

Malware Analysis Report [dofail.exe]

Prepared by: Sameer Patil

Mentors: Amit Malik & Monnappa KA

[SecurityXploded Student Mentorship Programme]

General Information

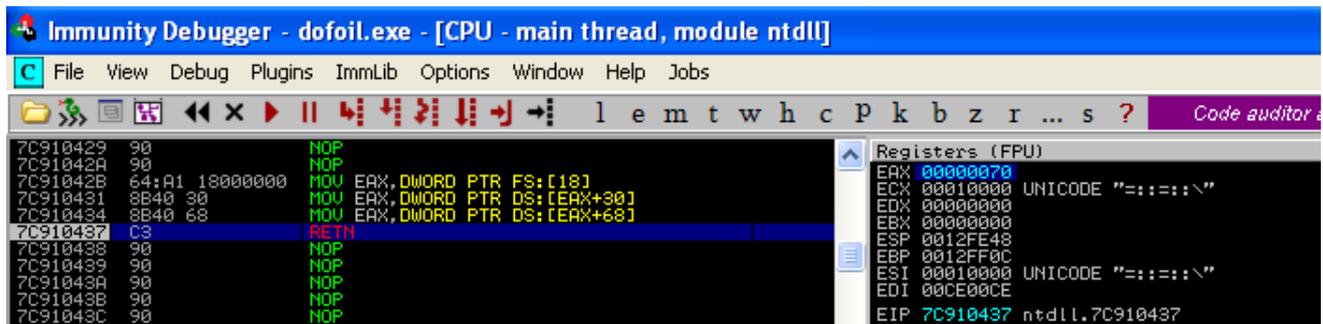
- File name: dofail.exe
- MD5: fbeb99d329cbaa396b148d37a32aae97
- SHA-1: 57747fd86269fa547a94e47d565a71285993e284
- File Size: 281 KB
- First submission on: 19-9-2012
- Identified as: TrojanDownloader:Win32/Dofail.R [Microsoft]
Trojan-Spy.Win32.Zbot.eqmk [Kaspersky Lab]
PWS-Zbot.gen.ala [McAfee]
TrojWare.Win32.Injector.VDK [Comodo]

Analysis Overview:

Dofail.exe is a Trojan dropper. It downloads and installs other programs without user consent. This may include installation of more powerful and latest malwares by the author. The Trojan also modifies the registry to trigger its execution at system startup itself. Its a very carefully written malware which also uses multiple anti-detection techniques.

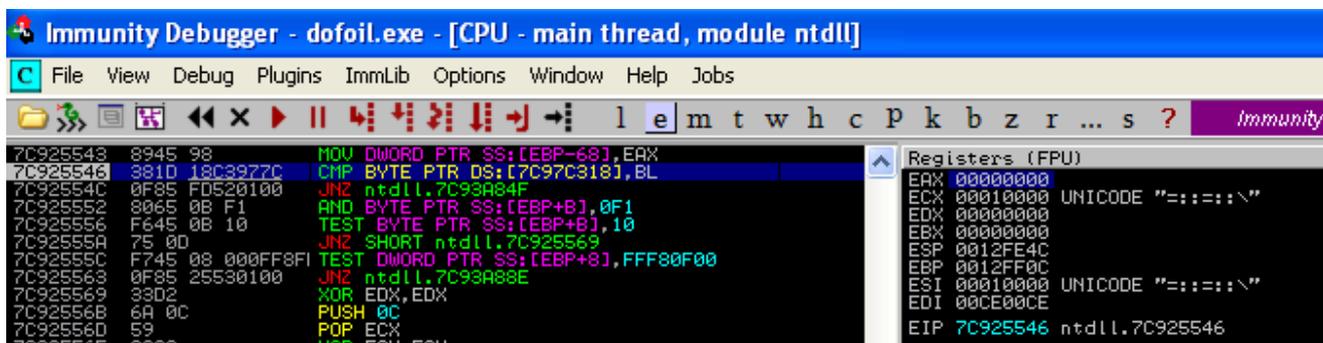
Technical Analysis

1. The malware deploys several debugger detection techniques to hide its activities.
 - a. One of them is the use of “PEB.NTGlobalFlag” which is present at offset 0x68 of PEB. The flag is set 0 if no debugger is running. If any debugger is running its value is 0x70 which signifies the following flags are set:
FLG_HEAP_ENABLE_TAIL_CHECK (0x10)
FLG_HEAP_ENABLE_FREE_CHECK (0x20)
FLG_HEAP_VALIDATE_PARAMETERS (0x40)



