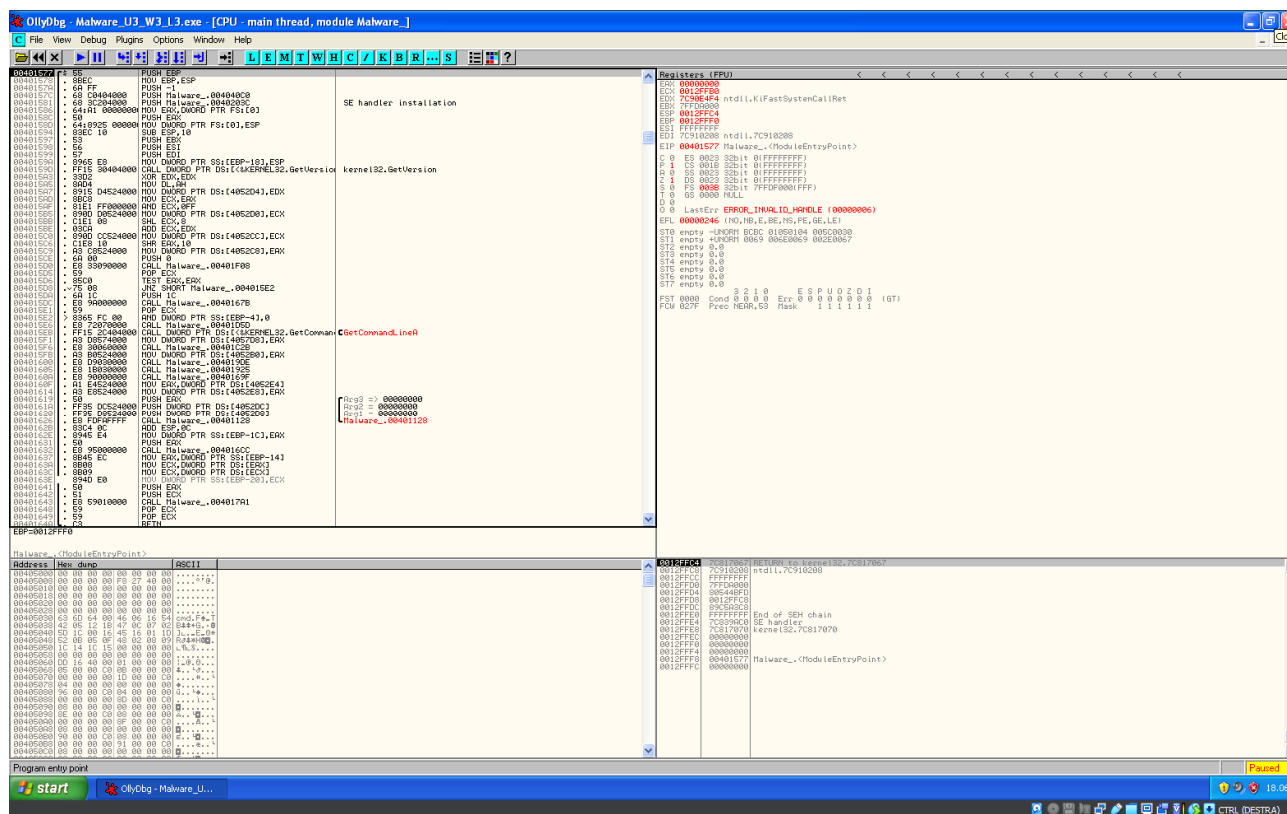


OllyDBG

- All'indirizzo 0040106E il Malware effettua una chiamata di funzione alla funzione "CreateProcess". Quel 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? Eseguite a questo punto uno "step-into". Indicate qual è ora il valore del registro EDX motivando la risposta. Che istruzione è stata eseguita?
- 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.

All'indirizzo 0040106E il Malware effettua una chiamata di funzione alla funzione "CreateProcess". Quel è il valore del parametro "CommandLine" che viene passato sullo stack ?

