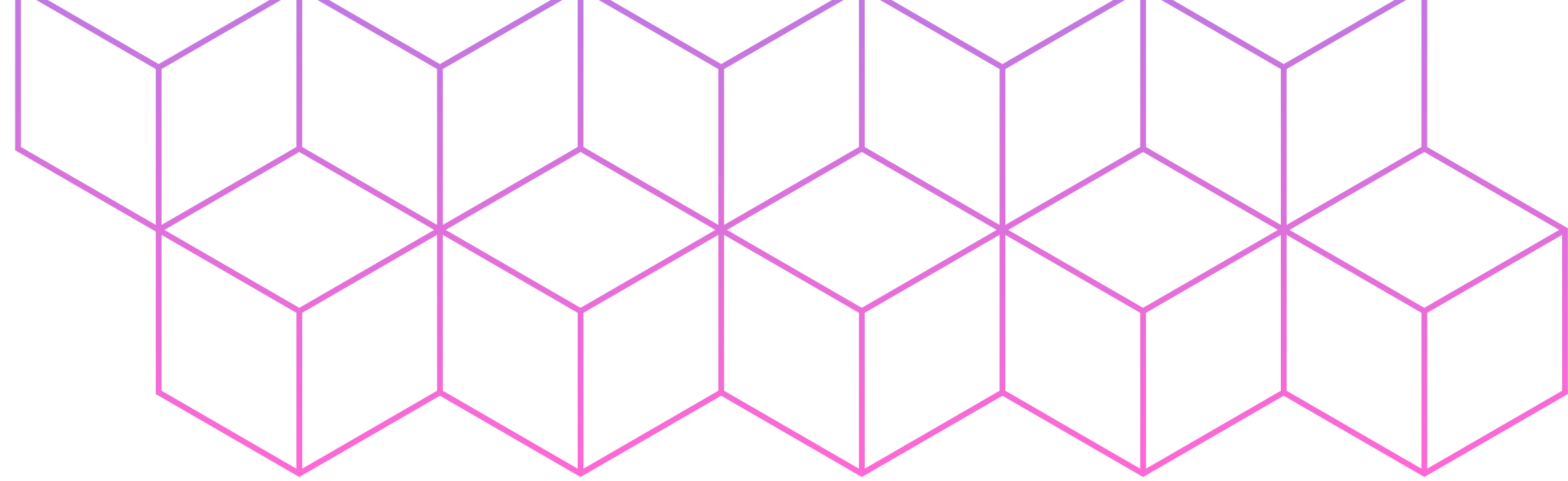
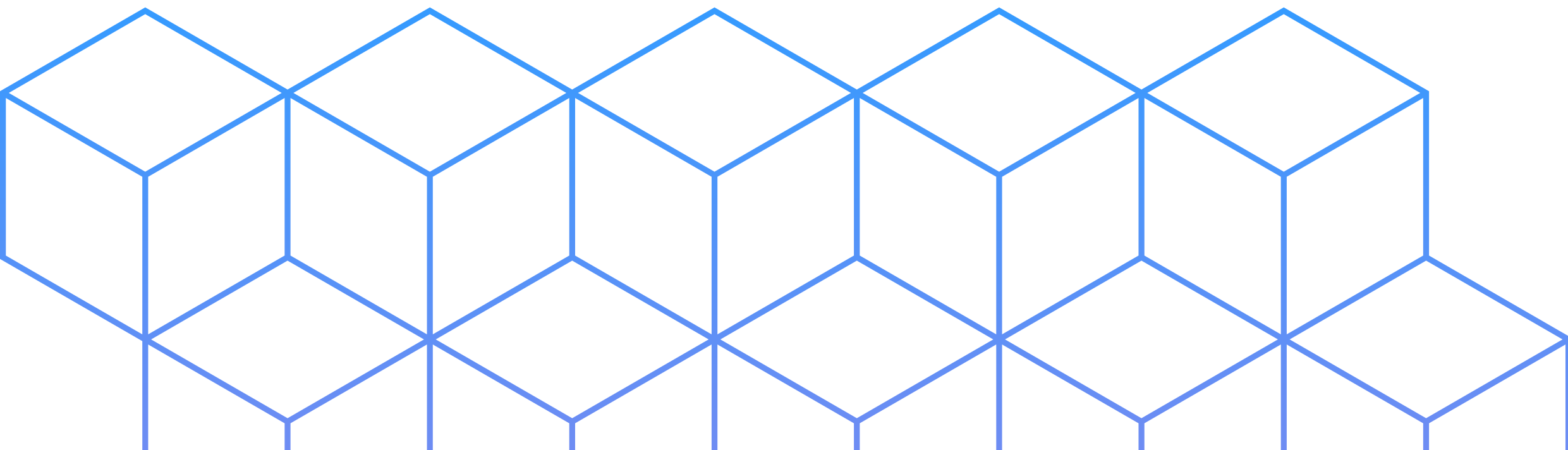


