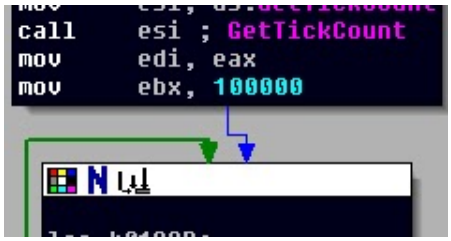
Disclosure of another 0day malware - Analysis of 2nd Dropper and 3rd Dropper (Part 2)

malware-reversing.com/2012/12/3-disclosure-of-another-0day-malware_15.html



R136a1 December 15, 2012 No comments

In the second Part of this series we analyze the downloaded file (2nd Dropper) and the dropped file (3rd Dropper). At time of this analysis the files weren't uploaded on Virustotal, so I guess the detection rates are very low, if at all.

2nd Dropper
Sample: msmvs.exe
Size: 80.388 Bytes
Timestamp: 25.07.2012 06:51:13
MD5: 66F368CAB3D5E64475A91F636C87AF15

3rd Dropper
Sample: conhost.dll
Size: 62.976 Bytes
Timestamp: 25.09.2012 08:23:13
MD5: F1704AAF08CD66A2AC6CF8810C9E07C2

2nd Dropper (msmvs.exe)

The file starts with a common Anti Debugging technique by using the GetTickCount() function twice and comparing the results. When you run the executable, the code between the two GetTickCount() functions is executed so fast that the functions return the same results. When you debug the code, the second GetTickCount() result is different from the first and the executable exits without doing anything harmful:

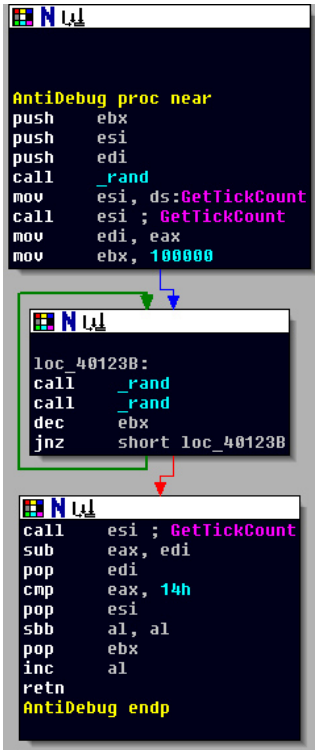


Figure 1: Simple Anti Debug trick

Next, it retrieves the temporary path with help of the GetTempPath() function and stores it for later use. Note that the temporary path can differ (http://msdn.microsoft.com/en-us/library/windows/desktop/aa364992%28v=vs.85%29.aspx):

"The GetTempPath function checks for the existence of environment variables in the following order and uses the first path found:

The path specified by the TMP environment variable.
The path specified by the TEMP environment variable.
The path specified by the USERPROFILE environment variable.
The Windows directory."

Then it gets the fully qualified path for its own file to open it afterwards (GetModuleFileName() + CreateFile()). If that fails the Dropper again exits without doing anything malicious. Now the Dropper sets a file pointer to the beginning of the file to be dropped (SetFilePointer()) and copies it into a buffer by using ReadFile() function. Thereafter it builds the string "C:\Documents and Settings\<Username>\Local Settings\Temp\conhost.dll" with the before retrieved temporary path. Then it checks if there is already the file "conhost.dll" in the temporary folder (probably to check for an older version of the malware) and renames it if it exists to "conhost.dll.tmp". There follows the decryption of the file in the buffer (file to be dropped) and finally the file is written to disk in the temporary folder as "conhost.dll". At last the file (.dll) is loaded by forming the string "rundll32.exe C:\Documents and Settings\<Username>\Local Settings\Temp\conhost.dll,Start" and using CreatProcess() function to start rundll32.exe program. Finally "conhost.dll" gets deleted from temporary folder to cover the tracks.

Now let's take a look into the 3rd Dropper.

