CIRCL » TR-25 Analysis - Turla / Pfinet / Snake

Archived: 2026-04-05 13:05:07 UTC

Overview

During the last weeks, various samples of *Uroburos* (also named *Urob*, *Turla*, *Sengoku*, *Snark* and *Pfinet*) were analyzed and reports have been published [1][2][3][4], also analyses about a suspected predecessor, *Agent.btz*, are public [5]. CIRCL analyzed an older version of *Turla*, known as a representative of the *Pfinet* malware family. The objective of this analysis is to gather additional *Indicators of Compromise* or behaviors in order to improve detection and to discover additional insights into the malware. This document is not considered a final release but a work-in-progress document.

Static Analysis

Sample A

Hashes:

Type of Hash	Hash
MD5	5b4a956c6ec246899b1d459838892493
SHA1	217b8fa45a24681551bd84b573795b5925b2573e
SHA-256	93742b415f28f57c61e7ce7d55208f71d5c4880dc66616da52f3c274b20b43b0
ssdeep	24576:D0MfCZaSyUS7YXz3aHUXXeJozanHZCfBvt9MSc99rdI+6cGHe:D02saHQXeManH81t9BONdI3VHe

VirusTotal results for sample A

AV product	Result
Bkav	W32.Clod24a.Trojan.ceee
MicroWorld-eScan	Dropped:Backdoor.Generic.252173
nProtect	Dropped:Backdoor.Generic.252173
McAfee	Artemis!5B4A956C6EC2
K7AntiVirus	Riskware (10a2c0f80)
K7GW	Trojan (00155adb1)
NANO-Antivirus	Trojan.Win64.Agent.lsivh

AV product	Result
F-Prot	W32/MalwareS.IHA
Symantec	Backdoor.Pfinet
Norman	Suspicious_Gen3.DGZV
TotalDefense	Win32/Pfinet.A
TrendMicro-HouseCall	TROJ_GEN.R27E1AH
Avast	Win32:Malware-gen
ClamAV	Trojan.Agent-126457
Kaspersky	Trojan.Win32.Genome.hitb
BitDefender	Dropped:Backdoor.Generic.252173
Agnitum	Trojan.Meredrop!A/hBhJu+uNc
Ad-Aware	Dropped:Backdoor.Generic.252173
Sophos	Mal/Generic-S
Comodo	TrojWare.Win32.Agent.czua
F-Secure	Dropped:Backdoor.Generic.252173
DrWeb	Trojan.Siggen.27969
VIPRE	Trojan.Win32.Generic!BT
AntiVir	TR/Agent.czua
TrendMicro	TROJ_GEN.R27E1AH
McAfee-GW-Edition	Artemis!5B4A956C6EC2
Emsisoft	Dropped:Backdoor.Generic.252173 (B)
Microsoft	Backdoor:WinNT/Pfinet.B
GData	Dropped:Backdoor.Generic.252173
Commtouch	W32/Risk.DWJW-7987
VBA32	Trojan.Agent2
Baidu-International	Trojan.Win32.Genome.aR
ESET-NOD32	a variant of Win32/Turla.AC
Ikarus	Trojan.Win32.Genome

AV product	Result
Fortinet	W32/Pfinet!tr
AVG	Generic16.BBMD
Panda	Trj/Hmir.F

Scanned: 2014-03-16 01:12:54 - 49 scans - 37 detections (75.0%)

File characteristics

Meta data

```
 Size:    1052672 bytes
Type:    PE32 executable (GUI) Intel 80386, for MS Windows
Date:    0x4AC5A74C [Fri Oct  2 07:10:04 2009 UTC]
EP:      0x4021bb .text 0/5
CRC:     Claimed: 0x0, Actual: 0x110f40 [SUSPICIOUS]
```

Resource entries

```
 Name      RVA      Size     Lang         Sublang             Type
------------------------------------------------------------------------------
BINARY    0xd190   0x3dc00  LANG_ENGLISH SUBLANG_ENGLISH_US  PE32 executable (DLL) (native) Intel 80386, for MS Windows
BINARY    0x4ad90  0x1d000  LANG_ENGLISH SUBLANG_ENGLISH_US  PE32 executable (DLL) (GUI) Intel 80386, for MS Windows
BINARY    0x67d90  0x21000  LANG_ENGLISH SUBLANG_ENGLISH_US  PE32 executable (DLL) (GUI) Intel 80386, for MS Windows
BINARY    0x88d90  0x1f9    LANG_ENGLISH SUBLANG_ENGLISH_US  ASCII text, with CRLF, LF line terminators
BINARY    0x88f90  0x37c00  LANG_ENGLISH SUBLANG_ENGLISH_US  PE32+ executable (DLL) (native) x86-64, for MS Windows
BINARY    0xc0b90  0x1bc00  LANG_ENGLISH SUBLANG_ENGLISH_US  PE32+ executable (DLL) (GUI) x86-64, for MS Windows
BINARY    0xdc790  0x24200  LANG_ENGLISH SUBLANG_ENGLISH_US  PE32+ executable (DLL) (GUI) x86-64, for MS Windows
```

Version info

No version information included.

Sections

```
 Name       VirtAddr     VirtSize     RawSize      Entropy
------------------------------------------------------------------------------
.text      0x1000       0x6f34       0x7000       6.582374
.rdata     0x8000       0x1fb8       0x2000       4.803196
.data      0xa000       0x26f4       0x1000       1.559595
.rsrc      0xd000       0xf3990      0xf4000      5.977919
.reloc     0x101000     0x188c       0x2000       2.462180
```

```
 SECTION 1 (.text   ):
                  virtual size                  : 00006F34 (  28468.)
                  virtual address               : 00001000
                  section size                  : 00007000 (  28672.)
                  offset to raw data for section: 00001000
                  offset to relocation          : 00000000
                  offset to line numbers        : 00000000
                  number of relocation entries  : 0
                  number of line number entries : 0
                  alignment                     : 0 byte(s)
                  Flags 60000020:
                    text only
                    Executable
                    Readable
SECTION 2 (.rdata  ):
                  virtual size                  : 00001FB8 (   8120.)
                  virtual address               : 00008000
                  section size                  : 00002000 (   8192.)
                  offset to raw data for section: 00008000
                  offset to relocation          : 00000000
                  offset to line numbers        : 00000000
                  number of relocation entries  : 0
                  number of line number entries : 0
                  alignment                     : 0 byte(s)
                  Flags 40000040:
                    data only
                    Readable
SECTION 3 (.data   ):
                  virtual size                  : 000026F4 (   9972.)
                  virtual address               : 0000A000
                  section size                  : 00001000 (   4096.)
                  offset to raw data for section: 0000A000
                  offset to relocation          : 00000000
                  offset to line numbers        : 00000000
                  number of relocation entries  : 0
                  number of line number entries : 0
                  alignment                     : 0 byte(s)
                  Flags C0000040:
                    data only
                    Readable
                    Writable
SECTION 4 (.rsrc   ):
                  virtual size                  : 000F3990 ( 997776.)
                  virtual address               : 0000D000
                  section size                  : 000F4000 ( 999424.)
                  offset to raw data for section: 0000B000
                  offset to relocation          : 00000000
                  offset to line numbers        : 00000000
                  number of relocation entries  : 0
                  number of line number entries : 0
```

```
                  alignment                    : 0 byte(s)
                  Flags 40000040:
                    data only
                    Readable
SECTION 5 (.reloc  ):
                  virtual size                 : 0000188C (   6284.)
                  virtual address              : 00101000
                  section size                 : 00002000 (   8192.)
                  offset to raw data for section: 000FF000
                  offset to relocation         : 00000000
                  offset to line numbers       : 00000000
                  number of relocation entries  : 0
                  number of line number entries : 0
                  alignment                    : 0 byte(s)
                  Flags 42000040:
                    data only
                    Discardable
                    Readable
```

Strings

The order of strings embedded in clear text in Sample A indicate that this file contains several other files, because the DOS stub (*!This program cannot be run in DOS mode.*) is present multiple times. We include interesting strings in the corresponding subsection.

Analysis - Installer

Sample A can be considered an installer or dropper. It drops files into the system and initializes the environment for production. First, it probes if a virtual disk

```
\DEVICE\IdeDrive1\
```

is present on the system. If not, the virtual disk is being created with file system *NTFS*, using *FormatEx* from Microsofts *fmifs.dll*.

```
 1int __cdecl create_virtual_disk()
 2{
 3  HMODULE hModule_fmifs.dll;
 4  int result;
 5  FARPROC FormatEx;
 6  WCHAR VirtualDisk;
 7
 8  result = 0;
 9  hModule_fmifs.dll = LoadLibraryA("fmifs.dll");
10  if ( hModule_fmifs.dll )
11  {
12    FormatEx = GetProcAddress(hModule_fmifs.dll, "FormatEx");
13    if ( FormatEx )
```

```
  {
    wsprintfW(&VirtualDisk, L"%S", "\\\\.\\IdeDrive1\\\\");
    (FormatEx)(&VirtualDisk, FMIFS_HARDDISK, L"NTFS", &gVirtualDiskName, 1, 0, FormatExCallback);
    result = gFormatExCallbackActionInfo != 0;
  }
  FreeLibrary(hModule_fmifs.dll);
}
else
{
  result = 0;
}
return result;
}
```

The presence of the malware's configuration file is tested:

```
\DEVICE\IdeDrive1\config.txt
```

If not found, it is dropped from the resource section *0x88d90*.

The following files are dropped depending on whether Windows is running in 32 bit or 64 bit.

```
 %SystemRoot%\$NtUninstallQ722833$\usbdev.sys (hidden)
\DEVICE\IdeDrive1\inetpub.dll
\DEVICE\IdeDrive1\cryptoapi.dll
```

Independently from the architecture, the file names of the dropped files are the same, but a specific version of the file is dropped according to the operating system architecture.

This is achieved by a logic similar to the following one. This is done for all files except the configuration file.

```
if ( IsWow64 )
      {
        res = create_from_resources("#162", "\\\\.\\IdeDrive1\\inetpub.dll");
        if ( last_error )
        {
          error = GetLastError();
          log(last_error, "ef1... %d, %d\n", res, error);
        }
        v29 = create_from_resources("#165", "\\\\.\\IdeDrive1\\cryptoapi.dll");
      }
```

The function *create_from_resources()* looks like:

```
int __cdecl create_from_resources(LPCSTR NameOfResource, LPCSTR lpSrc)
{
 HRSRC HRSRC;
 HGLOBAL hGlobal;
```

```
 5  DWORD SizeOfResource;
 6  HANDLE hFile;
 7  DWORD error;
 8  CHAR lpFileName;
 9  char pSecurityDescriptor;
10  DWORD NumberOfBytesWritten;
11  LPCVOID lpBuffer;
12
13  ExpandEnvironmentStringsA(lpSrc, &lpFileName, 0x104u);
14  HRSRC = FindResourceA(0, NameOfResource, "BINARY");
15  if ( !HRSRC )
16    return 0;
17  hGlobal = LoadResource(0, HRSRC);
18  if ( !hGlobal )
19    return 0;
20  lpBuffer = LockResource(hGlobal);
21  if ( !lpBuffer )
22    return 0;
23  SizeOfResource = SizeofResource(0, HRSRC);
24  hFile = CreateFileA(&lpFileName, GENERIC_WRITE, 0, 0, 2u, 0x80u, 0);
25  if ( hFile == -1 )
26  {
27    if ( last_error )
28    {
29      error = GetLastError();
30      log(last_error, "ex_fail... %d\n", error);
31    }
32    return 0;
33  }
34  WriteFile(hFile, lpBuffer, SizeOfResource, &NumberOfBytesWritten, 0);
35  CloseHandle(hFile);
36  if ( !InitializeSecurityDescriptor(&pSecurityDescriptor, 1u) )
37    return 0;
38  return SetFileSecurityA(&lpFileName, DACL_SECURITY_INFORMATION, &pSecurityDescriptor) != 0;
39}
```

Subsequently, after dropping the correct files, the malware makes itself persistent on the system and creates a service with the following parameters, which loads the file *usbdev.sys* as a kernel driver:

```
 In:           HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services:
Key:          usblink
Type:         1 (SERVICE_KERNEL_DRIVER)
Start:             1 (SERVICE_SYSTEM_START)
ErrorControl: 0 (SERVICE_ERROR_IGNORE)
Group:        Streams Drivers
DisplayName:  usblink
ImagePath:    \SystemRoot\$NtUninstallQ722833$\usbdev.sys
```

If during installation anything goes wrong, the registry keys are deleted. The files however are not.

