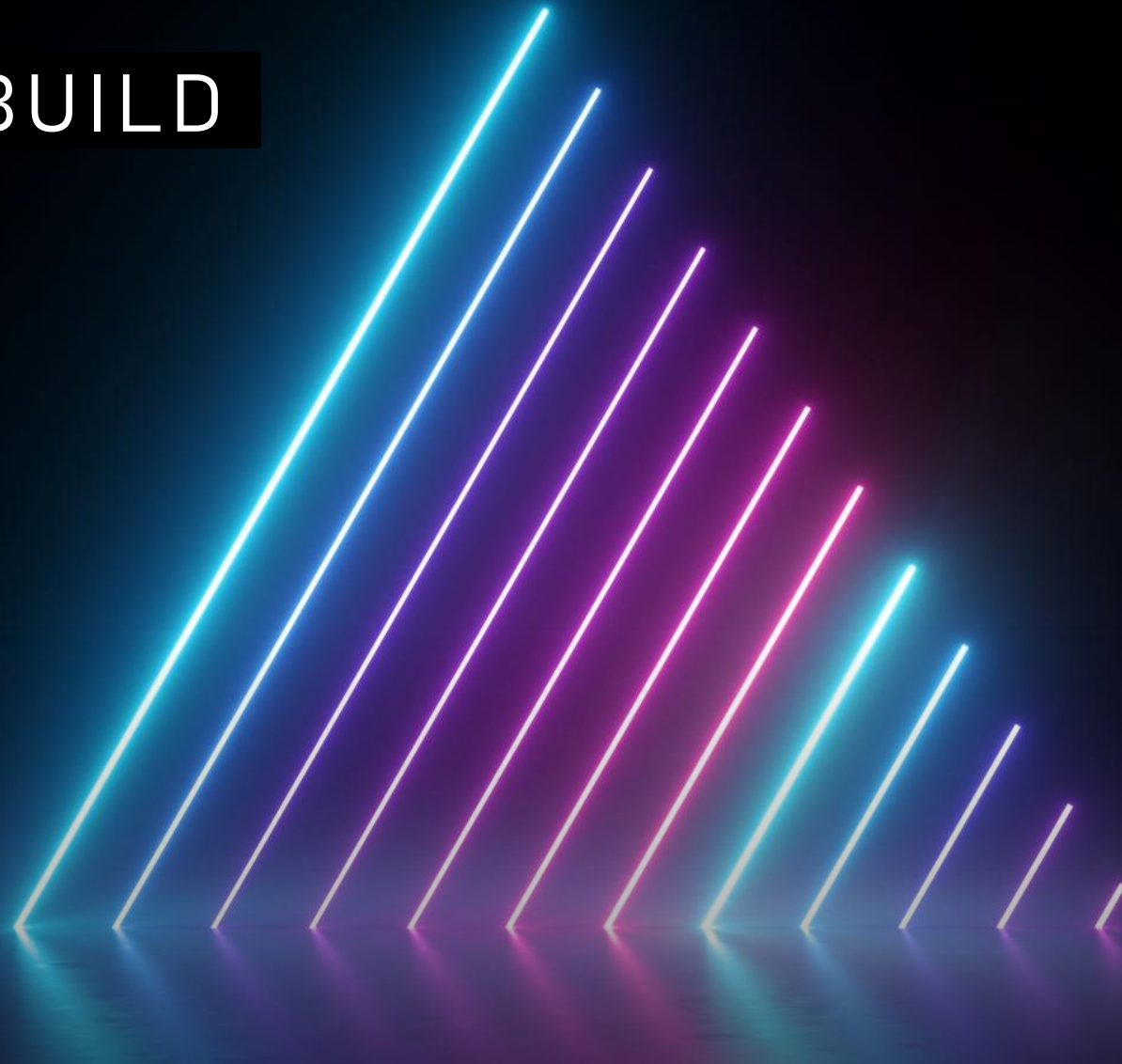
