



# S10/L1

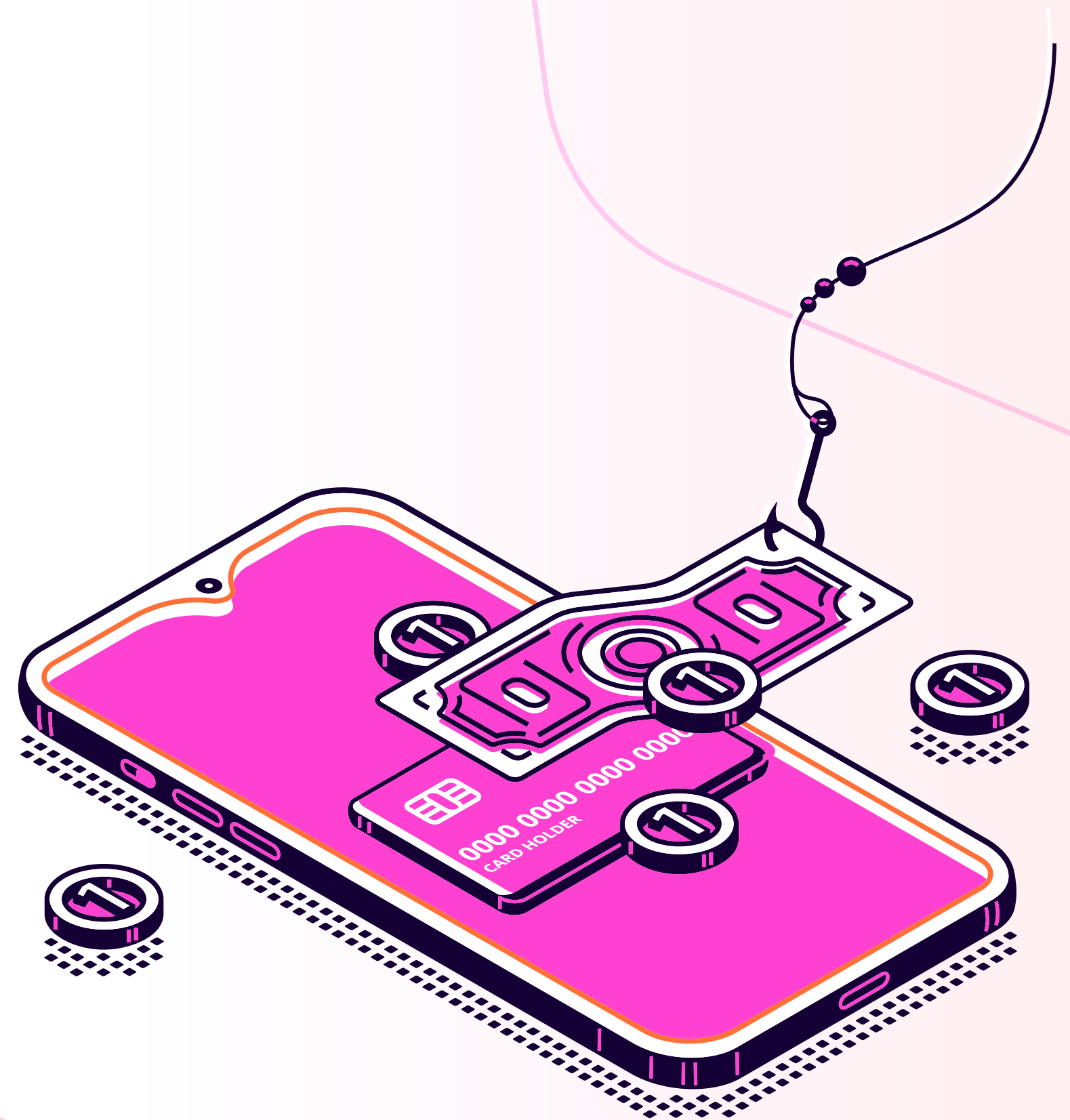
---

ANALISI MALWARE **STATICA** BASICA

# TRACCIA

Con riferimento al file eseguibile contenuto nella cartella «Esercizio\_Pratico\_U3\_W2\_L1» presente sul Desktop della vostra macchina virtuale dedicata all'analisi dei malware, rispondere ai seguenti quesiti:

- Indicare le **librerie** importate dal malware, fornendo una descrizione per ognuna di esse.
- Indicare le **sezioni** di cui si compone il malware, fornendo una descrizione per ognuna di essa.
- Aggiungere una **considerazione finale** sul malware in analisi in base alle informazioni raccolte.



