

# S11 L3

Davide Andreozzi

# 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**? **(1)**
- 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)**.
- BONUS: spiegare a grandi linee il funzionamento del malware

# 1) Primo punto

Andando all'indirizzo 0040106E c'è la chiamata alla funzione CreateProcessA che ha come valore del parametro CommandLine = «CMD», cioè il prompt di comando Windows.

00401061	. 6A 01	PUSH 1	InheritHandles = TRUE
00401063	. 6A 00	PUSH 0	pThreadSecurity = NULL
00401065	. 6A 00	PUSH 0	pProcessSecurity = NULL
00401067	. 68 30504000	PUSH Malware_.00405030	CommandLine = "cmd"
0040106C	. 6A 00	PUSH 0	ModuleFileName = NULL
0040106E	. FF15 04404000	CALL DWORD PTR DS:[&KERNEL32.CreateProcessA]	CreateProcessA

## 2) Secondo punto

Inserendo il break point all'indirizzo 004015A3 e andando ad eseguire un «play» possiamo notare che il valore EDX = 00001DB1.

00401565	. FD	STD	
00401566	. F2:AE	REPNE SCAS BYTE PTR ES:[EDI]	
00401568	. 47	INC EDI	
00401569	. 3B07	CMP BYTE PTR DS:[EDI],AL	
0040156B	. 74 04	JE SHORT Malware_.00401571	
0040156D	. 33C0	XOR EAX,EAX	
0040156F	. EB 02	JMP SHORT Malware_.00401573	
00401571	> 8BC7	MOV EAX,EDI	
00401573	> FC	CLO	
00401574	. 5F	POP EDI	
00401575	. C9	LEAVE	
00401576	. C3	RETN	
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]	SE handler installation
0040158C	. 50	PUSH EAX	
0040158D	. 64:8925 00000000	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-10],ESP	
0040159D	. FF15 30404000	CALL DWORD PTR DS:[<&KERNEL32.GetVersion>]	kernel32.GetVersion
004015A3	. 33D2	XOR EDX,EDX	
004015A5	. 8A04	MOV DL,AH	
004015A7	. 8915 D4524000	MOV DWORD PTR DS:[4052D4],EDX	
004015AD	. 8BC8	MOV ECX,EAX	

EAX	1DB10106
ECX	7EFDE000
EDX	00001DB1
EBX	7EFDE000
ESP	0018FF5C
EBP	0018FF88
ESI	00000000
EDI	00000000
EIP	004015A3 Malware_.004015A3
C 0	ES 002B 32bit 0(FFFFFFFF)
P 1	CS 0023 32bit 0(FFFFFFFF)
A 0	SS 002B 32bit 0(FFFFFFFF)
Z 0	DS 002B 32bit 0(FFFFFFFF)
S 0	FS 0053 32bit 7FDD000(FFF)
T 0	GS 002B 32bit 0(FFFFFFFF)
D 0	
O 0	LastErr ERROR_SUCCESS (00000000)
EFL	00000206 (NO,NB,NE,A,NS,PE,GE,0)
ST0	empty 0.0
ST1	empty 0.0
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

## 2) Secondo punto

Ora eseguiamo uno STEP-INTO dalla barra superiore del programma.

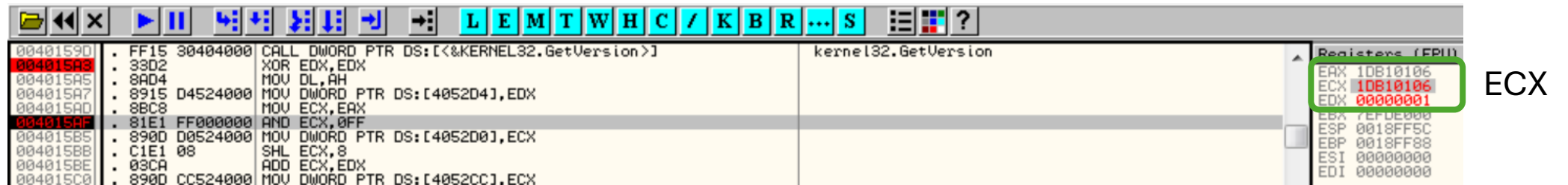
Dallo screen di seguito possiamo vedere che il valore di **EDX** è **0**, questo perché è stato eseguito il comando XOR EDX, EDX che essendo un'operazione tra due operandi identici serve ad inizializzare la variabile.

