета. Он содержит тип пакета (0x3EA), который сообщает серверу C2, какие данные находятся в пакете, хеш-код sha256 (69 35 ... 3C 4A) данных, идентификатор модуля (0x14), а также идентификатор жертвы. . Последующие данные, отмеченные синим цветом, начинаются с размера данных (0x398) следующих данных (от 10 55 52 4C до конца), которые являются сжатыми учетными данными веб-браузера.
Этот поток-модуль использует одиннадцать C2-серверов для получения данных, украденных с устройства жертвы. IP-адрес и порты этих серверов C2 зашифрованы в памяти и расшифрованы перед отправкой украденных данных. Три загруженных модуля имеют одинаковый список серверов C2, который можно найти в разделе «IOC» в конце этого анализа.

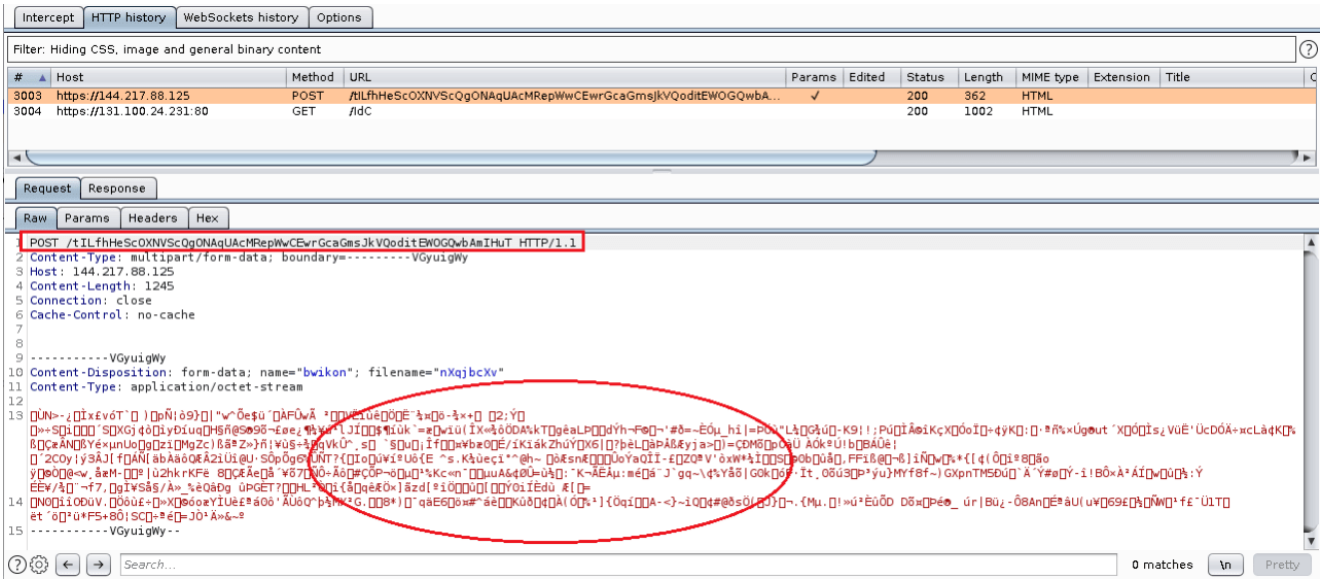

Рисунок 6.2 – Отображение перехваченного пакета на сервер C2 с зашифрованными данными

На рис. 6.2 показан снимок экрана прокси-инструмента, показывающий, как пакет с конфиденциальными данными украденной жертвы отправляется на ее C2-сервер. Он использует метод HTTP Post со случайным URL-адресом для отправки украденных данных в теле, которое состоит из экспортированного ключа длиной 40H в начале и зашифрованных данных, как показано на рисунке 6.2. Сервер C2 может расшифровать отправленные данные с помощью экспортированного ключа 40H.

Заключение

Во второй части этого анализа я начал с полученного пакета модуля от сервера C2 и объяснил структуру пакета. Далее я показал, как выполняется модуль (thread-module) во вновь созданном потоке. Затем мы рассмотрели, как модуль потока выполняет процесс очистки для выполнения функциональных модулей.
Обсуждая три полученных модуля, я подробно остановился на том, какие данные Emotet может украсть с устройства жертвы, например контактную информацию электронной почты из учетной записи электронной почты жертвы, настройки учетной записи электронной почты, учетные данные из почтового клиента жертвы и учетные данные, сохраненные в широкий выбор веб-браузеров.
Наконец, возвращаясь к модулю потока, Emotet считывает украденную информацию из заданных временных файлов. Затем он сжимает и шифрует данные, которые в конечном итоге отправляются с использованием метода HTTP Post на сервер C2.

Перевод вот ЭТОЙ статьи