

MALWARE ANALYSIS

OllyDBG

TRACCIA

ARGOMENTI TRATTATI NELLA PRESENTAZIONE

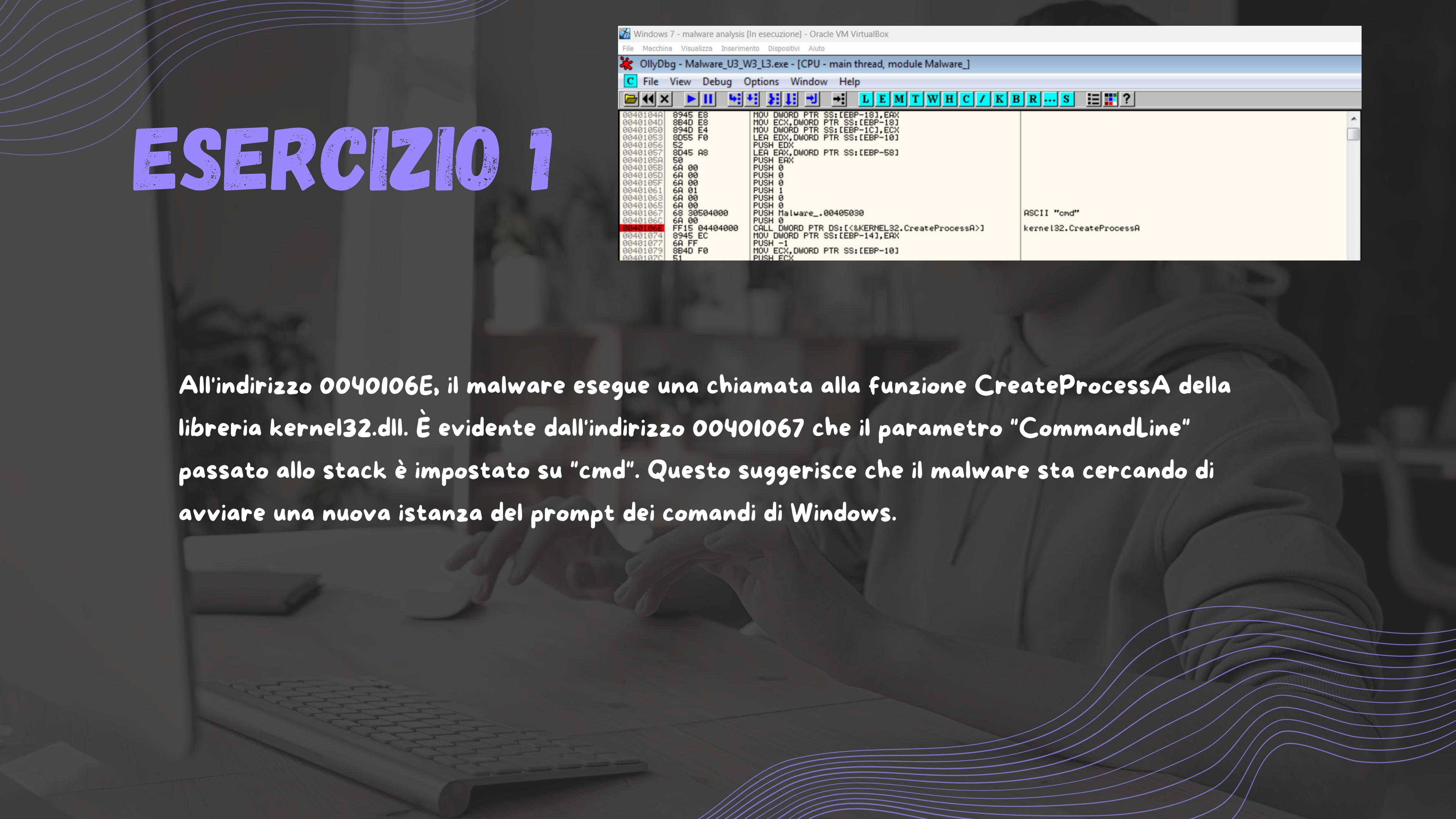
Fate riferimento al malware: Malware_U3_W3_L3, presente all'interno della cartella Esercizio_Pratico_U3_W3_L3 sul desktop della macchina virtuale dedicata all'analisi dei malware. Rispondete ai seguenti quesiti utilizzando OllyDBG.

