

Esercizio S11/L3

Malware analysis dinamica con Ollydbg

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

Punto 1

Ollydbg (Olly debug) è un software per windows in grado di analizzare un eseguibile mentre è in esecuzione. In tempo reale riesce a tracciare i registri, identificare le funzioni con i parametri e le variabili che vengono passate sullo stack.

Nella figura accanto possiamo vedere la finestra di ollydbg che mostra lo stack di memoria. Il parametro della "Commandline" passato sullo stack è "cmd", cioè la shell di windows.

0018FF6C	00000000	ModuleFileName = NULL
0018FF70	00405030	CommandLine = "cmd"
0018FF74	00000000	pProcessSecurity = NULL
0018FF78	00000000	pThreadSecurity = NULL
0018FF7C	00000001	InheritHandles = TRUE
0018FF80	00000000	CreationFlags = 0
0018FF84	00000000	pEnvironment = NULL
0018FF88	00000000	CurrentDir = NULL
0018FF8C	0018FF94	pStartupInfo = 0018FF94
0018FF90	0018FFDC	pProcessInfo = 0018FFDC
0018FF94	00000044	
0018FF98	00000000	

Punti 2, 3, 4 e 5

All'indirizzo 004015A3, come vediamo dalla prima figura, inizialmente il valore del registro EDX è 1DB1 (7601 convertito in decimale). Nella seconda figura il valore diventa 0, dopo aver effettuato lo step-into. Questo perchè viene eseguito lo XOR del registro EDX con se stesso, che da come risultato sempre 0.

```
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 7EFDD000(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,G)
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 Prec NEAR,S3 Mask 1 1 1 1 1 1
```

```
EAX 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 7EFDD000(FFF)
T 0 GS 002B 32bit 0(FFFFFFFF)
D 0
O 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
FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 0 (GT)
FCW 027F Prec NEAR,S3 Mask 1 1 1 1 1 1
```

Punti 6, 7 e 8

All'istruzione 004015AF inizialmente il valore di ECX è 1DB10106 (498139398 convertito in decimale) e poi, dopo lo step-into, diventa 6. L'istruzione che viene eseguita è l'AND, cioè il prodotto logico.

```
EAX 1DB10106
ECX 1DB10106
EDX 00000001
EBX 7EFDE000
ESP 0018FF5C
EBP 0018FF88
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)
V 1 DS 002B 32bit 0(FFFFFFFF)
Z 0 FS 0053 32bit 7EFDD000(FFF)
T 0 GS 002B 32bit 0(FFFFFFFF)
D 0
O 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
FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 (GT)
FCW 027F Prec NEAR,53 Mask 1 1 1 1 1 1
```

```
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(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
O 0 LastErr ERROR_SUCCESS (00000000)
EFL 00000206 (NO,NB,NE,A,NS,PE,GE,G)
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 (GT)
FCW 027F Prec NEAR,53 Mask 1 1 1 1 1 1
```

Bonus

Data la presenza di una istruzione che cerca di aprire una shell (Commandline) possiamo ipotizzare che il malware in questione sia un rootkit.