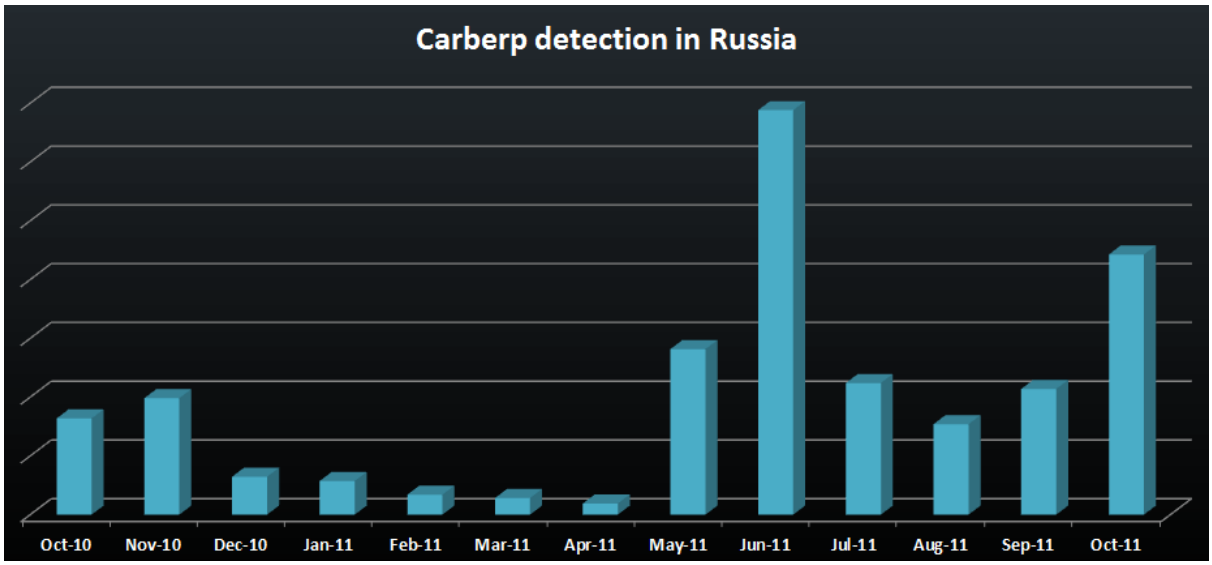
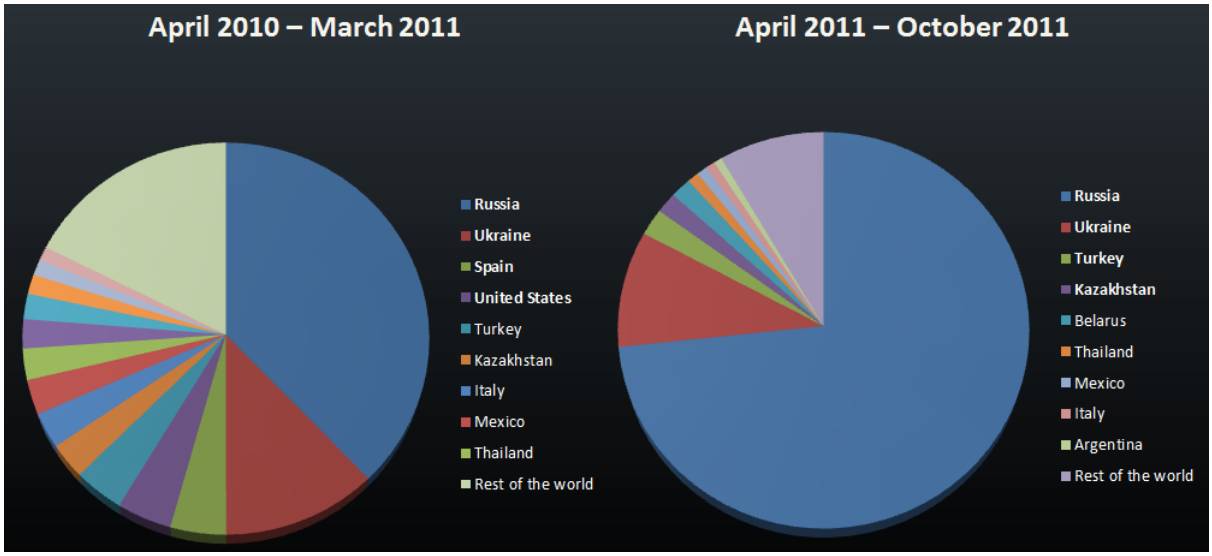
Банковский троянец Carberp приобрел буткит-функционал

Сегодня мы публикуем часть нашего расследования, посвященного вредоносной программе Carberp, которая направлена на системы дистанционного банковского обслуживания (ДБО).

