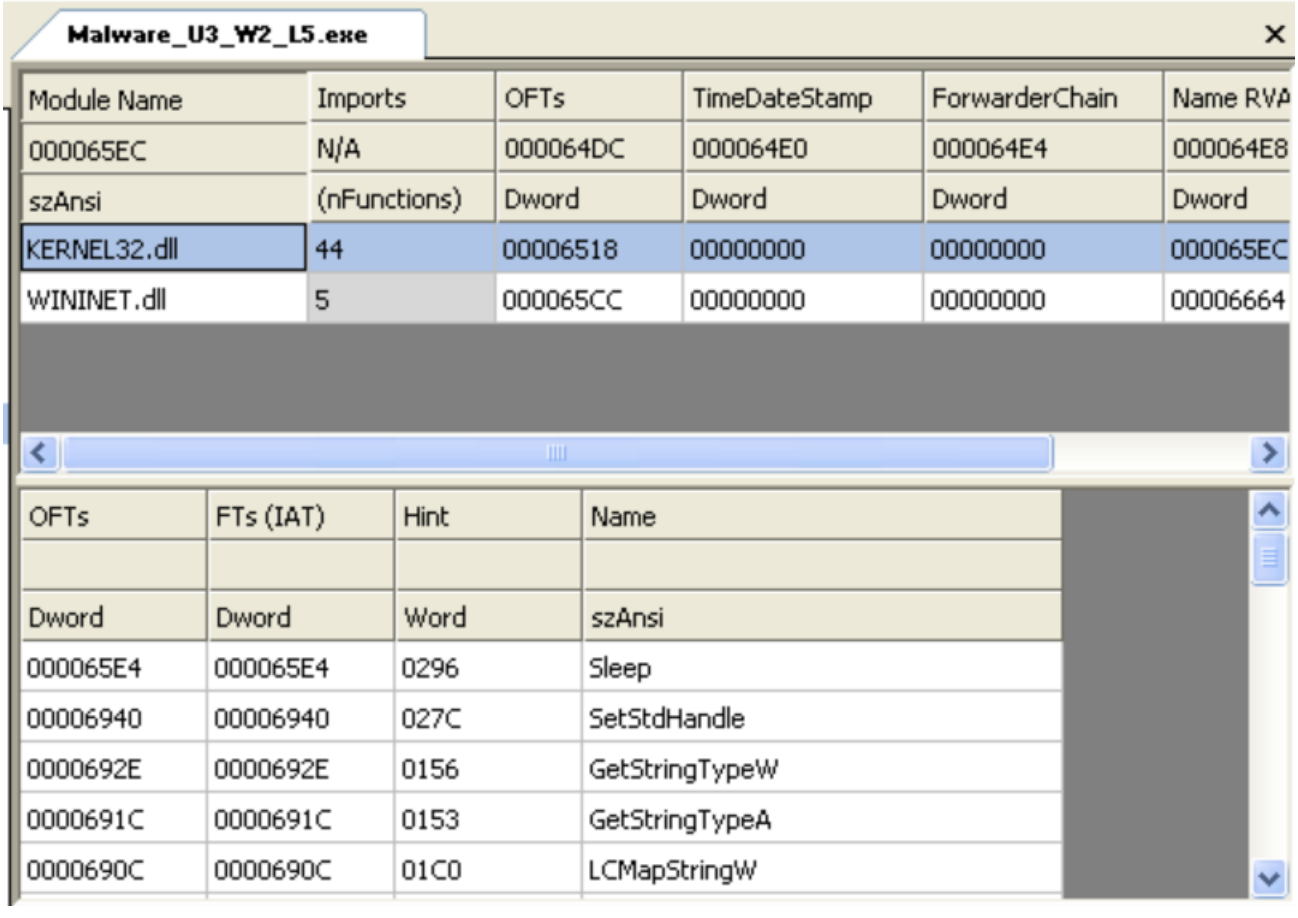
REPORT 12/01/2023

1)

KERNEL32.dll, WININET.dll



Malware_U3_W2_L5.exe

Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA
000065EC	N/A	000064DC	000064E0	000064E4	000064E8
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword
KERNEL32.dll	44	00006518	00000000	00000000	000065EC
WININET.dll	5	000065CC	00000000	00000000	00006664

OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
000065E4	000065E4	0296	Sleep
00006940	00006940	027C	SetStdHandle
0000692E	0000692E	0156	GetStringTypeW
0000691C	0000691C	0153	GetStringTypeA
0000690C	0000690C	01C0	LCMapStringW

KERNEL32.dll: è un file di libreria a collegamento dinamico (DLL) dei sistemi operativi Microsoft Windows. Il nome "kernel32" deriva da "kernel", che è la parte centrale di un sistema operativo. Questa DLL contiene un gran numero di funzioni essenziali per il corretto funzionamento del sistema operativo Windows.

