



S11L3

**TRACCIA:** FATE RIFERIMENTO AL MALWARE: MALWARE\_U3\_W3\_L3, PRESENTE ALL'INTERNO DELLA CARTELLA ESERCIZIO\_PRATICO\_U3\_W3\_L3 SUL DESKTOP DELLA MACCHINA VIRTUALE DEDICATA ALL'ANALISI DEI MALWARE. RISPONDETE AI SEGUENTI QUESITI UTILIZZANDO OLLYDBG. •

ALL'INDIRIZZO 0040106E IL MALWARE EFFETTUA UNA CHIAMATA DI FUNZIONE ALLA FUNZIONE «CREATEPROCESS». QUAL È IL VALORE DEL PARAMETRO «COMMANDLINE» CHE VIENE PASSATO SULLO STACK?

- INSERITE UN BREAKPOINT SOFTWARE ALL'INDIRIZZO 004015A3. QUAL È IL VALORE DEL REGISTRO EDX? (2) ESEGUITE A QUESTO PUNTO UNO «STEP-INTO». INDICATE QUAL È ORA IL VALORE DEL REGISTRO EDX (3) MOTIVANDO LA RISPOSTA (4). CHE ISTRUZIONE È STATA ESEGUITA? (5) •

- INSERITE UN SECONDO BREAKPOINT ALL'INDIRIZZO DI MEMORIA 004015AF. QUAL È IL VALORE DEL REGISTRO ECX? (6) ESEGUITE UN STEP-INTO. QUAL È ORA IL VALORE DI ECX? (7) SPIEGATE QUALE ISTRUZIONE È STATA ESEGUITA (8).