# Analisi statica.

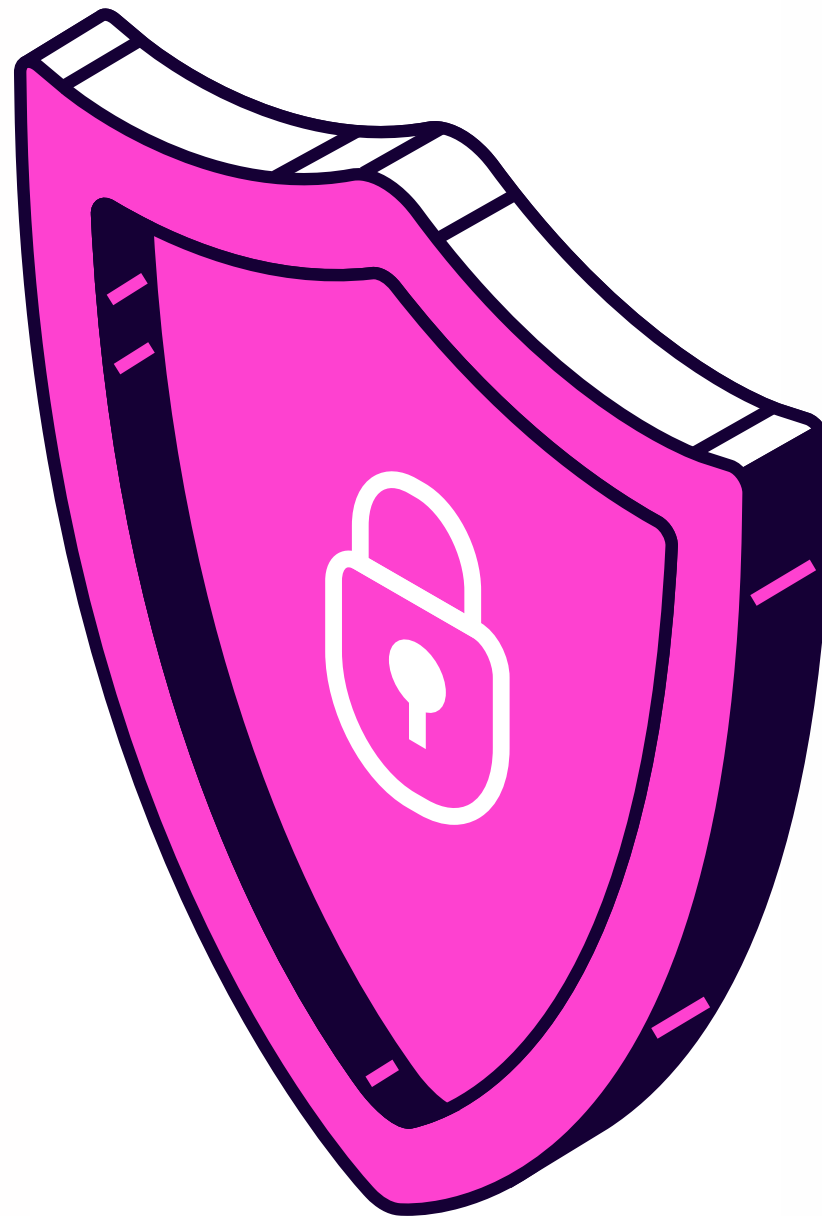
## Definizione

L'analisi statica non comporta l'esecuzione del software, e di base esamina il file eseguibile **senza visualizzare le istruzioni effettive**. Attraverso questo metodo possiamo sicuramente confermare se un file è dannoso, e fornire alcune informazioni sulla sua funzionalità. È semplice e rapida, ma inefficace contro malware sofisticati, e può perdersi comportamenti importanti. Da ciò che abbiamo appreso dalla teoria, però, risulta chiaro che per una corretta analisi di un malware bisogna effettuare più tipologie di analisi e non soffermarsi unicamente a quella statica.



# Setup Laboratorio

Per l'analisi in questione è stata predisposta una macchina virtuale appositamente progettata dal professore con i tool necessari e i file da analizzare. Io ci ho aggiunto alcuni strumenti che già conoscevo (**cmdr**) al fine di facilitarmi le operazioni.



01.

## Sistema Operativo

Il sistema operativo è **Windows 7** per lo scopo di questo corso.

02.

## Isolamento network

Ho aggiunto alla VM su proxmox una scheda di rete virtuale **non collegata a Internet** per isolare, all'occorrenza, la vm dall'esterno.

03.

