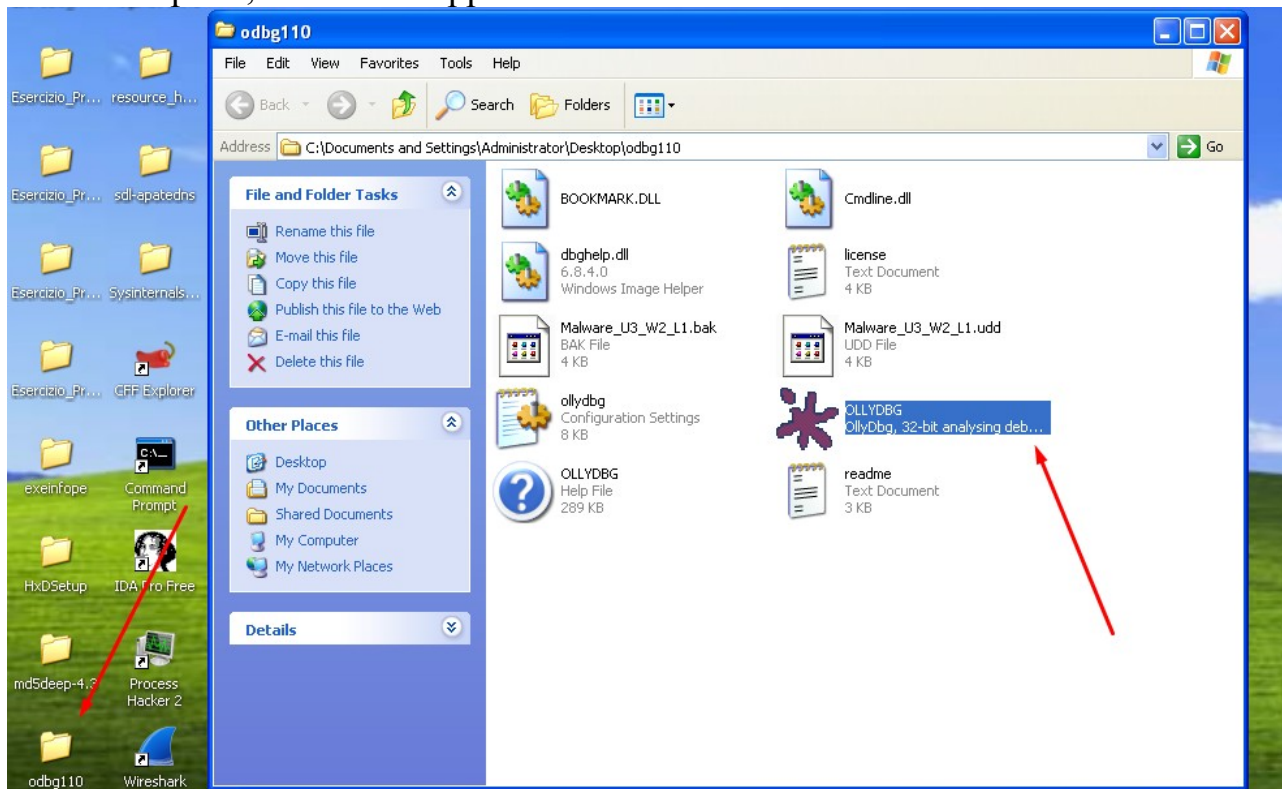
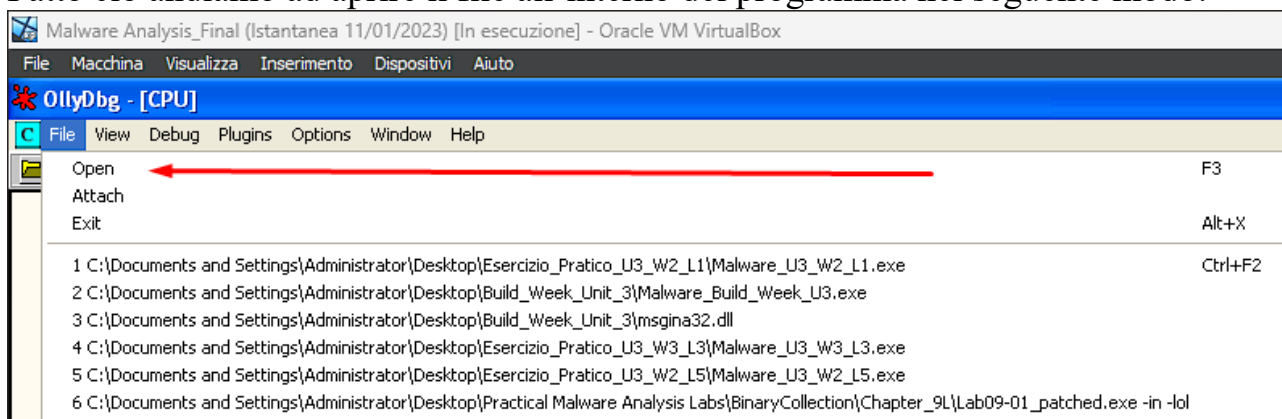