Alcune delle principali funzionalità fornite da kernel32.dll:

Gestione della memoria: kernel32.dll include funzioni per la gestione della memoria, come l'allocazione e la liberazione di blocchi di memoria, la copia della memoria e la gestione della memoria virtuale.

Gestione dei processi e dei thread: Fornisce funzioni per la creazione, la gestione e la terminazione di processi e thread. Include funzioni per la sincronizzazione, l'attesa di eventi e la gestione delle priorità dei thread.

Operazioni di I/O dei file: kernel32.dll offre funzioni per le operazioni di I/O dei file e dei dispositivi, tra cui la creazione, la lettura, la scrittura e la manipolazione degli attributi dei file.

Gestione degli errori: Include funzioni per la gestione degli errori e delle eccezioni, consentendo alle applicazioni di recuperare informazioni sugli errori, impostare modalità di errore e gestire le eccezioni.

Caricamento delle Dynamic Link Library (DLL): kernel32.dll è responsabile del caricamento e dello scaricamento dinamico delle DLL, consentendo alle applicazioni di utilizzare librerie esterne.

Informazioni e configurazione del sistema: Fornisce funzioni per recuperare informazioni sul sistema, come la versione del sistema, il tipo di processore e le impostazioni di configurazione.

Funzioni di data e ora: kernel32.dll include funzioni per lavorare con l'ora e la data, tra cui l'ottenimento dell'ora corrente del sistema, la conversione dei formati dell'ora e l'impostazione dell'ora del sistema.

Sicurezza e autenticazione: Offre funzioni relative alla sicurezza e all'autenticazione, tra cui funzioni di crittografia e decrittografia.

Multithreading e sincronizzazione: kernel32.dll fornisce funzioni per la gestione di oggetti di sincronizzazione come mutex, semafori e sezioni critiche, consentendo agli sviluppatori di creare applicazioni multithread.

kernel32.dll è un componente fondamentale del sistema operativo Windows e molte altre DLL e applicazioni dipendono dalle sue funzioni. Svolge un ruolo cruciale nel fornire un ambiente stabile e coerente per l'esecuzione delle applicazioni Windows.

WININET.dll: è un file di libreria a collegamento dinamico (DLL) nei sistemi operativi Microsoft Windows. È l'acronimo di Windows Internet Services e fornisce funzioni relative ai protocolli e alle comunicazioni Internet. Questa DLL è utilizzata principalmente dalle applicazioni per gestire varie operazioni di rete, come la connessione a Internet, il download di file e la gestione delle connessioni di rete.

Alcune delle principali funzionalità fornite da WININET.dll includono:

Operazioni HTTP e HTTPS: Supporta l'implementazione dei protocolli HTTP e HTTPS, consentendo alle applicazioni di inviare e ricevere dati sul web.

Operazioni FTP: WININET.dll facilita le operazioni FTP (File Transfer Protocol), consentendo alle applicazioni di caricare e scaricare file da e verso i server FTP.

Gestione degli URL: Aiuta ad analizzare e gestire gli URL (Uniform Resource Locator), consentendo alle applicazioni di lavorare con diversi tipi di indirizzi web.

Gestione della cache: La libreria gestisce la cache dei contenuti web per migliorare le prestazioni e ridurre la necessità di scaricare ripetutamente gli stessi dati.

Gestione dei cookie: WININET.dll gestisce i cookie, utilizzati per memorizzare informazioni sul lato client, spesso per mantenere le sessioni e le preferenze degli utenti.

Protocolli di sicurezza: Supporta vari protocolli di sicurezza, come SSL/TLS, per garantire una comunicazione sicura su Internet.

2)

Le sezioni di cui si compone il file eseguibile del malware sono qui sotto elencate:

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumb
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword
.text	00004A78	00001000	00005000	00001000	00000000	00000000
.rdata	0000095E	00006000	00001000	00006000	00000000	00000000
.data	00003F08	00007000	00003000	00007000	00000000	00000000

.text: contiene le istruzioni (le righe di codice) che la CPU eseguirà una volta che il software sarà avviato. Generalmente questa è l'unica sezione di un file eseguibile che viene eseguita dalla CPU, in quanto tutte le altre sezioni contengono dati o informazioni a supporto.

.rdata: include generalmente le informazioni circa le librerie e le funzioni importate ed esportate dall'eseguibile, informazione che come abbiamo visto possiamo ricavare con CFF Explorer.

.data: contiene tipicamente i dati / le variabili globali del programma eseguibile, che devono essere disponibili da qualsiasi parte del programma. Una variabile si dice globale quando non è definita all'interno di un contesto di una funzione, ma bensì è globalmente dichiarata ed è di conseguenza accessibile da qualsiasi funzione all'interno dell'eseguibile.

3)

Il codice assembly fornito è per una funzione che verifica la presenza di una connessione a Internet utilizzando la funzione InternetGetConnectedState. Ecco una ripartizione dei costrutti chiave:

Prologo:

```
.text:00401000    push  ebp
.text:00401001    mov   ebp, esp
```

Questo è il prologo della funzione standard. Salva il puntatore di base corrente (ebp) e imposta il puntatore di base sul puntatore di pila corrente (esp).