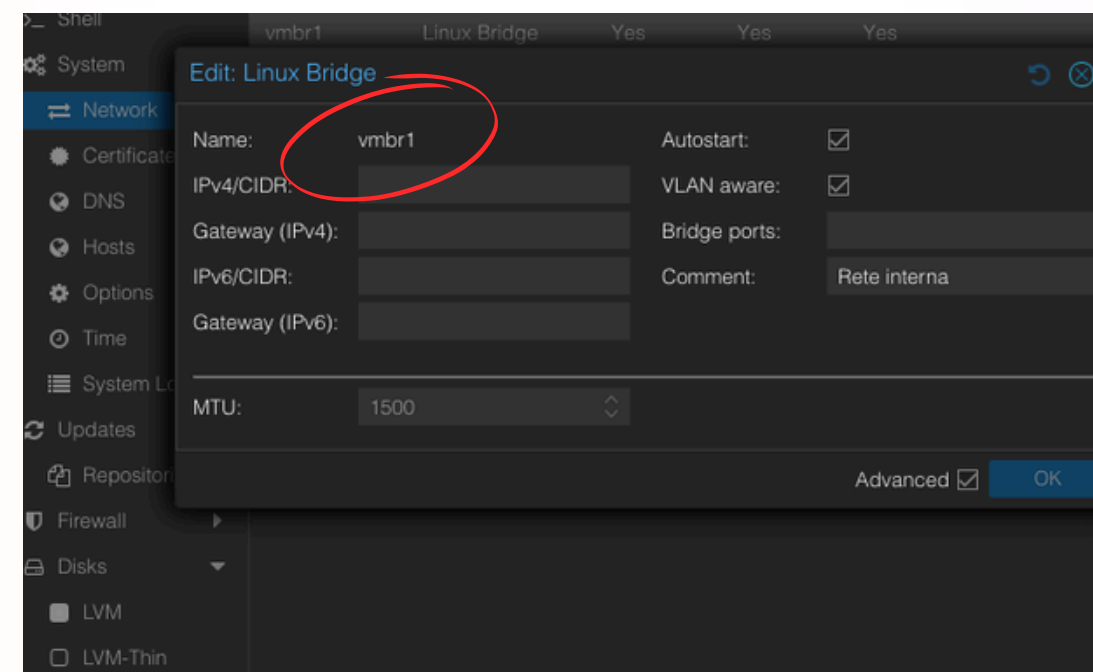
## Analisi del Malware

Il malware da analizzare oggi è:  
U3\_W2\_L1

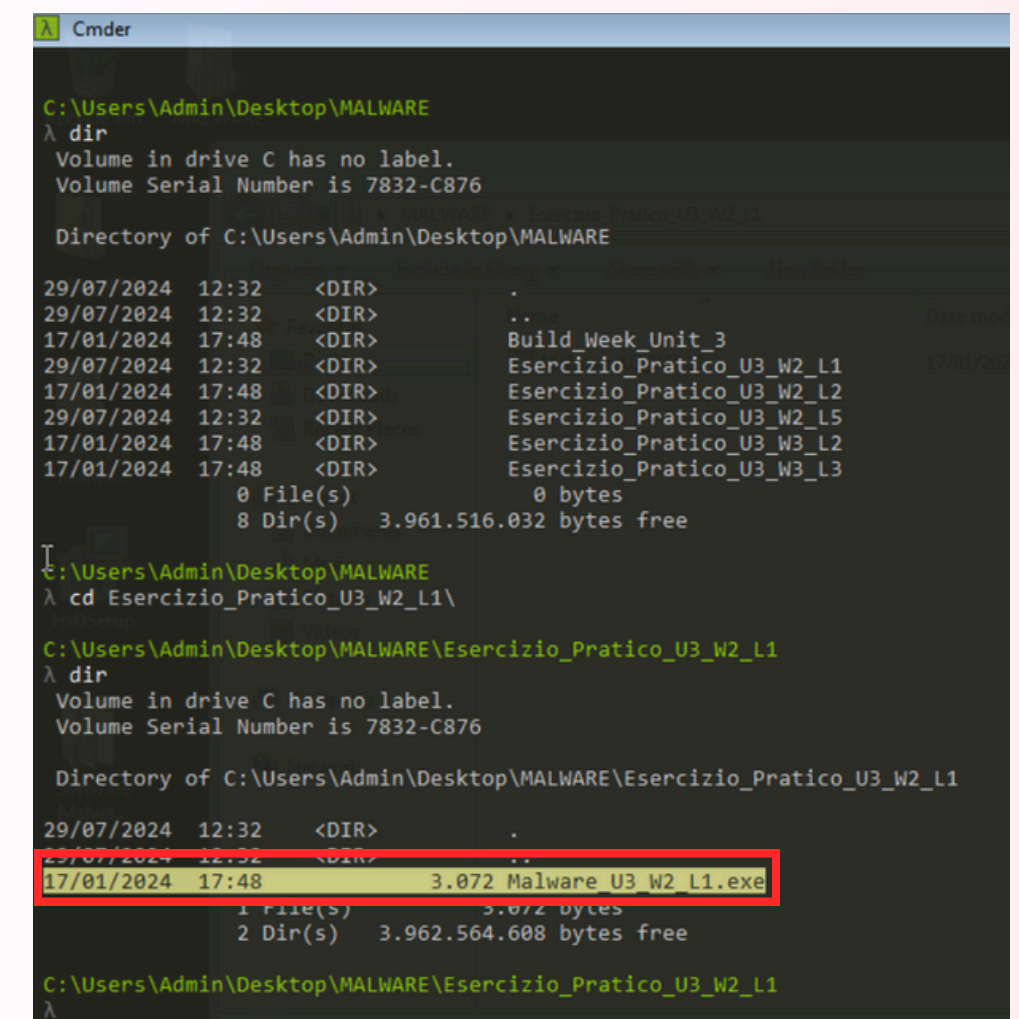
01.