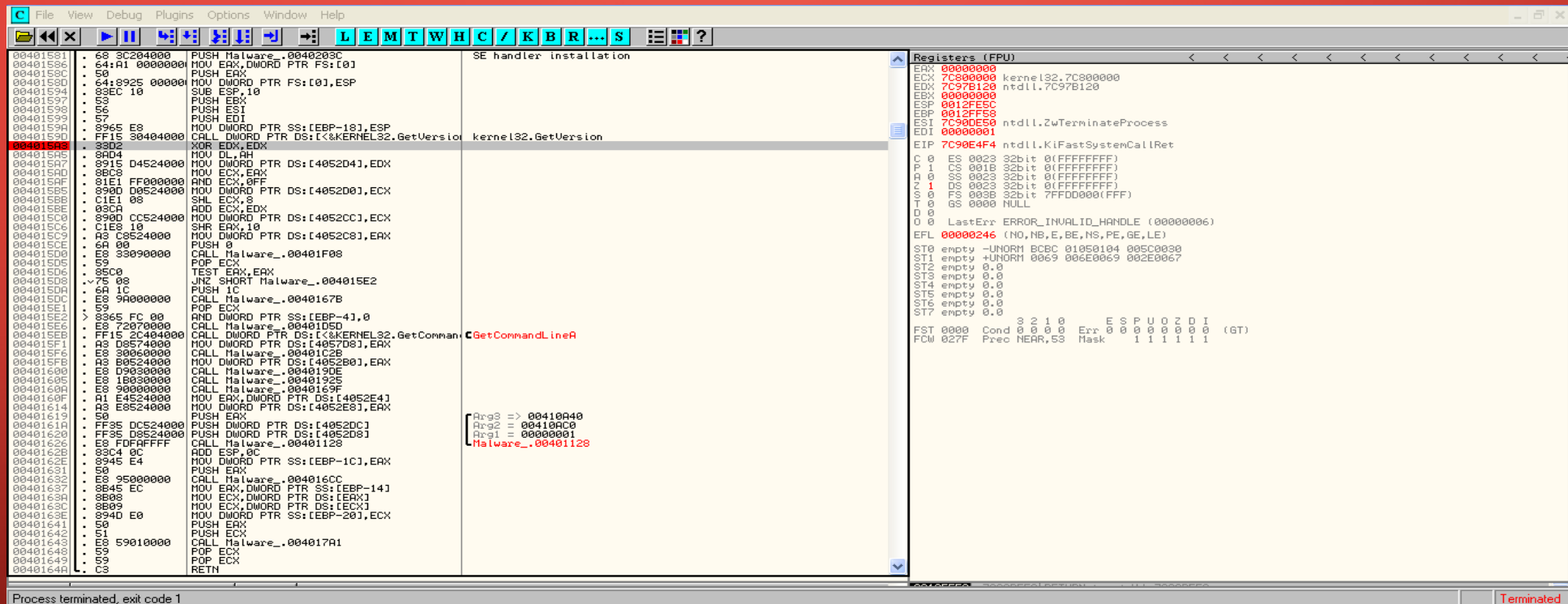
1) QUAL È IL VALORE DEL PARAMETRO «COMMANDLINE» CHE VIENE PASSATO SULLO STACK?  
IL VALORE DEL PARAMETRO È «CMD» OVVERO IL COMMAND PROMPT DI WINDOWS, COME SI NOTA  
NELLA FIGURA SOTTOSTANTE ALL'INDIRIZZO 00401067

```
00401047 . 8B45 E0 MOV EAX,DWORD PTR SS:[EBP-20]
0040104A . 8945 E8 MOV DWORD PTR SS:[EBP-18],EAX
0040104D . 8B4D E8 MOV ECK,DWORD PTR SS:[EBP-18]
00401050 . 894D E4 MOV DWORD PTR SS:[EBP-1C],ECK
00401053 . 8D55 F0 LEA EDX,DWORD PTR SS:[EBP-10]
00401056 . 52 PUSH EDX
00401057 . 8D45 A8 LEA EAX,DWORD PTR SS:[EBP-58]
0040105A . 50 PUSH EAX
0040105B . 6A 00 PUSH 0
0040105D . 6A 00 PUSH 0
0040105F . 6A 00 PUSH 0
00401061 . 6A 01 PUSH 1
00401063 . 6A 00 PUSH 0
00401065 . 6A 00 PUSH 0
00401067 . 68 30504000 PUSH Malware_.00405030
0040106C . 6A 00 PUSH 0
0040106E . FF15 04404000 CALL DWORD PTR DS:[<&KERNEL32.CreateProcessA>]
00401074 . 8945 EC MOV DWORD PTR SS:[EBP-14],EAX
00401077 . 6A FF PUSH -1
00401079 . 8B4D F0 MOV ECK,DWORD PTR SS:[EBP-10]
0040107C . 51 PUSH ECK
0040107D . FF15 00404000 CALL DWORD PTR DS:[<&KERNEL32.WaitForSingleObject>]
00401083 . 33C0 XOR EAX,EAX
00401085 . 8BE5 MOV ESP,EBP
00401087 . 5D POP EBP
00401088 . C3 RETN
00401089 . 55 PUSH EBP
0040108A . 8BEC MOV EBP,ESP
0040108C . 81EC 00010000 SUB ESP,100
00401092 . 57 PUSH EDI
00401093 . C785 F8FFFFFF MOV DWORD PTR SS:[EBP-100],0
0040109D . C685 00FFFFFF MOV BYTE PTR SS:[EBP-100],0
004010A4 . B9 3F000000 MOV ECK,3F
004010A9 . 33C0 XOR EAX,EAX
004010AB . 8DBD 01FFFFFF LEA EDI,DWORD PTR SS:[EBP-FF]
004010B1 . F3AB REP STOS DWORD PTR ES:[EDI]
004010B3 . 66AB STOS WORD PTR ES:[EDI]
004010B5 . AA STOS BYTE PTR ES:[EDI]
004010B6 . 8B45 08 MOV EAX,DWORD PTR SS:[EBP+8]
004010B9 . 50 PUSH EAX