Una volta aperto il nostro malware ci ritroveremo con una schermata di questo tipo



Ci rechiamo all'indirizzo di memoria richiesto **004015AF** dove possiamo vedere il valore del parametro "CommandLine"

00401026	. 6A 00	PUSH 0	
00401028	. 8D4D F0	LEA ECX, DWORD PTR SS:[EBP-10]	
0040102B	. 51	PUSH ECX	
0040102C	. E8 AF030000	CALL Malware_.004013E0	
00401031	. 83C4 0C	ADD ESP, 0C	
00401034	. C745 D4 010101	MOV DWORD PTR SS:[EBP-2C], 01010101	
0040103B	. 66:C745 D8 0000	MOV WORD PTR SS:[EBP-28], 0	
00401041	. 8B55 18	MOV EDI, DWORD PTR SS:[EBP+18]	
00401044	. 8955 10	MOV EDI, DWORD PTR SS:[EBP-20], EDI	
00401047	. 8B45 E0	MOV EAX, DWORD PTR SS:[EBP-20]	
0040104A	. 8945 E8	MOV DWORD PTR SS:[EBP-18], EAX	
0040104D	. 8B4D E8	MOV ECX, DWORD PTR SS:[EBP-18]	
00401050	. 894D E4	MOV DWORD PTR SS:[EBP-1C], ECX	
00401053	. 8D55 F0	LEA EDI, DWORD PTR SS:[EBP-10]	
00401056	. 52	PUSH EDI	pProcessInfo
00401057	. 8D45 A8	LEA EAX, DWORD PTR SS:[EBP-58]	pStartupInfo
0040105A	. 50	PUSH EAX	CurrentDir = NULL
0040105B	. 6A 00	PUSH 0	pEnvironment = NULL
0040105D	. 6A 00	PUSH 0	CreationFlags = 0
0040105F	. 6A 00	PUSH 0	InheritHandles = TRUE
00401061	. 6A 01	PUSH 1	pThreadSecurity = NULL
00401063	. 6A 00	PUSH 0	pProcessSecurity = NULL
00401065	. 6A 00	PUSH 0	CommandLine = "cmd"
00401067	. 68 30504000	PUSH Malware_.00405030	hModuleFileName = NULL
00401068	. 6A 00	PUSH 0	CreateProcessA
0040106E	. FF15 04404000	CALL DWORD PTR DS:[<&KERNEL32.CreatePro	Timeout = INFINITE
00401074	. 8945 EC	MOV DWORD PTR SS:[EBP-14], EAX	hObject
00401077	. 6A FF	PUSH -1	WaitForSingleObject
00401079	. 8B4D F0	MOV ECX, DWORD PTR SS:[EBP-10]	
0040107C	. 51	PUSH ECX	
0040107D	. FF15 00404000	CALL DWORD PTR DS:[<&KERNEL32.WaitForSi	
00401083	. 33C0	XOR EAX, EAX	
00401085	. 8B55	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 F8FEFFFF	MOV DWORD PTR SS:[EBP-108], 0	
0040109D	. C685 00FEFFFF	MOV BYTE PTR SS:[EBP-108], 0	
004010A4	. B9 3F000000	MOV ECX, 3F	
004010A9	. 33C0	XOR EAX, EAX	
004010AB	. 33C0	XOR EDI, DWORD PTR SS:[EBP-FF]	
004010B1	. F3:AB	REP STOS DWORD PTR ES:[EDI]	
004010B3	. 66:AB	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 FCFEFFFF	MOV DWORD PTR SS:[EBP-104], EAX	
004010C8	. C785 F8FEFFFF	MOV DWORD PTR SS:[EBP-108], 0	
004010D2	. 7EB 0F	JMP SHORT Malware_.004010E3	
004010D4	. 8B45 F8FEFFFF	MOV ECX, DWORD PTR SS:[EBP-108]	
004010D9	. 83C1 01	ADD ECX, 1	
004010DD	. 898D F8FEFFFF	MOV DWORD PTR SS:[EBP-108], ECX	
004010E3	. 83BD F8FEFFFF	CMP DWORD PTR SS:[EBP-108], 20	
004010EA	. 7D 31	JGE SHORT Malware_.0040111D	
004010EC	. 8B55 0C	MOV EDI, DWORD PTR SS:[EBP+C]	
004010EF	. 0395 F8FEFFFF	ADD EDI, DWORD PTR SS:[EBP-108]	
004010F5	. 8F8F8F8F	MOVX ECX, BYTE PTR DS:[F8F8F8F8]	