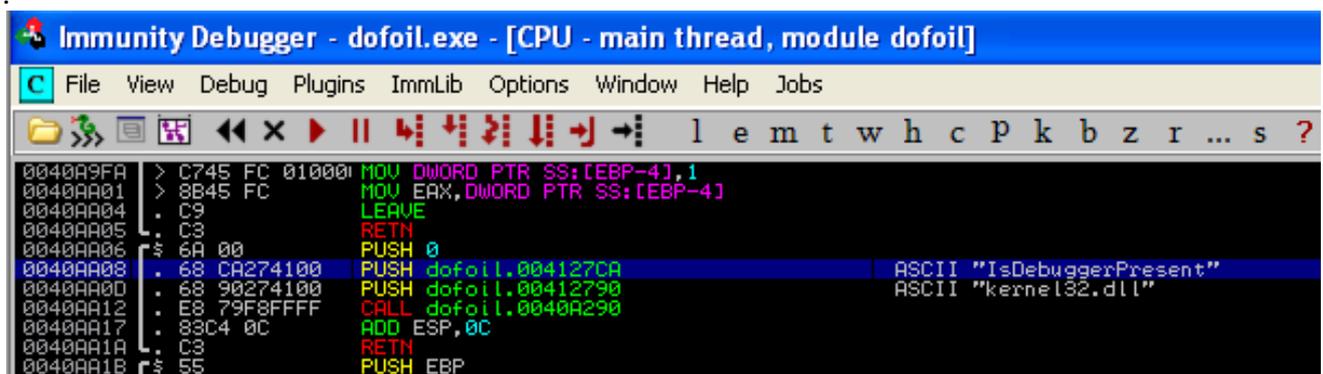
```
Immunity Debugger - dofoil.exe - [CPU - main thread, module ntdll]
File View Debug Plugins ImmLib Options Window Help Jobs
7C910429 90 NOP
7C91042A 90 NOP
7C91042B 64:A1 18000000 MOV EAX,DWORD PTR FS:[18]
7C910431 8B40 30 MOV EAX,DWORD PTR DS:[EAX+30]
7C910434 8B40 68 MOV EAX,DWORD PTR DS:[EAX+68]
7C910437 C3 RETN
7C910438 90 NOP
7C910439 90 NOP
7C91043A 90 NOP
7C91043B 90 NOP
7C91043C 90 NOP
Registers (FPU)
EAX 00000070
ECX 00010000 UNICODE "::::\"
EDX 00000000
EBX 00000000
ESP 0012FE48
EBP 0012FF0C
ESI 00010000 UNICODE "::::\"
EDI 00CE00CE
EIP 7C910437 ntdll.7C910437
```

This check can be bypassed by manually changing the value stored in EAX from “00000070” to 0.



```
Immunity Debugger - dofoil.exe - [CPU - main thread, module ntdll]
File View Debug Plugins ImmLib Options Window Help Jobs
7C925543 8945 98 MOV DWORD PTR SS:[EBP-68],EAX
7C925546 381D 18C3977C CMP BYTE PTR DS:[7C97C318],BL
7C92554C 0F85 FD520100 JNZ ntdll.7C93A84F
7C925552 8065 0B F1 AND BYTE PTR SS:[EBP+B],0F1
7C925556 F645 0B 10 TEST BYTE PTR SS:[EBP+B],10
7C92555A 75 0D JNZ SHORT ntdll.7C925569
7C92555C F745 08 00FF8F1 TEST DWORD PTR SS:[EBP+8],FFF80F00
7C925563 0F85 25530100 JNZ ntdll.7C93A88E
7C925569 33D2 XOR EDX,EDX
7C92556B 6A 0C PUSH 0C
7C92556D 59 POP ECX
7C92556F 33D2 XOR EDX,EDX
Registers (FPU)
EAX 00000000
ECX 00010000 UNICODE "::::\"
EDX 00000000
EBX 00000000
ESP 0012FE4C
EBP 0012FF0C
ESI 00010000 UNICODE "::::\"
EDI 00CE00CE
EIP 7C925546 ntdll.7C925546
```

