

2024



CS0124

REPORT

Week 10 Lesson 5

PREPARED BY : Bruno Falconi

Traccia:

Con riferimento al file **Malware_U3_W2_L5** presente all'interno della cartella «**Esercizio_Pratico_U3_W2_L5**» sul desktop della macchina virtuale dedicata per l'analisi dei malware, rispondere ai seguenti quesiti:

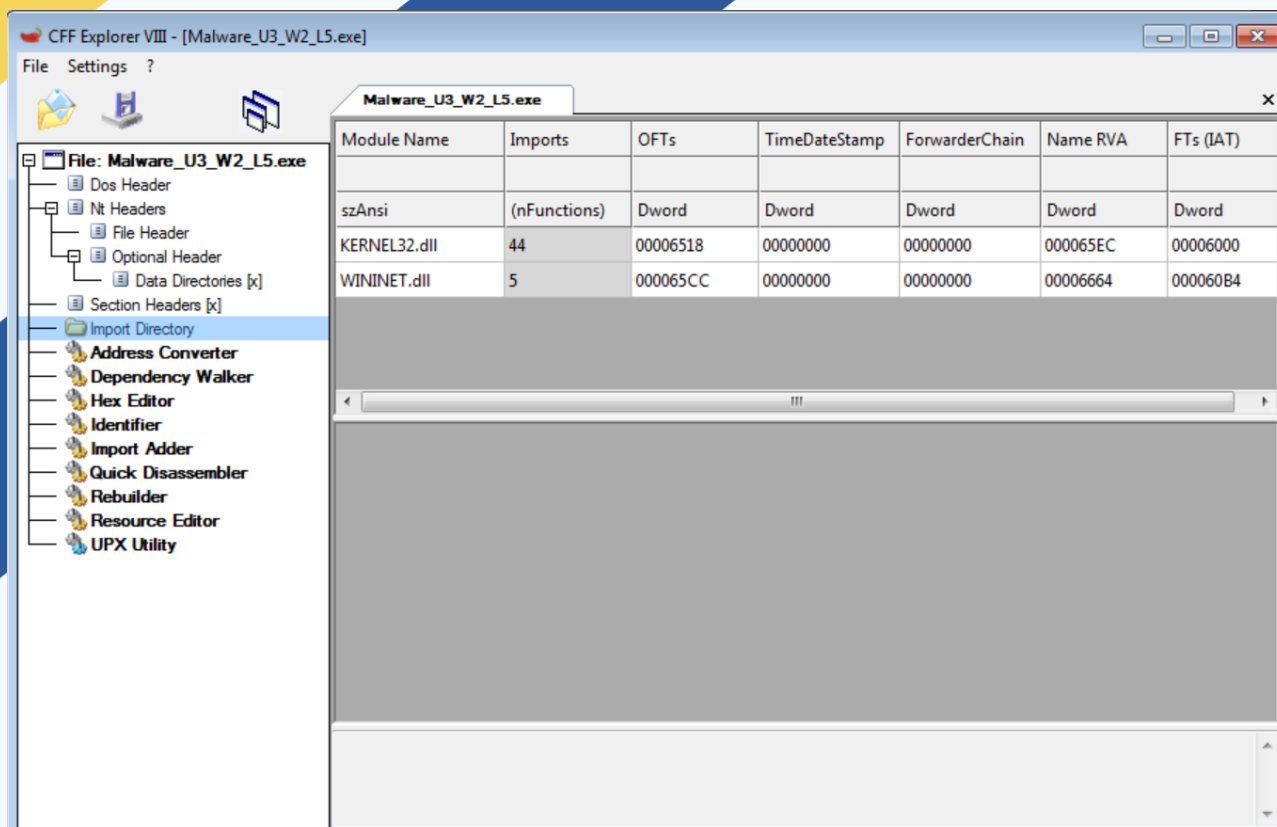
1. Quali **librerie** vengono importate dal file eseguibile?
 2. Quali sono le **sezioni** di cui si compone il file eseguibile del malware?
- Di entrambi dare una breve spiegazione

Con riferimento alla figura in slide 3, risponde ai seguenti quesiti:

3. Identificare i **costrutti** noti (creazione dello stack, eventuali cicli, altri costrutti)
4. **Ipotizzare il comportamento della funzionalità implementata**
5. BONUS fare tabella con significato delle singole righe di codice assembly

Analisi del File Eseguibile del Malware

1. Librerie Importate:



Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.dll	44	00006518	00000000	00000000	000065EC	00006000
WININET.dll	5	000065CC	00000000	00000000	00006664	000060B4

Il file eseguibile del malware importa due librerie principali:

KERNEL.dll: Questa libreria fornisce funzionalità di base del sistema operativo Windows, come la gestione della memoria, la creazione e il controllo dei processi e la gestione delle risorse di sistema.