02.



03.



# Strumenti utilizzati in questa sessione

Cmder non è un tool per l'analisi dei malware, ma è un **emulatore di terminale** (basato su ConEmu) che conosco bene e che ho installato per comodità, poiché offre funzionalità avanzate di gestione delle shell e supporto per comandi Unix-like anche su Windows.

md5deep

VirusTotal

CFF Explorer

HxD

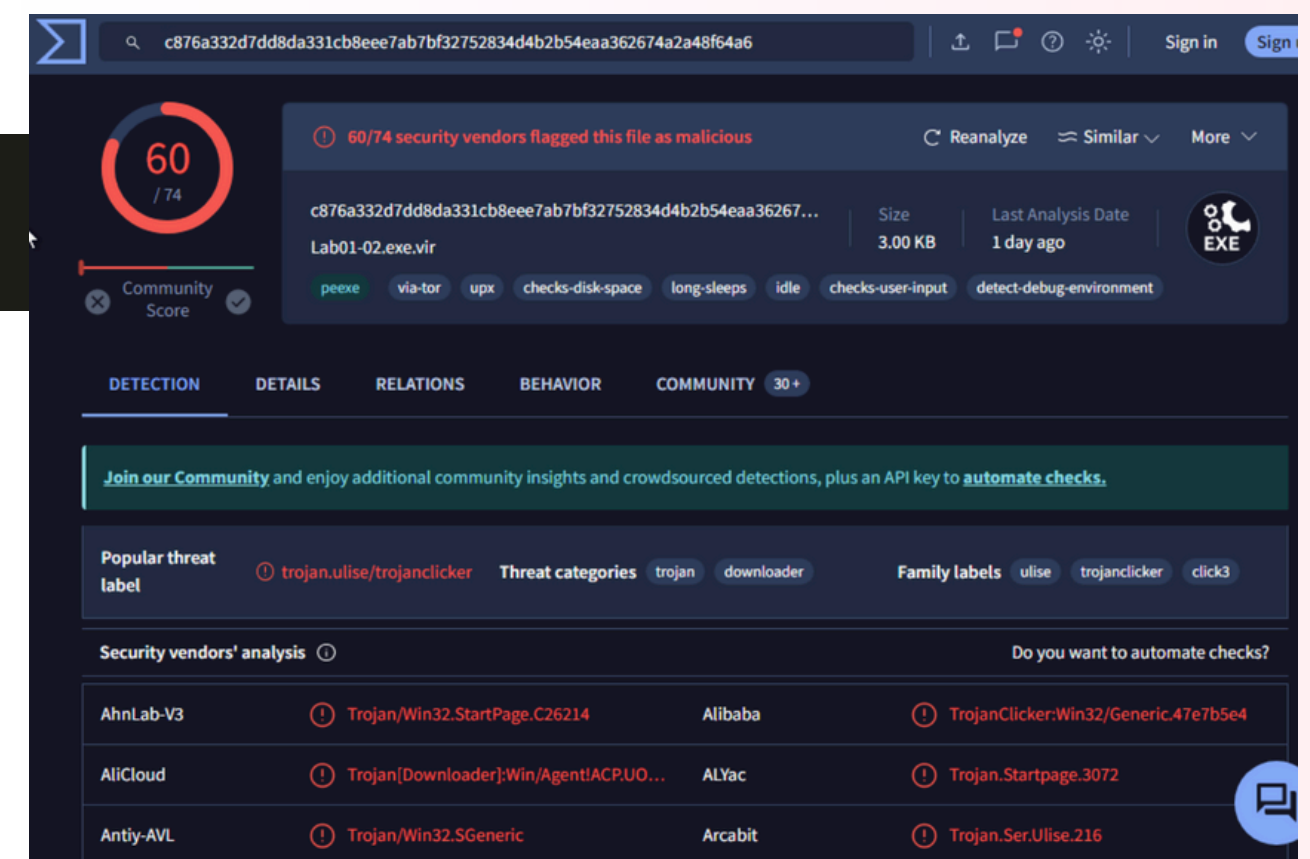
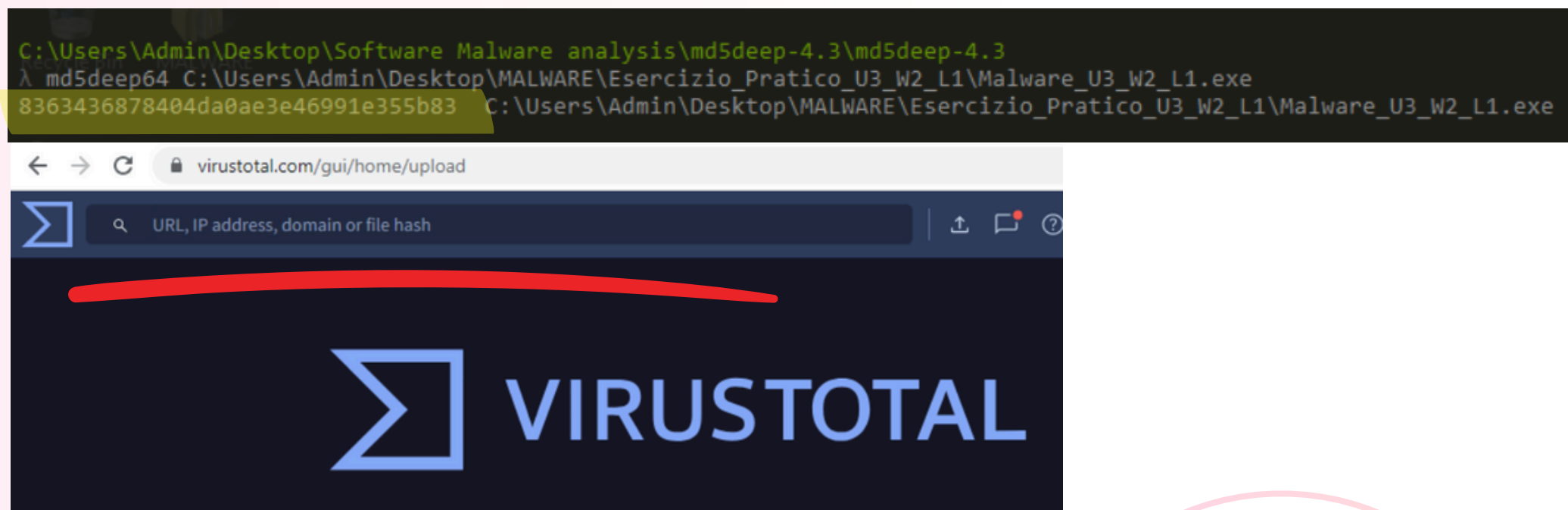
Cmder