- 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

INTRODUZIONE

Oggi esploreremo il comportamento di un malware utilizzando OllyDbg, un potente debugger per codice binario e assembly su piattaforma Microsoft Windows. OllyDbg permette di monitorare l'esecuzione di programmi in dettaglio, ispezionando registri, stack e memoria, e inserendo breakpoints strategici per analizzare il flusso di esecuzione. Per gli analisti di sicurezza informatica e i ricercatori di malware, OllyDbg rappresenta uno strumento indispensabile, consentendo l'identificazione accurata di comportamenti dannosi e la comprensione approfondita dei meccanismi interni utilizzati dai malware.

ESERCIZIO 1



The screenshot shows the OllyDbg debugger interface. The title bar reads "Windows 7 - malware analysis [In esecuzione] - Oracle VM VirtualBox" and "OllyDbg - Malware_U3_W3_L3.exe - [CPU - main thread, module Malware_]".

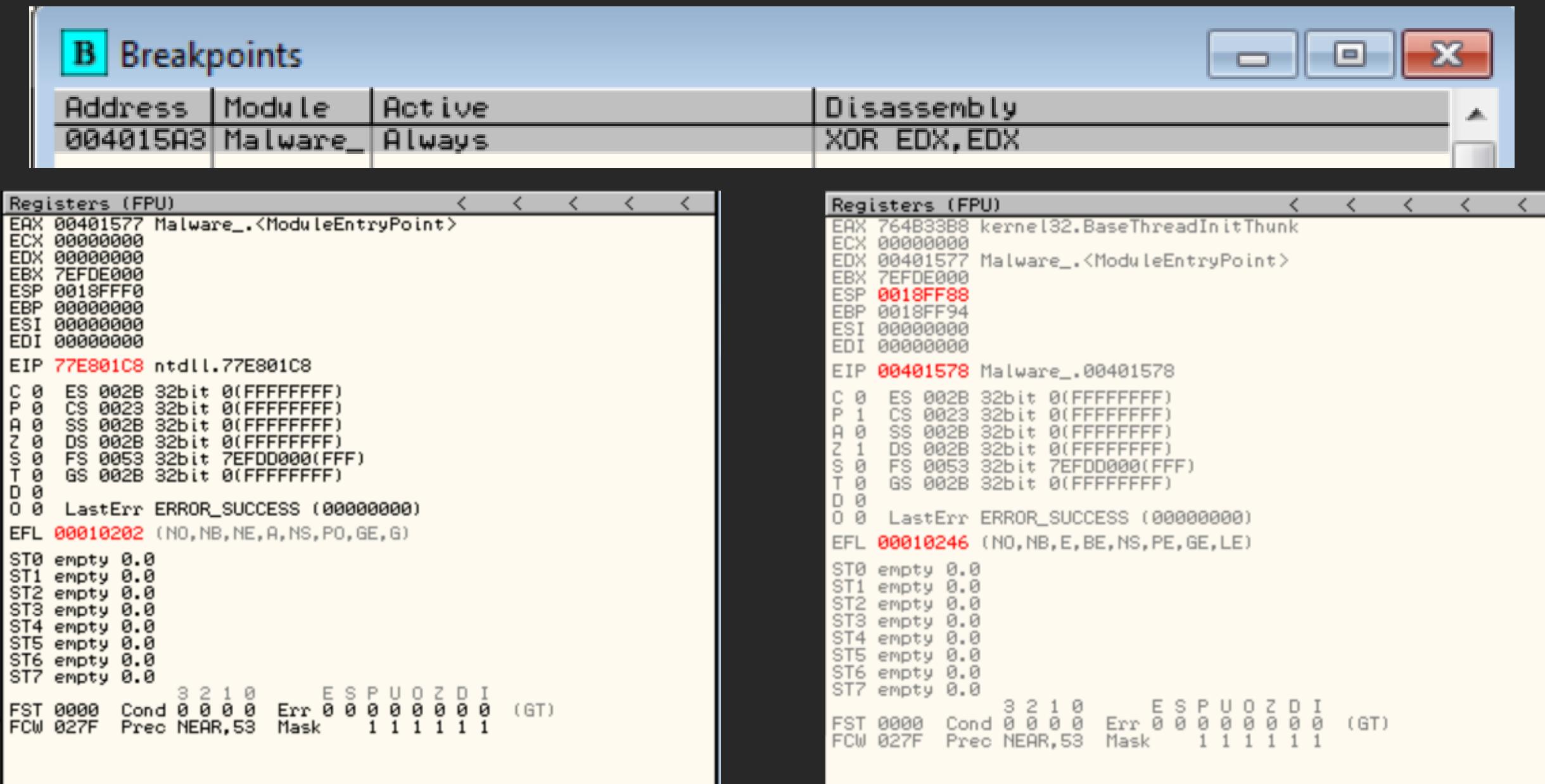
The assembly code window displays the following assembly instructions:

```
00401040 8945 E8      MOV DWORD PTR SS:[EBP-18], EAX
0040104D 884D E8      MOV ECX, DWORD PTR SS:[EBP-18]
00401050 8944 E4      MOV DWORD PTR SS:[EBP-1C], ECX
00401053 8055 F0      LEA EDX, DWORD PTR SS:[EBP-10]
00401056 52           PUSH EDX
00401057 8045 A8      LEA EAX, DWORD PTR SS:[EBP-58]
0040105A 50           PUSH EAX
0040105B 6A 00         PUSH 0
0040105D 6A 00         PUSH 0
0040105F 6A 00         PUSH 0
00401061 6A 01         PUSH 1
00401063 6A 00         PUSH 0
00401065 6A 00         PUSH 0
00401067 68 30504000  PUSH Malware_.00405030
0040106C 6A 00         PUSH 0
0040106E FF15 04404000 CALL DWORD PTR DS:[&KERNEL32.CreateProcessA]
00401074 8945 EC      MOV DWORD PTR SS:[EBP-14], EAX
00401077 6A FF         PUSH -1
00401079 884D F0      MOV ECX, DWORD PTR SS:[EBP-10]
0040107C 51           PUSH ECX
```

