

Report Analisi Malware

Introduzione:

In questo report, analizziamo il comportamento di un malware utilizzando **OllyDbg**, un debugger di livello utente per Microsoft Windows. **OllyDbg** è uno strumento potente per il debugging di codice binario e assembly, che permette di monitorare l'esecuzione di programmi, ispezionare registri, stack e memoria, e inserire breakpoints per analizzare il flusso di esecuzione in modo dettagliato.



Traccia



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

Quesito 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?

All'indirizzo 0040106E, il malware esegue una chiamata alla funzione CreateProcessA della libreria kernel32.dll. Il valore del parametro "CommandLine" passato allo stack è "cmd", come si osserva all'indirizzo 00401067. Questo parametro indica che il malware sta tentando di avviare il prompt dei comandi di Windows.

00401066	50	PUSH EDX	pCreateInfo
00401056	52	PUSH EDX	pStartupInfo
00401057	8D45 A8	LEA EAX, DWORD PTR SS:[EBP-58]	CurrentDir = NULL
0040105A	50	PUSH EAX	pEnvironment = NULL
0040105B	6A 00	PUSH 0	CreationFlags = 0
0040105D	6A 00	PUSH 0	InheritHandles = TRUE
0040105F	6A 00	PUSH 0	pThreadSecurity = NULL
00401061	6A 01	PUSH 1	pProcessSecurity = NULL
00401063	6A 00	PUSH 0	CommandLine = "cmd"
00401065	6A 00	PUSH 0	ModuleFileName = NULL
00401067	68 30504000	PUSH Malware_.00405030	CreateProcessA
0040106C	6A 00	PUSH 0	
0040106E	FF15 04404000	CALL DWORD PTR DS:[&KERNEL32.CreateProcessA]	

Quesito 2

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?

All'indirizzo 004015A3, inseriamo un breakpoint software. Prima di eseguire lo step-into, il valore del registro EDX è 00001DB1. Dopo aver eseguito lo step-into, il valore del registro EDX diventa 00000000. Questo cambiamento è dovuto all'istruzione XOR EDX, EDX che azzera il contenuto di EDX.

Address	Module	Active	Disassembly
004015A3	Malware_	Always	XOR EDX, EDX pushad

Prima

Registers (FPU)

ERX	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 7EFDD0000(FFF)
T 0 GS 002B 32bit 0(FFFFFFFF)
D 0
0 0 LastErr: ERROR_SUCCESS (00000000)
EFL 00000206 (NO,NB,NE,A,NS,PE,GE,BI)

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 0 0 (GT)
FCW 027F Preo NEAR,S3 Mask 1 1 1 1 1 1

Dopo

Registers (FPU)

ERX	1DB10106
ECX	7EFDE000
EDX	00000000
EBX	7EFDE000
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 7EFDD0000(FFF)
T 0 GS 002B 32bit 0(FFFFFFFF)
D 0
0 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

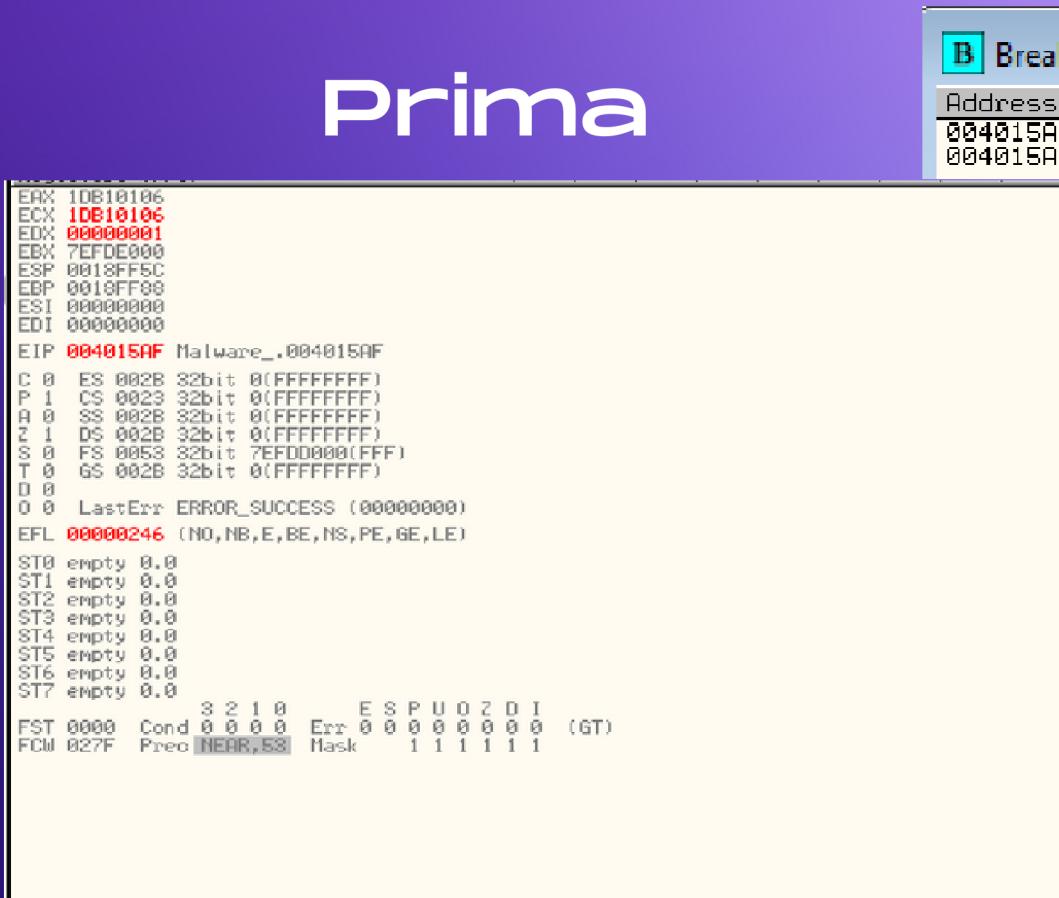
FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 0 (GT)
FCW 027F Preo NEAR,S3 Mask 1 1 1 1 1 1