file



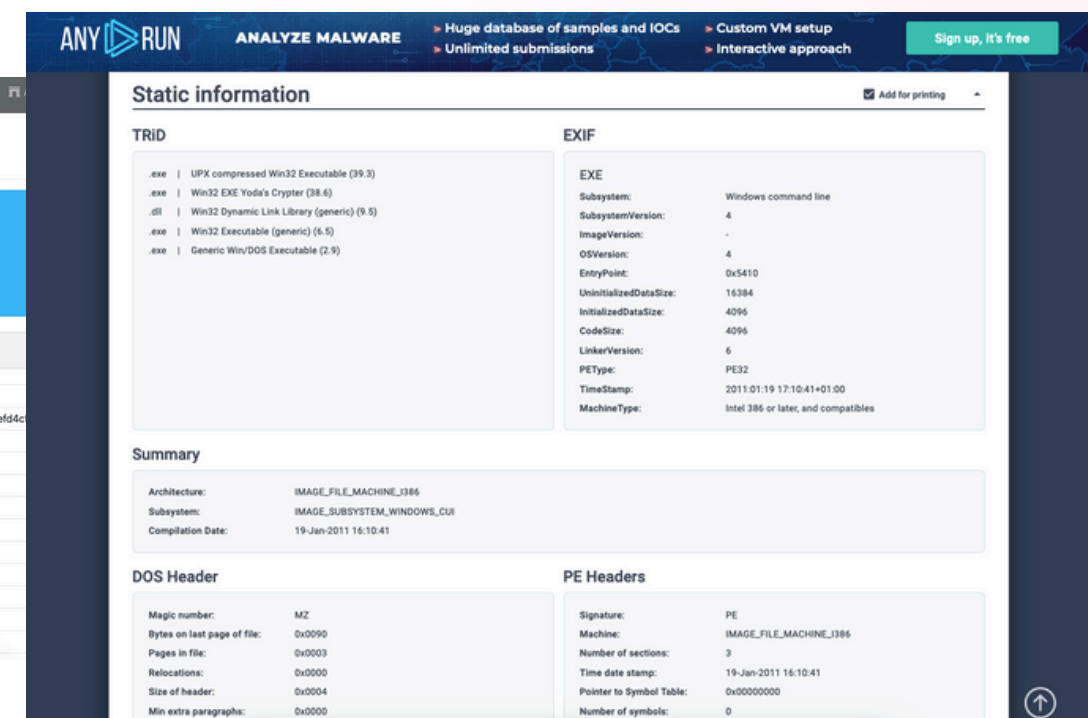
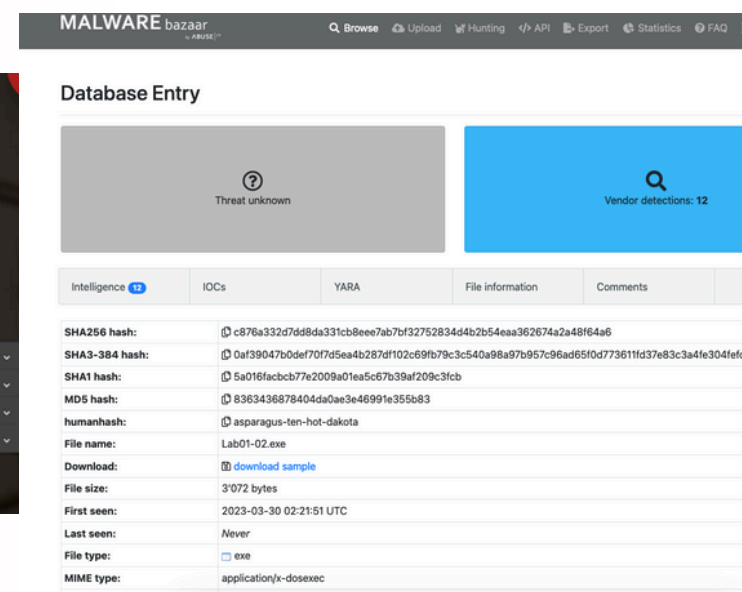
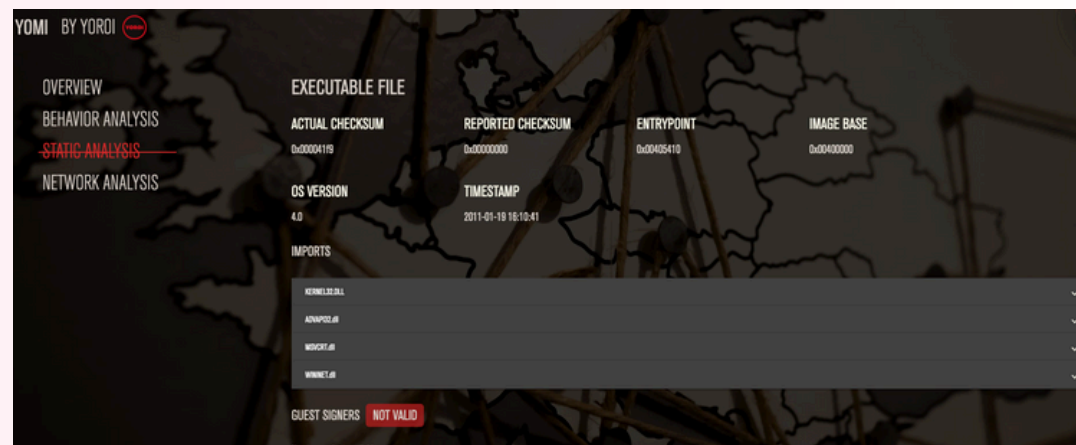
# Virus Total