During the installation process, extensive logging is ensuring good visibility on potential installation problems. The attacker uses english language for the logging, although he is lacking attention to detail when it comes to correct usage of the language, as the following examples demonstrate:

```
 win32 detect...                   (should be simple past)
x64 detect...                      (should be simple past)
CretaFileA(%s):           (should be CreateFileA)
Can`t open SERVICES key (that shouldn't be a backtick)
```

Language deficits are also demonstrated in other files of this collection. We show them in a separate chapter.

A list of dropped files is given in the next chapter.

Dropped files

Sample B - usbdev.sys (Resource: 101)

Hashes

Type of Hash	Hash
MD5	db93128bff2912a75b39ee117796cdc6
SHA1	418645c09002845a8554095b355f47907f762797
SHA-256	57b8c2f5cfeaca97da58cfcdaf10c88dbc2c987c436ddc1ad7b7ed31879cb665
ssdeep	3072:3B9f3bhj+FqCjAsWnQNCb/XzeQdRSFqfCeEmI/2XxjptNdjxjkMAE4E:3B9tQHWLrFfCZmI/MttB+E4

VirusTotal results for sample B

AV product	Result
Bkav	W32.Cloda11.Trojan.222a
MicroWorld-eScan	Backdoor.Generic.252173
nProtect	Trojan/W32.Agent2.252928
McAfee	Artemis!DB93128BFF29
K7GW	Trojan (0001140e1)
K7AntiVirus	Riskware (10a2c0f80)
Agnitum	Trojan.Agent2!HMPS2EOZWFE
F-Prot	W32/MalwareS.IHA

AV product	Result
Symantec	Backdoor.Pfinet
Norman	Suspicious_Gen3.DGZV
TrendMicro-HouseCall	TROJ_GEN.R27E1AH
Avast	Win32:Malware-gen
Kaspersky	Trojan.Win32.Agent2.flce
BitDefender	Backdoor.Generic.252173
Ad-Aware	Backdoor.Generic.252173
Sophos	Mal/Generic-S
F-Secure	Backdoor.Generic.252173
DrWeb	Trojan.Siggen1.51234
VIPRE	Trojan.Win32.Generic!BT
AntiVir	TR/Rootkit.Gen
TrendMicro	TROJ_GEN.R27E1AH
McAfee-GW-Edition	Artemis!DB93128BFF29
Emsisoft	Backdoor.Generic.252173 (B)
Jiangmin	Trojan/Agent.djjf
Antiy-AVL	Trojan/Win32.Agent2
Kingsoft	Win32.Troj.Agent2.(kcloud)
Microsoft	Backdoor:WinNT/Pfinet.B
GData	Backdoor.Generic.252173
Commtouch	W32/Risk.DWJW-7987
VBA32	Trojan.Agent2
Panda	Rootkit/Agent.IOO
ESET-NOD32	a variant of Win32/Turla.AC
Ikarus	Trojan.Win32.Agent
Fortinet	W32/Agent2.LDY!tr
AVG	Agent2.AHWF

AV product	Result
Baidu-International	Trojan.Win32.Agent.AFZ

Scanned: 2014-03-23 21:28:41 - 51 scans - 36 detections (70.0%)

File characteristics

Meta data

```
 Size:    252928 bytes
Type:    PE32 executable (DLL) (native) Intel 80386, for MS Windows
Date:    0x4AC48FC8 [Thu Oct  1 11:17:28 2009 UTC]
EP:      0x22d80 .text 0/5
CRC:     Claimed: 0x3e7fe, Actual: 0x3e7fe
```

Sections

```
 Name       VirtAddr     VirtSize     RawSize      Entropy
--------------------------------------------------------------------------------
.text      0x1000       0x28084      0x28200      6.325480
.basein    0x2a000      0x135        0x200        3.791369
.data      0x2b000      0x20e34      0x12600      1.335577
INIT       0x4c000      0xebc        0x1000       5.343628
.reloc     0x4d000      0x1de0       0x1e00       6.448244
```

Strings

Interesting strings:

```
 CsrClientCallServer
ExitThread
LdrGetProcedureAddress
ZwTerminateThread
\SystemRoot\system32\%s
IoCreateDevice
ModuleStart
ModuleStop
\??\%s\cryptoapi.dll
\??\%s\inetpub.dll
services.exe
iexplore.exe
firefox.exe
opera.exe
netscape.exe
mozilla.exe
msimn.exe
```

```
outlook.exe
adobeupdater.exe
```

Sample C - inetpub.dll (Resource: 102)

Hashes

Type of Hash	Hash
MD5	2145945b9b32b4ccbd498db50419b39b
SHA1	690f18810b0cbef06f7b864c7585bd6ed0d207e0
SHA-256	3de0ba77fa2d8b26e4226fd28edc3ab8448434d851f6b2b268ec072c5da92ade
ssdeep	3072:HPHvQByUS7Yqy7UKJm1Y3a3v/z61dmh9f3b/LAaulNA7:HPHqyUS7YqyIKH3aHz61Mh9jZulNC

VirusTotal results for sample C

AV product	Result
McAfee	Generic.dx!wel
K7AntiVirus	Riskware
Symantec	Backdoor.Pfinet
Norman	W32/Suspicious_Gen3.UANR
Avast	Win32:Malware-gen
eSafe	Win32.TRATRAPS
BitDefender	Backdoor.Generic.429659
F-Secure	Backdoor.Generic.429659
VIPRE	Trojan.Win32.Generic!BT
AntiVir	TR/ATRAPS.Gen
McAfee-GW-Edition	Generic.dx!wel
Emsisoft	Backdoor.SuspectCRC!IK
Antiy-AVL	Trojan/win32.agent.gen
GData	Backdoor.Generic.429659
AhnLab-V3	Backdoor/Win32.Pfinet

AV product	Result
PCTools	Backdoor.Pfinet
Ikarus	Backdoor.SuspectCRC
Panda	Trj/CI.A
Avast5	Win32:Malware-gen

Scanned: 2011-07-07 04:43:10 - 43 scans - 19 detections (44.0%)

File characteristics

Meta data

```
 Size:    118784 bytes
Type:    PE32 executable (DLL) (GUI) Intel 80386, for MS Windows
Date:    0x4AC5A6A4 [Fri Oct  2 07:07:16 2009 UTC]
EP:      0x20013857 .text 0/5
CRC:     Claimed: 0x0, Actual: 0x2cb10 [SUSPICIOUS]
```

Sections

```
 Name       VirtAddr     VirtSize     RawSize      Entropy
--------------------------------------------------------------------------------
.text      0x1000       0x12976      0x13000      6.509133
.basein    0x14000      0x97         0x1000       0.418760    [SUSPICIOUS]
.rdata     0x15000      0x4ede       0x5000       7.011329    [SUSPICIOUS]
.data      0x1a000      0x15f0       0x1000       5.453684
.reloc     0x1c000      0x152a       0x2000       4.423836
```

Exports

```
 Flags         : 00000000
Time stamp    : Fri Oct  2 09:07:16 2009
Version       : 0.0
DLL name      : CARBON.dll
Ordinals base : 1. (00000001)
# of Addresses: 2. (00000002)
# of Names    : 2. (00000002)
  1. 00002CB9 ModuleStart
  2. 0000266C ModuleStop
```