В этом месяце нам удалось собрать интересную информацию о новой модификации троянца из семейства Win32/TrojanDownloader.Carberp. Это одна из самых распространенных вредоносных программ в России, которая занимается хищением финансовых средств в системах удаленного банковского обслуживания и нацелена, в первую очередь, на атаки компаний, которые осуществляют ежедневно большое количество денежных транзакций. Мы уже рассказывали об этом троянце в нашей презентации “Cybercrime in Russia: Trends and issues” (http://go.eset.com/us/resources/white-papers/CARO_2011.pdf). Разработчики данного вредоносного ПО - одна из самых активных киберпреступных групп на территории России и близлежащих стран, которая занимается кражей денежных средств в RBS. Первые инциденты с троянцем Carberp были зафиксированы нами в начале 2010 года, а пик его активности в этом году приходится на конец весны и лето, причем с началом осени опять увеличилось число инцидентов, связанных с этой программой.



Абсолютным лидером по количеству зафиксированных инцидентов является Россия, что подтверждается нашей статистикой по регионам:



Денежные прибыли киберпреступной группы исчисляются миллионами долларов еженедельно, и, как показывает наш мониторинг, злоумышленники не скупятся вкладывать деньги в развитие своих вредоносных технологий. В частности, это показывает эволюция дропера Hodprot (http://go.eset.com/us/resources/white-papers/Hodprot-Report.pdf), который был замечен в установках Carberp, RDPdoor и Sheldor (http://blog.eset.com/2011/01/14/sheldor-shocked). Причем для открытия бекдора в системе и осуществления злоумышленниками денежных транзакций вручную Carberp тоже может устанавливать RDPdoor. На данный момент нами установлено, что эти вредоносные программы являются результатом деятельности одной киберкриминальной группы.

В начале текущего года на различных закрытых киберкриминальных форумах стала появляться информация о продаже билдера буткита, не обнаруживаемого антивирусными средствами. Стоимость этого билдера исчислялась несколькими десятками тысяч долларов и казалась фантастической в сравнении, например, с ценами на те же SpyEye и Zeus.

Ring0 bundle (Zerokit) for control million-strong botnet

Goto page 1, 2, 3, 4 Next

Post Reply darkode.com Forum Index » Projects

View previous topic
View next topic

Ring0 bundle (Zerokit) for control million-strong botnet

Author	Message
ring0 Joined: 21 May 2011 Posts: 12 Rep: 1752	QUOTE • Ring0 bundle (Zerokit) for control million-strong botnet I want to introduce new crazy **ring0 bundle** (**Zerokit** or **0kit**) for control million-strong botnet. Breaking down **all** nowadays-existing firewall with **full network blocking** (bypassing in ring0). Existence of the bundle is **not detected** by any of the antiviruses (the list http://www.matousec.com/projects/proactive-security-challenge/results.php), antirootkit-utilities (Tuluka, GMER, RKU, RootkitRevealer) also see nothing. Features: - Start of *.exe, *.dll (*.dll is in a pre-alpha stage) and shellcodes in a context of the chosen process. - Start of files from a disk and from the memory* (start from memory is in a pre-alpha stage). - Start of files with specified priveleges: CurrentUser and NT SYSTEM/AUTHORITY. - Granting the protected storehouse** for off-site (your) ring3-solutions for permanent existence in the system without need of crypt. - Survivability of the bundle, down to a reinstallation of the system. - All the components are stored outside of a file system and are invisible to OS. - Intuitively clear interface of admin-panel. - Protection against the abstraction of Admin Panel. - Impossibility of detection of the bundle in the working system by any of known AV/rootkit scanner, owing to the use of author's technologies of concealment. The unique opportunity of detection exists only at loading with livecd or scanning of a disk from the other computer. Thus the opportunity of detection is also extremely improbable, as own algorithms of a mutation are used. ** Start of a file from the memory allows to bypass all modern proactive protection and AV-scanners, that is, there is no necessity to crypt a file.* *** Protected storehouse is the original ciphered file system in which the certain quantity of files which will be started from the memory at each start of the OS can be stored.* The bundle consists of: - **Bootkit.** It is responsible for the start of the basic modules at a stage of loading of OS. - **Driver.** It is responsible for all infrastructure and implements componental business-logic on the basis of so-called mod (functional unit). That is, the driver is not a legacy driver (monolithic), and consists of the set of mods that allows to operate the bundle with maximum of flexibility, and to protect (hard to reverse), update and expand it. - **Dropper.** At the current moment it brake out all machines with the patches till January, 8th, 2011, except for XP x32/x64 where reloading is initiated. If the systems distinct from XP have latest updates reloading is initiated as well. - User friendly Admin Panel.