3rd Dropper (conhost.dll)

At first it also decrypts some function names, library names and other strings for subsequent use. Then it again dynamically resolves the API adresses of various functions (LoadLibrary() + GetProcAddress()). There follows the same anti (AV) emulation technique (two MMX instructions) as it was used in the initial Dropper and the Downloader.

```
push    esi
push    edi
push    offset unk_100077CC
push    0
push    796h
push    offset aKernel32_dll ; "KERNEL32.dll"
call    Decrypt
push    offset aKernel32_dll ; "KERNEL32.dll"
call    ds:LoadLibraryA
mov     var_ImageBaseKernel32, eax
push    offset aGetprocessheap ; "GetProcessHeap"
mov     eax, var_ImageBaseKernel32
push    eax             ; hModule
call    ds:GetProcAddress
call    eax
mov     addr_GetProcessHeap, eax
push    offset aHeapalloc ; "HeapAlloc"
mov     ecx, var_ImageBaseKernel32
push    ecx             ; hModule
call    ds:GetProcAddress
mov     addr_HeapAlloc, eax
push    offset aHeapfree ; "HeapFree"
mov     edx, var_ImageBaseKernel32
push    edx             ; hModule
call    ds:GetProcAddress
mov     addr_HeapFree, eax
push    offset aMsvcrt_dll ; "MSVCRT.dll"
call    ds:LoadLibraryA
mov     var_ImageBaseMSVCRT, eax
push    offset aMemset  ; "memset"
mov     eax, var_ImageBaseMSVCRT
```

Figure 2: Strings decryption and function address resolving

There is also the same information retrieved and stored into a string as we saw in the Downloader:

"<ComputerName><VolumeSerialNumber>-<OSMajorVersion>_<OSMinorVersion>"

In the Registry Key "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer" it creates a REG_BINARY entry with name "IP" and 6 random hex values (used as Encryption Key) and 14 Null byte values that were created before with help of GetTickCount(), srand() and rand() functions:

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer] -> "IP" = hex:2d,3c,3e,1c,84,30,00,00,00,00,00,00,00,00,00,00,00,00,00,00

With help of this six random bytes the following strings are encrypted and appended to the REG_BINARY "IP" entry. We can see two more IPs to C&C servers, one is used in the final Payload (**200.74.244.118**), the other isn't available anymore (see Appendix for whois information):

"600000"
"600000"
"<ComputerName><VolumeSerialNumber>-<OSMajorVersion>_<OSMinorVersion>"
"200.74.244.118"
"123.100.229.59"
"bint"
"10000"
"600000"
"1"

In my case I get the following encrypted binary hex values in "IP":

2d 3c 3e 1c 84 30 06 06 14 0e 0e 04 05 06 01 00 00 00 00 00 6e 72 2c 94 6c 54 6e 0d 2f 95 6e 6d 1a 71 2d 91 0d 66 79 2b 2f d2 7d 79 0f 69 3d 81 79 18 08 79 21 b3 2f 12 5f 1b 3c be 29 02 46 07 3c b5 2e 7e 6b 12 35 af 3d 68 6a 05 3e b6 3a 06 2b 53 68 ed af 41 79 15 70 9d 90 5f 09 18 0f 85

The same strings are encrpted and added to the Registry Key "HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer" but with other 6 random hex values as encryption Key.