Strings

```
 \\.\IdeDrive1\\config.txt
ReceiveTimeout
SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings
NAME
object_id
VERSION
User
Carbon v3.51
OPER|Wrong config: bad address|
Mozilla/4.0 (compatible; MSIE 6.0)
OPER|Wrong config: no port|
OPER|Wrong config: empty address|
address
CW_INET
quantity
user_winmax
user_winmin
ST|Carbon v3.51|
\\.\IdeDrive1\\log.txt
Global\MSMMC.StartupEnvironment.PPT
Global\411A5195CD73A8a710E4BB16842FA42C
Global\881F0621AC59C4c035A5DC92158AB85E
Global\MSCTF.Shared.MUTEX.RPM
Global\WindowsShellHWDetection
Global\MSDBG.Global.MUTEX.ATF
TR|%d|
$Id: hide_module_win32.c 10189 2008-11-25 14:25:41Z gilg $
ZwWow64ReadVirtualMemory64
$Id: load_lib_win32.c 10180 2008-11-20 12:13:01Z gilg $
\SysWOW64\
\System32\
CreateRemoteThread
ZwTerminateThread
LdrGetProcedureAddress
ExitThread
$Id: mutex.c 3940 2006-03-20 16:47:16Z vlad $
$Id: rw_lock.c 4482 2006-08-30 13:07:14Z vlad $
%x-%x-%x-%x
%02d/%02d/%02d|%02d:%02d:%02d|%s|u|
search.google.com
www.easports.com
www.sun.com
www.dell.com
www.3com.com
www.altavista.com
www.hp.com
search.microsoft.com
windowsupdate.microsoft.com
www.microsoft.com
www.asus.com
```

```
www.eagames.com
www.google.com
www.astalavista.com
www.bbc.com
www.yahoo.com
CreateToolhelp32Snapshot() failed: %d
OPER|Sniffer '%s' running... ooopppsss...|
snoop.exe
ettercap.exe
wireshark.exe
ethereal.exe
windump.exe
tcpdump.exe
HTTP/1.1
%sauth.cgi?mode=query&id=%u:%u:%u:%u&serv=%s&lang=en&q=%u-%u&date=%s
%Y-%m-%d
%sdefault.asp?act=%u&id=%u&item=%u&event_id=%u&cln=%u&flt=%u&serv=%s&t=%ld&mode=query&lang=en&date=%s
lastconnect
timestop
.bak
\\.\IdeDrive1\\
D:AI
@OPER|Wrong timeout: high < low|
Mem alloc err
P|-1|%d|NULL|%d|
P|0|%s|%d|HC=%d
HC|%d|
P|-1|%d|%s|%d|
\\.\IdeDrive1\\Results\result.txt
POST
HTTP/1.0
A|-1|%u|%s|%s|
%u|%s|%s
Task %d failed %s,%d
\\.\IdeDrive1\\Results\
207.46.249.57
207.46.249.56
207.46.250.119
microsoft.com
207.46.253.125
207.46.18.94
update.microsoft.com
G|0|%d|%d|
%u|%s|%s|%s
OPER|Wrong config|
S|0|%s|
S|-1|%d|%s|
logperiod
lastsend
logmax
```

```
logmin
CopyFile(%s, %s):%d
CrPr(),WL(),AU() error: %d
CrPr() WaitForSingleObject() error: %d
CrPr() wait timeout %d msec exceeded: %d
T|-1|%d|%d|
Task not execute. Arg file failed.
WORKDATA
run_task
DELETE
COMPRESSION
RESULT
stdout
CONFIG
cmd.exe
time2task
m_recv() RESULT failed.
A|-1|%u|%s|%d|
active_con
m_send() TASK failed.
OBJECT ACK failed.
Internal task %d obj %s not equal robj %s... very strange!!!
m_recv() OBJECT failed.
m_send() OBJECT failed.
m_send() WHO failed.
AUTH failed.
m_recv() AUTH failed.
m_send() AUTH failed.
m_connect() failed.
m_setoptlist() failed.
net_password=
net_user=
allow=*everyone
write_peer_nfo=%c%s%c
frag_no_scrambling=1
frag_size=32768
m_create() failed.
frag.np
\\%s\pipe\comnode
W|2|%s|%d|
127.0.0.1
m_send() ZERO failed.
Trans task %d obj %s ACTIVE fail robj %s
net_password=%s
net_user=%s
\\%s\pipe\%s
frag.tcp
%s:%d
W|1|%s|%d|
%u|%s|%s|%s|%s|%d|%s|%s
```

```
\\.\IdeDrive1\\Tasks\task_system.txt
%u|%s|%s|%s|%s|%d
\\.\IdeDrive1\\Tasks\task.txt
%u|%s|%s|%s|%s
\\.\IdeDrive1\\Tasks\
W|0|%s|%d|
W|-1|%s|%d|
start
T|e|%d|
T|s|%d|
task_max
task_min
I|%d|
        reconstructing block ...
%6d unresolved strings
        depth %6d has
        bucket sorting ...
        %d pointers, %d sorted, %d scanned
        qsort [0x%x, 0x%x]   done %d   this %d
        main sort initialise ...
    too repetitive; using fallback sorting algorithm
      %d work, %d block, ratio %5.2f
CONFIG_ERROR
OUTBUFF_FULL
UNEXPECTED_EOF
IO_ERROR
DATA_ERROR_MAGIC
DATA_ERROR
MEM_ERROR
PARAM_ERROR
SEQUENCE_ERROR
codes %d
code lengths %d,
selectors %d,
      bytes: mapping %d,
      pass %d: size is %d, grp uses are
      initial group %d, [%d .. %d], has %d syms (%4.1f%%)
Y@      %d in block, %d after MTF & 1-2 coding, %d+2 syms in use
    final combined CRC = 0x%x
    block %d: crc = 0x%8x, combined CRC = 0x%8x, size = %d
$Id: b2_to_m2_stub.c 5273 2007-01-23 17:41:15Z vlad $
$Id: b_tcp.c 8474 2007-09-19 15:40:39Z vlad $
TCP: closed.
TCP: connecting...
Y1N0
nodelay
TCP: send
TCP: recv
%s:%u
nodelay=1
```

```
TCP: resolved %s
TCP: resolving host name...
$Id: l1_check.c 4477 2006-08-28 15:58:21Z vlad $
$Id: m2_to_b2_stub.c 4477 2006-08-28 15:58:21Z vlad $
$Id: m_frag.c 8715 2007-11-29 16:04:46Z urik $
peer_frag_size
frag_no_scrambling
frag_size
Frag: send
$Id: m_np.c 8825 2008-01-10 13:13:15Z vlad $
\\.\pipe\
no_server_hijack
imp_level
net_password
net_user
write_peer_nfo
read_peer_nfo
*everyone
allow
$Id: np_win32_common.c 4483 2006-08-30 13:13:51Z vlad $
anonymous
every1
\ipc$
\pipe\
$Id: t_byte1.c 5324 2007-01-30 12:45:35Z vlad $
frag
$Id: t_manager.c 8715 2007-11-29 16:04:46Z urik $
transports
$Id: t_message1.c 5290 2007-01-26 11:15:03Z vlad $
licence error
```

Sample D - cryptoapi.dll (Resource: 105)

Hashes

Type of Hash	Hash
MD5	a67311ec502593630307a5f3c220dc59
SHA1	74b0c62737f43b0138cfae0d0972178a14fbea10
SHA-256	67bc775cc1a58930201ef247ace86cc5c8569057d4911a8e910ac2263c8eb880
ssdeep	3072:/eZCuX04e/tmjQFFTNna3bFy99f3bay/FjIJA:/eZbUIj4zaLFw9/JI+

VirusTotal results for sample D

AV product	Result
CAT-QuickHeal	Backdoor.Pfinet

AV product	Result
McAfee	Generic.dx!ueu
K7AntiVirus	Riskware
VirusBuster	Backdoor.Agent!JK8atQHb1PQ
Symantec	Backdoor.Pfinet
Norman	W32/Suspicious_Gen3.JVLR
TrendMicro-HouseCall	TROJ_GEN.R47C3JS
Avast	Win32:Malware-gen
Kaspersky	UDS:DangerousObject.Multi.Generic
BitDefender	Backdoor.Generic.264016
Emsisoft	Backdoor.SuspectCRC!IK
Comodo	UnclassifiedMalware
F-Secure	Backdoor.Generic.264016
VIPRE	Trojan.Win32.Generic!BT
AntiVir	TR/ATRAPS.Gen
TrendMicro	TROJ_GEN.R47C3JS
McAfee-GW-Edition	Heuristic.BehavesLike.Win32.Suspicious.H
GData	Backdoor.Generic.264016
AhnLab-V3	Backdoor/Win32.Pfinet
PCTools	Backdoor.Pfinet
Ikarus	Backdoor.SuspectCRC
Panda	Trj/CI.A
Avast5	Win32:Malware-gen

Scanned: 2011-05-08 11:16:36 - 42 scans - 23 detections (54.0%)

File characteristics

Meta data

```
 Size:    135168 bytes
Type:    PE32 executable (DLL) (GUI) Intel 80386, for MS Windows
Date:    0x4AC5A662 [Fri Oct  2 07:06:10 2009 UTC]
```

```
EP:      0x20015d85 .text 0/5
CRC:     Claimed: 0x0, Actual: 0x2ccd6 [SUSPICIOUS]
```

Exports

```
 Flags         : 00000000
Time stamp    : Fri Oct  2 09:06:07 2009
Version       : 0.0
DLL name      : carbon_system.dll
Ordinals base : 1. (00000001)
# of Addresses: 1. (00000001)
# of Names    : 1. (00000001)
  1. 00002655 ModuleStart
```

Sections

```
 Name       VirtAddr     VirtSize     RawSize      Entropy
-------------------------------------------------------------------------------
.text      0x1000       0x150d5      0x16000      6.417399
.basein    0x17000      0x97         0x1000       0.418760    [SUSPICIOUS]
.rdata     0x18000      0x5380       0x6000       6.450645
.data      0x1e000      0x15e0       0x1000       5.450370
.reloc     0x20000      0x15e4       0x2000       4.991237
```