WININET.dll: Questa libreria è coinvolta nella comunicazione su Internet e fornisce funzioni per l'accesso a risorse su Internet, come il download di file, l'invio di dati a server remoti e la verifica dello stato della connessione Internet.

Funzioni Importate:

Tra le funzioni importate dal malware troviamo:

InternetOpenUrlA: Utilizzata per aprire un'URL specificato in una sessione Internet, suggerendo che il malware potrebbe scaricare ulteriori componenti dannosi o inviare dati a un server remoto.

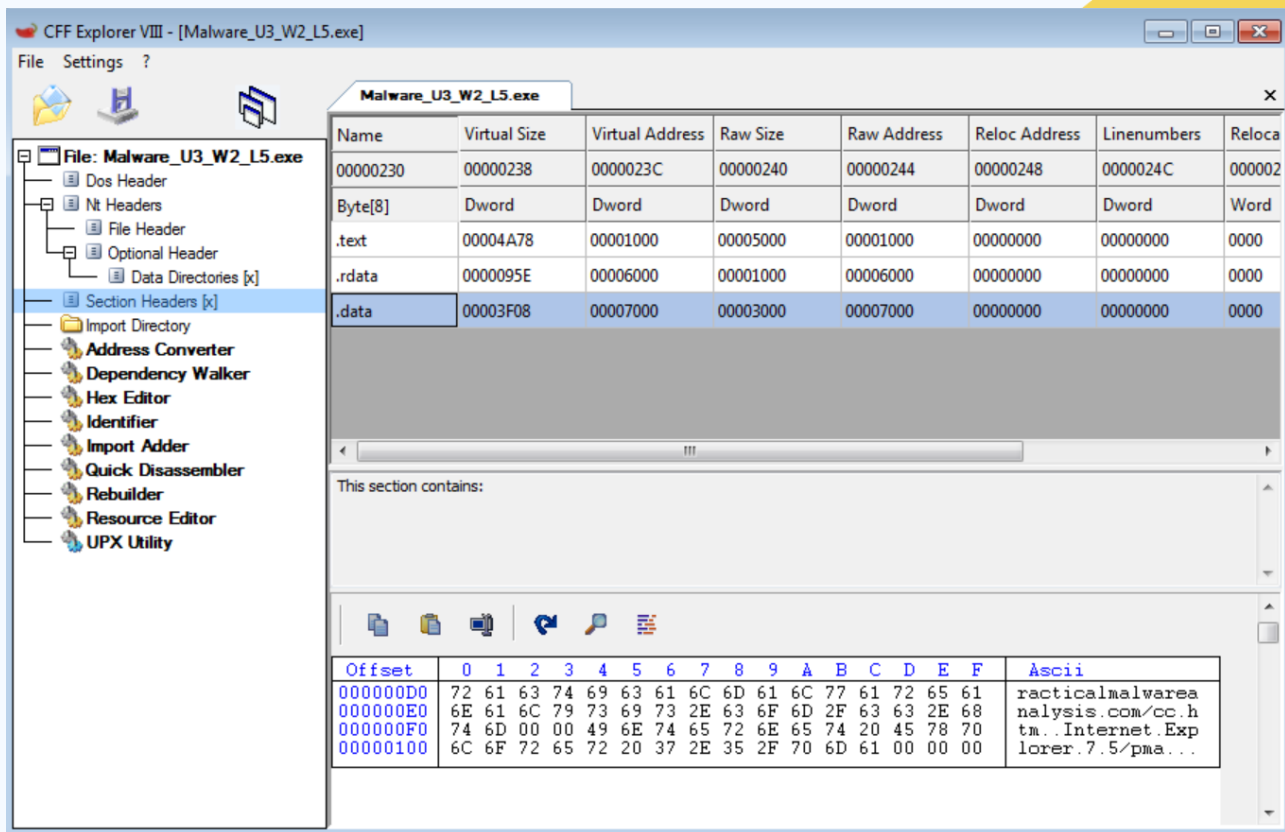
InternetCloseHandle: Utilizzata per chiudere un handle precedentemente aperto, indicando una corretta gestione delle risorse di sistema.

InternetReadFile: Utilizzata per leggere i dati da una risorsa Internet, suggerendo che il malware potrebbe acquisire informazioni da risorse remote.

InternetGetConnectedState: Utilizzata per determinare lo stato della connessione Internet, indicando che il malware potrebbe verificare la disponibilità di connessione a Internet prima di eseguire determinate azioni.

InternetOpenA: Utilizzata per inizializzare una sessione Internet, suggerendo che il malware potrebbe avviare una connessione Internet prima di eseguire altre operazioni di rete.

2. Sezioni del File Eseguitabile:



Il file eseguibile del malware è composto dalle seguenti sezioni:

.text: Contiene il codice eseguibile del programma, che definisce il comportamento del malware.

.rdata: Contiene dati in sola lettura utilizzati dal programma durante l'esecuzione.