The instruction at address 0040106E is highlighted in red. To its right, the ASCII string "cmd" is shown, followed by the function name "kernel32.CreateProcessA".

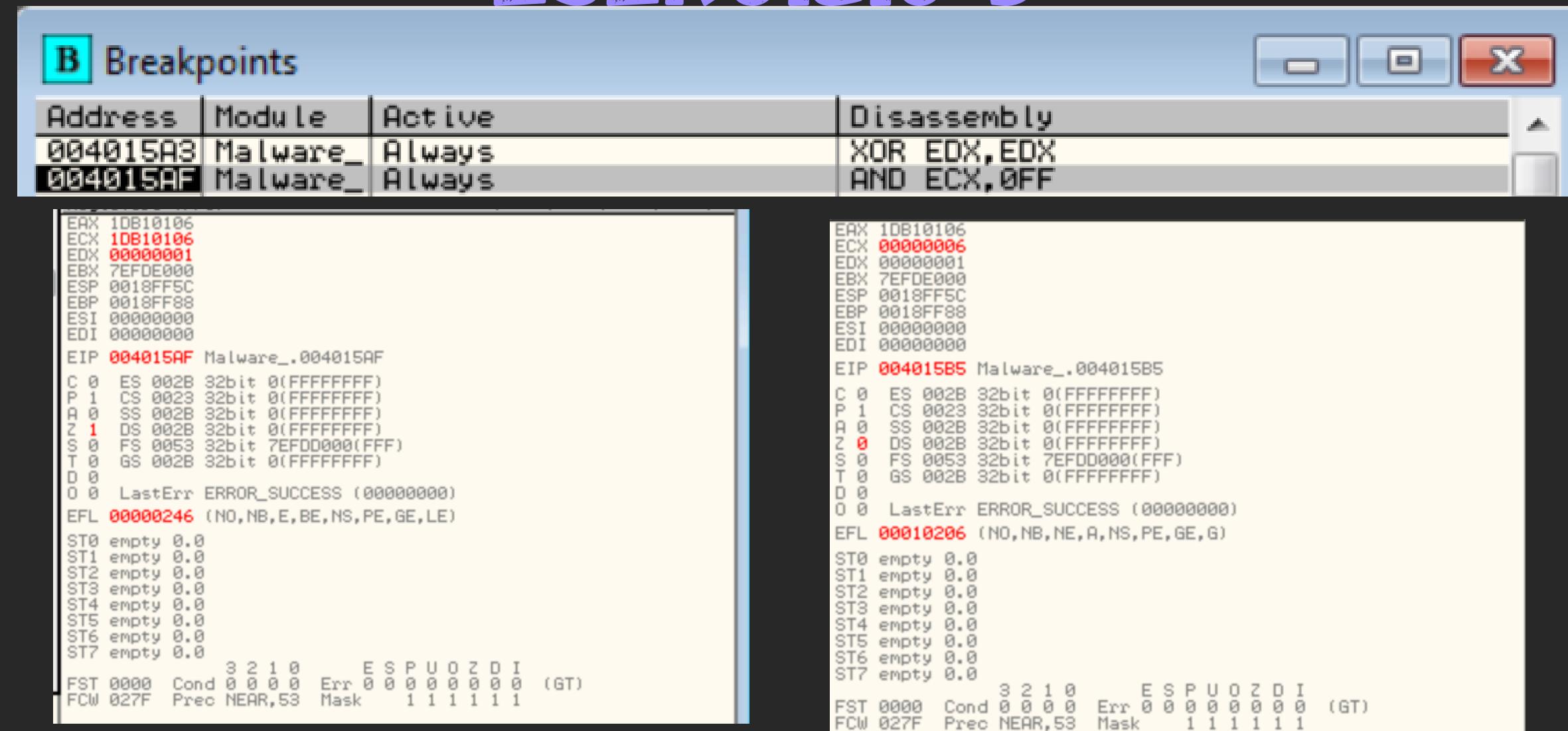
All'indirizzo 0040106E, il malware esegue una chiamata alla funzione CreateProcessA della libreria kernel32.dll. È evidente dall'indirizzo 00401067 che il parametro "CommandLine" passato allo stack è impostato su "cmd". Questo suggerisce che il malware sta cercando di avviare una nuova istanza del prompt dei comandi di Windows.