Strings

```
 $Id: t_utils.c 5503 2007-02-26 13:14:30Z vlad $
$Id: t_status.c 5666 2007-03-19 16:18:00Z vlad $
$Id: t_message1.c 5290 2007-01-26 11:15:03Z vlad $
$Id: t_manager.c 8715 2007-11-29 16:04:46Z urik $
$Id: t_byte1.c 5324 2007-01-30 12:45:35Z vlad $
$Id: np_win32_common.c 4483 2006-08-30 13:13:51Z vlad $
$Id: m_np.c 8825 2008-01-10 13:13:15Z vlad $
$Id: m_frag.c 8715 2007-11-29 16:04:46Z urik $
$Id: m2_to_b2_stub.c 4477 2006-08-28 15:58:21Z vlad $
$Id: l1_check.c 4477 2006-08-28 15:58:21Z vlad $
$Id: b_tcp.c 8474 2007-09-19 15:40:39Z vlad $
$Id: b2_to_m2_stub.c 5273 2007-01-23 17:41:15Z vlad $
$Id: thread.c 4593 2006-10-12 11:43:29Z urik $
$Id: rw_lock.c 4482 2006-08-30 13:07:14Z vlad $
$Id: mutex.c 3940 2006-03-20 16:47:16Z vlad $
$Id: load_lib_win32.c 10180 2008-11-20 12:13:01Z gilg $
$Id: hide_module_win32.c 10189 2008-11-25 14:25:41Z gilg $
\\.\IdeDrive1\\Tasks\
\\.\IdeDrive1\\Results\
Global\MSDBG.Global.MUTEX.ATF
Global\WindowsShellHWDetection
```

```
Global\MSCTF.Shared.MUTEX.RPM
Global\881F0621AC59C4c035A5DC92158AB85E
Global\411A5195CD73A8a710E4BB16842FA42C
Global\MSMMC.StartupEnvironment.PPT
\\.\IdeDrive1\\log.txt
TR|%d|
SR|%d|
ST|Carbon v3.61|
\\.\IdeDrive1\\*.bak
\\.\IdeDrive1\\
\\.\IdeDrive1\\Tasks\task.txt
\\.\IdeDrive1\\Tasks\task_system.txt
\\.\IdeDrive1\\Tasks\*.tmp
\\.\IdeDrive1\\config.txt
sys_winmin
TIME
sys_winmax
\\.\IdeDrive1\\restrans.txt
quantity
CW_LOCAL
address
object
D:(A;OICIID;GRGWGX;;;WD)
Carbon v3.61
System
VERSION
object_id
NAME
CW_INET
logperiod
OPER|Survive me, i`m close to death... free space less than 5%%...|
OPER|Low space... free space less than 10%%...|
ZwWow64ReadVirtualMemory64
ExitThread
LdrGetProcedureAddress
ZwTerminateThread
CreateRemoteThread
\System32\
\SysWOW64\
OPER|Wrong timeout: high < low|
%02d/%02d/%02d|%02d:%02d:%02d|%s|s|
CreateToolhelp32Snapshot() failed: %d
tcpdump.exe
windump.exe
ethereal.exe
wireshark.exe
ettercap.exe
snoop.exe
OPER|Sniffer '%s' running... ooopppsss...|
%x-%x-%x-%x
```

```
run_task_system
WORKDATA
\\.\IdeDrive1\\Results\result.txt
I|%d|
task_min
task_max
T|s|%d|
%u|1|%s|%s
%u|2|%s|%s|%s
T|e|%d|
start
time2task
cmd.exe
CONFIG
stdout
RESULT
COMPRESSION
DELETE
%u|%s|%s
%u|%s|%s|%s
Task not execute. Arg file failed.
T|-1|%d|%d|
AS_USER:LogonUser():%d
AS_USER:DuplicateTokenEx():%d
explorer.exe
AS_CUR_USER:OpenProcessToken():%d
AS_CUR_USER:DuplicateTokenEx():%d
CrPr() wait timeout %d msec exceeded: %d
CrPr() WaitForSingleObject() error: %d
CrPr(),WL(),AU():%d
CopyFile(%s, %s):%d
Memory allocation error. Use no compression
frag.np
\\.\Global\PIPE\comnode
frag_size=32768
frag_no_scrambling=1
allow=*everyone
active_con
frag.tcp/%s:445
frag.np/%s
\\.\IdeDrive1\\logtrans.txt
A|2|%s|
W|%s|%s|
m_send() ZERO1 failed
W|%s|%s|%s|
\*.tmp
m_send() ZERO2 failed
R|%s|%d|
\\%s\pipe\comnode
frag.tcp
```

```
net_user=
net_password=
write_peer_nfo=%c%s%c
P|0|%s|%d|
P|-1|%d|%s|%d|
P|-1|%d|%d|
nodelay=N
W|-1|%d|%s|
SEND AUTH
W|-1|%d|%s|%s|
RECV AUTH
AUTH FAILED
SEND WHO
SEND OBJECT_ID
logmin
logmax
lastsend
S|0|%s|
S|-1|%d|%s|
Task %d failed %s, %d
A|-1|%u|%s|%s|
timestop
lastconnect
.bak
%u:%u:%u:%u:%u
Freeze Ok.
\$NtUninstallQ722833$\usbdev.sys
\\.\IdeDrive1\\usbdev.bak
\\.\IdeDrive1\\inetpub.bak
\\.\IdeDrive1\\inetpub.dll
\\.\IdeDrive1\\cryptoapi.bak
\\.\IdeDrive1\\cryptoapi.dll
Update Ok.
Update failed =((  Can`t create file.
\\.\IdeDrive1\\Plugins\
Can't create file '%s', error %d =((
Create plugin '%s' OK.
Create plugin '%s' failed. Write error, %d.
PLUGINS
Find existing record.
not_started|%d
Config update success.
enable%s
Config record error: %s = %s.
Plugin not found in config.
Plugin already loaded.
ModuleStart
can`t find entry point.
loadlibrary() failed.
Plugin start failed, %d
```

```
try to run dll with user priv.
can`t get characs.
Plugin not PE format.
Plugin start success.
Plugin start failed.
disable%s
removed%s
Plugin not loaded.
Plugin deleted.
Plugin delete failed, %d.
Plugin terminated.
Plugin terminate failed, %d.
ModuleStop
Plugin dll stop success.
Plugin dll stop failed.
Plugin freelib success.
Plugin freelib failed, %d.
Internal command not support =((
%u|1|%s
G|0|%d|%d|
W|0|%s|%d|
A|0|%s|%d|
%u|%s|%s|%s|%s
%u|%s|%s|%s|%s|%d|%s|%s
%u|%s|%s|%s|%s|%d
W|1|%s|%d|
A|1|%s|%d|
%s:%d
\\%s\pipe\%s
m_create() failed.
net_user=%s
net_password=%s
m_setoptlist() failed.
m_connect() failed.
m_send() AUTH failed.
m_recv() AUTH failed.
AUTH failed.
m_send() WHO failed.
m_send() OBJECT failed.
m_recv() OBJECT failed.
Trans task %d for obj %s ACTIVE fail robj=%s
OBJECT ACK failed.
m_send() TASK failed.
m_recv() WIN RESULT failed.
m_recv() ACT RESULT failed.
m_send() ACT RESULT failed.
enable
L|-1|can`t find entry point %s|
L|-1|loadlibrary() failed %d|
L|-1|%s|%d|
```

```
L|-1|try to run dll %s with user priv|
L|-1|can`t get characs %s|
L|-1|not PE format %s|
L|-1| parse error %s|
L|-1|  parse error %s|
L|0|%s|
L|-1|AS_CUR_USER:OpenProcessToken():%d, %s|
L|-1|AS_CUR_USER:DuplicateTokenEx():%d, %s|
L|-1|AS_CUR_USER:LogonUser():%d, %s|
L|-1|wrong priv %s|
L|-1|CreateProcessAsUser():%d, %s|
D:AI
TCP: resolving host name...
TCP: resolved %s
TCP: closed.
TCP: connecting...
nodelay
Y1N0
TCP: send
TCP: recv
%s:%u
Frag: send
frag_size
frag_no_scrambling
peer_frag_size
\\.\pipe\
allow
*everyone
read_peer_nfo
write_peer_nfo
net_user
net_password
imp_level
no_server_hijack
every1
anonymous
\pipe\
\ipc$
frag
transports
licence error
```

Sample E - usbdev.sys - x64 - (Resouce: 161)

Hashes

Type of Hash	Hash
MD5	62e9839bf0b81d7774a3606112b318e8

Type of Hash	Hash
SHA1	6f2e50c5f03e73e77484d5845d64d952b038a12b
SHA-256	39050386f17b2d34bdbd118eec62ed6b2f386e21500a740362454ed73ea362e8
ssdeep	3072:S9f3buYUVKa6a1206K55kL+tkA3qkQQ0dwZATH:S9iYUImo06KXkL+qA6kf0dwK

VirusTotal results for sample E

AV product	Result
McAfee+Artemis	Pfinet
nProtect	Trojan/W32.Agent.228352.W
McAfee	Pfinet
F-Prot	W32/Pfinet.A
a-squared	Backdoor.Pfinet!IK
Avast	Win32:Malware-gen
ClamAV	Trojan.Agent-126457
Kaspersky	Trojan.Win32.Agent.czua
BitDefender	Trojan.Generic.2617254
Comodo	TrojWare.Win32.Agent.czua
F-Secure	Trojan:W64/Carbys.gen!A
DrWeb	Trojan.Siggen.27969
TrendMicro	TROJ_PFINET.A
Authentium	W32/Pfinet.A
Jiangmin	Trojan/Agent.dcrw
Antiy-AVL	Trojan/Win32.Agent.gen
Symantec	Backdoor.Pfinet
Microsoft	Backdoor:WinNT/Pfinet.B
GData	Trojan.Generic.2617254
VBA32	Trojan.Win32.Agent.czua
PCTools	Backdoor.Pfinet
Ikarus	Backdoor.Pfinet

AV product	Result
AVG	Agent2.YKW
Panda	Rootkit/Agent.MXI

Scanned: 2009-12-27 12:15:01 - 40 scans - 24 detections (60.0%)

File characteristics

Meta data

```
 Size:    228352 bytes
Type:    PE32+ executable (DLL) (native) x86-64, for MS Windows
Date:    0x4AC48FE7 [Thu Oct  1 11:17:59 2009 UTC]
EP:      0x21454 .text 0/6
CRC:     Claimed: 0x397f7, Actual: 0x397f7
```

Sections

```
 Name       VirtAddr     VirtSize     RawSize      Entropy
--------------------------------------------------------------------------------
.text      0x1000       0x2126c      0x21400      6.518352
.basein    0x23000      0xc7         0x200        2.902918
.data      0x24000      0x23a3c      0x13400      1.284443
.pdata     0x48000      0x10b0       0x1200       5.035513
INIT       0x4a000      0x10ce       0x1200       4.944873
.reloc     0x4c000      0x99a        0xa00        4.576183
```

Strings

The strings correspond mostly to the ones of Sample B.

Sample F - inetpub.dll - x64 (Resource: 162)

Hashes

Type of Hash	Hash
MD5	e1ee88eda1d399822587eb58eac9b347
SHA1	32287d26656587c6848902dbed8086c153d94ee7
SHA-256	92c2023095420de3ca7d53a55ed689e7c0086195dc06a4369e0ee58a803c17bb

Type of Hash	Hash
ssdeep	3072:vr84EaVK9B9MklzeALxqS6kcLyHFQ+vYnb9f3bkrlESXdMQyFc8:QPp9B9MkllLMScLmsb9IKrF1

VirusTotal results for sample F

AV product	Result
Symantec	Backdoor.Pfinet

Scanned: 2014-03-23 21:27:06 - 51 scans - 1 detections (1.0%)

File characteristics

Meta data

```
 Size:    113664 bytes
Type:    PE32+ executable (DLL) (GUI) x86-64, for MS Windows
Date:    0x4AC5A6C2 [Fri Oct  2 07:07:46 2009 UTC]
EP:      0x200149d0 .text 0/5
CRC:     Claimed: 0x0, Actual: 0x1e6b8 [SUSPICIOUS]
```

Sections

```
 Name       VirtAddr     VirtSize     RawSize      Entropy
--------------------------------------------------------------------------------
.text      0x1000       0x13b8d      0x13c00      6.247940
.rdata     0x15000      0x582e       0x5a00       6.692290
.data      0x1b000      0x1ae0       0x1400       4.598089
.pdata     0x1d000      0x8c4        0xa00        4.522066
.reloc     0x1e000      0x248        0x400        2.325587
```

Strings

The strings correspond mostly to the ones of Sample C.

Sample G - cryptoapi.dll - x64 (Resource: 165)

Hashes

Type of Hash	Hash
MD5	a7853bab983ede28959a30653baec74a
SHA1	eee11da421c7268e799bd938937e7ef754a895bf
SHA-256	0e3842bd092db5c0c70c62e8351649d6e3f75e97d39bbfd0c0975b8c462a65ca
ssdeep	3072:U/ylCK5WUZFspUjcF65zlEzEOflC9Pw6OPEH66kcXF9f3b6ivgCUHXM:1gWWUrg3ANOP+6cXF9/u

VirusTotal results for sample G

AV product	Result
Symantec	Backdoor.Pfinet
AntiVir	TR/ATRAPS.Gen2

Scanned: 2014-03-23 21:26:59 - 51 scans - 2 detections (3.0%)

File characteristics

Meta data

```
 Size:    147968 bytes
Type:    PE32+ executable (DLL) (GUI) x86-64, for MS Windows
Date:    0x4AC5A685 [Fri Oct  2 07:06:45 2009 UTC]
EP:      0x2001bd80 .text 0/6
CRC:     Claimed: 0x0, Actual: 0x32c9f [SUSPICIOUS]
```

Sections

```
 Name       VirtAddr     VirtSize     RawSize      Entropy
--------------------------------------------------------------------------------
.text      0x1000       0x1af6d      0x1b000      6.195387
.basein    0x1c000      0xc7         0x200        2.902918
.rdata     0x1d000      0x66f0       0x6800       6.585248
.data      0x24000      0x1b00       0x1400       4.647566
.pdata     0x26000      0xad4        0xc00        4.848795
.reloc     0x27000      0x2a6        0x400        2.344107
```

Strings

The strings correspond mostly to the ones of Sample D.