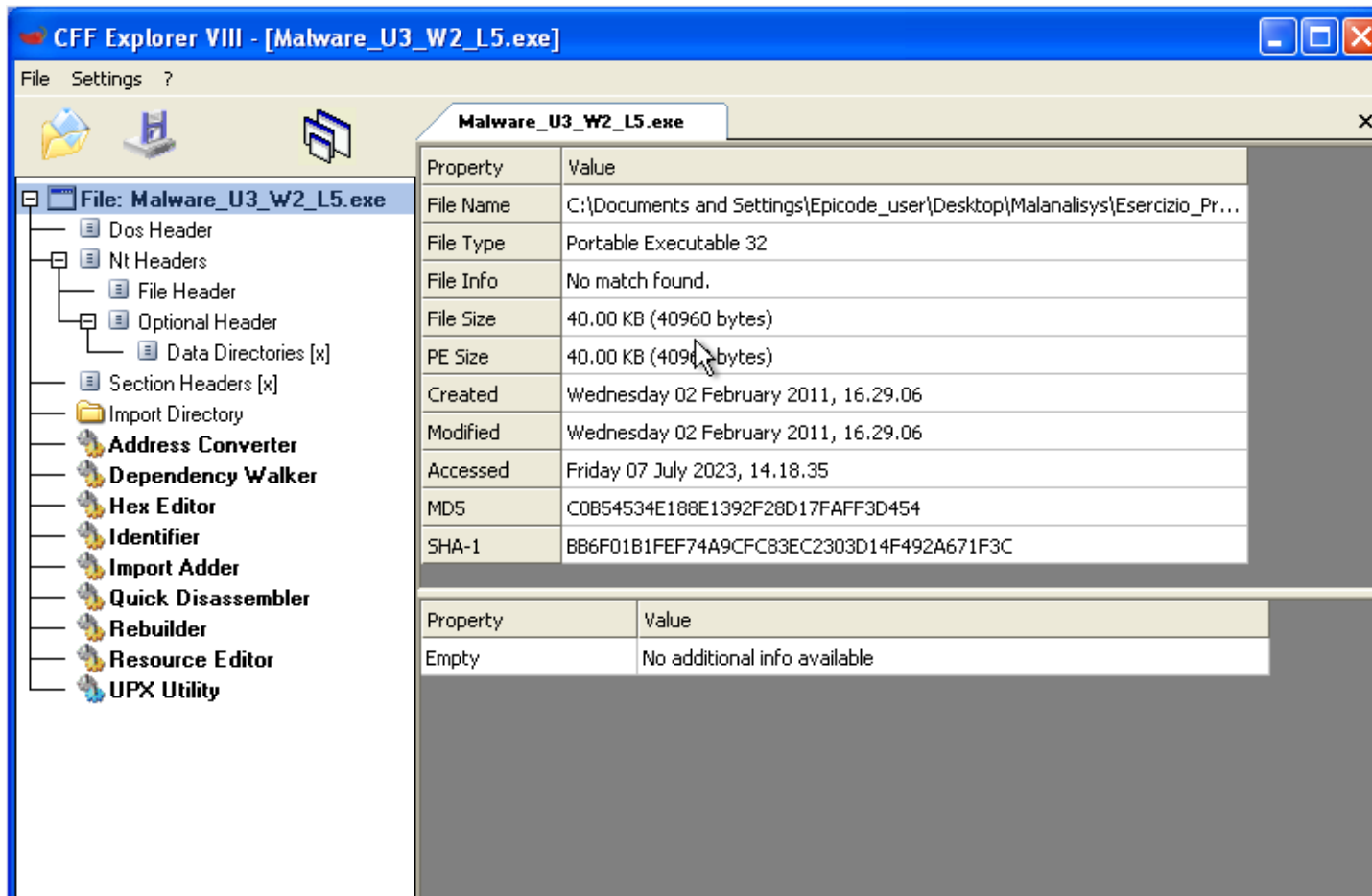


