

Emanuele Di Stefano S11-L3

OllyDBG

Esercizio S11 L3 – Malware Analysis

Traccia:

Fate riferimento al malware Malware_U3_W3_L3, presente nella cartella Esercizio_Pratico_U3_W3_L3 sul desktop della macchina virtuale dedicata all'analisi dei malware. Rispondete ai seguenti quesiti utilizzando OllyDBG.

1. 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?
2. Inserite un breakpoint software all'indirizzo 004015A3. Qual è il valore del registro EDX? Eseguite uno step-into. Indicate qual è ora il valore del registro EDX, motivando la risposta. Che istruzione è stata eseguita?
3. Inserite un secondo breakpoint all'indirizzo di memoria 004015AF. Qual è il valore del registro ECX? Eseguite uno step-into. Qual è ora il valore di ECX? Spiegate quale istruzione è stata eseguita.
4. Bonus: Spiegare a grandi linee il funzionamento del malware.

Esercizio OllyDBG - Analisi del malware "Malware_U3_W3_L3"

1. Analisi del valore del parametro CommandLine:

- Indirizzo di riferimento: 0040106E
- Ho navigato all'indirizzo 0040106E nel codice, dove viene effettuata una chiamata alla funzione CreateProcessA. (Vedi Screenshot 1)
- Analizzando lo stack, ho identificato che il valore del parametro CommandLine, che viene passato alla funzione CreateProcessA, è "cmd". Questo è chiaramente visibile nella sezione dello stack dove i valori esadecimali corrispondenti alla stringa "cmd" (63 6D 64) sono mostrati. (Vedi Screenshot 2 e Screenshot 3)

1

```
Windows 7 - malware analysis [In esecuzione] - Oracle VM VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto
* OllyDbg - Malware_U3_W3_L3.exe - [CPU - main thread, module ntdll]
File View Debug Options Window Help
[<] [>] [<<] [>>] [X] [?] [L] [M] [T] [W] [H] [C] / [K] [B] [R] ... S [?]
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 30 04 00 PUSH Malware_.00404000
0040106E FF15 00404000 CALL DWORD PTR DS:[&KERNEL32.CreateProcessA]
00401074 89 55 EC MOU EDI,DWORD PTR SS:[EBP-14],ERX
00401077 5A FF PUSH -1
00401079 8B 4D F0 MOU ECX,DWORD PTR SS:[EBP-10]
0040107C 51 PUSH ECX
0040107D FF15 00404000 CALL DWORD PTR DS:[&KERNEL32.WaitForSingleObject]
00401083 33 C0 XOR EAX,EAX
00401085 8BE5 MOU ESP,ERX
00401087 5D POP EBX
00401088 C3 RETN
00401089 55 PUSH EBX
0040108A 8B EC MOU EBP,ESP
0040108C 81 EC 00 01 0000 SUB ESP,100
00401092 57 PUSH EDI
00401093 C785 F8 FE FFFF 00000000 MOU DWORD PTR SS:[EBP-108],0
00401094 F0 00 FF FF 00000000 MOU BYTE PTR SS:[EBP-100],0
```

2

3

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..@
00405040	50 1C 00 16 45 16 01 10	JL..E@#
00405048	52 AB A5 AF 48 02 08 09	Ret@H@.

```
DS:[00404004]=76B41072 (kernel32.CreateProcessA)
```