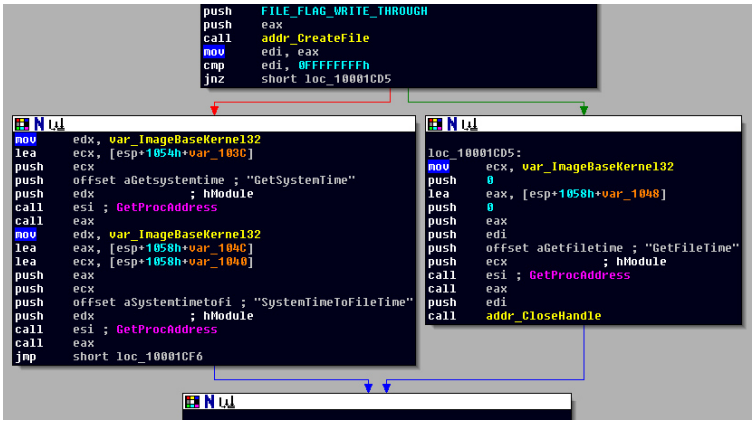


Figure 3: Get "els.dll" or system file time

As we already saw in the initial Dropper, the 3rd Dropper also searches for the file "els.dll" in system directory to get its file time and uses system time if it fails. And as we saw in the 2nd Dropper (conhost.dll), the 3rd Dropper also uses the same technique to load the file to be dropped (final Payload) into a buffer for decryption by using GetModuleFileName(), CreateFile(), SetFilePointer() and ReadFile() functions. Then it looks if there is already a netui.dll in system folder (C:\Windows\System32) and creates the final Payload by writing the bytes from the buffer into the netui.dll file, if this is not the case. Thereafter it sets the file's time to one of the two retrieved before. I think the malware author chosed the name "netui.dll", because in Windows XP system directory there exist the files "netui0.dll", "netui1.dll", "netui2.dll", so it doesn't look suspicious.

Then the Windows Service "Network User Interface" is created with description "Provides user network interface service for secure connections" and "netui.dll" as application. Again as we saw in the initial Dropper, this is done by using "C:\WINDOWS\system32\svchost.exe -k NtShvcs" as

application path and registering "NtShvcs" as a Service in registry. The following registry keys with the appropriate values are created:

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\svchost\NtShvcs
 |-> CoInitializeSecurityParam = 0x00000000

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Network User Interface
 |-> parameters
 Value: ServiceDll = C:\WINDOWS\system32\netui.dll
 Value: ServiceDllUnloadOnStop = 0x00000001

HKEY_LOCAL_MACHINE\software\microsoft\windows nt\currentversion\svchost\ntsvcs
 Value: NtShvcs = Network User Interface

Then the service is started by using the StartService() function. If for some reason the Service creation failed, the malware startup is realized by adding its class to the SharedTaskScheduler registry key:

HKEY_LOCAL_MACHINE\Software\Classes\CLSID\{61113868-6B5D-4195-8966-B26462B909FA}
 |-> InProcServer32 = C:\WINDOWS\system32\netui.dll
 Value: ThreadingModel = Apartment

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Explorer\SharedTaskScheduler
 Value: {61113868-6B5D-4195-8966-B26462B909FA} = NetWork User Interface

This way the netui.dll file is automatically loaded on every Windows startup. And if for some reason the creation of this persistency technique also failed, a simple entry of rundll32.exe with the appropriate parameter is set to the Run key:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
 Value: RUNDLL32.EXE "C:\WINDOWS\system32\netui.dll",Init1

Now if the installation and setup of the file in the system directory (C:\Windows\System32) failed, the malware tries to accomplish the same procedures as above in the local application data folder (SHGetFolderPath() -> C:\Documents and Settings\<Username>\Local Settings\Application Data).

That's all of the 3rd Dropper's functionality. In this second Part we analyzed the downloaded file which turned out to be just another Dropper. This file drops yet another Dropper, which in turn drops the final Payload. So let's move to the most interesting part, the final Payload.

Appendix

Whois for 200.74.244.118:

IP Location: Panama Panama Panama Cyber Cast International S.a.
ASN: AS27956
Resolve Host: host-200-74-244-118.ccipanama.com

IP Address: 200.74.244.118 [Whois] [Reverse-Ip] [Ping] [DNS Lookup] [Traceroute]

NetRange: 200.0.0.0 - 200.255.255.255
CIDR: 200.0.0.0/8
OriginAS:
NetName: LACNIC-200
NetHandle: NET-200-0-0-0-1
Parent:
NetType: Allocated to LACNIC
Comment: This IP address range is under LACNIC responsibility for further
Comment: allocations to users in LACNIC region.
Comment: Please see http://www.lacnic.net/ for further details, or check the
Comment: WHOIS server located at http://whois.lacnic.net
RegDate: 2002-07-27
Updated: 2010-07-21
Ref: http://whois.arin.net/rest/net/NET-200-0-0-0-1

OrgName: Latin American and Caribbean IP address Regional Registry
OrgId: LACNIC
Address: Rambla Republica de Mexico 6125
City: Montevideo
StateProv:
PostalCode: 11400
Country: UY
RegDate: 2002-07-27
Updated: 2011-09-24
Ref: http://whois.arin.net/rest/org/LACNIC