Sample H - config.txt

Hashes

Type of Hash	Hash
MD5	08cbc46302179c4cda4ec2f41fc9a965
SHA1	6a905818f9473835ac90fc38b9ce3958bfb664d6
SHA-256	3576035105b4714433331dff1f39a50d55f4548701b6ab8343a16869903ebc3c

Content

```
[NAME]
object_id=

[TIME]
user_winmin =  600000
user_winmax = 1200000
sys_winmin = 3600000
sys_winmax = 3700000
task_min = 20000
task_max = 30000
checkmin = 60000
checkmax = 70000
logmin =  600000
logmax = 1200000
lastconnect=
timestop=
active_con = 900000
time2task=3600000

[CW_LOCAL]
quantity = 0

[CW_INET]
quantity = 0

[TRANSPORT]
user_pipe = \\.\pipe\userpipe
system_pipe = \\.\pipe\iehelper

[DHCP]
server = 135

```

```
37
38[LOG]
39lastsend =
40logperiod = 7200
41
42[WORKDATA]
43run_task=
44run_task_system=
```

Analysis - Payload

Sample B - usbdev.sys (Resource: 101)

A very extensive analysis of a similar kernel module of Sample B (usbdev.sys) has been documented in 'Uroburos: the snake rootkit' [2] by deresz and tecamac.

Sample B also checks for the presence of infection markers in form of events:

```
 .text:00023210                push    ebp
.text:00023211                mov     ebp, esp
.text:00023213                sub     esp, 130h
.text:00023219                mov     [ebp+string.Length], 70h
.text:0002321F                mov     [ebp+string.MaximumLength], 72h
.text:00023225                mov     [ebp+string.Buffer], offset aBasenamedobjec ; "\\BaseNamedObjects\\{B93DFED5-9A3B
.text:0002322C                lea     eax, [ebp+var_110]
.text:00023232                mov     [ebp+SecurityDescriptor], eax
.text:00023235                mov     [ebp+ObjectAttributes.Length], 18h
.text:0002323F                mov     [ebp+ObjectAttributes.RootDirectory], 0
.text:00023249                mov     [ebp+ObjectAttributes.Attributes], 40h
.text:00023253                lea     ecx, [ebp+string]
.text:00023256                mov     [ebp+ObjectAttributes.ObjectName], ecx
.text:0002325C                mov     [ebp+ObjectAttributes.SecurityDescriptor], 0
.text:00023266                mov     [ebp+ObjectAttributes.SecurityQualityOfService], 0
.text:00023270                lea     edx, [ebp+ObjectAttributes]
.text:00023276                push    edx             ; ObjectAttributes
.text:00023277                push    1F0003h         ; DesiredAccess
.text:0002327C                lea     eax, [ebp+EventHandle]
.text:00023282                push    eax             ; EventHandle
.text:00023283                call    ZwOpenEvent
```

or as pseudo-code:

```
1  string.Length = 0x70;
2  string.MaximumLength = 0x72;
3  string.Buffer = L"\\BaseNamedObjects\\{B93DFED5-9A3B-459b-A617-59FD9FAD693E}";
4  SecurityDescriptor = &v4;
5  ObjectAttributes.Length = 24;
6  ObjectAttributes.RootDirectory = 0;
```

```
 7  ObjectAttributes.Attributes = OBJ_CASE_INSENSITIVE;
 8  ObjectAttributes.ObjectName = &string;
 9  ObjectAttributes.SecurityDescriptor = 0;
10  ObjectAttributes.SecurityQualityOfService = 0;
11  if ( ZwOpenEvent(&EventHandle, 0x1F0003u, &ObjectAttributes) )
12  {
13    ...
```

That means, the famous Agent.btz marker

```
\BaseNamedObjects\{B93DFED5-9A3B-459b-A617-59FD9FAD693E}
```

is checked directly using a UNICODE_STRING structure without using RtlInitUnicodeString(). A brief comparison with other samples, like

Type of Hash	**Hash**
MD5	57770d70b704811e8ac13893337cea32
SHA1	0e6dff1007b6a5f744b2bc90978496328c95ed11
SHA-256	65fdaf08e562611ce58f1d427f198f8743d88a68e1c4d92afe6dc6251e8a3112

or

Type of Hash	**Hash**
MD5	06a3f5df6ac23db15ba52581a38c725b
SHA1	a6cc9d9034637192d264cb4e9b6b83b70cc36da9
SHA-256	43e71b993d6e7c977caaf2ed7610a71758734d87ec2ceb20a84e573ea05a01b3

shows, that this marker is checked in the same way.

The analysis of this kernel module by *deresz* and *tecamac* is very detailed. We advise the interested reader to work through their document to understand all the details.

Implemented transports

In this module, the following transport or communication modules are present:

- Type 1: tcp
- Type 2: np, m2b

-> TODO: Compare this with the observed transports in

- userland modules
- modules described in other reports

Disassembler Library

This sample contains a large chunk of code taken from the *Udis86 Disassembler Library for x86 / x86-64* project[6]

RawDisk1, RawDisk2 and fixdata.dat

The devices

- \Device\RawDisk1
- \Device\RawDisk2

and the file

- \SystemRoot\$NtUninstallQ722833$\fixdata.dat

are already known from other reports.

If the file fixdata.dat could successfully be created within the function

```
1NTSTATUS create\_fixdata_dat()
2{
3  char v1;
4  NTSTATUS error;
5  OBJECT_ATTRIBUTES ObjectAttributes;
6  LARGE_INTEGER AllocationSize;
7  UNICODE_STRING Name;
8  UINT_PTR ViewSize;
9  __int64 FileInformation;
10 struct _IO_STATUS_BLOCK IoStatusBlock;
11
12 Name.Length = 0x58;
13 Name.MaximumLength = 0x5A;
14 Name.Buffer = L"\\SystemRoot\\$NtUninstallQ722833$\\fixdata.dat";
15 ObjectAttributes.Length = 24;
16 ObjectAttributes.RootDirectory = 0;
17 ObjectAttributes.Attributes = OBJ_CASE_INSENSITIVE;
18 ObjectAttributes.ObjectName = &Name;
19 ObjectAttributes.SecurityDescriptor = 0;
20 ObjectAttributes.SecurityQualityOfService = 0;
21 AllocationSize = 0x6400000i64;
22 error = call_IoCreateFile(
23           &FileHandle,
24           FILE_ADD_FILE|FILE_LIST_DIRECTORY,
25           &ObjectAttributes,
26           &IoStatusBlock,
27           &AllocationSize,
28           FILE_ATTRIBUTE_NORMAL,
29           0,
30           FILE_OPEN_IF,
31           FILE_RANDOM_ACCESS|FILE_NON_DIRECTORY_FILE|FILE_SYNCHRONOUS_IO_NONALERT|FILE_NO_INTERMEDIATE_B
32           0,
33           0);
34 if ( !error )
```

```
35  {
36    dword_5BDEC = FileHandle;
37    if ( IoStatusBlock.Information == 2 )
38    {
39      FileInformation = AllocationSize.QuadPart;
40      error = ZwSetInformationFile(FileHandle, &IoStatusBlock, &FileInformation, 8u, FileEndOfFileInformat
41      if ( error )
42        goto LABEL_10;
43      v1 = 1;
44    }
45    else
46    {
47      v1 = 0;
48    }
49    ObjectAttributes.Length = 24;
50    ObjectAttributes.RootDirectory = 0;
51    ObjectAttributes.Attributes = 0;
52    ObjectAttributes.ObjectName = 0;
53    ObjectAttributes.SecurityDescriptor = 0;
54    ObjectAttributes.SecurityQualityOfService = 0;
55    error = ZwCreateSection(&gSectionHandle, 6u, &ObjectAttributes, 0, 4u, 0x18000000u, FileHandle);
56    if ( !error )
57    {
58      ViewSize = 0;
59      error = ZwMapViewOfSection(gSectionHandle, 0xFFFFFFFF, &BaseAddress_0, 0, 0, 0, &ViewSize, ViewUnmap
60      if ( !error )
61      {
62        gViewSize = ViewSize;
63        dword_4FBD4[0] = 0;
64        if ( v1 )
65          sub_2F6E0(0, gViewSize, 2, gViewSize >> 15, 32, 0x200u);
66      }
67    }
68  }
69LABEL_10:
70  if ( error )
71  {
72    if ( BaseAddress_0 )
73    {
74      ZwUnmapViewOfSection(0xFFFFFFFF, BaseAddress_0);
75      BaseAddress_0 = 0;
76    }
77    if ( gSectionHandle )
78    {
79      ZwClose_1(gSectionHandle);
80      gSectionHandle = 0;
81    }
82    ZwClose_1(FileHandle);
83    FileHandle = 0;
84  }
```

```
85  return error;
86}
```

also the devices are created within this function:

```
 1NTSTATUS create_file_rawdisk()
 2{
 3  NTSTATUS ERROR;
 4  OBJECT_ATTRIBUTES ObjectAttributes;
 5  LSA_UNICODE_STRING DestinationString;
 6  UINT_PTR ViewSize;
 7
 8  if ( disks_initialized )
 9  {
10    ERROR = 0;
11  }
12  else if ( DriverObject )
13  {
14    sub_2DFD0(&Lock);
15    KeInitializeEvent(&Event, SynchronizationEvent, 0);
16    sub_2DFB0(&ListHead);
17    ERROR = sub_2F490();
18    if ( !ERROR )
19    {
20      RtlInitUnicodeString(&DestinationString, L"\\Device\\RawDisk1");
21      ERROR = IoCreateDevice(
22                DriverObject,
23                0,
24                &DestinationString,
25                FILE_DEVICE_DISK,
26                FILE_REMOVABLE_MEDIA,
27                0,
28                &DeviceObject_RawDisk1);
29      if ( !ERROR )
30      {
31        ERROR = call_SeSetSecurityDescriptorInfo(DeviceObject_RawDisk1);
32        if ( !ERROR )
33        {
34          DeviceObject_RawDisk1->Flags = (DeviceObject_RawDisk1->Flags | 0x10);
35          DeviceObject_RawDisk1->Flags = DeviceObject_RawDisk1->Flags & 0xFFFFFF7F;
36          ObjectAttributes.Length = 24;
37          ObjectAttributes.RootDirectory = 0;
38          ObjectAttributes.Attributes = 0;
39          ObjectAttributes.ObjectName = 0;
40          ObjectAttributes.SecurityDescriptor = 0;
41          ObjectAttributes.SecurityQualityOfService = 0;
42          MaximumSize = 0x1000000i64;
43          ERROR = ZwCreateSection(&SectionHandle, 6u, &ObjectAttributes, &MaximumSize, 4u, 0x18000000u, 0
44          if ( !ERROR )
45          {
```

```
46            ViewSize = MaximumSize.LowPart;
47            ERROR = ZwMapViewOfSection(SectionHandle, 0xFFFFFFFF, &BaseAddress, 0, 0, 0, &ViewSize, ViewU
48            if ( !ERROR )
49            {
50              MaximumSize = ViewSize;
51              RtlInitUnicodeString(&DestinationString, L"\\Device\\RawDisk2");
52              ERROR = IoCreateDevice(
53                        DriverObject,
54                        0,
55                        &DestinationString,
56                        FILE_DEVICE_DISK,
57                        FILE_REMOVABLE_MEDIA,
58                        0,
59                        &DeviceObject_RawDisk2);
60              if ( !ERROR )
61              {
62                ERROR = call_SeSetSecurityDescriptorInfo(DeviceObject_RawDisk2);
63                if ( !ERROR )
64                {
65                  DeviceObject_RawDisk2->Flags = (DeviceObject_RawDisk2->Flags | 0x10);
66                  DeviceObject_RawDisk2->Flags = DeviceObject_RawDisk2->Flags & 0xFFFFFF7F;
67                  sub_2F6E0(1, MaximumSize.LowPart, 2, MaximumSize.LowPart >> 15, 32, 0x200u);
68                  byte_4FBBD = 0;
69                  ERROR = create_system_threads(&handle, sub_2EFB0, 0, 0);
70                  disks_initialized = 1;
71                }
72              }
73            }
74          }
75        }
76      }
77    }
78    if ( ERROR )
79    {
80      if ( DeviceObject_RawDisk1 )
81      {
82        IoDeleteDevice(DeviceObject_RawDisk1);
83        DeviceObject_RawDisk1 = 0;
84      }
85      if ( DeviceObject_RawDisk2 )
86      {
87        IoDeleteDevice(DeviceObject_RawDisk2);
88        DeviceObject_RawDisk2 = 0;
89      }
90      if ( BaseAddress )
91      {
92        ZwUnmapViewOfSection(0xFFFFFFFF, BaseAddress);
93        BaseAddress = 0;
94      }
95      if ( SectionHandle )
```

```
 96      {
 97        ZwClose_1(SectionHandle);
 98        SectionHandle = 0;
 99      }
100    }
101  }
102  else
103  {
104    ERROR = 0xC0000001;
105  }
106  return ERROR;
107}
```