Report OllyDBG

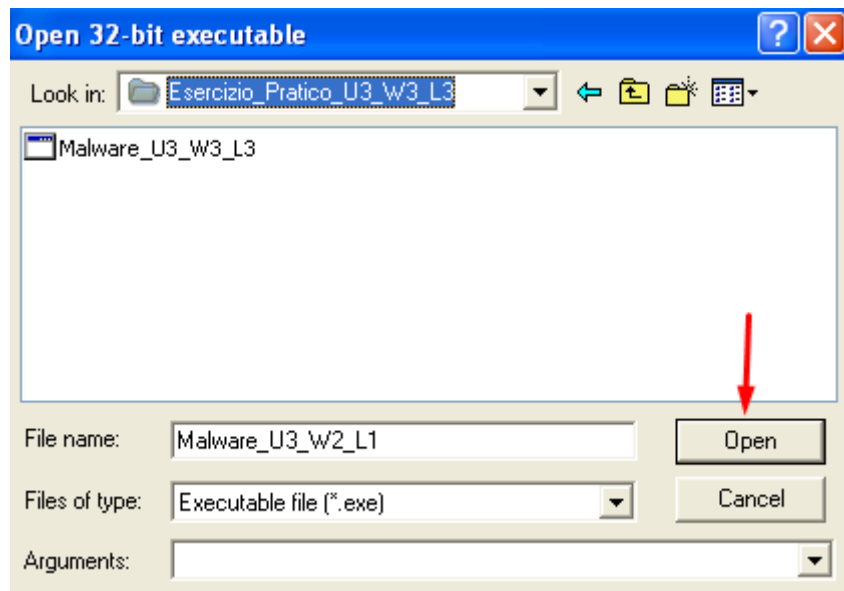
Come richiesto dall'esercizio odierno, andiamo ad aprire OllyDBG per fare debugging del file .exe richiesto. Lo ritroveremo della nostra cartella sul desktop ed una volta aperta, facciamo doppio click e diamo l'OK:



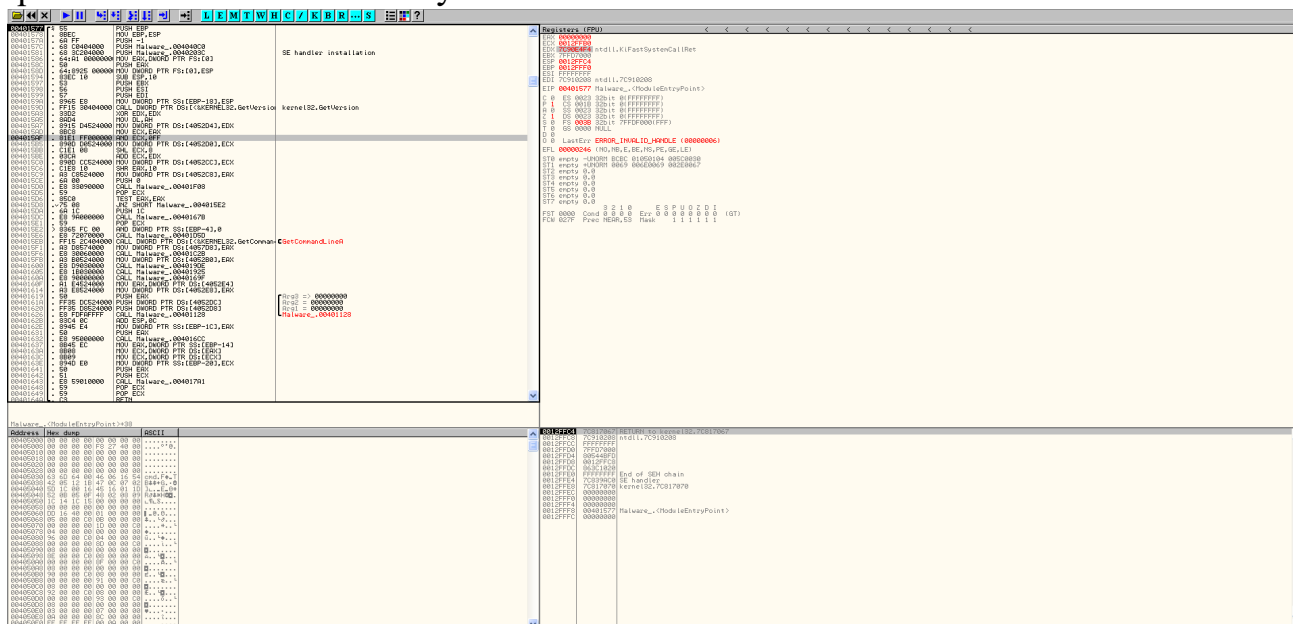
Fatto ciò andiamo ad aprire il file all'interno del programma nel seguente modo:



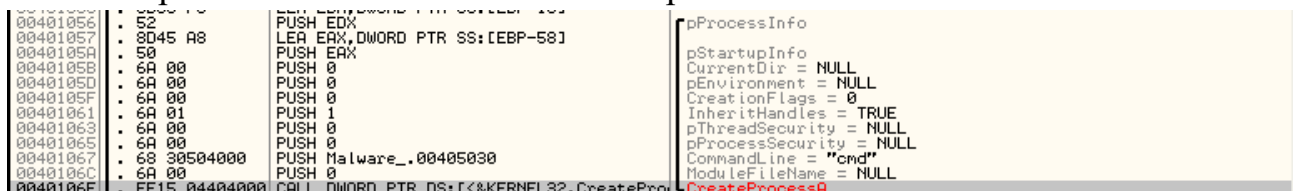
Dopodiché andiamo nella cartella del file in questione ed apriamolo:



Adesso ci ritroveremo davanti diverse finestre tra cui quella più importante ovvero quella del codice in Assembly:



Come richiesto, saliamo fino alla stringa con indirizzo 0040106E e possiamo notare che come parametro al **Comand Line** viene passato **cmd**:



Ora andiamo a creare un breakpoint all'indirizzo 004015A3 facendo doppio click dove indicato dalla freccia ed andiamo avanti con con la freccia blu **Run Program**:

00401577	55	PUSH EBP	
00401578	8BEC	MOV EBP,ESP	
00401579	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:[<&KERNEL32.GetVersion	kernel32.GetVersion
004015A6	33D2	XOR EDX,EDX	

Inizialmente il valore di EDX era 2600, ma dopo l'operatore logico XOR, il risultato verrà 0 poiché l'operatore finché avrà 2 valori uguali, darà come risultato 0. Infatti possiamo vedere un prima ed un dopo del valore cliccando il pulsante *step-in*:

00401577	55	PUSH EBP	
00401578	8BEC	MOV EBP,ESP	
00401579	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:[<&KERNEL32.GetVersion	kernel32.GetVersion
004015A6	33D2	XOR EDX,EDX	

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:[<&KERNEL32.GetVersion	kernel32.GetVersion
004015A6	33D2	XOR EDX,EDX	
004015A7	8AD4	MOV DL,AH	

Registers (FPU)	
EAX	0A280105
ECX	7FFD7000
EDX	00000000
EBX	7FFD7000
ESP	0012FF94
EBP	0012FFC0
ESI	FFFFFFFF
EDI	7C910208 ntdll.7C910208
EIP	004015A5 Malware_.004015A5

00401577	55	PUSH EBP	
00401578	8BEC	MOV EBP,ESP	
00401579	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:[<&KERNEL32.GetVersion	kernel32.GetVersion
004015A6	33D2	XOR EDX,EDX	
004015A7	8AD4	MOV DL,AH	
004015A8	8BC9	MOV ECX,EBX	
004015AB	81E1 FF000000	AND ECX,0FF	

Registers (FPU)	
EAX	0A280105
ECX	00000000
EDX	00000000
EBX	7FFD7000
ESP	0012FF94
EBP	0012FFC0
ESI	FFFFFFFF
EDI	7C910208 ntdll.7C910208
EIP	004015AF Malware_.004015AF
C 0	ES 0023 32bit 0 (FFFFFFFF)
P 1	CS 0018 32bit 0 (FFFFFFFF)
A 0	SS 0023 32bit 0 (FFFFFFFF)
Z 1	DS 0023 32bit 0 (FFFFFFFF)
S 0	FS 003B 32bit 7FDE000 (FFF)
0 0	GS 0000 NULL
LastErr ERROR_INVALID_HANDLE (00000006)	