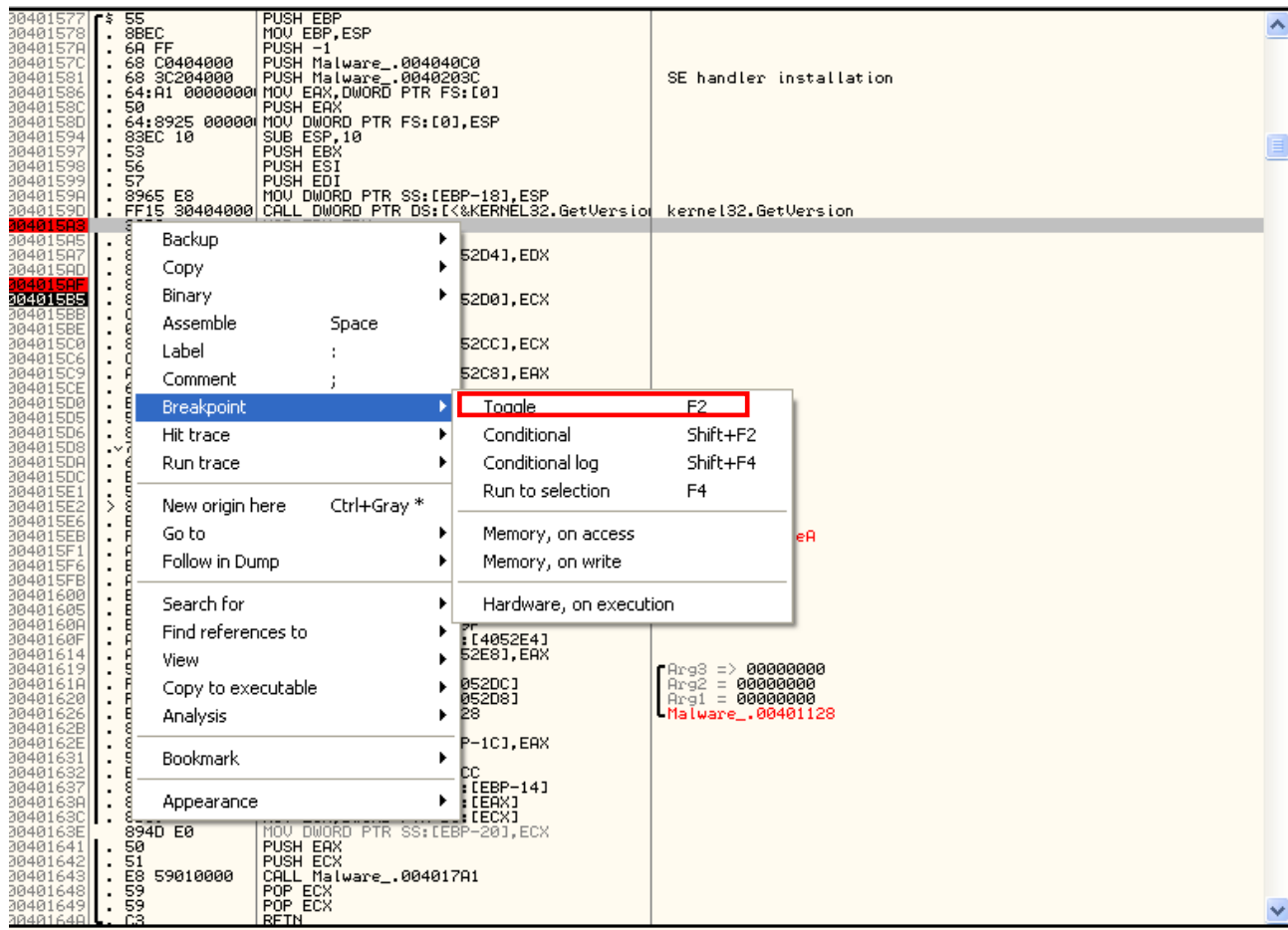
Inserite un breakpoint software all'indirizzo 004015A3. Qual è il valore del registro EDX? Eseguite a questo punto uno "step-into". Indicate qual è ora il valore del registro EDX motivando la risposta. Che istruzione è stata eseguita?