Decryption of string for VFS drive

The authors demonstrate that they have a sense of humor. In the following example, they decrypt (XOR) the strings used to assemble the locations of where to drop the other components of the malware to. The final destinations are:

- \.\IdeDrive1\cryptoapi.dll
- \.\IdeDrive1\inetpub.dll

But have a closer look at how they decrypt the string:

```
 [...]
.text:0001E122                mov     [ebp+xor_key], 4E415341h ; key
.text:0001E129                mov     [ebp+part_1], 7253605h   ; part 1 encrypted
.text:0001E130                mov     [ebp+part_2], 3C282524h  ; part 2 encrypted
[...]
.text:0001E17B                mov     eax, [ebp+part_1]
.text:0001E17E                xor     eax, [ebp+xor_key]       ; decrypt part 1: IdeD
.text:0001E181                mov     [ebp+part_1], eax
[...]
.text:0001E184                mov     ecx, [ebp+part_2]
.text:0001E18A                xor     ecx, [ebp+xor_key]       ; decrypt part 2: rive
.text:0001E18D                mov     [ebp+part_2], ecx
[...]
```

They are seriously using a key 0x4E415341 to decrypt the string. 0x4E415341 is ASCII for ‘NASA’. That’s how they decrypt and assemble the string IdeDrive, appending a ‘1’ in the next step and using if for creating the destination. Full excerpt below:

```
 [...]
.text:0001E11B                mov     [ebp+var_20], 0
.text:0001E122                mov     [ebp+xor_key], 4E415341h
.text:0001E129                mov     [ebp+part_1], 7253605h
.text:0001E130                mov     [ebp+part_2], 3C282524h
.text:0001E13A                xor     eax, eax
.text:0001E13C                mov     [ebp+drive], eax
.text:0001E142                mov     [ebp+var_338], eax
```

```
.text:0001E148                mov     [ebp+var_334], ax
.text:0001E14F                push    104h            ; size_t
.text:0001E154                push    0               ; int
.text:0001E156                lea     ecx, [ebp+cryptoapi.dll]
.text:0001E15C                push    ecx             ; void *
.text:0001E15D                call    memset
.text:0001E162                add     esp, 0Ch
.text:0001E165                push    104h            ; size_t
.text:0001E16A                push    0               ; int
.text:0001E16C                lea     edx, [ebp+inetpub.dll]
.text:0001E172                push    edx             ; void *
.text:0001E173                call    memset
.text:0001E178                add     esp, 0Ch
.text:0001E17B                mov     eax, [ebp+part_1]
.text:0001E17E                xor     eax, [ebp+xor_key]
.text:0001E181                mov     [ebp+part_1], eax
.text:0001E184                mov     ecx, [ebp+part_2]
.text:0001E18A                xor     ecx, [ebp+xor_key]
.text:0001E18D                mov     [ebp+part_2], ecx
.text:0001E193                mov     edx, [ebp+part_1]
.text:0001E196                push    edx
.text:0001E197                call    order_bytes
.text:0001E19C                mov     [ebp+part_1], eax
.text:0001E19F                mov     eax, [ebp+part_1]
.text:0001E1A2                mov     [ebp+part_1], eax
.text:0001E1A5                mov     ecx, [ebp+part_2]
.text:0001E1AB                push    ecx
.text:0001E1AC                call    order_bytes
.text:0001E1B1                mov     [ebp+part_2], eax
.text:0001E1B7                mov     edx, [ebp+part_2]
.text:0001E1BD                mov     [ebp+part_2], edx
.text:0001E1C3                mov     eax, [ebp+part_1]
.text:0001E1C6                mov     [ebp+drive], eax
.text:0001E1CC                mov     ecx, [ebp+part_2]
.text:0001E1D2                mov     [ebp+var_338], ecx
.text:0001E1D8                lea     edx, [ebp+drive]
.text:0001E1DE                add     edx, 0FFFFFFFFh
.text:0001E1E1                mov     [ebp+var_454], edx
.text:0001E1E7                mov     eax, [ebp+var_454]
.text:0001E1ED                mov     cl, [eax+1]
.text:0001E1F0                mov     [ebp+var_455], cl
.text:0001E1F6                add     [ebp+var_454], 1
.text:0001E1FD                cmp     [ebp+var_455], 0
.text:0001E204                jnz     short loc_1E1E7
.text:0001E206                mov     edi, [ebp+var_454]
.text:0001E20C                mov     dx, word ptr ds:a1 ; "1"
.text:0001E213                mov     [edi], dx
.text:0001E216                lea     eax, [ebp+drive]
.text:0001E21C                push    eax
.text:0001E21D                push    offset a??SCryptoapi_d ; "\\??\\%s\\cryptoapi.dll"
```

```
.text:0001E222                lea     ecx, [ebp+cryptoapi.dll]
.text:0001E228                push    ecx             ; char *
.text:0001E229                call    sprintf
.text:0001E22E                add     esp, 0Ch
.text:0001E231                lea     edx, [ebp+drive]
.text:0001E237                push    edx
.text:0001E238                push    offset a??SInetpub_dll ; "\\??\\%s\\inetpub.dll"
.text:0001E23D                lea     eax, [ebp+inetpub.dll]
.text:0001E243                push    eax             ; char *
.text:0001E244                call    sprintf
[...]
```

To describe

```
 \Registry\Machine\usblink_export

HKEY_LOCAL_MACHINE\usblink_export

(also LEGACY_usblink and usblink?)
```

Potentially old code

The malware checks if the queried process has one of the following names

```
 1bool __stdcall match_list_of_programs_by_name(char *a1)
 2{
 3  return !stricmp(a1, "iexplore.exe")
 4      || !stricmp(a1, "firefox.exe")
 5      || !stricmp(a1, "opera.exe")
 6      || !stricmp(a1, "netscape.exe")
 7      || !stricmp(a1, "mozilla.exe")
 8      || !stricmp(a1, "msimn.exe")
 9      || !stricmp(a1, "outlook.exe")
10      || !stricmp(a1, "adobeupdater.exe");
11}
```

and if so, it would call *pulse_event_wininet_activate()*.

```
1char __stdcall check_proces_and_activate_wininet(int a1, int a2, int a3)
2{
3[...]
4     if ( match_list_of_programs_by_name(&program_name) )
5        pulse_event_wininet_activate();
6[...]
7}
```

The event *\BaseNamedObjects\wininet_activate* is then created and pulsed.

```
NTSTATUS pulse_event_wininet_activate()
{
  NTSTATUS result;
  LSA_UNICODE_STRING DestinationString;
  OBJECT_ATTRIBUTES ObjectAttributes;
  HANDLE EventHandle;
  wchar_t SourceString;

  swprintf(&SourceString, L"\\BaseNamedObjects\\%S", "wininet_activate");
  RtlInitUnicodeString(&DestinationString, &SourceString);
  ObjectAttributes.Length = 24;
  ObjectAttributes.RootDirectory = 0;
  ObjectAttributes.Attributes = 0;
  ObjectAttributes.ObjectName = &DestinationString;
  ObjectAttributes.SecurityDescriptor = 0;
  ObjectAttributes.SecurityQualityOfService = 0;
  result = ZwOpenEvent(&EventHandle, 2u, &ObjectAttributes);
  if ( !result )
  {
    result = ZwPulseEvent(EventHandle, 0);
    ZwClose_1(EventHandle);
  }
  return result;
}
```

There are no references to this event, neither in this module nor in the other analyzed modules. Microsoft mentions in the documentation of the PulseEvent function [7]:

> Note This function is unreliable and should not be used. It exists mainly for backward compatibility. For more information, see Remarks.

So it could well be that this part is old code and was forgotten to be removed.

Applying work-around for bugs related to AMD Athlon and AGP graphics port

From Microsoft Support article *AGP program may hang when using page size extension on Athlon processor* [8] the following excerpt:

> The following workaround for this issue prevents Memory Manager from using the processor's Page Size Extension feature and may affect the performance of some programs, depending on the paging behavior. This registry value also limits non-paged pool to a maximum of 128 megabytes (MB) instead of 256 MB.

```
int __stdcall disable_processors_page_size_extension_feature(int a1)
{
  name[0] = 0xA8;
  name[1] = 0xAA;
  *&name[2] = L"\\Registry\\Machine\\System\\CurrentControlSet\\Control\\Session Manager\\Memory Managemen
  ValueName.Length = 32;
  ValueName.MaximumLength = 34;
```

```
 8  ValueName.Buffer = L"LargePageMinimum";
 9  Data = -1;
10  v2 = sub_19110();
11  if ( !v2 )
12  {
13    ObjectAttributes.Length = 24;
14    ObjectAttributes.RootDirectory = 0;
15    ObjectAttributes.Attributes = OBJ_CASE_INSENSITIVE;
16    ObjectAttributes.ObjectName = name;
17    ObjectAttributes.SecurityDescriptor = 0;
18    ObjectAttributes.SecurityQualityOfService = 0;
19    if ( !ZwOpenKey(&KeyHandle, 2u, &ObjectAttributes) )
20    {
21      ZwSetValueKey(KeyHandle, &ValueName, 0, 4u, &Data, 4u);
22      ZwClose_1(KeyHandle);
23    }
24}
25
```

Sample D - cryptoapi.dll (Resource: 105)

Original filename: carbon_system.dll

Internal name: Carbon v3.61

This component first initializes the winsock subsystem by calling WSAStartup. Right after it creates directories on the VFS:

```
 CreateDirectoryA("\\\\.\\IdeDrive1\\\\Tasks\\", (LPSECURITY_ATTRIBUTES)&Dst);
CreateDirectoryA("\\\\.\\IdeDrive1\\\\Results\\", (LPSECURITY_ATTRIBUTES)&Dst);
```

Sample D is the next file in the logical execution order, as it creates the following mutexes, which are also accessed by Sample E. Sample D can be considered the main userland module, a control unit that sets up the communication with the kernel module and has the ability to load plugins dynamically during runtime. The internal name of this module, *carbon_system.dll*, supports this observation.