PROGETTO BUILD

WEEK 07/07



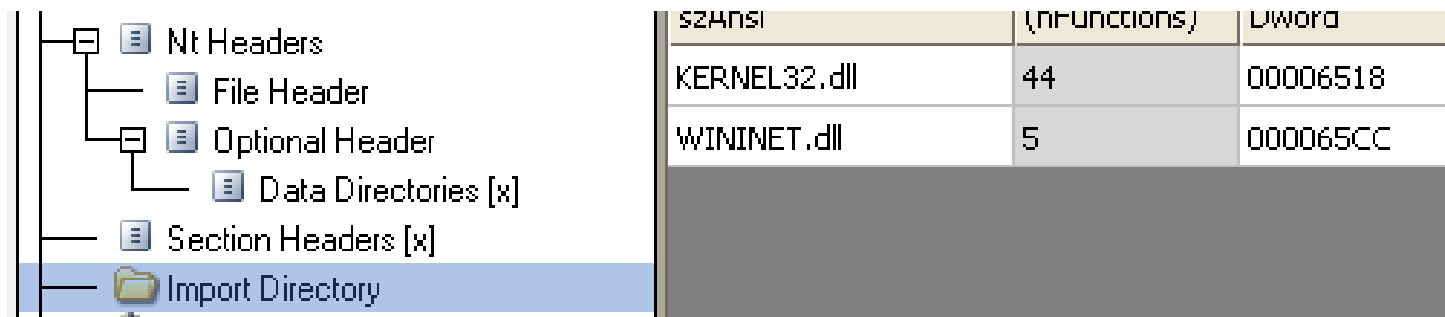
Lo scopo dell'esercizio è quello di analizzare e descrivere diverse situazioni con l'ausilio dei tool conosciuti.



PARTE 1

Esaminare e descrivere le librerie importate dal malware, per prima cosa apriamo il nostro file .exe con cff explorer;

PUNTO1



The screenshot shows the PE Explorer interface. On the left, a tree view displays the file structure: Nt Headers, File Header, Optional Header, Data Directories [x], Section Headers [x], and Import Directory (selected). On the right, a table lists the imported DLLs.

szANSI	(nFunctions)	DWORD
KERNEL32.dll	44	00006518
WININET.dll	5	000065CC

La libreria "**kernel32.dll**" è una componente essenziale del sistema operativo Windows. Contiene una vasta gamma di funzioni che sono utilizzate dai programmi per interagire con il sistema operativo.

Di conseguenza è una libreria che consente al sistema operativo di fare certe operazioni quasi indispensabili, ma di conseguenza un malware potrebbe sfruttare la libreria per eseguire una serie di azioni dannose sul sistema operativo Windows. Ecco alcuni esempi di cosa potrebbe fare un malware utilizzando questa libreria:

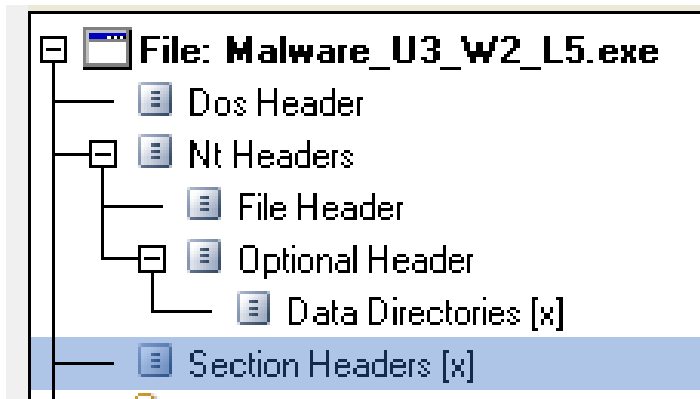
1. **Infiltrazione**: il malware potrebbe utilizzare funzioni come "CreateProcess" o "LoadLibrary" per eseguire se stesso o altri componenti dannosi nel sistema.
2. **Modifica dei processi**: il malware potrebbe utilizzare funzioni come "OpenProcess" o "TerminateProcess" per manipolare i processi in esecuzione, terminarli o iniettare codice dannoso all'interno di essi.
3. **Interazione con il file system**: il malware potrebbe utilizzare funzioni come "CreateFile", "ReadFile" o "WriteFile" per accedere, leggere o modificare file sensibili o critici del sistema.
4. **Allocazione di memoria**: il malware potrebbe utilizzare funzioni come "VirtualAlloc" per allocare spazio di memoria nel sistema e utilizzarlo per scopi dannosi come l'iniezione di codice maligno o l'esecuzione di payload dannosi.
5. **Evasione delle misure di sicurezza**: la libreria offre funzioni per manipolare le eccezioni e gestire gli errori. Un malware potrebbe sfruttare queste funzioni per eludere le misure di sicurezza o mascherare la sua presenza.
6. **Manipolazione del registro di sistema**: il malware potrebbe utilizzare funzioni come "RegOpenKey" o "RegSetValue" per modificare le voci del registro di sistema, compromettendo così le impostazioni del sistema o l'integrità delle applicazioni.
7. **Rilevamento e bypass delle analisi antivirus**: il malware potrebbe utilizzare funzioni come "GetSystemInfo" o "GetVersionEx" per ottenere informazioni sul sistema operativo e regolare il suo comportamento per evitare la rilevazione da parte degli strumenti antivirus.