Quesito 3

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.

All'indirizzo 004015AF, inseriamo un secondo breakpoint. Prima di eseguire lo step-into, il valore del registro ECX è 1DB10106. Dopo aver eseguito lo step-into, il valore del registro ECX diventa 00000006. Questo cambiamento è dovuto all'istruzione AND ECX, FF, che esegue un'operazione logica AND tra ECX e il valore esadecimale FF, mantenendo solo i bit meno significativi.

Prima



Registers (FPU) < <
EAX 1DB10106
ECX 1DB10106
EDX 00000001
EBX 7EFDE000
ESP 0018FF5C
EBP 0018FF98
ESI 00000000
EDI 00000000

EIP 004015AF Malware_.004015AF
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 7EFDD000(FFF)
T 0 GS 002B 32bit 0(FFFFFFFF)
D 0
D 0 LastErr ERROR_SUCCESS (00000000)
EFL 00000246 (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

3 2 1 0 E S P U 0 2 D I
FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 0 (GT)
FCM 027F Preo NEAR,B3 Mask 1 1 1 1 1 1

Dopo



Registers (FPU) < <
EAX 1DB10106
ECX 00000006
EDX 00000001
EBX 7EFDE000
ESP 0018FF5C
EBP 0018FF98
ESI 00000000
EDI 00000000

EIP 004015B5 Malware_.004015B5
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 7EFDD000(FFF)
T 0 GS 002B 32bit 0(FFFFFFFF)
D 0
D 0 LastErr ERROR_SUCCESS (00000000)
EFL 00010206 (NO,NB,NE,R,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

8 2 1 0 E S P U 0 2 D I
FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 0 (GT)
FCM 027F Preo NEAR,B3 Mask 1 1 1 1 1 1

Quesito bonus

Funzionamento generale del malware:

1. Creazione di Processi:

- Il malware utilizza la funzione CreateProcessA per creare nuovi processi. In questo caso, crea un'istanza del Command Prompt (cmd). Ciò consente al malware di eseguire comandi di sistema arbitrari, che possono includere il lancio di ulteriori payloads, l'esecuzione di comandi dannosi o la manipolazione di file e configurazioni di sistema.

2. Manipolazione dei Registri:

- Le istruzioni come XOR EDX, EDX e AND ECX, FF sono utilizzate per manipolare i registri. XOR EDX, EDX azzerà il registro EDX, mentre AND ECX, FF isola l'ultimo byte di ECX, azzerando i restanti bit. Queste tecniche sono utilizzate per preparare i valori nei registri per ulteriori operazioni e per evitare il rilevamento.

3. Utilizzo di Funzioni di Sistema:

- Il malware fa uso di diverse funzioni di sistema, tra cui GetVersion, GetCommandLineA, WaitForSingleObject, ecc. Queste funzioni gli permettono di raccogliere informazioni sul sistema, eseguire comandi specifici e sincronizzarsi con altri processi.

4. Networking:

- Funzioni come WSAStartup, WSASocketA, connect, closesocket, ecc., indicano che il malware tenta di stabilire connessioni di rete. Queste connessioni possono essere utilizzate per comunicare con un server di comando e controllo (C&C), scaricare ulteriori componenti del malware o esfiltrare dati dal sistema compromesso.

5. Persistenza:

- Il malware può includere meccanismi per garantirsi la persistenza sul sistema infetto. Ciò può includere la modifica di chiavi di registro, la creazione di task schedulati o l'utilizzo di tecniche di iniezione di codice per assicurarsi che venga eseguito anche dopo un riavvio del sistema.