Mutexes from cryptoapi.dll

```
 Global\\MSMMC.StartupEnvironment.PPT
Global\\411A5195CD73A8a710E4BB16842FA42C
Global\\881F0621AC59C4c035A5DC92158AB85E
Global\\MSCTF.Shared.MUTEX.RPM
Global\\WindowsShellHWDetection
Global\\MSDBG.Global.MUTEX.ATF
```

For reading or writing operations on files, exclusive access is ensured by locking them with mutexes:

- Global\MSMMC.StartupEnvironment.PPT is used for operations on the configuration file.

- Global\411A5195CD73A8a710E4BB16842FA42C is used to exclusively access temporary files
- Global\MSDBG.Global.MUTEX.ATF is used to exclusively access \.\IdeDrive1\log.txt
- Global\WindowsShellHWDetection is used to exclusively access \.\IdeDrive1\Results\result.txt
- Global\MSCTF.Shared.MUTEX.RPM is used to exclusively access \.\IdeDrive1\Tasks\task.txt
- Global\881F0621AC59C4c035A5DC92158AB85E is used to exclusively access \.\IdeDrive1\Tasks\task_system.txt

During the startup of the ModuleStart() function, 6 threads are being started. The first two are:

- get_initialization_parameters_create_GUID_and_check_Packet_Capturing()
- periodic_free_space_check_and_write_log()

These serve the purpose of initializing the environment for the malware and running maintenance and log tasks.

Then a function *load_transports()* is called (more later), and then four more threads are started:

- read_config_start_thread_start()
- thread 5 - handles *frag.np*/*frag.tcp* requests
- thread 6 - handles *frag.np*/*frag.tcp* requests
- execute_plugin() - starts a new thread, calling a DLLs export *ModuleStart* from the \.\IdeDrive1\\Plugins\ directory

load_transports()

In this module, the following transport or communication modules are present:

- Type 1: tcp, b2m
- Type 2: np, frag, m2b

each associated with a bunch of functions:

```
 np_functions    func_obj_3 <44h, offset sub_2000FAF9, offset sub_2000FB13, \
.data:2001EE30                   offset sub_2000FB2B, offset sub_2000FC37, \
.data:2001EE30                   offset sub_2000FC91, offset sub_2000FD8E, \
.data:2001EE30                   offset sub_2000FECC, offset sub_20010798, \
.data:2001EE30                   offset sub_20010046, offset sub_2001030F, \
.data:2001EE30                   offset sub_200103BA, offset sub_200103DB, \
.data:2001EE30                   offset sub_2000EB1A, offset sub_2001077D, \
.data:2001EE30                   offset sub_20010798, offset sub_2001079E>

frag_functions  func_obj <4Ch, offset sub_2000DA6E, offset return, \
.data:2001EE78                 offset sub_2000EC14, offset sub_2000EC9E, \
.data:2001EE78                 offset sub_2000ECB2, offset sub_2000ECF3, \
.data:2001EE78                 offset sub_2000ED69, offset sub_2000F5D4, \
.data:2001EE78                 offset sub_2000F4F9, offset sub_2000EDF5, \
.data:2001EE78                 offset sub_2000F185, offset sub_2000F5EB, \
.data:2001EE78                 offset sub_2000EB1A, offset sub_2001077D, \
.data:2001EE78                 offset sub_2000F48B, offset sub_2000F4DA, 0, 0, 0>

m2b_functions   func_obj <4Ch, offset sub_2000DA6E, offset return, \
.data:2001EEC8                 offset sub_2000E8C8, offset sub_2000E93B, \
.data:2001EEC8                 offset sub_2000DB2B, offset sub_2000E94A, \
```

```
.data:2001EEC8                 offset sub_2000E956, offset sub_2000E9B5, \
.data:2001EEC8                 offset sub_2000E9C7, offset sub_2000E9D9, \
.data:2001EEC8                 offset sub_2000EA0C, offset sub_2000EADE, \
.data:2001EEC8                 offset sub_2000EB1A, offset sub_2000EB26, \
.data:2001EEC8                 offset sub_2000EB47, offset sub_2000EB66, \
.data:2001EEC8                 offset sub_2000EB85, offset sub_2000EBE5, 0>

tcp_functions   func_obj_2 <40h, offset sub_2000DDD6, offset WSACleanup, \
.data:2001EF18                   offset sub_2000DE03, offset sub_2000E0FE, \
.data:2001EF18                   offset sub_2000E14A, offset sub_2000E156, \
.data:2001EF18                   offset sub_2000E1D3, offset sub_20010798, \
.data:2001EF18                   offset sub_2000E288, offset sub_2000E31F, \
.data:2001EF18                   offset sub_2000E499, offset sub_2001077D, \
.data:2001EF18                   offset sub_2000E634, offset sub_2000E661, \
.data:2001EF18                   offset sub_2000E715>

b2m_functions   func_obj_2 <40h, offset sub_2000DA6E, offset return, \
.data:2001EF58                   offset sub_2000DA71, offset sub_2000DAF9, \
.data:2001EF58                   offset sub_2000DB2B, offset sub_2000DB44, \
.data:2001EF58                   offset sub_2000DB54, offset sub_2000DBB2, \
.data:2001EF58                   offset sub_2000DBC7, offset sub_2000DBDC, \
.data:2001EF58                   offset sub_2000DBF6, offset sub_2000DD63, \
.data:2001EF58                   offset sub_2000DD84, offset sub_2000DDA2, \
.data:2001EF58                   offset sub_2000DDC0>
```

TODO: these functions need to be analyzed and described

Other reports mention different other transports that are not present in this collection.

Transport (Type)	**CIRCL**	**BAE**	**deresz/tecamac**
tcp (1)	x		x
b2m (1)	x		
np (2)	x		x
enc (2)			x
reliable (2)			x
frag	x	x	x
m2b (2)	x		x
m2d (2)			x
t2m (3)			x
udp (4)			x
doms (4)			x

Transport (Type)	CIRCL	BAE	deresz/tecamac
domc (4)			x

frag.np and frag.tcp replies:

```
 SEND AUTH
RECV AUTH
AUTH FAILED
SEND WHO
SEND OBJECT_ID
```

frag.np/frag.tcp options:

```
 frag_size=32768
frag_no_scrambling=1
allow=*everyone
active_con
net_user=
net_password=
write_peer_nfo=%c%s%c
nodelay=N
```

Files from cryptoapi.dll

```
 \\.\IdeDrive1\
\\.\IdeDrive1\log.txt
\\.\IdeDrive1\*.bak
\\.\IdeDrive1\Tasks\\task.txt
\\.\IdeDrive1\Tasks\\task_system.txt
\\.\IdeDrive1\Tasks\\*.tmp
\\.\IdeDrive1\config.txt
\\.\IdeDrive1\restrans.txt
\\.\IdeDrive1\Tasks\\
\\.\IdeDrive1\Results\\
\\.\IdeDrive1\logtrans.txt
\\.\IdeDrive1\usbdev.bak
\\.\IdeDrive1\inetpub.bak
\\.\IdeDrive1\inetpub.dll
\\.\IdeDrive1\cryptoapi.bak
\\.\IdeDrive1\cryptoapi.dll
\\.\IdeDrive1\Plugins\\
```

Pipes from cryptoapi.dll

```
 \\\\.\\Global\\PIPE\\comnode
\\\\%s\\pipe\\comnode
```

```
\\\\%s\\pipe\\%s
```

Custom error codes, shared in sample B, C and D (E and F to be check)

```
 CUSTOM_ERROR_01  = 21590001h
CUSTOM_ERROR_02  = 21590002h            ; WAIT_TIMEOUT?
CUSTOM_ERROR_03  = 21590003h            ; BROKEN_PIPE?
CUSTOM_ERROR_04  = 21590004h
CUSTOM_ERROR_05  = 21590005h
CUSTOM_ERROR_06  = 21590006h
CUSTOM_ERROR_07  = 21590007h
CUSTOM_ERROR_08  = 21590008h
CUSTOM_ERROR_09  = 21590009h
CUSTOM_ERROR_0A  = 2159000Ah
CUSTOM_ERROR_0B  = 2159000Bh            ; INVALID_USER_BUFFER?
CUSTOM_ERROR_0D  = 2159000Dh
CUSTOM_ERROR_64  = 21590064h
CUSTOM_ERROR_65  = 21590065h
CUSTOM_ERROR_66  = 21590066h
CUSTOM_ERROR_67  = 21590067h
CUSTOM_ERROR_68  = 21590068h
CUSTOM_ERROR_69  = 21590069h
CUSTOM_ERROR_C9  = 215900C9h            ; NO_VALID_ADDR?
CUSTOM_ERROR_CA  = 215900CAh            ; NO_VALID_PORT?
CUSTOM_ERROR_CB  = 215900CBh
CUSTOM_ERROR_CC  = 215900CCh
```

Sample C - inetpub.dll (Resource: 102)

Original filename: CARBON.dll

Internal name: Carbon v3.51

Files from inetpub.dll

```
 \\.\IdeDrive1\config.txt
\\.\IdeDrive1\Tasks\\task.txt
\\.\IdeDrive1\Tasks\\task_system.txt
\\.\IdeDrive1\log.txt
\\.\IdeDrive1\Results\result.txt
```

Mutexes from inetpub.dll

```
 Global\\MSMMC.StartupEnvironment.PPT
Global\\411A5195CD73A8a710E4BB16842FA42C
Global\\881F0621AC59C4c035A5DC92158AB85E
Global\\MSCTF.Shared.MUTEX.RPM
```

```
Global\\WindowsShellHWDetection
Global\\MSDBG.Global.MUTEX.ATF
```

thread 2:

In a 10 minutes loop check server availability by doing a HTTP POST (HTTP/1.0) to a server/port configured in

```
\\.\IdeDrive1\config.txt
```

in *CW_INET* section *address* with user agent

```
Mozilla/4.0 (compatible; MSIE 6.0)
```

but only if a valid internet connection was successfully probed:

```
char isInternetConnectionWorking()
{
  char result;
  HINTERNET hInternetOpen;

  result = 0;
  if ( InternetAttemptConnect(0) )
  {
    result = 0;
  }
  else
  {
    hInternetOpen = InternetOpenA("Mozilla/4.0 (compatible; MSIE 6.0)", 0, 0, 0, 0);
    if ( hInternetOpen )
    {
      if ( HttpConnect(hInternetOpen, "update.microsoft.com")
        || HttpConnect(hInternetOpen, "windowsupdate.microsoft.com")
        || HttpConnect(hInternetOpen, "207.46.18.94")
        || HttpConnect(hInternetOpen, "207.46.253.125")
        || HttpConnect(hInternetOpen, "microsoft.com")
        || HttpConnect(hInternetOpen, "207.46.250.119")
        || HttpConnect(hInternetOpen, "207.46.249.56")
        || HttpConnect(hInternetOpen, "207.46.249.57") )
        result = 1;
      InternetCloseHandle(hInternetOpen);
    }
    else
    {
      result = 0;
    }
  }
  return result;
}
```

thread 3:

The actions described below are only taken if the following programs are *not* running

- tcpdump.exe
- windump.exe
- ethereal.exe
- wireshark.exe
- ettercap.exe
- snoop.exe

The following is the main (endless) loop of this thread:

```
LOOP:
   if ( do_HTTP_GET(hInternetConnect, &base_string) )
   {
     while ( isCapturingPackets() == 1 )
       Sleep(0xEA60u);
     while ( sub_20009871(hInternetConnect, ::Dest, &lpszServerName, &base_string) )
       ;
     while ( sub_200075C0(hInternetConnect, ::Dest, &lpszServerName, &base_string) )
       Sleep(0x3E8u);
     goto LOOP;
   }
```

It starts in *do_HTTP_GET()* with a *HTTP GET* (*HTTP/1.1*) to server/port taken from

```
\\.\IdeDrive1\config.txt
```

in *CW_INET* section *address* with user agent

```
Mozilla/4.0 (compatible; MSIE 6.0)
```

with script name and query as follows:

```
auth.cgi?mode=query&id=%u:%u:%u:%u&serv=%s&lang=en&q=%u-%u&date=%s
```

where the format strings are filled in accordingly.

```
serv=
```

is filled pseudorandomly with a host from the following list:

- www.yahoo.com
- www.bbc.com
- www.astalavista.com
- www.google.com

- www.eagames.com
- www.asus.com
- www.microsoft.com
- windowsupdate.microsoft.com
- search.microsoft.com
- www.hp.com
- www.altavista.com
- www.3com.com
- www.dell.com
- www.sun.com
- www.easports.com
- search.google.com

perhaps to make a reasonable appearance or to mislead log analysts who filter out common domain names.