Dopodiché andiamo ad inserire un secondo breakpoint all'indirizzo 004015AF ed andiamo a controllare il valore attuale di ECX, il quale è in decimale 170393861:

00401577	55	PUSH EBP	
00401578	8BEC	MOV EBP,ESP	
00401579	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:[<&KERNEL32.GetVersion	kernel32.GetVersion
004015A6	33D2	XOR EDX,EDX	
004015A7	8AD4	MOV DL,AH	
004015A8	8BC9	MOV ECX,EBX	
004015AB	81E1 FF000000	AND ECX,0FF	
004015AC	8900 D0524000	MOV DWORD PTR DS:[4052D0],ECX	

Registers (FPU)	
EAX	0A280105
ECX	00000005
EDX	00000000
EBX	7FFD7000
ESP	0012FF94
EBP	0012FFC0
ESI	FFFFFFFF
EDI	7C910208 ntdll.7C910208
EIP	004015B5 Malware_.004015B5
C 0	ES 0023 32bit 0 (FFFFFFFF)
P 1	CS 0018 32bit 0 (FFFFFFFF)
A 0	SS 0023 32bit 0 (FFFFFFFF)
Z 0	DS 0023 32bit 0 (FFFFFFFF)
S 0	FS 003B 32bit 7FDE000 (FFF)
0 0	GS 0000 NULL
LastErr ERROR_INVALID_HANDLE (00000006)	

La stringa va ad eseguire l'operatore logico AND con il valore in decimale 170393861 (ECX) e 255 (0FF), dando come risultato 00000005:

00401577	55	PUSH EBP	
00401578	8BEC	MOV EBP,ESP	
00401579	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:[<&KERNEL32.GetVersion	kernel32.GetVersion
004015A6	33D2	XOR EDX,EDX	
004015A7	8AD4	MOV DL,AH	
004015A8	8BC9	MOV ECX,EBX	
004015AB	81E1 FF000000	AND ECX,0FF	
004015AC	8900 D0524000	MOV DWORD PTR DS:[4052D0],ECX	

Registers (FPU)	
EAX	0A280105
ECX	00000005
EDX	00000000
EBX	7FFD7000
ESP	0012FF94
EBP	0012FFC0
ESI	FFFFFFFF
EDI	7C910208 ntdll.7C910208
EIP	004015B5 Malware_.004015B5
C 0	ES 0023 32bit 0 (FFFFFFFF)
P 1	CS 0018 32bit 0 (FFFFFFFF)
A 0	SS 0023 32bit 0 (FFFFFFFF)
Z 0	DS 0023 32bit 0 (FFFFFFFF)
S 0	FS 003B 32bit 7FDE000 (FFF)
0 0	GS 0000 NULL
LastErr ERROR_INVALID_HANDLE (00000006)	

Se si vuole fare il calcolo dell'operatore AND, si può andare alla seguente URL (<https://toolslick.com/math/bitwise/and-calculator>) e possiamo inserire i nostri valori e ci darà il valore come di seguito:

Calculation

Type	Decimal	Binary
Input 1	170393861	1010001010000000000100000101
Input 2	000000255	0000000000000000000001111111
Final Result	000000005	0000000000000000000000000101

Ipotizzo che la funzione di questo Malware sia quello di creare una backdoor per avere modo di eseguire codici in remoto, e lo deduco dal fatto che il Malware crea un Socket andandosi a connettere ad esso, e queste sono le stringhe di codice da cui l'ho dedotto:

```
00401284 |> 6A 00 | PUSH 0 | Flags = 0
00401286 |. 6A 00 | PUSH 0 | Group = 0
00401288 |. 6A 00 | PUSH 0 | pWSAProtocol = NULL
0040128A |. 6A 06 | PUSH 6 | Protocol = IPPROTO_TCP
0040128C |. 6A 01 | PUSH 1 | Type = SOCK_STREAM
0040128E |. 6A 02 | PUSH 2 | Family = AF_INET
00401290 |. FF15 A0404000 | CALL DWORD PTR DS:[&WS2_32.WSASocketA] | WSASocketA
```