00401562	. 8A45 0C	MOV AL, BYTE PTR SS:[EBP+C]	
00401565	. FD	STD	
00401566	. F2:AE	REPNE SCAS BYTE PTR ES:[EDI]	
00401568	. 47	INC EDI	
00401569	. 3807	CMP BYTE PTR DS:[EDI], AL	
0040156B	~74 04	JE SHORT Malware_.00401571	
0040156D	. 33C0	XOR EAX, EAX	
0040156F	~EB 02	JMP SHORT Malware_.00401573	
00401571	> 8BC7	MOV EAX, EDI	
00401573	> FC	CLD	
00401574	. 5F	POP EDI	
00401575	. C9	LEAVE	
00401576	. C3	RETN	
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]	SE handler installation
0040158C	. 50	PUSH EAX	
0040158D	. 64:8925 00000000	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	. 9965 E8	MOV DWORD PTR SS:[EBP-10], ESP	
0040159D	. FF15 30404000	CALL DWORD PTR DS:[<&KERNEL32.GetVersion>]	kernel32.GetVersion
004015A5	. 33D2	XOR EDX, EDX	
004015A5	. 8AD4	MOV DL, AH	
004015A7	. 8915 D4524000	MOV DWORD PTR DS:[4052D4], EDX	
004015AD	. 8BC8	MOV ECX, EAX	

**Registers (FPU)**  
EAX 10B10106  
ECX 7EFD0000  
EDX 00000000  
EBX 7EFD0000  
ESP 0018FF5C  
EBP 0018FF88  
ESI 00000000  
EDI 00000000  
EIP 004015A5 Malware\_.004015A5  
C 0 ES 002B 32bit 0(FFFFFFFF)  
P 1 CS 0023 32bit 0(FFFFFFFF)  
A 0 SS 002B 32bit 0(FFFFFFFF)  
Z 1 DS 002B 32bit 0(FFFFFFFF)  
S 0 FS 0053 32bit 7EFD0000(FFF)  
T 0 GS 002B 32bit 0(FFFFFFFF)  
D 0  
O 0 LastErr ERROR\_SUCCESS (00000000)  
EFL 00010246 (NO, NB, E, BE, NS, PE, GE, LE)  
ST0 empty 0.0  
ST1 empty 0.0  
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

### 3) Terzo punto

PRIMA



The screenshot shows a debugger window with assembly code and a registers pane. The assembly code is as follows:

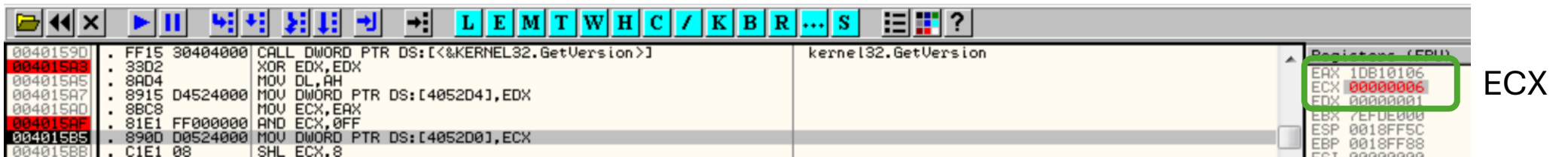
Address	Disassembly	Comment
0040159D	FF15 30404000	CALL DWORD PTR DS:[<&KERNEL32.GetVersion>]
004015A3	33D2	XOR EDX,EDX
004015A5	8AD4	MOV DL,AH
004015A7	8915 D4524000	MOV DWORD PTR DS:[4052D4],EDX
004015AD	8BC8	MOV ECX,EAX
004015AF	81E1 FF000000	AND ECX,0FF
004015B5	890D D0524000	MOV DWORD PTR DS:[4052D0],ECX
004015BB	C1E1 08	SHL ECX,8
004015BE	03CA	ADD ECX,EDX
004015C0	890D CC524000	MOV DWORD PTR DS:[4052CC],ECX

The registers pane on the right shows the following values:

Register	Value
EAX	1DB10106
ECX	1DB10106
EDX	00000001
EBX	7EFD0000
ESP	0018FF5C
EBP	0018FF88
ESI	00000000
EDI	00000000

The ECX register is highlighted with a green box and labeled "ECX".

DOPO



The screenshot shows the same debugger window after the function call. The assembly code is the same as before, but the registers pane shows updated values:

Register	Value
EAX	1DB10106
ECX	00000006
EDX	00000001
EBX	7EFD0000
ESP	0018FF5C
EBP	0018FF88
ESI	00000000
EDI	00000000

The ECX register is highlighted with a green box and labeled "ECX".

### 3) Terzo punto

In questo caso ECX = 1DB10106, eseguendo uno STEP-INTO il valore di ECX diventa 00000006.

L'istruzione va a eseguire un operatore logico AND.

L'operazione è la seguente AND ECX, 0FF

Andiamo a convertire in binario ECX e FF

(ECX) = 1DB10106	0001 1101 1011 0001 0000 0001 0000 0110
FF	0000 0000 0000 0000 0000 0000 1111 1111
<b>Risultato AND logico</b>	<b>0000 0000 0000 0000 0000 0000 0000 0110</b>

Il risultato dell'operazione convertito in esadecimale sarà **6**

## 4) Bonus

Il malware va a leggere e modificare i file di registro, ha accesso alla connessione internet quindi molto probabilmente potrebbe scaricare e uploadare dati su un server o installare una backdoor. Può aprire il prompt dei comandi di windows per controllare la macchina.