La libreria "**wininet.dll**" è una componente di Windows che fornisce funzionalità per l'interazione con Internet. È utilizzata principalmente per l'accesso e la gestione delle risorse di rete, come il download di file, la comunicazione con server Web e l'invio di richieste HTTP.

Però dato che viene importata da un malware ecco in quale modo potrebbe essere utilizzata negativamente:

1. **Comunicazione con server di comando e controllo**: Il malware può utilizzare la libreria per connettersi a un server remoto e ricevere istruzioni dannose.
2. **Download di file dannosi**: Il malware può utilizzare la libreria per scaricare e distribuire file dannosi sul sistema infetto.
3. **Iniezione di codice maligno**: Il malware può usare la libreria per iniettare codice maligno in processi legittimi.
4. **Furto di informazioni sensibili**: Utilizzando questa libreria, il malware può inviare informazioni personali a un server remoto controllato dagli attaccanti.
5. **Spoofing o attacchi di phishing**: Il malware può creare pagine o e-mail fasulle per ingannare gli utenti e ottenere informazioni personali.
6. **Attacchi DoS**: Il malware può utilizzare la libreria per sovraccaricare un server con un'elevata quantità di richieste, rendendolo inaccessibile.

PUNTO 2



Byte[8]	Dword	Dword	Dword	Dword	Dword
.text	00004A78	00001000	00005000	00001000	00000000
.rdata	0000095E	00006000	00001000	00006000	00000000
.data	00003F08	00007000	00003000	00007000	00000000

La sezione **.text** contiene le istruzioni che la CPU eseguirà una volta che il software sarà avviato. Generalmente questa è l'unica sezione di un file exe che viene eseguita dalla CPU, in quanto tutte le rimanenti sezioni contengono dati o informazioni di supporto.

La sezione **.rdata** include le informazioni su librerie e funzioni importate ed esportate dal file, che appunto vediamo su cff explorer.

La sezione **.data** contiene solitamente i dati/variabili globali del programma eseguibile che devono essere disponibili ovunque nel programma. Perciò queste variabili non sono dichiarate all'interno di funzioni ma sono accessibili da qualsiasi zona dell'eseguibile.

PUNTO 3:

Apertura dello stack

Realizzazione di un
costrutto con if

```
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:
push    offset aError1_1NoInte ; "Error 1.1: No Internet\n"
call    sub_40117F
add     esp, 4
xor     eax, eax
```

```
loc_40103A:
mov     esp, ebp
pop     ebp
ret     0
sub_401000 endp
```

Qui è stata fatta la chiusura
dello stack

Entrambi i codici
richiamano delle
funzioni

PUNTO 4:

Per prima cosa nel codice sono presenti 2 puntatori uno sulla cima dello stack e uno in fondo, secondariamente vediamo che il codice verifica che ci sia connettività verso internet con il comando **Internetgetconnectedstate**, poi probabilmente con il costrutto if verifica la risposta del comando per capire se andare avanti con le funzioni o tornare indietro.