Ci spostiamo all'indirizzo **004015A3** e leggiamo sulla destra il valore del registro EDX

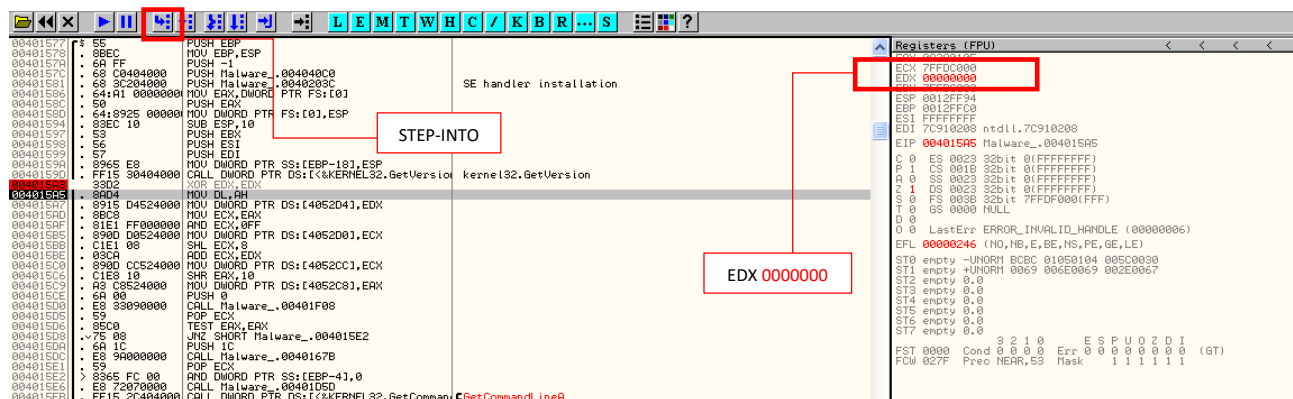
00401599	. 57	PUSH EDI	
0040159A	. 8965 E8	MOV DWORD PTR SS:[EBP-18], ESP	
0040159D	. FF15 00404000	CALL DWORD PTR DS:[<&KERNEL32.GetVersion	kernel32.GetVersion
004015A3	. 33D2	XOR EDI, EDI	
004015A5	. 8B4D	MOV DI, AH	
004015A7	. 8915 D4524000	MOV DWORD PTR DS:[4052D4], EDI	
004015AD	. 8BC8	MOV ECX, EAX	
004015AE	. 01E1 F8000000	AND ECX, 00000001	

Registers (FPU)	
EAX	0A280105
ECX	7EEDC000
EDX	00000A28
EBX	7FFDC000
ESP	0012FF94
EBP	0012FFC0
ESI	FFFFFFFF
EDI	7C910208 ntdll.7C910208
EIP	004015A3 Malware_.004015A3
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 7FFDF000(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
<div> <div>3 2 1 0</div> <div>E S P U O Z D I</div> </div> <div> <div>FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 0 (GT)</div> <div>FCW 027F Prec NEAR, 53 Mask 1 1 1 1 1 1</div> </div>	

Inseriamo un **breakpoint** all'indirizzo, cliccando con il tasto destro >> Breakpoint >> Toggle



Inserito il **breakpoint** (lo notiamo dal colore rosso) eseguiamo uno **step-into** e notiamo come il valore di EDX si sia azzerato. Il risultato ottenuto è dovuto al fatto che la locazione di memoria precedente contiene un'istruzione **XOR**. Operatore logico che mette a confronto destinatario e sorgente, e se i due valori sono identici l'istruzione darà come risultato **0**



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.

Muoviamoci all'indirizzo di memoria 004015AF e leggiamo il valore del registro ECX