- b. Another method used is “Kernel32.IsDebuggerPresent” API as shown below. Here also we make the value of EAX register as 0 to bypass the protection.



```
Immunity Debugger - dofoil.exe - [CPU - main thread, module dofoil]
File View Debug Plugins ImmLib Options Window Help Jobs
0040A9FA > C745 FC 010001 MOV DWORD PTR SS:[EBP-4],1
0040AA01 > 8B45 FC MOV EAX,DWORD PTR SS:[EBP-4]
0040AA04 . C9 LEAVE
0040AA05 . C3 RETN
0040AA06 $ 6A 00 PUSH 0
0040AA08 . 68 CA274100 PUSH dofoil.004127CA
0040AA0D . 68 90274100 PUSH dofoil.00412790
0040AA12 . E8 79F8FFFF CALL dofoil.0040A290
0040AA17 . 83C4 0C ADD ESP,0C
0040AA1A . C3 RETN
0040AA1B $ 55 PUSH EBP
Registers (FPU)
EAX 00000000
ECX 00010000 UNICODE "::::\"
EDX 00000000
EBX 00000000
ESP 0012FE4C
EBP 0012FF0C
ESI 00010000 UNICODE "::::\"
EDI 00CE00CE
EIP 0040A9FA dofoil.0040A9FA
```

2. After we bypass these tests the malware unpacks itself and generates more code at different locations in memory.
 - a. After unpacking, it tries to find out whether the malware is being run inside an emulator. It accesses the registry through “Advapi32.RegOpenKeyExW” API and looks for keys present in “System\ControlSet001\Services\Disk\Enum”. Enum key stores values for the various drives present in the system. The malware checks for the presence of emulators through strings like vmware, vbox, virtual, qemu etc.

```

0040C4BC . 57      PUSH EDI
0040C4BD . BE 08020000  MOV ESI, 208
0040C4C2 . 56      PUSH ESI
0040C4C3 . C745 E4 B12D4  MOV DWORD PTR SS:[EBP-1C], dofoil.00412D1  UNICODE "VMWARE"
0040C4CA . C745 E8 BF2D4  MOV DWORD PTR SS:[EBP-18], dofoil.00412D1  UNICODE "VBOX"
0040C4D1 . C745 EC C92D4  MOV DWORD PTR SS:[EBP-14], dofoil.00412D1  UNICODE "VIRTUAL"
0040C4D8 . C745 F0 D92D4  MOV DWORD PTR SS:[EBP-10], dofoil.00412D1  UNICODE "QEMU"
0040C4DF . E8 4A7BFFFF  CALL dofoil.0040402E
0040C4E4 . 56      PUSH ESI
0040C4E5 . 33FF     XOR EDI, EDI
0040C4E7 . 8BD8     MOV EBX, EAX
0040C4E9 . 57      PUSH EDI
0040C4EA . 53      PUSH EBX
0040C4EB . E8 E07BFFFF  CALL dofoil.004040D0
0040C4F0 . 8D45 FC    LEA EAX, DWORD PTR SS:[EBP-4]
0040C4F3 . 50      PUSH EAX
0040C4F4 . 68 19000200  PUSH 19000200
0040C4F9 . 57      PUSH EDI
0040C4FA . 68 E32D4100  PUSH dofoil.00412DE3
0040C4FF . 68 02000080  PUSH 80000002
0040C504 . 6A 05      PUSH 5
0040C506 . 8975 F8    MOV DWORD PTR SS:[EBP-8], ESI
0040C509 . 68 2A2A4100  PUSH dofoil.00412A2A
0040C50E . BE 042A4100  MOV ESI, dofoil.00412A04
0040C513 . 56      PUSH ESI
0040C514 . E8 770DFFFF  CALL dofoil.0040A290
0040C519 . 83C4 30    ADD ESP, 30
  
```

- b. It also checks whether malware is being run inside Anubis Sandbox by searching for the string “InsideTM”.

```

0040C481 . 83C4 28    ADD ESP, 28
0040C484 . 85C0     TEST EAX, EAX
0040C486 . 74 27     JE SHORT dofoil.0040C4AF
0040C488 . 68 912D4100  PUSH dofoil.00412D91  UNICODE "InsideTM"
0040C48D . 57      PUSH EDI
0040C48E . E8 5EEAFFFF  CALL dofoil.0040AEF1
0040C493 . 59      POP ECX
0040C494 . 59      POP ECX
  