ESERCIZIO 2



All'indirizzo `004015A3`, abbiamo impostato un breakpoint software. Prima di eseguire lo step-into, il valore del registro `EDX` era `00001DB1`. Dopo aver eseguito lo step-into, il valore del registro `EDX` è cambiato a `00000000`. Questo cambiamento è causato dall'istruzione `XOR EDX, EDX` che azzera il contenuto di `EDX`. L'istruzione `XOR EDX, EDX` è un'operazione comune utilizzata per azzerare un registro senza utilizzare un valore costante. L'effetto di questa operazione è che il contenuto di `EDX` viene impostato a zero, indipendentemente dal suo valore precedente. Pertanto, il valore di `EDX` è cambiato da `00001DB1` a `00000000` dopo l'esecuzione di questa istruzione `XOR`.

ESERCIZIO 3



All'indirizzo 004015AF, abbiamo inserito un secondo breakpoint. Prima di eseguire lo step-into, il valore del registro ECX era 1DB10106. Dopo aver eseguito lo step-into, il valore del registro ECX è cambiato a 00000006. Questo cambiamento è dovuto all'istruzione AND ECX, FF, che esegue un'operazione logica AND tra ECX e il valore esadecimale FF.

L'operazione AND logico tra ECX e FF comporta la seguente manipolazione dei bit:

- Il valore binario di FF è 11111111.
- L'operazione AND viene eseguita bit per bit tra ECX (1DB10106) e FF.
- Solo i bit corrispondenti che sono entrambi impostati su 1 in entrambi gli operandi rimarranno impostati a 1 nel risultato finale.

Quindi, applicando l'AND logico:

- ECX (1DB10106) AND FF (11111111) = 00000006.

Questo significa che l'istruzione AND ha preservato solo i bit meno significativi di ECX, che erano impostati a 06 dopo l'esecuzione.

FUNZIONAMENTO

1. Creazione di Processi: Il malware utilizza la funzione `CreateProcessA` per avviare nuovi processi. Un esempio comune è la creazione di un'istanza del Command Prompt (cmd). Questo consente al malware di eseguire comandi di sistema arbitrari, facilitando l'esecuzione di payload aggiuntivi, comandi dannosi o la manipolazione di file e configurazioni di sistema.
2. Manipolazione dei Registri: Il malware manipola i registri utilizzando istruzioni come `XOR EDX, EDX` e `AND ECX, FF`. `XOR EDX, EDX` azzerà completamente il registro EDX, mentre `AND ECX, FF` isola e preserva solo il byte meno significativo di ECX, azzerando gli altri bit. Queste tecniche preparano i registri per ulteriori operazioni e possono contribuire a evitare il rilevamento da parte delle tecniche di analisi statica e dinamica.
3. Utilizzo di Funzioni di Sistema: Il malware sfrutta varie funzioni di sistema come `GetVersion`, `GetCommandLineA`, `WaitForSingleObject` e altre. Queste funzioni consentono al malware di raccogliere informazioni sul sistema ospite, eseguire comandi specifici e sincronizzarsi con altri processi in esecuzione sul sistema.
4. Networking: Il malware utilizza funzioni di networking come `WSAStartup`, `WSASocketA`, `connect`, `closesocket`, ecc. Queste indicano che il malware sta tentando 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 include meccanismi per garantire la sua persistenza sul sistema infetto anche dopo un riavvio. Questo 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 il malware venga eseguito automaticamente ad ogni avvio del sistema. Questi componenti sono tipici di molti malware, che combinano tecniche avanzate di programmazione e sfruttano le vulnerabilità dei sistemi per scopi dannosi come il furto di dati, il controllo remoto del sistema infetto, o il sabotaggio delle operazioni normali del sistema.

THANK YOU

Antonio Perna