Il primo passo che ho effettuato, è stato quello di **ricercare informazioni** sul malware in rete. Nello specifico, su **VirusTotal**.  
Attenzione: non ho effettuato l'upload dell'intero file, bensì, ho estrapolato prima il suo hash con il tool **md5deep** e l'ho incollato nel servizio Virus Total. L'utilizzo dell'hash MD5 è stato molto più rapido. Essendo il file in questione già stato analizzato, ho ottenuto immediatamente i risultati della scansione senza dover caricare il file.



# INFORMATION GATHERING

Facendo ricerche banali sui motori di ricerca utilizzando l'hash, si trovano altre informazioni utili provenienti da servizi che hanno analizzato il malware, oltre virus total. In questi report si vedono chiaramente quali sono le librerie importanti dal malware e le sezioni relative, ma noi **dobbiamo confermarlo sul campo**.



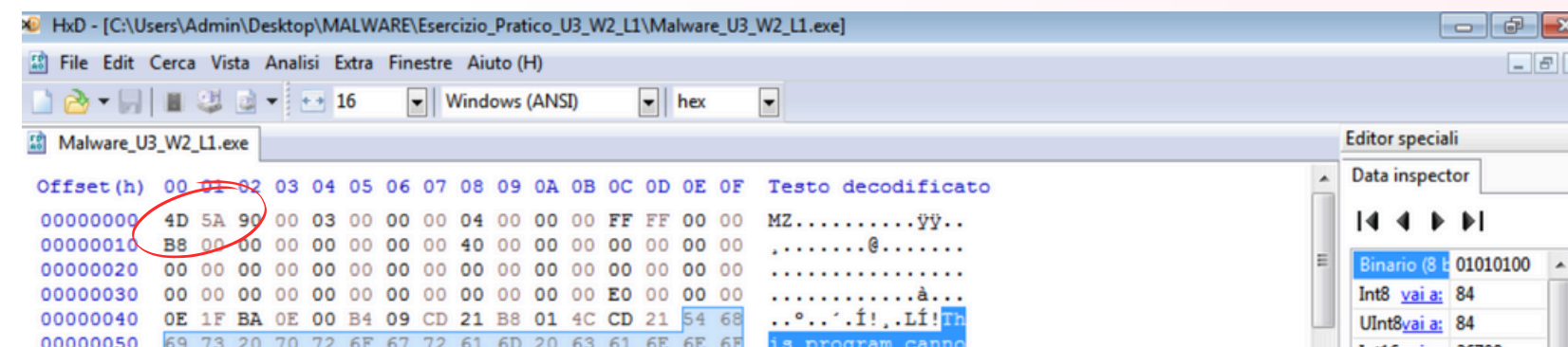


# ANALISI

Come abbiamo constatato dalla teoria, windows utilizza nella maggior parte dei casi dei file in formato PE (Portable Executable). Il formato PE al suo interno contiene delle informazioni necessarie al sistema operativo per capire come gestire il codice del file, come ad esempio le librerie, moduli o altre risorse. Questi file PE includono **sezioni** come il codice eseguibile, i dati, le risorse e le tabelle di importazione/esportazione che indicano [le librerie esterne e le funzioni utilizzate dal programma](#). Il malware è in **.exe**, **e se avesse avuto estensione jpg ad esempio?** Facciamo una verifica che sia davvero un PE usando il tool da riga di comando file, o analizzando l'intestazione dell'eseguibile:

```
C:\Users\Admin\Desktop\Software Malware analysis\md5deep-4.3\md5deep-4.3
λ file C:\Users\Admin\Desktop\MALWARE\Esercizio_Pratico_U3_W2_L1\Malware_U3_W2_L1.exe
C:\Users\Admin\Desktop\MALWARE\Esercizio_Pratico_U3_W2_L1\Malware_U3_W2_L1.exe: PE32 executable (console) Intel 80386, for MS
Windows, UPX compressed, 3 sections
```