2

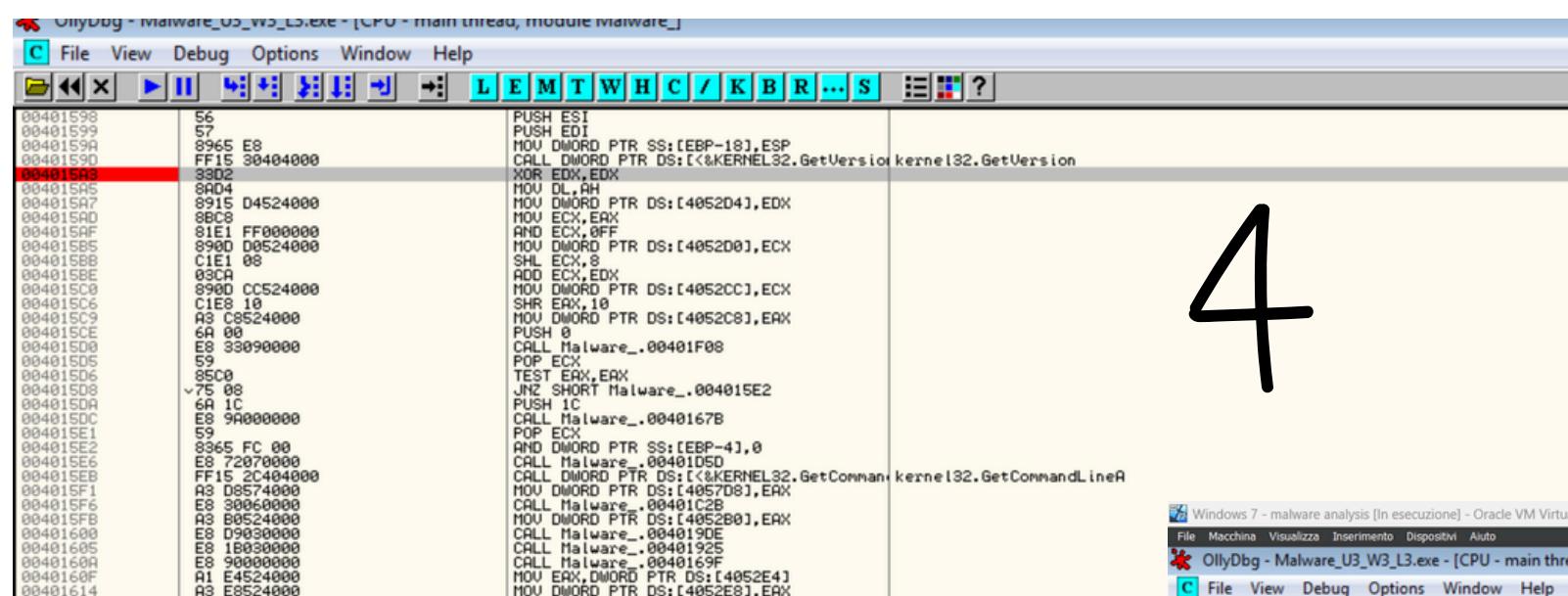
3

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..@
00405040	50 1C 00 16 45 16 01 10	JL..E@#
00405048	52 AB A5 AF 48 02 08 09	Ret@H@.

```
ASCII "cmd"
```

2. Impostazione del breakpoint all'indirizzo 004015A3:

- Indirizzo di riferimento: 004015A3
- Ho impostato un breakpoint software all'indirizzo 004015A3. (Vedi Screenshot 4)
- Quando l'esecuzione del programma è stata fermata al breakpoint, ho verificato che il valore del registro EDX fosse 00000001. Questo valore indica che EDX è stato azzerato in un'operazione di XOR con sé stesso (EDX = EDX XOR EDX). (Vedi Screenshot 5)
- Dopo aver eseguito uno "step-into" premendo F7, il valore del registro EDX è rimasto invariato. Questo perché l'istruzione che è stata eseguita (XOR EDX,EDX) non ha alterato il registro EDX in un modo significativo dato che l'operazione XOR tra EDX e sé stesso restituisce sempre 0. (Vedi Screenshot 6)