Сколько всего киберкриминальных групп успело приобрести этот инструментарий - нам не известно. Но у нас был в наличии образец установщика буткит-части в скомпилированном виде, который грузил драйвер заглушку. Нам удалось обнаружить идентичность этого кода с некоторыми другими вредоносными программами. На данный момент о его использовании известно только в бутките Rovnix (http://blog.eset.com/2011/08/23/hasta-la-vista-bootkit-exploiting-the-vbr), который первым стал массово применять этот функционал. И с конца лета началось тестирование (распространение в очень ограниченном количестве) специальной версии Carberp с буткит-частью. На то, что бот работает в режиме тестирования, указывало два фактора: большое количество отладочных сообщений о процессе установки и очень маленькая распространенность. Нам удалось получить доступ к логам на командном центре управления ботами с буткит-частью и подтвердить факт активного процесса тестирования:

Total bots: 2831

Sort
Status
Step
Alias
Other
Del

ID	step	info	status	data
TEST_BK_KIT_EXPLORER0D9493DFECAE8C4B0	6	BkInstall	FALSE	0000-00-00 00:00:00
TEST_BK_KIT_EXPLORER08D7BD1230A905D00	6	BkInstall	FALSE	0000-00-00 00:00:00
123213oob	1	infa	false	0000-00-00 00:00:00
TEST_BK_EX_MY_DRV0F1B889AC4F21B5CA	6	BkInstall	FALSE	0000-00-00 00:00:00
TEST_BK_EX_MY_DRV0049C4497DE79EC77	6	BkInstall	FALSE	0000-00-00 00:00:00
TEST_BK_EX_MY_DRV082A52B2218EEED1A	6	BkInstall	FALSE	0000-00-00 00:00:00
TEST_BK_EX_MY_DRV06F0743BC19E94740	6	BkInstall	FALSE	0000-00-00 00:00:00
TEST_BK_EX_MY_DRV0DA631E2FA5B562AF	6	BkInstall	FALSE	0000-00-00 00:00:00
TEST_BK_EX_MY_DRV079943F8A64F9587B	6	BkInstall	FALSE	0000-00-00 00:00:00
TEST_BK_EX_MY_DRV09A01A1B010A8035A	6	BkInstall	FALSE	0000-00-00 00:00:00
TEST_BK_EX_MY_DRV07AA547C0940C1901	3	BkInstall0 GetLastError = 0	FALSE	0000-00-00 00:00:00
TEST_BK_EX_ORIG_DRV0B61FDB428F96A87B	6	BkInstall	FALSE	0000-00-00 00:00:00
TEST_BK_EX_ORIG_DRV0AE10F7A3602E42CB	6	BkInstall	FALSE	0000-00-00 00:00:00
TEST_BK_EX_ORIG_DRV06627C6A2AB3A2480	1	IsUserAdmin	FALSE	0000-00-00 00:00:00