004010BA . E8 81030000 CALL Malware_.00401440
004010BF . 83C4 04 ADD ESP,4
004010C2 . 8985 FCFFFFFF MOV DWORD PTR SS:[EBP-104],EAX
004010C8 . C785 F8FFFFFF MOV DWORD PTR SS:[EBP-100],0
004010D2 . 7EB 0F JMP SHORT Malware_.004010E3
004010D4 . 8B8D F8FFFFFF MOV ECK,DWORD PTR SS:[EBP-100]
004010D8 . 83C1 01 ADD ECK,1
004010DD . 898D F8FFFFFF MOV DWORD PTR SS:[EBP-100],ECK
004010E3 . 8B8D F8FFFFFF CMP DWORD PTR SS:[EBP-100],20
004010EA . 7D 31 JGE SHORT Malware_.0040111D
004010EC . 8B55 0C MOV EDX,DWORD PTR SS:[EBP+C]
004010EF . 0395 F8FFFFFF ADD EDX,DWORD PTR SS:[EBP-100]
004010F5 . 0FBEB0 MOVSX ECX,BYTE PTR DS:[EDI]
004010F8 . 8B85 F8FFFFFF MOV EAX,DWORD PTR SS:[EBP-100]
004010FE . 99 CDQ
004010FF . F7BD FCFFFFFF IDIV DWORD PTR SS:[EBP-104]
00401105 . 8B45 08 MOV EAX,DWORD PTR SS:[EBP+8]

pProcessInfo
pStartupInfo
CurrentDir = NULL
pEnvironment = NULL
CreationFlags = 0
InheritHandles = TRUE
pThreadSecurity = NULL
pProcessSecurity = NULL
CommandLine = "cmd"
ModuleFileName = NULL
CreateProcessA
Timeout = INFINITE
hObject
WaitForSingleObject
```

```
Registers (FPU)
EAX 00000000
ECX 0012FFB0
EDX 7C90E4F4 ntdll.KiFastSystemCallRet
EBX 7FFD8000
ESP 0012FFC4
EBP 0012FFF0
ESI FFFFFFFF
EDI 7C910208 ntdll.7C910208
EIP 00401577 Malware_.<ModuleEntryPoint>
C 0 ES 0023 32bit 0(FFFFFFFF)
P 1 CS 001B 32bit 0(FFFFFFFF)
A 0 SS 0023 32bit 0(FFFFFFFF)
Z 1 DS 0023 32bit 0(FFFFFFFF)
S 0 FS 003B 32bit 7FDF000(FFF)
T 0 GS 0000 NULL
D 0
O 0 LastErr ERROR_INVALID_HANDLE (00000006)
EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)
ST0 empty -UNORM BCBC 01050104 005C0030
ST1 empty +UNORM 0069 006E0069 002E0067
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0
FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 0 (GT)
FCW 027F Prec NEAR,53 Mask 1 1 1 1 1 1
```