Impostazione della pila:

```
.text:00401003    push ecx
.text:00401004    push 0        ; dwReserved
.text:00401006    push 0        ; lpdwFlags
```

Spinge i valori sullo stack, eventualmente come parametri per la successiva chiamata di funzione (InternetGetConnectedState).

Chiamata di funzione:

```
.text:00401008    call  ds:InternetGetConnectedState
```

Richiama la funzione InternetGetConnectedState, che verifica la presenza di una connessione a Internet.

Gestione del valore di ritorno:

```
.text:0040100E    mov  [ebp+var_4], eax
.text:00401011    cmp [ebp+var_4], 0
.text:00401015    jz short loc_40102B
```

Memorizza il valore di ritorno in [ebp+var_4], lo confronta con zero e salta a loc_40102B se il risultato è zero (indicando l'assenza di connessione a Internet).

Caso di successo:

```
push offset aSuccessInterne ; "Success: Internet Connection\n"
call sub_40117F
add esp, 4
mov eax, 1
jmp short loc_40103A
```

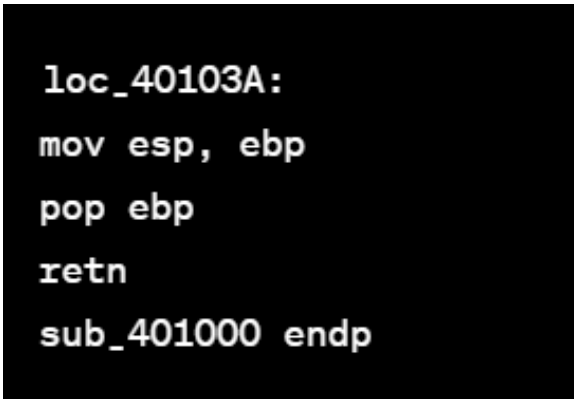
Se il controllo della connessione a Internet ha successo, stampa un messaggio di successo, imposta eax a 1 e salta a loc_40103A.

Caso di errore:

```
loc_40102b:                ; "Error  1.1: No Internet\n"
push offset aError1_NoInte
call sub_40117F
add esp, 4
xor eax, eax
```

Se la verifica della connessione a Internet fallisce, viene stampato un messaggio di errore, si cancella eax a 0 e si continua con loc_40103A.

Epilogo:



Ripristina lo stack e il puntatore alla base ai loro valori originali e ritorna dalla funzione.

Si noti che alcuni dettagli dipendono dal contesto, come le definizioni delle stringhe e delle funzioni come sub_40117F. Se si dispone di tali dettagli, la comprensione sarà più completa.

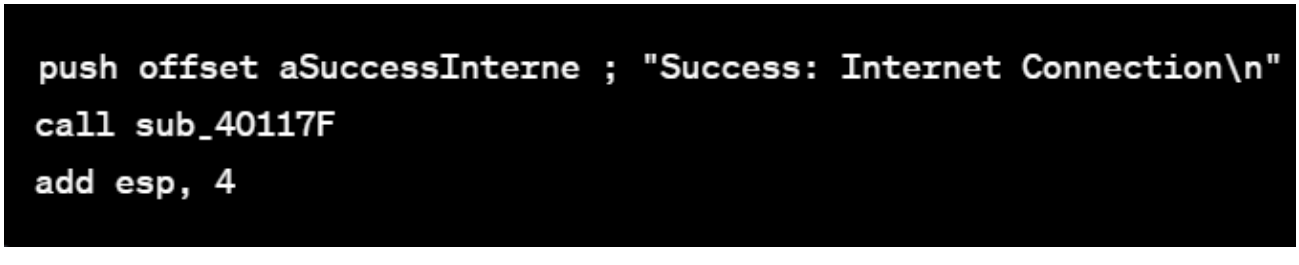
4)

InternetGetConnectedState:

È una funzione API di Windows utilizzata per verificare se è disponibile una connessione a Internet.

sub_40117F:

4a)



4b)

```
loc_40102b:                ; "Error  1.1: No Internet\n"
push offset aError1_NoInte
call sub_40117F
add esp, 4
```

Il codice suggerisce l'esistenza di una subroutine (funzione) all'indirizzo sub_40117F responsabile della stampa dei messaggi di successo e di errore. L'esatta implementazione di questa funzione non è fornita nel frammento di codice fornito.

È importante notare che il codice fa riferimento a funzioni come InternetGetConnectedState e sub_40117F, ma i dettagli dell'implementazione effettiva di queste funzioni non sono mostrati nello snippet fornito. Per comprendere appieno il codice e il suo comportamento, è necessario avere accesso alle definizioni o alle implementazioni di queste funzioni.

Simone F Caracci