TEST_BK_EX_ORIG_DRV0523F20AD27008003	6	BkInstall	FALSE	0000-00-00 00:00:00
TEST_BK_EX_ORIG_DRV03E797730D59441E7	6	BkInstall	FALSE	0000-00-00 00:00:00
TEST_BK_EX_ORIG_DRV0F7988F6217265D14	1	probapera	false	0000-00-00 00:00:00
TEST_BK_EX_ORIG_DRV0F7988F6317265D14	1	probapera	false	0000-00-00 00:00:00
TEST_TEST_TEST0123324234243	1	infa	false	0000-00-00 00:00:00
TEST_BK_EX_CHANGE_DRV01E6A389EE0D306DA	2	SetSystemPrivileges	FALSE	0000-00-00 00:00:00
TEST_BK_EX_CHANGE_DRV08C893A82AB121144	6	BkInstall	FALSE	0000-00-00 00:00:00
TEST_BK_EX_CHANGE_DRV074B2240F14F7F098	6	BkInstall	FALSE	0000-00-00 00:00:00
TEST_BK_EX_CHANGE_DRV0018A1BBAC95DCF46	2	SetSystemPrivileges	FALSE	0000-00-00 00:00:00
TEST_BK_EX_CHANGE_DRV0143930074B642759	6	BkInstall	FALSE	0000-00-00 00:00:00
TEST_BK_EX_CHANGE_DRV0598877EB08A14360	6	BkInstall	FALSE	0000-00-00 00:00:00
TEST_BK_EX_CHANGE_DRV0D8781E848009A04A	6	BkInstall	FALSE	0000-00-00 00:00:00
TEST_BK_EX_CHANGE_DRV05910FAB2AB121144	6	BkInstall	FALSE	0000-00-00 00:00:00
TEST_BK_EX_CHANGE_DRV09FC9B32DCEBACF5A	6	BkInstall	FALSE	0000-00-00 00:00:00
TEST_BK_EX_CHANGE_DRV039034BD2E81688D0	6	BkInstall	FALSE	0000-00-00 00:00:00
TEST_BK_EX_CHANGE_DRV0AC2F4C7B405B2000	6	BkInstall	FALSE	0000-00-00 00:00:00
TEST_BK_EX_CHANGE_DRV0E75B71B1CF9C074E	6	BkInstall	FALSE	0000-00-00 00:00:00
TEST_BK_EX_CHANGE_DRV0804FAAA06CB8B686	6	BkInstall	FALSE	0000-00-00 00:00:00
TEST_BK_EX_CHANGE_DRV0AC37DCBF566138A1	6	BkInstall	FALSE	0000-00-00 00:00:00
NEW_BK_TEST012B7B297A8FC6244	2	SetSystemPrivileges	FALSE	0000-00-00 00:00:00
NEW_BK_TEST0B6424B774E7188FC	6	BkInstall	FALSE	0000-00-00 00:00:00
NEW_BK_TEST0A29E1011ACCF989B	6	BkInstall	FALSE	0000-00-00 00:00:00
NEW_BK_TEST099E961A9D26824C0	6	BkInstall	FALSE	0000-00-00 00:00:00
NEW_BK_TEST0084B77CA30C0481F	3	BkInstall0 GetLastError = 0	FALSE	0000-00-00 00:00:00
NEW_BK_TEST_CHECKED0809EB7F457A58CC6	6	BkInstall	FALSE	0000-00-00 00:00:00
NEW_BK_TEST_CHECKED089583D04428F269B	6	BkInstall	FALSE	0000-00-00 00:00:00
NEW_BK_TEST_CHECKED0DC3B31D927AC1529	6	BkInstall	FALSE	0000-00-00 00:00:00

(C&C для тестовой версии бота)