INSERITE UN BREAKPOINT SOFTWARE ALL'INDIRIZZO 004015A3. QUAL È IL VALORE DEL REGISTRO EDX? ESEGUITE A QUESTO PUNTO UNO «STEP-INTO». INDICATE QUAL È ORA IL VALORE DEL REGISTRO EDX MOTIVANDO LA RISPOSTA. CHE ISTRUZIONE È STATA ESEGUITA? UNA VOLTA CONFIGURATO IL BREAKPOINT, CLICCHIAMO SU «PLAY»,



The screenshot shows a debugger window with the following components:

- Assembly View:** Displays assembly instructions with their addresses. The instruction at address 004015A3 is highlighted in red: `CALL DWORD PTR DS:[<&KERNEL32.GetVersion`. Other instructions include `PUSH Malware_.0040203C`, `MOV EAX, DWORD PTR FS:[0]`, `PUSH EAX`, `MOV DWORD PTR FS:[0], ESP`, `SUB ESP, 10`, `PUSH EAX`, `PUSH ESI`, `PUSH EDI`, `MOV DWORD PTR SS:[EBP-18], ESP`, `CALL DWORD PTR DS:[<&KERNEL32.GetVersion`, `XOR EDX, EDX`, `MOV DL, AH`, `MOV DWORD PTR DS:[4052D4], EDX`, `MOV ECX, EAX`, `AND ECX, 0FF`, `MOV DWORD PTR DS:[4052D0], ECX`, `SHL ECX, 8`, `ADD ECX, EDX`, `MOV DWORD PTR DS:[4052CC], ECX`, `SHR EAX, 10`, `MOV DWORD PTR DS:[4052C8], EAX`, `PUSH 0`, `CALL Malware_.00401F08`, `POP ECX`, `TEST EAX, EAX`, `JNZ SHORT Malware_.004015E2`, `PUSH 1C`, `CALL Malware_.0040167B`, `POP ECX`, `AND DWORD PTR SS:[EBP-4], 0`, `CALL Malware_.00401D5D`, `CALL DWORD PTR DS:[<&KERNEL32.GetComm`, `MOV DWORD PTR DS:[4057D8], EAX`, `CALL Malware_.00401C2B`, `MOV DWORD PTR DS:[4052B0], EAX`, `CALL Malware_.004019DE`, `CALL Malware_.00401925`, `CALL Malware_.0040169F`, `MOV EAX, DWORD PTR DS:[4052E4]`, `MOV DWORD PTR DS:[4052E8], EAX`, `PUSH EAX`, `PUSH DWORD PTR DS:[4052DC]`, `PUSH DWORD PTR DS:[4052D8]`, `CALL Malware_.00401128`, `ADD ESP, 0C`, `MOV DWORD PTR SS:[EBP-1C], EAX`, `PUSH EAX`, `CALL Malware_.004016CC`, `MOV EAX, DWORD PTR SS:[EBP-14]`, `MOV ECX, DWORD PTR DS:[EAX]`, `MOV ECX, DWORD PTR DS:[ECX]`, `MOV DWORD PTR SS:[EBP-20], ECX`, `PUSH EAX`, `PUSH ECX`, `CALL Malware_.004017A1`, `POP ECX`, `POP ECX`, `POP ECX`, `RETN`.
- Registers (FPU):** Shows the current values of the registers. The `EDX` register is highlighted in red and contains the value `00000000`. Other registers include `EAX` (00000000), `ECX` (7C900000), `EDX` (7C97B120), `EBX` (00000000), `ESP` (0012F5C0), `EBP` (0012F5C0), `ESI` (7C900E50), `EDI` (00000001), `EIP` (7C90E4F4), `C` (0), `P` (1), `A` (0), `Z` (1), `S` (0), `O` (0), `D` (0), `I` (0), `ST0` (empty), `ST1` (empty), `ST2` (empty), `ST3` (empty), `ST4` (empty), `ST5` (empty), `ST6` (empty), `ST7` (empty), `FST` (0000), `FCW` (027F).
- Process Information:** Shows the process name `kernel32.dll` and the `SE handler installation` status.
- Process Termination:** The status bar at the bottom indicates "Process terminated, exit code 1".



IL PROGRAMMA SI FERMA ALL'ISTRUZIONE XOR EDX,EDX. PRIMA CHE L'ISTRUZIONE VENGA ESEGUITA IL VALORE DEL REGISTRO È «00000A28». DOPO LO STEP-INTO, VIENE ESEGUITA L'ISTRUZIONE XOR EDX,EDX. QUINDI, DOPO LO STEP-INTO IL VALORE DI EDX SARÀ 0. P

The screenshot shows a debugger window with the following components:

- Assembly View:** Displays assembly instructions with their addresses and hex values. The instruction at address 00401593, `XOR EDX,EDX`, is highlighted in red. The instruction at address 004015A5, `MOV DL,AH`, is also highlighted. The instruction at address 00401590, `CALL DWORD PTR DS:[<&KERNEL32.GetVersion`, is labeled as `kernel32.GetVersion`.
- Registers (FPU):** Shows the current values of the registers. The `EDX` register is highlighted and shows the value `00000000`. Other registers like `EAX`, `ECX`, `EBX`, `ESP`, `EBP`, `ESI`, `EDI`, `EIP`, `C`, `P`, `A`, `Z`, `S`, `T`, `D`, `O`, `EFL`, `ST0`, `ST1`, `ST2`, `ST3`, `ST4`, `ST5`, `ST6`, and `ST7` are also visible.
- Memory Dump:** Shows a hex dump of memory starting at address 00401500. The dump includes hex values and their corresponding ASCII representations.

ESEGUENDO L'AND LOGICO TRA I BIT UNO AD UNO 0000 0000 0000 0000 0000 0000 0000 0101 CHE IN ESADECIMALE È 00000005 ECCO SPIEGATO IL VALORE DI ECX DOPO L'ISTRUZIONE AND ECX, 0FF NELLA RIQUADRO IN ALTO A DESTRA

The screenshot displays a debugger interface with two main panels. The left panel shows the assembly code, and the right panel shows the registers.

**Assembly Code:**

| Address  | Disassembly     | Comment                                    |
|----------|-----------------|--------------------------------------------|
| 00401577 | 55              | PUSH EBP                                   |
| 00401578 | 8BEC            | MOV EBP, ESP                               |
| 0040157A | 6A FF           | PUSH -1                                    |
| 0040157C | 68 C0404000     | PUSH Malware_.004040C0                     |
| 00401581 | 68 3C204000     | PUSH Malware_.0040203C                     |
| 00401586 | 64: A1 00000000 | MOV EAX, DWORD PTR FS:[0]                  |
| 0040158C | 50              | PUSH EAX                                   |
| 0040158D | 64: 8925 000000 | MOV DWORD PTR FS:[0], ESP                  |
| 00401594 | 83EC 10         | SUB ESP, 10                                |
| 00401597 | 53              | PUSH EBX                                   |
| 00401598 | 56              | PUSH ESI                                   |
| 00401599 | 57              | PUSH EDI                                   |
| 0040159A | 8965 E8         | MOV DWORD PTR SS:[EBP-18], ESP             |
| 0040159D | FF15 30404000   | CALL DWORD PTR DS:[C:\&KERNEL32.GetVersion |
| 004015A3 | 33D2            | XOR EDX, EDX                               |
| 004015A5 | 8AD4            | MOV DL, AH                                 |
| 004015A7 | 8915 D4524000   | MOV DWORD PTR DS:[4052D4], EDX             |
| 004015AD | 8BC8            | MOV ECX, EAX                               |
| 004015B5 | 81E1 FF000000   | AND ECX, 0FF                               |
| 004015B8 | 8900 D0524000   | MOV DWORD PTR DS:[4052D0], ECX             |
| 004015BB | C1E1 08         | SHL ECX, 8                                 |
| 004015BE | 03CA            | ADD ECX, EDX                               |
| 004015C0 | 8900 CC524000   | MOV DWORD PTR DS:[4052CC], ECX             |
| 004015C6 | C1E8 10         | SHR EAX, 10                                |
| 004015C9 | A3 C8524000     | MOV DWORD PTR DS:[4052C8], EAX             |
| 004015CE | 6A 00           | PUSH 0                                     |
| 004015D0 | E8 33090000     | CALL Malware_.00401F08                     |
| 004015D5 | 59              | POP ECX                                    |
| 004015D6 | 85C0            | TEST EAX, EAX                              |
| 004015D8 | 75 08           | JNZ SHORT Malware_.004015E2                |