**Epicode**



# Malware Analysis

## OllyDBG



- Traccia
- 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? 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 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 un step-into. Qual è ora il valore di ECX? Spiegate quale istruzione è stata eseguita.
- BONUS: spiegare a grandi linee il funzionamento del malware



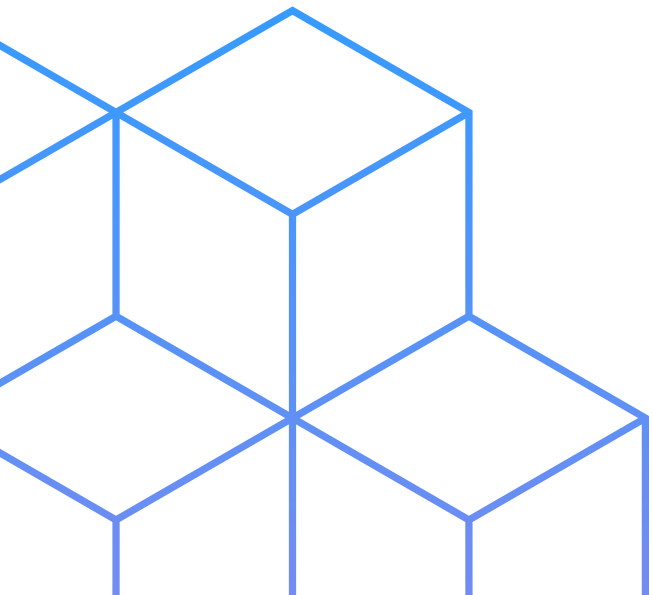
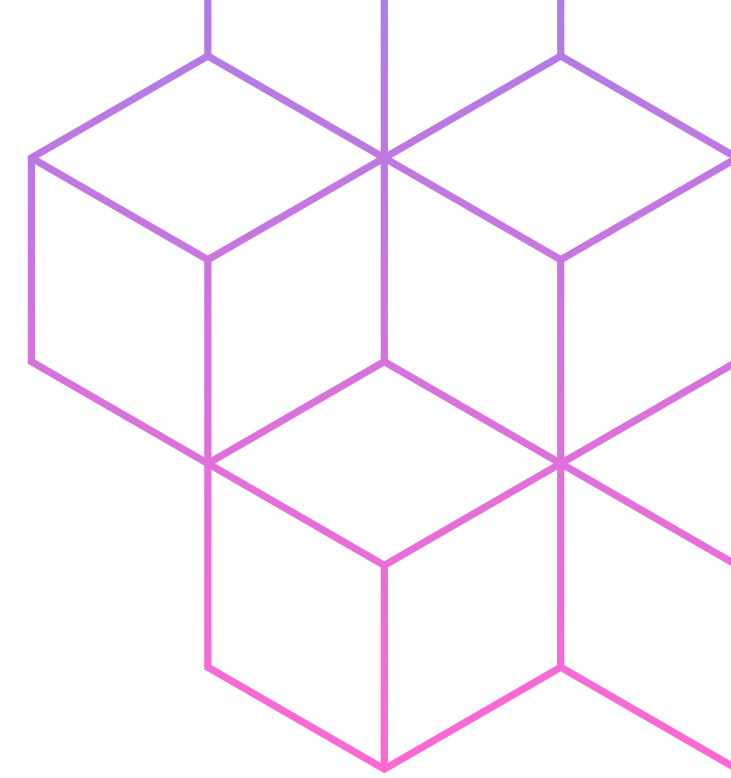
# Indice



# 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



# Qual è il valore del parametro «CommandLine» che viene passato sullo stack? (1)

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.CreatePro	CreateProcessA
00401074	. 8945 EC	MOV DWORD PTR SS:[EBP-14],EAX	
00401077	. 6A FF	PUSH -1	Timeout = INFINITE
00401079	. 8B4D F0	MOV ECX,DWORD PTR SS:[EBP-10]	
0040107C	. 51	PUSH ECX	hObject
0040107D	. FF15 00404000	CALL DWORD PTR DS:[&KERNEL32.WaitForSi	WaitForSingleObject

Il valore del parametro e "CMD", dunque il prompt dei comandi di Windows, possiamo notarlo nella figura all'indirizzo 00401067

Spostandoci all'allocazione di memoria 004015A3 andiamo ad impostare un breakpoint su una specifica istruzione nel programma.