Большое количество отладочных сообщений было найдено и в установщике новой версии Carberp:

```
_IsWow64Process@4•
VBR•
\PHYSICALDRIVE0•
\PHYSICALDRIVE0•
BKSETUP: Payload of %u bytes successfully written at sector %x.
\Device\Harddisk0\Partition%u•
\Device\Harddisk0\Partition%u•
NTPS    •
BKSETUP_%04x: BK setup dll version 2.1.
BKSETUP_%04x: Attached to a 32-bit process at 0xxx.
BKSETUP_%04x: Detached from a 32-bit process.
{%08X-%04X-%04X-%04X-%08X%04X}•
IsWow64Process•
KERNEL32.DLL•
open•
%lu.bat•
"%s"•
attrib -r -s -h%1
:klabel
del %1
if exist %1 goto klabel
del %0
Software\Classes\CLSID\•
runas•
BKSETUP: Failed generating program key name.
BKSETUP: Already installed.
BKSETUP: OS not supported.
BKSETUP: Not enough privileges to complete installation.
BKSETUP: No joined payload found.
BKSETUP: Installation failed because of unknown reason.
BKSETUP: Successfully installed.
BKSETUP: Version: 1.0
BKSETUP: Started as win32 process 0xxx.
BKSETUP: Process 0xxx finished with status %u.
BKSETUP: Version: 1.0
BKSETUP: Started as win32 process 0xxx
BKSETUP: Process 0xxx finished with status %u
```

(строки внутри распакованного кода Carberp)

Сама буткит-часть осталась без существенных модификаций и практически полностью идентична с тем, что мы уже видели у Rovnix. Технические подробности о работе буткита можно увидеть на слайдах нашего доклада «Defeating x64: Modern Trends of Kernel-Mode Rootkits» (http://go.eset.com/us/resources/white-papers/Ekoparty2011_preso.pdf), который мы представили на конференции Ekoparty осенью этого года.

А вот установщик претерпел значительные модификации. Кроме установки в систему буткит-функционала Carberp теперь еще эксплуатируется ряд уязвимостей для повышения привилегий. В первую очередь, это связано с тем, что для установки буткита требуются привилегии администратора или системы. Во время атаки на корпоративных пользователей (а

именно это является целью злоумышленников), имеющих доступ к ДБО, у них может просто не быть прав на запуск исполняемого файла с высокими привилегиями, и тут уже не поможет атака с использованием методов социальной инженерии. Поэтому установщик сейчас эксплуатирует четыре разных уязвимости для повышения локальных привилегий в системе:

- MS10-073 (win32k.sys KeyboardLayout vuln)
- MS10-092 (Task Scheduler vuln)
- MS11-011 (win32k.sys SystemDefaultEUDCFont vuln)
- .NET Runtime Optimization vuln (http://osvdb.org/show/osvdb/71013)

```
  zero = 0;
  status = 0;
  osVerInfo.dwOSVersionInfoSize = 156;
  osVerInfo.dwMajorVersion = 0;
  mem_set(&osVerInfo.dwMinorVersion, 0, 0x94u);
  result = GetVersionEx(&osVerInfo);
  if ( result )
  {
    if ( osVerInfo.dwMajorVersion == 5 )        // WinXP
    {
      status = ExploitKeyboardLayoutVuln();     // MS10-073
      if ( status )
        goto NEXT_STEP;
      imBase = GetImageBaseSelf();
      if ( !CheckPE(imBase) )
      {
        size = 0;
        data = GetDataFromSection("DROPER_DLL", &size);
        if ( data )
        {
          if ( size )
          {
            hDll = _GenTempFileName();
            WriteDataInFile(hDll, data, size);
            status = BypassHIPS(hDll);          // AddPrintProvidor
            CheckName(hDll);
            zero = 0;
          }
        }
      }
      if ( status != zero )
        goto NEXT_STEP;
      exp_status = Exploit_dotNetVuln(ModFileName);// .NET Runtime Optimization Vuln
    }
    else
    {
      if ( osVerInfo.dwMajorVersion != 6 )      // Vista or Win2008
        goto NEXT_STEP;
      if ( !osVerInfo.dwMinorVersion )
      {
        if ( ExploitTaskSchedVuln(ModFileName) )// MS10-092
          status = 2;
      }
      if ( osVerInfo.dwMinorVersion != 1 )      // Win7 or Win2008 R2
        goto NEXT_STEP;
      exp_status = ExploitEUDCFontVuln();       // MS11-011
    }
    status = exp_status;
NEXT_STEP:
    result = status;
  }
  return result;
}
```