**Registers (FPU):**

| Register | Value                                   |
|----------|-----------------------------------------|
| EAX      | 0A280105                                |
| ECX      | 00000005                                |
| EDX      | 00000001                                |
| EBX      | 7FFDF000                                |
| ESP      | 0012FF94                                |
| EBP      | 0012FFC0                                |
| ESI      | FFFFFFFF                                |
| EDI      | 7C910208 ntdll.7C910208                 |
| EIP      | 004015B5 Malware_.004015B5              |
| C 0      | ES 0023 32bit 0(FFFFFFFF)               |
| P 1      | CS 001B 32bit 0(FFFFFFFF)               |
| A 0      | SS 0023 32bit 0(FFFFFFFF)               |
| Z 0      | DS 0023 32bit 0(FFFFFFFF)               |
| S 0      | FS 003B 32bit 7FFDE000(FFF)             |
| T 0      | GS 0000 NULL                            |
| D 0      |                                         |
| O 0      | LastErr ERROR_INVALID_HANDLE (00000006) |
| EFL      | 00000206 (NO, NB, NE, A, NS, PE, GE, G) |
| ST0      | empty -UNORM BCBC 01050104 005C0030     |
| ST1      | empty +UNORM 0069 006E0069 002E0067     |
| ST2      | empty 0.0                               |
| ST3      | empty 0.0                               |
| ST4      | empty 0.0                               |
| ST5      | empty 0.0                               |
| ST6      | empty 0.0                               |
| ST7      | empty 0.0                               |

**Hex dump:**

| Address  | Hex                     | dump | ASCII    |
|----------|-------------------------|------|----------|
| 00405000 | 00 00 00 00 00 00 00 00 |      | .....    |
| 00405008 | 00 00 00 00 F8 27 40 00 |      | .....@.  |
| 00405010 | 00 00 00 00 00 00 00 00 |      | .....    |
| 00405018 | 00 00 00 00 00 00 00 00 |      | .....    |
| 00405020 | 00 00 00 00 00 00 00 00 |      | .....    |
| 00405028 | 00 00 00 00 00 00 00 00 |      | .....    |
| 00405030 | 63 6D 64 00 46 06 16 54 |      | cmd.F.T  |
| 00405038 | 42 05 12 1B 47 0C 07 02 |      | B*+G..0  |
| 00405040 | 5D 1C 00 15 45 16 01 1D |      | lL..E.0# |
| 00405048 | 52 0E 05 0F 48 02 03 09 |      | R0*H00.  |
| 00405050 | 1C 14 1C 15 00 00 00 00 |      | LtLS.... |
| 00405058 | 00 00 00 00 00 00 00 00 |      | .....    |
| 00405060 | 0D 16 40 00 01 00 00 00 |      | ...0.0.. |
| 00405068 | 05 00 00 C0 0B 00 00 00 |      | ...0.0.. |
| 00405070 | 00 00 00 00 1D 00 00 C0 |      | ...0.0.. |
| 00405078 | 04 00 00 00 00 00 00 00 |      | ...0.0.. |
| 00405080 | 36 00 00 C0 04 00 00 00 |      | ...0.0.. |
| 00405088 | 00 00 00 00 8D 00 00 C0 |      | ...0.0.. |
| 00405090 | 00 00 00 00 00 00 00 00 |      | .....    |
| 00405098 | 3E 00 00 C0 08 00 00 00 |      | ...0.0.. |
| 004050A0 | 00 00 00 00 8F 00 00 C0 |      | ...0.0.. |
| 004050A8 | 00 00 00 00 00 00 00 00 |      | .....    |
| 004050B0 | 30 00 00 C0 05 00 00 00 |      | ...0.0.. |
| 004050B8 | 00 00 00 00 31 00 00 C0 |      | ...0.0.. |
| 004050C0 | 00 00 00 00 00 00 00 00 |      | .....    |
| 004050C8 | 32 00 00 C0 08 00 00 00 |      | ...0.0.. |
| 004050D0 | 00 00 00 00 23 00 00 C0 |      | ...0.0.. |