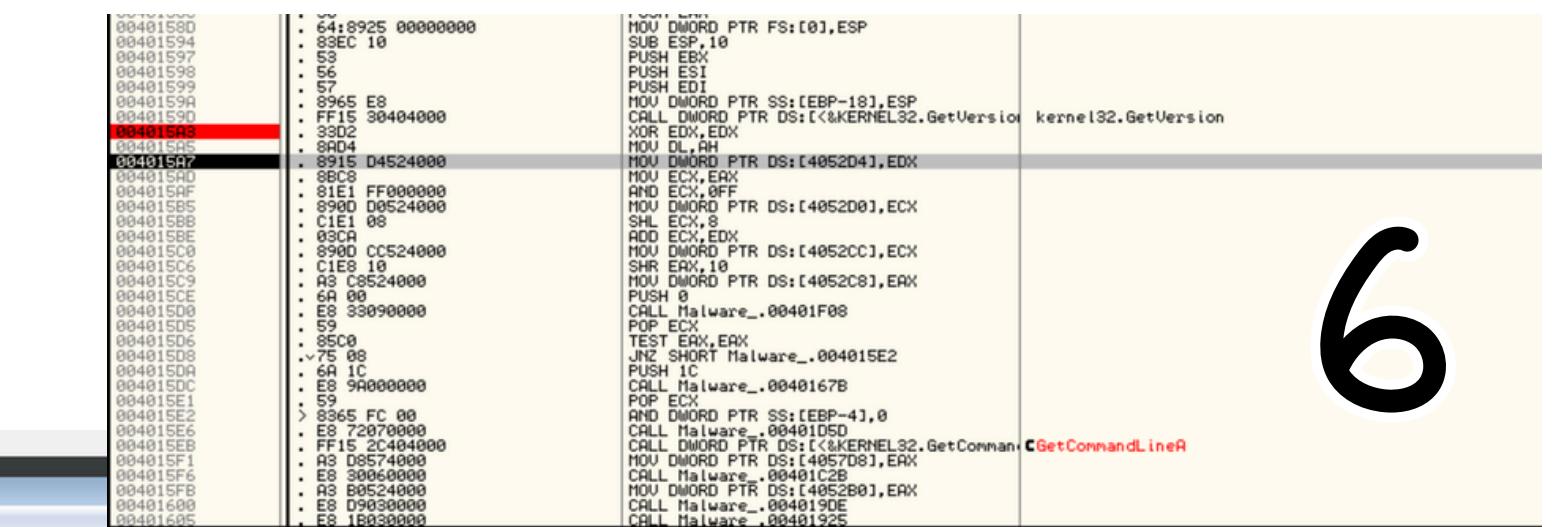


```

00401598 56 PUSH ESI
00401599 57 PUSH EDI
0040159A 9965 E8 MOV DWORD PTR SS:[EBP-18],ESP
0040159B FF15 90404000 CALL DWORD PTR DS:[<&KERNEL32.GetVersion kernel32.GetVersion
0040159C XOR EDX,EDX
0040159D 8904 MOU DL,AH
0040159E 8915 D4524000 MOU EDX,EDX
0040159F 8BC8 MOU ECX,EAX
004015A0 81E1 FF000000 AND ECX,OFF
004015A1 8900 D0524000 MOU DWORD PTR DS:[4052D0],ECX
004015A2 C1E1 08 SHL ECX,8
004015A3 03CA ADD ECX,EDX
004015A4 8900 CC524000 MOU DWORD PTR DS:[4052CC],ECX
004015A5 C1E8 10 SHR ECX,10
004015A6 A3 C9524000 MOU DWORD PTR DS:[4052C8],EAX
004015A7 8A 08 PUSH 0
004015A8 8900900000 CALL Malware_.00401F08
004015A9 59 POP ECX
004015A9 FF15 2C494000 CALL DWORD PTR DS:[<&KERNEL32.GetCommandLineA kernel32.GetCommandLineA
004015AB A3 D574000 MOU DWORD PTR DS:[4057D8],EAX
004015AC 88 30060000 CALL Malware_.00401C2B
004015AD 88 30060000 CALL Malware_.00401925
004015AE 88 1B030000 CALL Malware_.00401925
004015AF 88 90000000 CALL Malware_.0040169F
004015B0 A1 E4524000 MOU EDX,DWORD PTR DS:[4052E4]
004015B1 A3 E0524000 MOU DWORD PTR DS:[4052E8],ERX