Еще одним интересным моментом работы дропера является снятие различных хуков и спласов со следующих системных функций перед непосредственным запуском буткит-установщика:

```
v1 = hash_ntdll_ZwSetContextThread;
v2 = hash_ntdll_ZwGetContextThread;
v3 = hash_ntdll_ZwUnmapViewOfSection;
v4 = hash_ntdll_ZwMapViewOfSection;
v5 = hash_ntdll_ZwAllocateVirtualMemory;
v6 = hash_ntdll_ZwWriteVirtualMemory;
v7 = hash_ntdll_ZwProtectVirtualMemory;
v8 = hash_ntdll_ZwCreateThread;
v9 = hash_ntdll_ZwOpenProcess;
v10 = hash_ntdll_ZwOpenThread;
v11 = hash_ntdll_ZwQueueApcThread;
v12 = hash_ntdll_ZwTerminateProcess;
v13 = hash_ntdll_ZwTerminateThread;
v14 = hash_ntdll_ZwResumeThread;
v15 = hash_ntdll_ZwQueryDirectoryFile;
v16 = hash_ntdll_ZwCreateProcess;
v17 = hash_ntdll_ZwCreateProcessEx;
v18 = hash_ntdll_ZwCreateFile;
v19 = hash_ntdll_ZwDeviceIoControlFile;
v20 = hash_ntdll_ZwClose;
v21 = hash_ntdll_ZwSetInformationProcess;
v23 = hash_kernel32_CreateRemoteThread;
v24 = hash_kernel32_WriteProcessMemory;
v25 = hash_kernel32_VirtualProtectEx;
v26 = hash_kernel32_VirtualAllocEx;
v27 = hash_kernel32_SetThreadContext;
v28 = hash_kernel32_CreateProcessA;
v29 = hash_kernel32_CreateProcessInternalA;
v30 = hash_kernel32_CreateProcessInternalW;
v31 = hash_kernel32_CreateFileA;
v32 = hash_kernel32_CreateFileW;
v33 = hash_kernel32_CopyFileA;
v34 = hash_kernel32_CopyFileW;
v35 = hash_kernel32_CopyFileExW;
v37 = hash_ws2_32_connect;
v38 = hash_ws2_32_send;
v39 = hash_ws2_32_recv;
v40 = hash_ws2_32_gethostbyname;
RestoreSplicing(L"ntdll.dll", &v1, 1);
RestoreSplicing(L"kernel32.dll", &v23, 1);
return RestoreSplicing(L"ws2_32.dll", &v37, 1);
```

Это своего рода обход различных песочниц и средств мониторинга, которые используют хуки в пользовательском режиме.

В процессе работы буткит кода в обход проверок цифровой подписи и других защитных механизмов загружается вредоносный драйвер. Задачей этого драйвера является внедрение полезной нагрузки в адресное пространство user-mode процессов.

После распаковки оригинального файла драйвера можно обнаружить дату сборки и версию компилятора:

Field Name	Data Value	Description
Machine	014Ch	i386®
Number of Sections	0005h	
Time Date Stamp	4EB7F565h	07/11/2011 15:12:37
Pointer to Symbol Table	00000000h	
Number of Symbols	00000000h	
Size of Optional Header	00E0h	
Characteristics	0102h	
Magic	010Bh	PE32
Linker Version	0009h	9.0

По этим данным видно, что файл был собран в начале текущего месяца, и сейчас, видимо, происходит его обкатка на стабильность работы. В драйвере после компиляции осталась интересная строка, указывающая к директории с исходными кодами на компьютере разработчиков:

```
d:\programming\commerc\c++\bootkit_archiv\bk22\kloader\Release\i386\kloader.pdb
CurrentControlSet\Services\null
```

Судя по всему, это не первая и не последняя версия для тестов, которая появилась за последнее время. Также мы не исключаем возможность того, что дальнейшей поддержкой этого кода занимается автор, который ранее выставлял его на продажу. Так как кроме Rovnix и этой версии Carberp вредоносных программ на базе данного кода нам неизвестно, и раньше низкоуровневой части не было ни в одной версии Carberp, это позволяет предположить, что у них не было разработчиков такой квалификации.

Основной задачей этого драйвера является внедрение в адресное пространство пользовательских процессов динамической библиотеки с полезной нагрузкой.