```

- c. Further anti-analysis techniques check for various other reverse-engineering softwares like OllyDbg, Process Explorer, Process Monitor, WinDbg, Cain & Abel, TCPView, Portmon etc.

```

Immunity Debugger - dofoil.exe - [CPU - main thread, module dofoil]
File View Debug Plugins ImmLib Options Window Help Jobs
l e m t w h c p k b z r ... s ?
0040A97C . 57 PUSH EDI
0040A97D . BE 852C4100 MOV ESI,dofoil.00412C85 UNICODE "PROCEXPL"
0040A982 . E8 4DFDFFFF CALL dofoil.0040A6D4
0040A987 . 59 POP ECX
0040A988 . 85C0 TEST EAX,EAX
0040A98A . 75 10 JNZ SHORT dofoil.0040A99C
0040A98C . 57 PUSH EDI
0040A98D . BE 972C4100 MOV ESI,dofoil.00412C97 UNICODE "PROCMON_WINDOW_CLASS"
0040A992 . E8 3DFDFFFF CALL dofoil.0040A6D4
0040A997 . 59 POP ECX
0040A998 . 85C0 TEST EAX,EAX
0040A99A . 74 02 JE SHORT dofoil.0040A99E
0040A99C . > 8BC7 MOV EAX,EDI
0040A99E . > 5F POP EDI
0040A99F . 5E POP ESI
0040A9A0 . RETN
0040A9A1 . 56 PUSH ESI
0040A9A2 . 6A 01 PUSH 1
0040A9A4 . BE 192C4100 MOV ESI,dofoil.00412C19 UNICODE "icu_dbg"
0040A9A9 . E8 26FDFFFF CALL dofoil.0040A6D4
0040A9AE . 59 POP ECX
0040A9AF . 85C0 TEST EAX,EAX
0040A9B1 . 75 24 JNZ SHORT dofoil.0040A9D7
0040A9B3 . 6A 01 PUSH 1
0040A9B5 . BE 292C4100 MOV ESI,dofoil.00412C29 UNICODE "OLLYDBG"
0040A9B8 . E8 15FDFFFF CALL dofoil.0040A6D4
0040A9BF . 59 POP ECX
0040A9C0 . 85C0 TEST EAX,EAX
0040A9C2 . 75 13 JNZ SHORT dofoil.0040A9D7
0040A9C4 . 6A 01 PUSH 1
0040A9C6 . BE 392C4100 MOV ESI,dofoil.00412C39 UNICODE "WinDbgFrameClass"
0040A9CB . E8 04FDFFFF CALL dofoil.0040A6D4
0040A9D0 . 59 POP ECX
0040A9D1 . 85C0 TEST EAX,EAX
0040A9D3 . 75 02 JNZ SHORT dofoil.0040A9D7

```

```

Immunity Debugger - dofoil.exe - [CPU - main thread, module dofoil]
File View Debug Plugins ImmLib Options Window Help Jobs
l e m t w h c p k b z r ... s ?
0040A8F1 . C3 RETN
0040A8F2 . 56 PUSH ESI
0040A8F3 . 6A 01 PUSH 1
0040A8F5 . BE C12C4100 MOV ESI,dofoil.00412CC1 UNICODE "gdkWindowTemp"
0040A8FA . E8 05FDFFFF CALL dofoil.0040A6D4
0040A8FF . 59 POP ECX
0040A900 . 85C0 TEST EAX,EAX
0040A902 . 75 43 JNZ SHORT dofoil.0040A947
0040A904 . 6A 01 PUSH 1
0040A906 . BE D02C4100 MOV ESI,dofoil.00412CD0 UNICODE "gdkWindowTopLevel"
0040A90B . E8 C4FDFFFF CALL dofoil.0040A6D4
0040A910 . 59 POP ECX
0040A911 . 85C0 TEST EAX,EAX
0040A913 . 75 32 JNZ SHORT dofoil.0040A947
0040A915 . 6A 01 PUSH 1
0040A917 . BE 012D4100 MOV ESI,dofoil.00412D01 UNICODE "TCPViewClass"
0040A91C . E8 B3FDFFFF CALL dofoil.0040A6D4
0040A921 . 59 POP ECX
0040A922 . 85C0 TEST EAX,EAX
0040A924 . 75 21 JNZ SHORT dofoil.0040A947
0040A926 . 6A 01 PUSH 1
0040A928 . BE 1B2D4100 MOV ESI,dofoil.00412D1B UNICODE "PortmonClass"
0040A92D . E8 A2FDFFFF CALL dofoil.0040A6D4
0040A932 . 59 POP ECX
0040A933 . 85C0 TEST EAX,EAX
0040A935 . 75 10 JNZ SHORT dofoil.0040A947
0040A937 . 89 352D4100 MOV ECX,dofoil.00412D35 UNICODE "cain.exe"
0040A93C . E8 87FCFFFF CALL dofoil.0040A5C8
0040A941 . 85C0 TEST EAX,EAX
0040A943 . 75 02 JNZ SHORT dofoil.0040A947
0040A945 . 5E POP ESI
0040A946 . C3 RETN