```

4

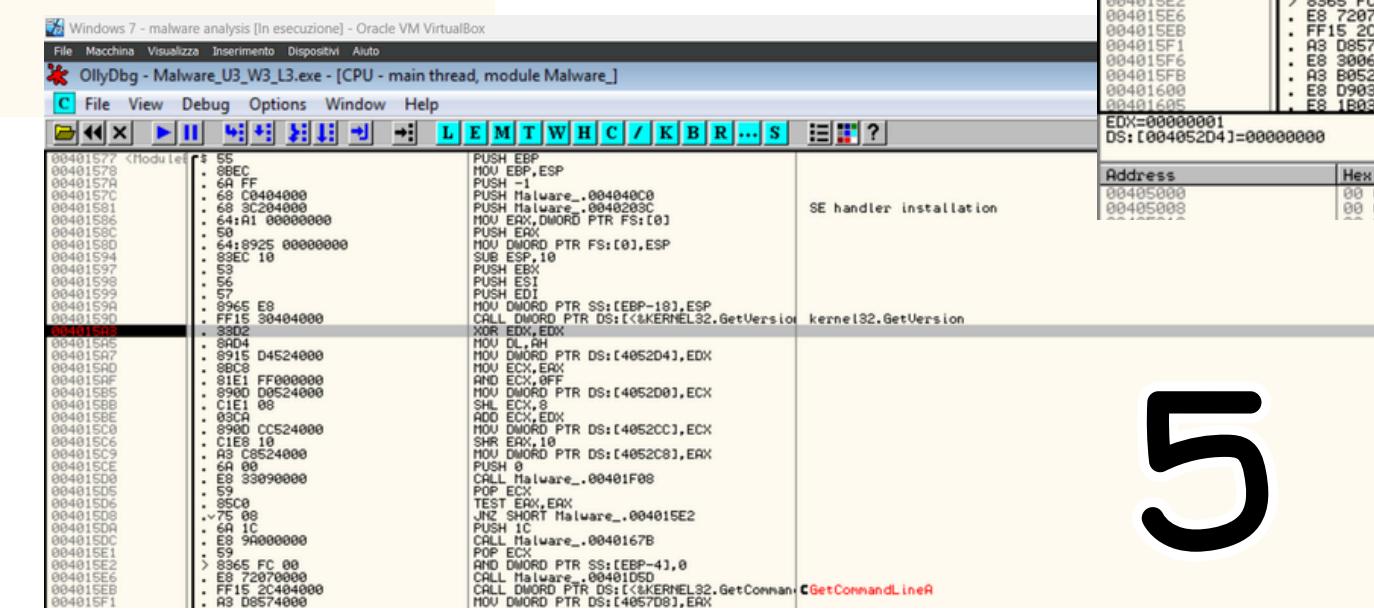


```

004015B2 8904 MOU DL,AH
004015B3 8915 D4524000 MOU EDX,EDX
004015B4 8BC8 MOU ECX,EAX
004015B5 81E1 FF000000 AND ECX,OFF
004015B6 8900 D0524000 MOU DWORD PTR DS:[4052D0],ECX
004015B7 C1E1 08 SHL ECX,8
004015B8 03CA ADD ECX,EDX
004015B9 8900 CC524000 MOU DWORD PTR DS:[4052CC],ECX
004015BA C1E8 10 SHR ECX,10
004015BB A3 C9524000 MOU DWORD PTR DS:[4052C8],EAX
004015BC 8A 08 PUSH 0
004015BD 8900900000 CALL Malware_.00401F08
004015BE 59 POP ECX
004015BF 8900900000 TEST ECX,ECX
004015C0 75 08 JNZ SHORT Malware_.004015E2
004015C1 6A 1C PUSH 1C
004015C2 E8 94000000 CALL Malware_.0040167B
004015C3 59 POP ECX
004015C4 8904 FC 00 AND DWORD PTR SS:[EBP-41,0
004015C5 8900900000 CALL Malware_.00401F08
004015C6 FF15 2C494000 CALL DWORD PTR DS:[<&KERNEL32.GetCommandLineA kernel32.GetCommandLineA
004015C7 A3 D574000 MOU DWORD PTR DS:[4057D8],EAX
004015C8 88 30060000 CALL Malware_.00401C2B
004015C9 88 30060000 CALL Malware_.00401925
004015CA 88 1B030000 CALL Malware_.00401925
004015CB 88 90000000 CALL Malware_.0040169F
004015CC A1 E4524000 MOU EDX,DWORD PTR DS:[4052E4]
004015CD A3 E0524000 MOU DWORD PTR DS:[4052E8],ERX