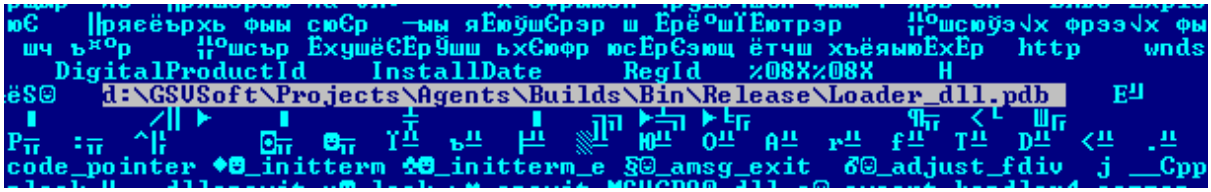
```
NTSTATUS __stdcall _DriverEntry(PDRIVER_OBJECT DriverObject, PUNICODE_STRING RegistryPath)
{
  NTSTATUS Status; // eax@1
  NTSTATUS _Status; // esi@2

  DbgPrint("DriverStartup");
  Status = InitializeInternalList(&dword_16250, 208, 0, 0);
  if ( Status >= 0 )
  {
    ExtractPayloadFromImage();
    _InitializeInternalList();
    _Status = PsSetCreateProcessNotifyRoutine(CreateProcessNotifyRoutine, 0);
    if ( _Status >= 0
      && (_Status = PsSetLoadImageNotifyRoutine(LoadImageNotifyRoutine),
          DriverObject->DriverUnload = DriverUnload,
          _Status >= 0) )
    {
      StartSystemThread();
      Status = _Status;
    }
    else
    {
      PsSetCreateProcessNotifyRoutine(CreateProcessNotifyRoutine, 1u);
      FreeInternalList(dword_16250);
      Status = _Status;
    }
  }
  return Status;
}
```

После распаковки динамической библиотеки с полезной нагрузкой можно также увидеть дату компиляции, из которой становится понятным, что она была собрана в разное время, если сравнивать с драйвером.

Field Name	Data Value	Description
Machine	014Ch	i386®
Number of Sections	0005h	
Time Date Stamp	4EB16817h	02/11/2011 15:56:07
Pointer to Symbol Table	00000000h	
Number of Symbols	00000000h	
Size of Optional Header	00E0h	
Characteristics	2102h	
Magic	010Bh	PE32
Linker Version	0009h	9.0

Внутри нашлась интересная строковая константа, указывающая на совершенно другую директорию с проектом и интересной корневой директорией с названием «GSVSoft».



Интересно, что даты компиляции и пути к директориям проектов отличаются, что косвенно говорит о том, что работали две разные команды разработчиков. Помимо этого, с именем «GSVSoft» существует вполне реальный домен (gsvsoft.ru), но делать какие-то выводы, основываясь только на этом, нельзя.

Основной функционал динамической библиотеки во многом схож с предыдущей версией Carberp. Основной задачей этого модуля является внедрение JS-скриптов на страницы с интернет банкингом, загрузка дополнительных плагинов и взаимодействие с командным центром.

Все вышенаписанное наводит на мысли о том, что текущие методы сокрытия в системе троянца Carberp не устраивают его разработчиков, и они готовы вкладывать деньги в развитие собственных технологий обхода защитного ПО. Так эволюция этой вредоносной программы за два года сделала его номером один по количеству инцидентов с клиентами российских банков, причем страдают от этого не только коммерческие организации - есть инциденты и в государственных учреждениях. С новым буткит функционалом Carberp выходит на одну ступень по сложности анализа и обнаружению с такими угрозами, как TDL4 и Rovnix.

Несмотря на то, что сейчас основной целью Carberp являются клиенты российских банков, эта ситуация может быстро измениться, так как технологических препятствий для атак на банки в других странах у данной киберкриминальной группы нет. И выход в другие регионы не происходит, вероятнее всего, из-за того, что их пока все устраивает и в таком виде. Но в ближайшем будущем они могут найти партнеров по атакам в других регионах. В прошлом году уже была зафиксирована активность Carberp в европейских странах, но потом она стала снижаться.