ReferralServer: whois://whois.lacnic.net

OrgAbuseHandle: LACNIC-ARIN
OrgAbuseName: LACNIC Whois Info
OrgAbusePhone: 999-999-9999
OrgAbuseEmail: whois-contact@lacnic.net
OrgAbuseRef: http://whois.arin.net/rest/poc/LACNIC-ARIN

OrgTechHandle: LACNIC-ARIN
OrgTechName: LACNIC Whois Info
OrgTechPhone: 999-999-9999
OrgTechEmail: whois-contact@lacnic.net
OrgTechRef: http://whois.arin.net/rest/poc/LACNIC-ARIN

== Additional Information From whois://whois.lacnic.net ==

inetnum: 200.74.240/21
status: allocated
aut-num: N/A

owner: Cyber Cast International, S.A.
ownerid: PA-CCIS-LACNIC
responsible: Jorge Moreno
address: Addison House Plaza Suite 20, 507, 264-0852
address: 6-3783 - Panama - PA
country: PA
phone: +507 264-0852 []
owner-c: CCS2
tech-c: CCS2
abuse-c: CCS2
inetrev: 200.74.244/24
nserver: NS1.CYBERCASTCO.COM
nsstat: 20121103 AA
nslastaa: 20121103
nserver: NS2.CYBERCASTCO.COM
nsstat: 20121103 AA
nslastaa: 20121103
created: 20090331
changed: 20090331

nic-hdl: CCS2
person: Cyber Cast International, S.A.
e-mail: info@ccipanama.com
address: Addison House Plaza Suite 20, 507, 264-0852
address: 6-3783 - panama - pa
country: PA
phone: +507 264-0852 []
created: 20050405
changed: 20080923

Whois for 123.100.229.59:

IP Location: Australia Australia Sydney Ozfrontiers Pty Ltd
ASN: AS55736
IP Address: 123.100.229.59 [Whois] [Reverse-Ip] [Ping] [DNS Lookup] [Traceroute]

inetnum: 123.100.228.0 - 123.100.229.255
netname: OZFRONTIERS-COLO-SYD-AU
country: AU
descr: Ozfrontiers Pty Ltd
descr: A Subsidiary of Webvisions Pte Ltd (Singapore)
descr: Sydney, Australia
descr: Dedicated and Co-location Servers
descr: For spam/abuse issues, please e-mail abuse@webvisions.com
admin-c: IP6-AP
tech-c: MH352-AP

tech-c: JK1424-AP
status: ASSIGNED NON-PORTABLE
changed: jason.koh@webvisions.com 20070123
mnt-by: MAINT-SG-WEBVISIONS
source: APNIC

person: Indra Pramana
address: Webvisions Pte Ltd
address: 75 Science Park Drive
address: #02-06/08 Cintech II
address: Singapore Science Park I
address: Singapore 118255
country: SG
phone: +65-6773-9492
fax-no: +65-6773-9389
e-mail: indra@webvisions.com
nic-hdl: IP6-AP
mnt-by: MAINT-SG-WEBVISIONS
changed: indra@webvisions.com 20020719
source: APNIC

person: Mohamad Zulkifli Hanafi
nic-hdl: MH352-AP
e-mail: zukifli@webvisions.com
address: 75 Science Park Drive
address: #02-06/08 Cintech II
address: Singapore Science Park I
address: Singapore 118255
phone: +65-6773-9550
fax-no: +65-6773-9389
country: SG
changed: indra@webvisions.com 20030303
mnt-by: MAINT-SG-WEBVISIONS
source: APNIC

person: Jason Koh
nic-hdl: JK1424-AP
e-mail: jason.koh@webvisions.com
address: 75 Science Park Drive
address: #02-06/08 Cintech II
address: Singapore Science Park I
address: Singapore 118255
phone: +65-6773-9490
fax-no: +65-6773-9389
country: SG
changed: indra@webvisions.com 20070123

mnt-by: MAINT-SG-WEBVISIONS
source: APNIC