Inizialmente il regis

Inizialmente il registro EDX ha valore 00401577



**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?**

Interrompendo l'esecuzione del programma possiamo controllare il valore del registro EDX e verificiamo che sembrerebbe essere 00001DB1. Continuiamo avviando l'esecuzione del programma cliccando sul tasto "play" nella barra degli strumenti. Attraverso la finestra "Registers FPU" è stato possibile verificare che il valore nel registro EDX sia rimasto lo stesso.

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
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	
004015C6	. C1E8 10	SHR EAX,10	
004015C9	. A3 C8524000	MOV DWORD PTR DS:[4052C8],EAX	
004015CE	. 6A 00	PUSH 0	

Registers (FPU)  
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 7EFD0000(FFF)  
T 0 GS 002B 32bit 0(FFFFFFFF)

Utilizzando la funzione "Step-into" possiamo entrare nel codice della funzione in analisi. In questa fase possiamo notare che il valore del registro EDX è cambiato a 0. Questo cambiamento è dovuto all'operazione logica XOR nel codice, che restituisce sempre 0 quando applicata a due valori uguali. In questo caso, l'operazione XOR ha annullato il valore precedente di EDX, impostandolo a zero.

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
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	

ECX 7EFDE000  
EDX 00000000  
EBX 7EFDE000  
ESP 0018FF5C  
EBP 0018FF88  
ESI 00000000  
EDI 00000000  
EIP 004015A5 Malwar  
C 0 ES 002B 32bit  
P 1 CS 0023 32bit

**Inserite un secondo breakpoint all'indirizzo di memoria 004015AF. Qual è il valore del registro ECX? Eseguite un step-into. Qual è ora il valore di ECX? Spiegate quale istruzione è stata eseguita.**

00401597

0040159A

0040159D

004015A3

004015A5

004015A7

004015AD

004015AF

004015B5

004015B8

004015BE

004015C0

004015C6

004015C9

004015CE

004015D0

004015D5

. 8965 E8

. FF15 30404000

. 33D2

. 8AD4

. 8915 D4524000

. 8BC8

. 81E1 FF000000

. 890D D0524000

. C1E1 08

. 03CA

. 890D CC524000

. C1E8 10

. A3 C8524000

. 6A 00

. E8 33090000

. 59

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

kernel32.GetVersion

ECX 1DB10106

EDX 00000001

EBX 7EFDE000

ESP 0018FF5C

EBP 0018FF88

ESI 00000000

EDI 00000000

EIP 004015AF Malware\_.00

C 0 ES 002B 32bit 0(FFF

P 1 CS 0023 32bit 0(FFF

A 0 SS 002B 32bit 0(FFF

Z 1 DS 002B 32bit 0(FFF

S 0 FS 0053 32bit 7EFDD

T 0 GS 002B 32bit 0(FFF

D 0

O 0 LastErr ERROR SUCCE

Configuriamo il secondo breakpoint all'indirizzo di memoria 004015AF. Il valore del registro ECX è «1DB10106».

004015A7

004015AD

004015AF

004015B5

004015B8

004015BE

004015C0

004015C6

004015C9

004015CE

004015D0

004015D5

. 8915 D4524000

. 8BC8

. 81E1 FF000000

. 890D D0524000

. C1E1 08

. 03CA

. 890D CC524000

. C1E8 10

. A3 C8524000

. 6A 00

. E8 33090000

. 59

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

Registers (FPU)