Essendo un file .exe che cerca una connessione ad internet potrebbe, scaricare file in background sulla macchina, connettersi verso siti infetti senza autorizzazione, o inviare file a **server di proprietà dell'hacker**;

Di conseguenza possiamo ipotizzare che il codice possa creare una **backdoor** oppure anche essere un **trojan**.

PUNTO 5:

`push ecx` → Con il push il valore all'interno del registro ecx viene posto in cima allo stack.

`call ds:internetgetconnectedstate` → Richiama una funzione che verifica la connettività internet della macchina, facendo delle ricerche ho visto che questa funzione è contenuta nella libreria wininet.

`jz short loc_40102b` → Questa istruzione fa parte del controllo if e controlla l'output del comando `cmp` precedente, nel caso fosse 1 allora viene fatto un salto (short-short jump) nella allocazione di memoria descritta, altrimenti in caso contrario se fosse 0 andrebbe avanti con il codice seguente.

`pop` → rimuove il contenuto di un registro dal nostro stack.

`retn` → Effettua il ritorno alla funzione che è stata chiamata in precedenza.

BONUS:



File: IEXPLORE.EXE

- Dos Header
- Nt Headers
 - File Header
 - Optional Header
 - Data Directories [x]
- Section Headers [x]
- Export Directory
- Import Directory
- Resource Directory
- Debug Directory
- Address Converter
- Dependency Walker
- Hex Editor
- Identifier
- Import Adder
- Quick Disassembler
- Rebuilder
- Resource Editor
- UPX Utility

File Name	C:\Programmi\Internet Explorer\IEXPLORE.EXE
File Type	Portable Executable 32
File Info	No match found.
File Size	91.00 KB (93184 bytes)
PE Size	91.00 KB (93184 bytes)
Created	Friday 15 July 2022, 15.05.47
Modified	Monday 14 April 2008, 14.00.00
Accessed	Friday 07 July 2023, 16.12.58
MD5	173E49AEBB665C0577D751BA55F84B6C
SHA-1	7EBABA5B1CDDA4906EC56EE05715B3993DA199FA

Property	Value
CompanyName	Microsoft Corporation
FileDescription	Internet Explorer
FileVersion	6.00.2900.5512 (xpsp.080413-2105)
InternalName	ieexplore
LegalCopyright	© Microsoft Corporation. Tutti i diritti riservati.
OriginalFilename	IEXPLORE.EXE
ProductName	Sistema operativo Microsoft® Windows®

Ho analizzato con hybrid analisys e virus total

Analysis Overview

[Request Report Deletion](#)

Submission name: iexplore.exe ⓘ
Size: 624KiB
Type: peexe executable ⓘ
Mime: application/x-dosexec
SHA256: b18a0d4beba606bf30f5010ba3c72abafac80d5f303a8bffb24d7f7b78b786e6 ⓘ
Last Anti-Virus Scan: 07/07/2023 12:48:58 (UTC)

[whitelisted](#)[Link](#) [Twitter](#) [E-Mail](#)

Anti-Virus Results

[Up-to-date](#)

CrowdStrike Falcon

CLEAN
Static Analysis and ML ⓘ
Last Update: 07/07/2023 12:48:58 (UTC)

MetaDefender

CLEAN
Multi Scan Analysis
Last Update: 07/07/2023 12:48:58 (UTC)

VirusTotal

CLEAN
Multi Scan Analysis
Last Update: 07/07/2023 12:48:58 (UTC)

Come prima cosa vediamo che il nostro file pericoloso è rilasciato da microsoft con anche i diritti riservati, possiamo vedere l'autore.



✓ No security vendors and no sandboxes flagged this file as malicious

b18a0d4beba606bf30f5010ba3c72abafac80d5f303a8bffb24d7f7b78b786e6

IEXPLORE.EXE

Size
623.84 KB

peexe via-tor overlay runtime-modules signed detect-debug-environment idle direct-cpu-clock-access checks-user-input