```

5



```

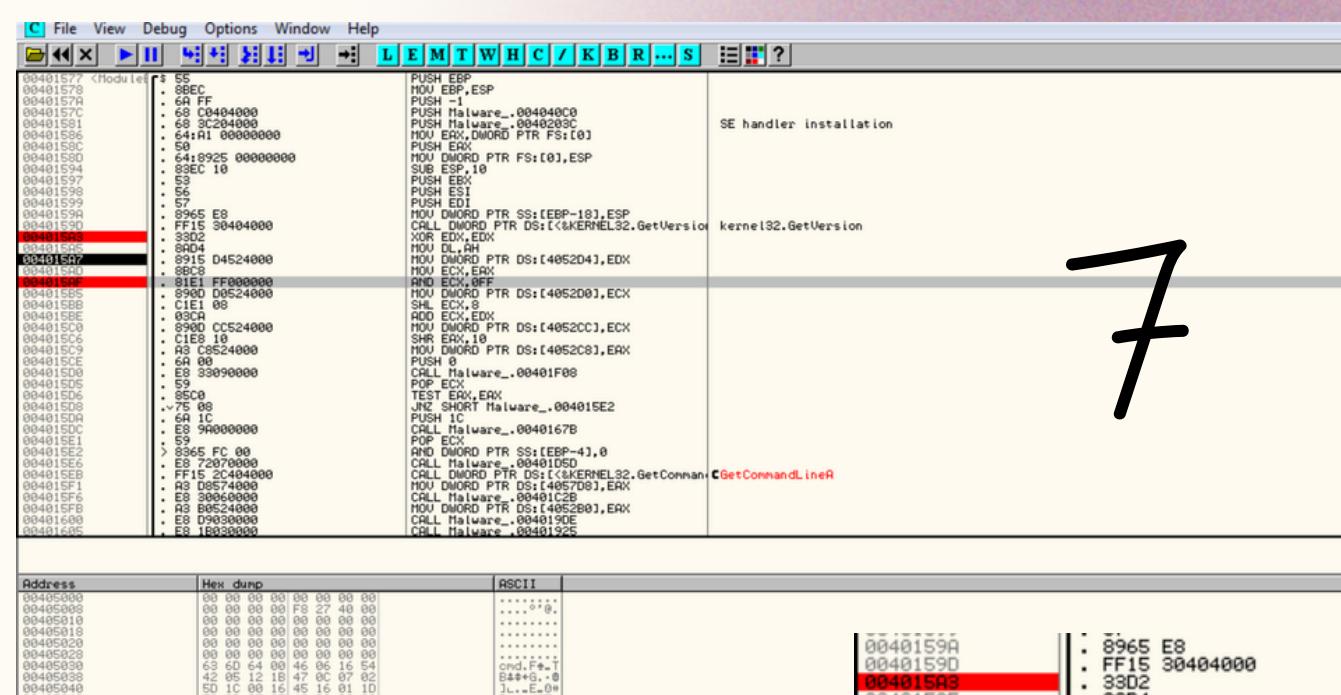
004015D2 8904 MOU DL,AH
004015D3 8915 D4524000 MOU EDX,EDX
004015D4 8BC8 MOU ECX,EAX
004015D5 81E1 FF000000 AND ECX,OFF
004015D6 8900 D0524000 MOU DWORD PTR DS:[4052D0],ECX
004015D7 C1E1 08 SHL ECX,8
004015D8 03CA ADD ECX,EDX
004015D9 8900 CC524000 MOU DWORD PTR DS:[4052CC],ECX
004015DA C1E8 10 SHR ECX,10
004015DB A3 C9524000 MOU DWORD PTR DS:[4052C8],EAX
004015DC 8A 08 PUSH 0
004015DD 8900900000 CALL Malware_.00401F08
004015DE 59 POP ECX
004015DF 8900900000 TEST ECX,ECX
004015E0 75 08 JNZ SHORT Malware_.004015E2
004015E1 6A 1C PUSH 1C
004015E2 E8 94000000 CALL Malware_.0040167B
004015E3 59 POP ECX
004015E4 8904 FC 00 AND DWORD PTR SS:[EBP-41,0
004015E5 8900900000 CALL Malware_.00401F08
004015E6 FF15 2C494000 CALL DWORD PTR DS:[<&KERNEL32.GetCommandLineA kernel32.GetCommandLineA
004015E7 A3 D574000 MOU DWORD PTR DS:[4057D8],EAX
004015E8 88 30060000 CALL Malware_.00401C2B
004015E9 88 30060000 CALL Malware_.00401925
004015EA 88 1B030000 CALL Malware_.00401925
004015EB 88 90000000 CALL Malware_.0040169F
004015EC A1 E4524000 MOU EDX,DWORD PTR DS:[4052E4]
004015ED A3 E0524000 MOU DWORD PTR DS:[4052E8],ERX

```

6

3. Impostazione del secondo breakpoint all'indirizzo 004015AF:

- Indirizzo di riferimento: 004015AF
 - Ho impostato un secondo breakpoint all'indirizzo 004015AF. (Vedi Screenshot 7)
 - Quando il programma è stato eseguito fino a questo breakpoint, ho osservato che il valore del registro ECX era 00000006. (Vedi Screenshot 8)
 - Dopo aver eseguito uno "step-into", il valore del registro ECX è cambiato in 1D181006. (Vedi Screenshot 9)
L'istruzione che è stata eseguita ha caricato un nuovo valore in ECX, che può essere il risultato di un'operazione logica o aritmetica su ECX.



7



8

4. BONUS: Analisi del comportamento generale del malware:

Dall'analisi condotta, il malware sembra eseguire operazioni di verifica delle versioni del sistema operativo e manipolazioni di registri per gestire processi e probabilmente eseguire comandi specifici come cmd. La funzione CreateProcessA viene utilizzata per avviare nuovi processi, il che indica che il malware potrebbe tentare di eseguire un altro programma o script maligno tramite il comando cmd.