In questo caso, leggendo la documentazione ufficiale di Microsoft e banalmente Wikipedia, ho appreso che i primi due byte (**4D 5A** in esadecimale, corrispondenti alla stringa ASCII '**MZ**') servono come 'numero magico' per identificare il file come eseguibile per ambienti **MS-DOS**. Questi due byte rappresentano le iniziali di Mark Zbikowski, uno dei principali sviluppatori di MS-DOS.



[https://en.wikipedia.org/wiki/DOS\\_MZ\\_executable](https://en.wikipedia.org/wiki/DOS_MZ_executable)

[https://en.wikipedia.org/wiki/Magic\\_number\\_\(programming\)](https://en.wikipedia.org/wiki/Magic_number_(programming))

# ANALISI

Adesso che conosciamo esattamente la tipologia, proseguiamo con l’analisi utilizzando **CFF Explorer**, per controllare le librerie importate e le sezioni. **Ecco le librerie importante dinamicamente dal malware:**

CFF Explorer VIII - [Malware\_U3\_W2\_L1.exe]

File: Malware\_U3\_W2\_L1.exe

Dos Header

NT Headers

File Header

Optional Header

Data Directories [x]

Section Headers [x]

Import Directory

Address Converter

Dependency Walker

Hex Editor

Identifier

Import Adder

Quick Disassembler

Rebuilder

Resource Editor

UPX Utility

Property

Value

File Name

C:\Users\Admin\Desktop\MALWARE\Esercizio\_Pratico\_U3\_W2

File Type

Portable Executable 32

File Info

UPX v3.0

File Size

3.00 KB (3072 bytes)

PE Size

3.00 KB (3072 bytes)

Created

Wednesday 17 January 2024, 17.48.16

Modified

Wednesday 17 January 2024, 17.48.16

Accessed

Wednesday 17 January 2024, 17.48.16

MD5

8363436878404DA0AE3E46991E355B83

SHA-1

5A016FACBCB77E2009A01EA5C67B39AF209C3FCB

Property

Value

Empty

No additional info available

Malware\_U3\_W2\_L1.exe

Member	Offset	Size	Value
e_magic	00000000	Word	5A4D
e_cblp	00000002	Word	0090
e_cp	00000004	Word	0003
e_crlc	00000006	Word	0000
e_cparhdr	00000008	Word	0004
e_minalloc	0000000A	Word	0000
e_maxalloc	0000000C	Word	FFFF
e_ss	0000000E	Word	0000
e_sp	00000010	Word	00B8
e_csum	00000012	Word	0000
e_ip	00000014	Word	0000
e_cs	00000016	Word	0000
e_lfarlc	00000018	Word	0040
e_ovno	0000001A	Word	0000
e_res	0000001C	Word	0000
	0000001E	Word	0000
	00000020	Word	0000
	00000022	Word	0000
e_oemid	00000024	Word	0000
e_oeminfo	00000026	Word	0000
e_res2	00000028	Word	0000
	0000002A	Word	0000
	0000002C	Word	0000
	0000002E	Word	0000
	00000030	Word	0000