.data: Contiene dati modificabili durante l'esecuzione del programma, come variabili globali e strutture dati.

Il malware potrebbe essere classificato come un trojan. Un Trojan è un tipo di malware progettato per sembrare legittimo ma che in realtà esegue funzionalità dannose sul computer infetto. Ecco perché il malware analizzato potrebbe essere considerato un trojan:

Funzionalità Distruttive o Dannose: Il malware sembra essere progettato per eseguire azioni dannose, come il download di ulteriori componenti dannosi da Internet o l'invio di dati sensibili a server remoti.

Comunicazione su Internet: Il malware utilizza funzioni per comunicare su Internet, come l'apertura di URL e la lettura di file da risorse remote, suggerendo un possibile ruolo di controllo remoto o di trasferimento di dati sensibili a server controllati dagli attaccanti.

Potenziale di Controllo Remoto: La capacità del malware di comunicare su Internet e di eseguire azioni dannose potrebbe consentire a un attaccante di controllare il sistema infetto da remoto, sfruttando il malware come un mezzo per eseguire operazioni non autorizzate.

3. Identificazione dei Costrutti Noti:

- Creazione dello Stack: Utilizzo delle istruzioni "push ebp" e "mov ebp, esp" per creare un nuovo frame di stack e salvare il puntatore dello stack.
- Utilizzo di Condizioni e Salti: Le istruzioni "cmp" e "jz" sono utilizzate per condizionare l'esecuzione del codice in base al risultato della comparazione. Ciò indica la presenza di una struttura di controllo condizionale, potenzialmente un if-else.
- Chiamate a Funzioni Esterne: Le istruzioni "call" vengono utilizzate per chiamare funzioni esterne al modulo, come "ds:InternetGetConnectedState" e "sub_40117F".
- Ritorno dalla Sottofunzione: La direttiva **ret** **sub_401000** indica il ritorno dalla sottofunzione corrente alla funzione identificata con l'etichetta "sub_401000".

4. Ipotesizzare il Comportamento della Funzionalità Implementata:

- Il codice sembra verificare lo stato della connessione Internet utilizzando la funzione "InternetGetConnectedState".
- Se lo stato della connessione è attivo, viene eseguito un blocco di codice per indicare il successo.
- Se non c'è connessione Internet, viene eseguito un altro blocco di codice per indicare l'errore.
- Infine, la sottofunzione termina restituendo il controllo alla funzione chiamante identificata con l'etichetta "sub_401000".

5. Tabella con Significato delle Singole Righe di Codice Assembly:

Push ebp

Mov ebp, esp

Push ecx

Push 0 ;dwReserved

Push 0 ;lpdwFlags

Call ds:InternetGetConnectedState

Mov [ebp+var_4], eax

Cmp [ebp+var_4], 0

Jz short loc_40102B

Push offset aSuccessInterne ; "Success: Internet Connection\n"

Call sub_40117F

Add esp, 4

Mov eax, 1

```

    Jmp short loc_40103A
loc_40102B:    ; "Error 1.1 : No Internet\n"
    Push offset aError1_1NoInte
    Call Sub_40117F
    Add esp, 4
    Xo reax, eax
loc_40103A:
    Mov esp, ebp
    Pop ebp
    Retn sub_401000 endp

```

Righe di Codice	Significato
push ebp	Salva il valore del puntatore dello stack precedente
mov ebp, esp	Crea un nuovo frame dello stack
push ecx	Salva il registro ECX sullo stack
push 0	Mette il valore 0 sullo stack
push 0	Mette il valore 0 sullo stack
call ds:InternetGetConnectedState	Chiama la funzione InternetGetConnectedState
mov [ebp+var_4], eax	Salva il risultato della funzione in [ebp+var_4]
cmp [ebp+var_4], 0	Confronta il risultato con 0
jz short loc_40102B	Salta a "loc_40102B" se il risultato è zero
push offset aSuccessInterne	Mette l'indirizzo della stringa sullo stack
call sub_40117F	Chiama la subroutine sub_40117F
add esp, 4	Libera spazio dallo stack
mov eax, 1	Mette il valore 1 nel registro EAX
jmp short loc_40103A	Salta a "loc_40103A"
loc_40102B:	Etichetta per gestire il caso di mancanza di connessione Internet
push offset aError1_1NoInte	Mette l'indirizzo della stringa sullo stack
call Sub_40117F	Chiama la subroutine Sub_40117F

Righe di Codice	Significato
add esp, 4	Libera spazio dallo stack
xor eax, eax	Mette il valore 0 nel registro EAX
loc_40103A:	Etichetta per gestire il successo
mov esp, ebp	Ripristina il puntatore dello stack
pop ebp	Ripristina il valore del puntatore dello stack precedente
retn sub_401000	Ritorna alla funzione chiamante identificata con l'etichetta "sub_401000"