When a successful handle is returned, a file is being downloaded and stored in the virtual file system.

What follows is a *GET* in *HTTP/1.0* on

```
default.asp?act=%u&id=%u&item=%u&event_id=%u&cln=%u&flt=%u&serv=%s&t=%ld&mode=query&lang=en&date=%s
```

This code is part of *sub_20009871*, which continues to serve the *frag.np/frag.tcp* part.

In *sub_200075C0* another *POST* in *HTTP/1.0* to

```
default.asp?act=%u&id=%u&item=%u&event_id=%u&cln=%u&flt=%u&serv=%s&t=%ld&mode=query&lang=en&date=%s
```

follows.

The purpose of the two functions is not clear, yet.

load_transports()

In this module, the following transport or communication modules are present:

- Type 1: tcp, b2m
- Type 2: np, frag, m2b

This corresponds to the transports found in Sample D.

3rd party code

bzip2/libbzip2

The compiled code of bzip2/libbzip2, a program and library for lossless block-sorting data compression, was identified, coming from http://svn.apache.org/repos/asf/labs/axmake/trunk/src/libuc++/srclib/bzip2/compress.c.

> bzip2/libbzip2 version 1.0.5 of 10 December 2007

> Copyright (C) 1996-2007 Julian Seward jseward@bzip.org

Using the source code without including the author's Copyright statement, the conditions and the disclaimer is an infringement of the software license:

http://svn.apache.org/repos/asf/labs/axmake/trunk/src/libuc++/srclib/bzip2/LICENSE

Other analysis

Analysis of check-in messages

Check-in messages of Sample C and D (unique)

```
 $Id: b2_to_m2_stub.c 5273 2007-01-23 17:41:15Z vlad $
$Id: b_tcp.c 8474 2007-09-19 15:40:39Z vlad $
$Id: hide_module_win32.c 10189 2008-11-25 14:25:41Z gilg $
$Id: l1_check.c 4477 2006-08-28 15:58:21Z vlad $
$Id: load_lib_win32.c 10180 2008-11-20 12:13:01Z gilg $
$Id: m2_to_b2_stub.c 4477 2006-08-28 15:58:21Z vlad $
$Id: m_frag.c 8715 2007-11-29 16:04:46Z urik $
$Id: m_np.c 8825 2008-01-10 13:13:15Z vlad $
$Id: mutex.c 3940 2006-03-20 16:47:16Z vlad $
$Id: np_win32_common.c 4483 2006-08-30 13:13:51Z vlad $
$Id: rw_lock.c 4482 2006-08-30 13:07:14Z vlad $
$Id: t_byte1.c 5324 2007-01-30 12:45:35Z vlad $
$Id: t_manager.c 8715 2007-11-29 16:04:46Z urik $
$Id: t_message1.c 5290 2007-01-26 11:15:03Z vlad $
$Id: t_status.c 5666 2007-03-19 16:18:00Z vlad $
$Id: t_utils.c 5503 2007-02-26 13:14:30Z vlad $
$Id: thread.c 4593 2006-10-12 11:43:29Z urik $
```

Developers

Sample C and D contain author names of three people:

- vlad
- gilg
- urik

Newer samples, for instance the one from BAE, contain only two:

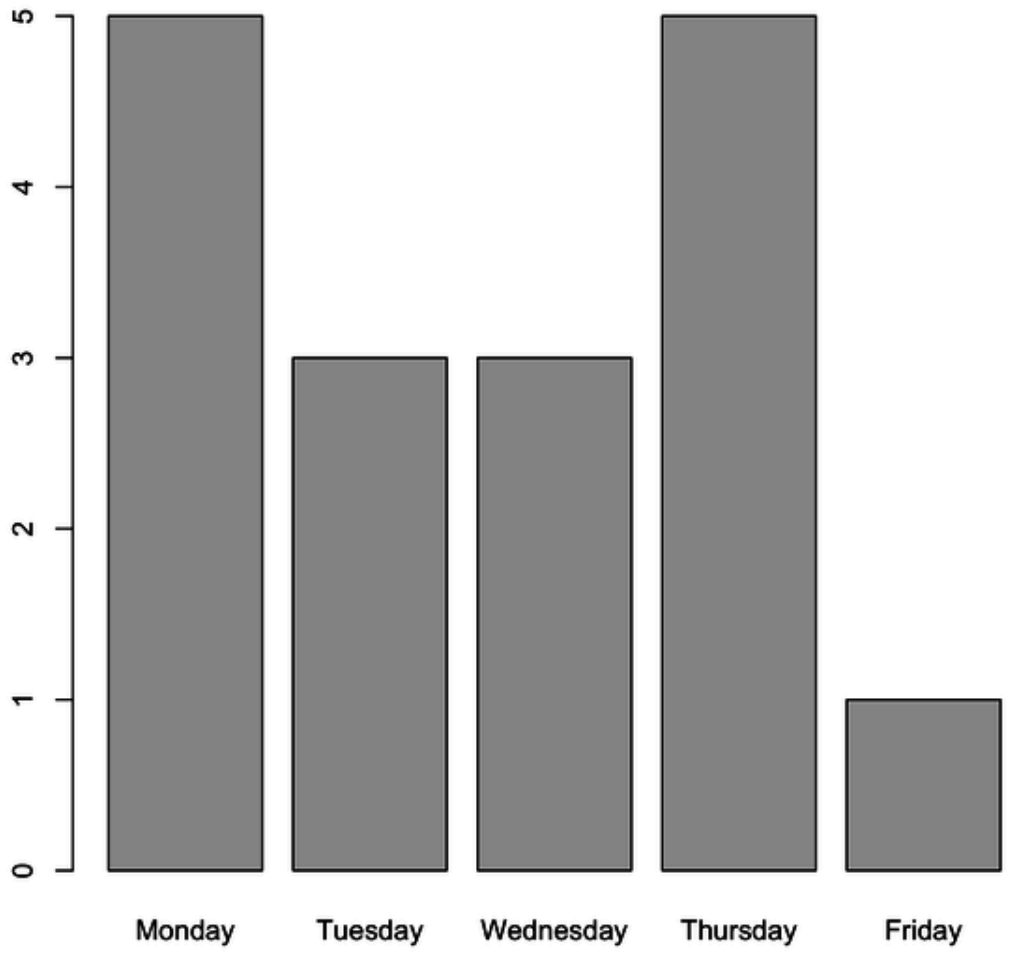
- vlad
- gilg

Check-in period

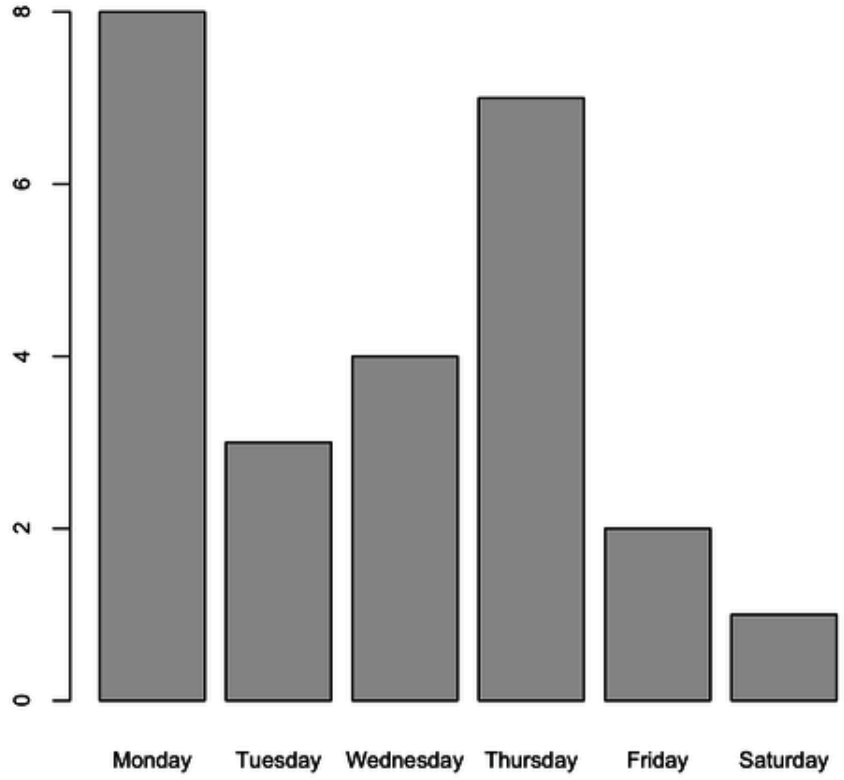
First check-in: 2006-03-20

Last check-in: 2008-11-25

Check-in dates



When incorporating the check-in dates of the BAE sample, the following graph shows that someone checked-in a file once during a Saturday.



Language deficits

A small collection of strings demonstrates the language deficits, mainly distinguishable as:

- Use of backticks instead of apostrophes by some of the developers
- Problems using past tense by some developers
- Spelling
- Mistranslated terms
- Oversights

Examples:

```
 win32 detect...
x64 detect...
CretaFileA(%s):
Can`t open SERVICES key
error has been suddenly occured
timeout condition has been occured inside call of function
OPER|Survive me, i`m close to death... free space less than 5%%...|\n
OPER|Sniffer '%s' running... ooopppsss...|\n
Task not execute. Arg file failed.
Update failed =((  Can`t create file.
can`t get characs.\n
Internal command not support =((\n
L|-1|can`t get characs %s|\n
```

Recommendations

- CIRCL recommends to review the IOCs of this report and compare them with servers in the infrastructure of your organization which produce log files including proxies, A/V and system logs. As this family of malware might be difficult to detect from a network perspective, we recommend to perform check of the indicators at the system level.

Classification of this document

TLP:WHITE information may be distributed without restriction, subject to copyright controls.

Revision

- Version 0.9 July 10, 2014 work-in-progress (not a final release) (TLP:WHITE)

References