EAX 1DB10106

ECX 00000006

EDX 00000001

EBX 7EFDE000

ESP 0018FF5C

EBP 0018FF88

ESI 00000000

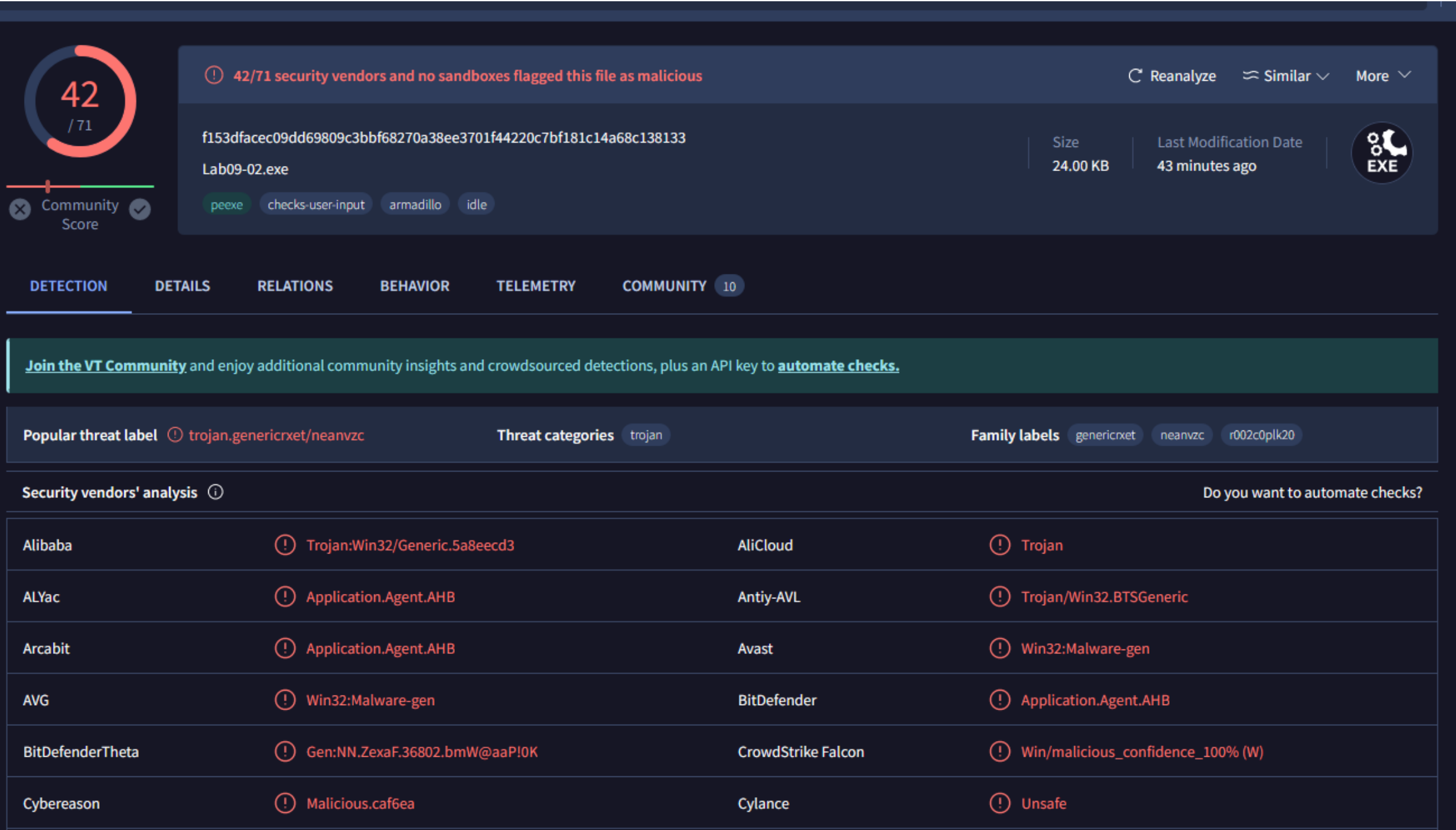
EDI 00000000

EIP 004015B5 Malware\_.004015B5

C 0 ES 002B 32bit 0(FFFFFFF)

Dopo lo step-into il valore del registro ECX è stato modificato in «00000006» in quanto è stata eseguita l'istruzione AND ECX, FF. Se l'operazione restituisce un risultato vero, il valore di ECX viene modificato

# BONUS: spiegare a grandi linee il funzionamento del malware



Grazie a VirusTotal possiamo scoprire che il malware viene identificato molto probabilmente come trojan.