Address	Disassembly	Comment
00401577	PUSH ESP	
00401578	MOV EBP, ESP	
00401579	PUSH -1	
0040157C	PUSH Malware_.004040C0	
00401581	PUSH Malware_.0040203C	
00401586	MOV EAX, DWORD PTR FS:[0]	SE handler installation
0040158C	PUSH EAX	
00401590	MOV DWORD PTR FS:[0], ESP	
00401594	SUB ESP, 10	
00401597	PUSH EAX	
00401598	PUSH ESI	
00401599	PUSH EDI	
0040159A	MOV DWORD PTR SS:[EBP-10], ESP	
00401599	CALL DWORD PTR DS:[&kernel32.GetVersion]	kernel32.GetVersion
0040159C	XOR EDI, EDI	
004015A7	MOV DWORD PTR DS:[4052D4], EDI	
004015B2	AND ECX, OFF	
004015B5	MOV DWORD PTR DS:[4052D0], ECX	
004015B8	SHL ECX, 8	
004015BC	ADD ECX, EDI	
004015C0	MOV DWORD PTR DS:[4052CC], ECX	
004015C6	SHR EAX, 10	
004015C9	MOV DWORD PTR DS:[4052C8], EAX	
004015CE	PUSH 0	
004015D0	CALL Malware_.00401F08	
004015D5	POP ECX	
004015D6	TEST EAX, EAX	
004015D8	JNZ SHORT Malware_.004015E2	
004015DA	PUSH 1C	
004015DD	CALL Malware_.0040167B	
004015E1	POP ECX	
004015E2	AND DWORD PTR SS:[EBP-4], 0	

Registers (FPU)

ECX	00280105
EDX	77F0C000
ESP	0012FF94
EBP	0012FFC0
EI1	FFFFFFFF
EDI	7C910208 ntdll.7C910208
EIP	004015AF Malware_.004015AF
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 7FFDF000(FFF)
T 0	GS 0000 NULL
D 0	
0 0	LastErr ERROR_INVALID_HANDLE (00000006)
EPL	00000246 (NO, NB, E, BE, HS, PE, GE, LE)
ST0	empty -UNORM ECBC 01050104 00C00030
ST1	empty +UNORM 0069 00E00069 00E00067
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 (GT)
FCW 027F	Prec NEAR, S3 Mask 1 1 1 1 1 1

Eseguiamo un altro step-into e notiamo il cambiamento del valore di ECX

Address	Disassembly	Comment
00401577	PUSH ESP	
00401578	MOV EBP, ESP	
00401579	PUSH -1	
0040157C	PUSH Malware_.004040C0	
00401581	PUSH Malware_.0040203C	
00401586	MOV EAX, DWORD PTR FS:[0]	SE handler installation
0040158C	PUSH EAX	
00401590	MOV DWORD PTR FS:[0], ESP	
00401594	SUB ESP, 10	
00401597	PUSH EAX	
00401598	PUSH ESI	
00401599	PUSH EDI	
0040159A	MOV DWORD PTR SS:[EBP-10], ESP	
00401599	CALL DWORD PTR DS:[&kernel32.GetVersion]	kernel32.GetVersion
0040159C	XOR EDI, EDI	
004015A7	MOV DWORD PTR DS:[4052D4], EDI	
004015B2	AND ECX, OFF	
004015B5	MOV DWORD PTR DS:[4052D0], ECX	
004015B8	SHL ECX, 8	
004015BC	ADD ECX, EDI	
004015C0	MOV DWORD PTR DS:[4052CC], ECX	
004015C6	SHR EAX, 10	
004015C9	MOV DWORD PTR DS:[4052C8], EAX	
004015CE	PUSH 0	
004015D0	CALL Malware_.00401F08	
004015D5	POP ECX	
004015D6	TEST EAX, EAX	
004015D8	JNZ SHORT Malware_.004015E2	
004015DA	PUSH 1C	
004015DD	CALL Malware_.0040167B	
004015E1	POP ECX	
004015E2	AND DWORD PTR SS:[EBP-4], 0	

Registers (FPU)

ECX	00000005
EDX	77F0C000
ESP	0012FF94
EBP	0012FFC0
EI1	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 7FFDF000(FFF)
T 0	GS 0000 NULL
D 0	
0 0	LastErr ERROR_INVALID_HANDLE (00000006)
EPL	00000206 (NO, NB, NE, A, NS, PE, GE, 0)
ST0	empty -UNORM ECBC 01050104 00C00030
ST1	empty +UNORM 0069 00E00069 00E00067
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 (GT)
FCW 027F	Prec NEAR, S3 Mask 1 1 1 1 1 1

Il cambio di valore è avvenuto per l'utilizzo dell'operatore logico AND nel punto precedente. Questa istruzione è un prodotto logico che esegue la and logica dei 2 operandi, in questo caso ECX, OFF