```

3. There are lots of strings dumped in the memory which can tell what the malware does.

a. Modifies the registry key

“HKLM\Software\Microsoft\Windows\CurrentVersion\Run” to add its existence during system startup. This key maintains configurations for all the users of the system. So if one user of the system is infected, others get infected too.

```
004003FC . 50          PUSH EAX
004003FD . 68 6B2E4100 PUSH dofoil.00412E6B      UNICODE "Software\Microsoft\Windows\CurrentVersion\Run"
00400402 . 68 02000080 PUSH 80000002
00400407 . EB 15          JMP SHORT dofoil.0040D41E
```

b. Modifies the userinit.exe file to run the malware process when the user logs in.

Userinit.exe runs various logon scripts, re-establishes network connections and also starts explorer.exe.

```
0040033E . E8 8D60FFFF CALL dofoil.00404000
00400343 . 83C4 10      ADD ESP,10
00400346 . 68 332F4100 PUSH dofoil.00412F33      UNICODE "C:\Windows\system32\userinit.exe,"
0040034B . 56          PUSH ESI
0040034C . FF4F 30044080 CALL dofoil.00408080
```

Further it also checks the registry to find whether Winlogon runs userinit.exe after the user logs in.

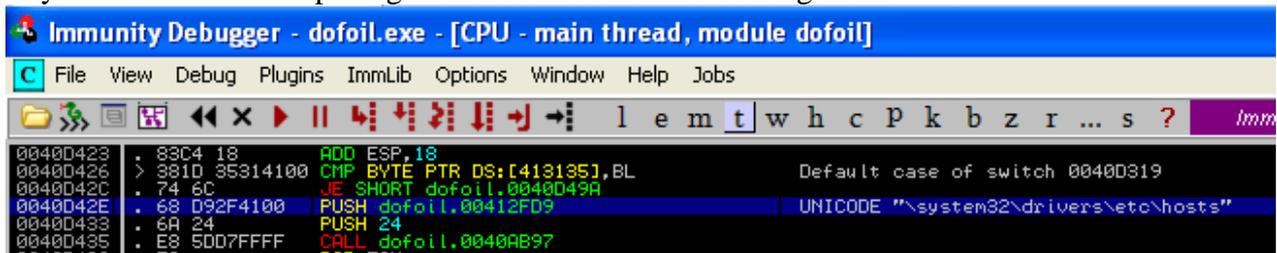
```
00400368 . 56          PUSH ESI
00400367 . 68 772F4100 PUSH dofoil.00412F77      UNICODE "Userinit"
0040036C . 68 C72E4100 PUSH dofoil.00412EC7      UNICODE "Software\Microsoft\Windows NT\CurrentVersion\Winlogon"
00400371 . 68 02000080 PUSH 80000002
00400376 . E8 02FFFFFF CALL dofoil.00408080
```

c. It downloads other malicious files from the internet by using

“Wininet.InternetReadFile” API.

```
0040C050 . 8D45 9C      LEA EAX,DWORD PTR SS:[EBP-64]
0040C053 . 50          PUSH EAX
0040C054 . FF75 90      PUSH DWORD PTR SS:[EBP-70]
0040C057 . 6A 04      PUSH 4
0040C059 . 68 B52B4100 PUSH dofoil.00412BB5      ASCII "InternetReadFile"
0040C05E . 56          PUSH ESI
0040C05F . E8 2CE2FFFF CALL dofoil.0040A290
0040C064 . 83C4 3C      ADD ESP,3C
0040C067 . 395D 98      CMP DWORD PTR SS:[EBP-68],EBX
```

d. Possibly it also performs local DNS poisoning by modifying the “hosts” file. This may trick the user into opening malicious websites instead of genuine sites.



```
Immunity Debugger - dofoil.exe - [CPU - main thread, module dofoil]
File View Debug Plugins ImmLib Options Window Help Jobs
l e m t w h c P k b z r ... s ? Imm
0040D423 . 83C4 18      ADD ESP,18
0040D426 > 3B1D 35314100 CMP BYTE PTR DS:[413135],BL      Default case of switch 0040D319
0040D42C . 74 6C      JE SHORT dofoil.0040D49A
0040D42E . 68 D92F4100 PUSH dofoil.00412FD9      UNICODE "\\system32\drivers\etc\hosts"
0040D433 . 6A 24      PUSH 24
0040D435 . E8 50D7FFFF CALL dofoil.0040AB97
```