Malware\_U3\_W2\_L1.exe

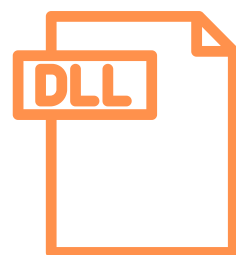
Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
00000A98	N/A	00000A00	00000A04	00000A08	00000A0C	00000A10
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.DLL	6	00000000	00000000	00000000	00006098	00006064
ADVAPI32.dll	1	00000000	00000000	00000000	000060A5	00006080
MSVCRT.dll	1	00000000	00000000	00000000	000060B2	00006088
WININET.dll	1	00000000	00000000	00000000	000060BD	00006090

OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
N/A	000060C8	0000	LoadLibraryA
N/A	000060D6	0000	GetProcAddress
N/A	000060E6	0000	VirtualProtect
N/A	000060F6	0000	VirtualAlloc
N/A	00006104	0000	VirtualFree
N/A	00006112	0000	ExitProcess

funzioni

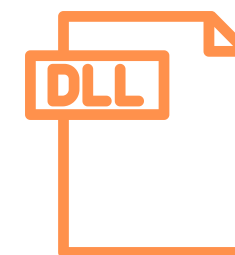
# Analisi dll

Queste sono le DLL trovate durante l'analisi statica del malware



## Kernel32.dll

Gestisce operazioni di base come aprire e salvare file, gestire la memoria e avviare programmi. È essenziale per il funzionamento del sistema operativo.



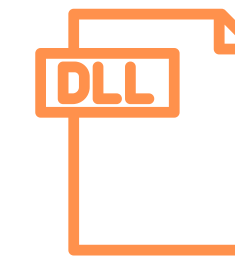
## Advapi32.dll

Si occupa di funzioni avanzate come la sicurezza dei file e la gestione dei servizi di sistema. Aiuta a configurare e controllare vari aspetti del sistema operativo.



## Msvcrt.dll

Fornisce funzioni di base del linguaggio C, come leggere e scrivere dati e fare calcoli. È utilizzata dai programmi per operazioni comuni.

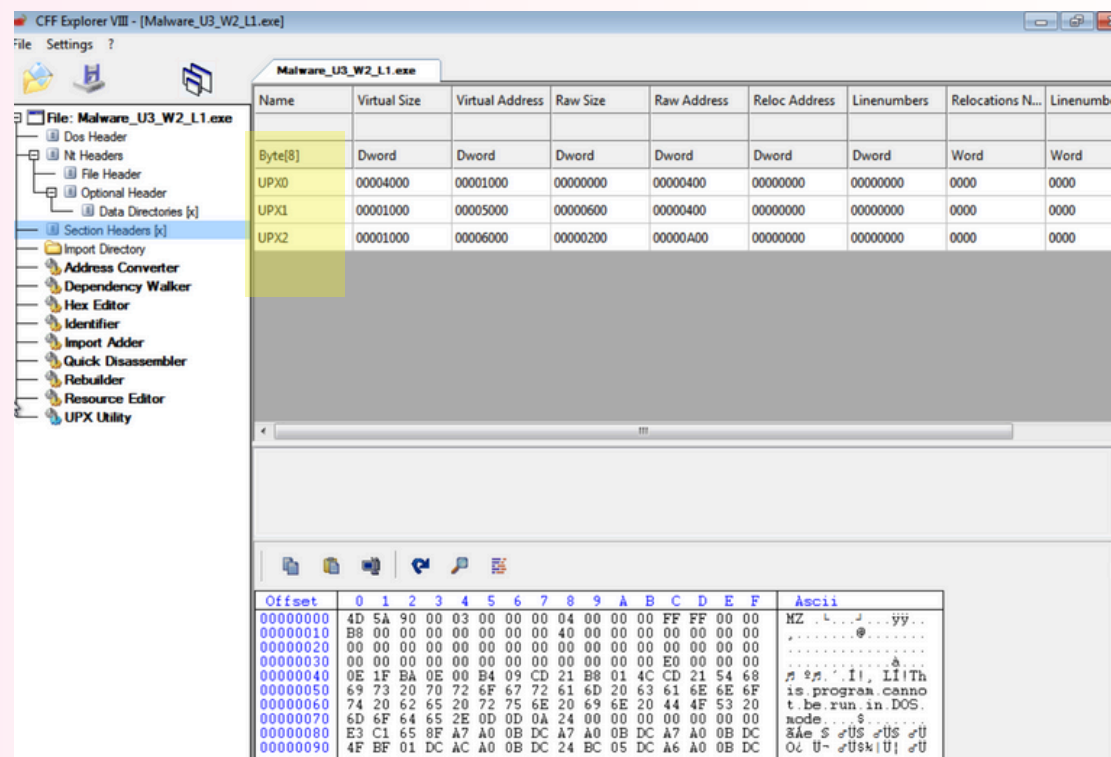


## Wininet.dll

Gestisce la connessione a internet e le comunicazioni di rete, come navigare su siti web e scaricare file. Aiuta le applicazioni a comunicare online.

# ANALISI

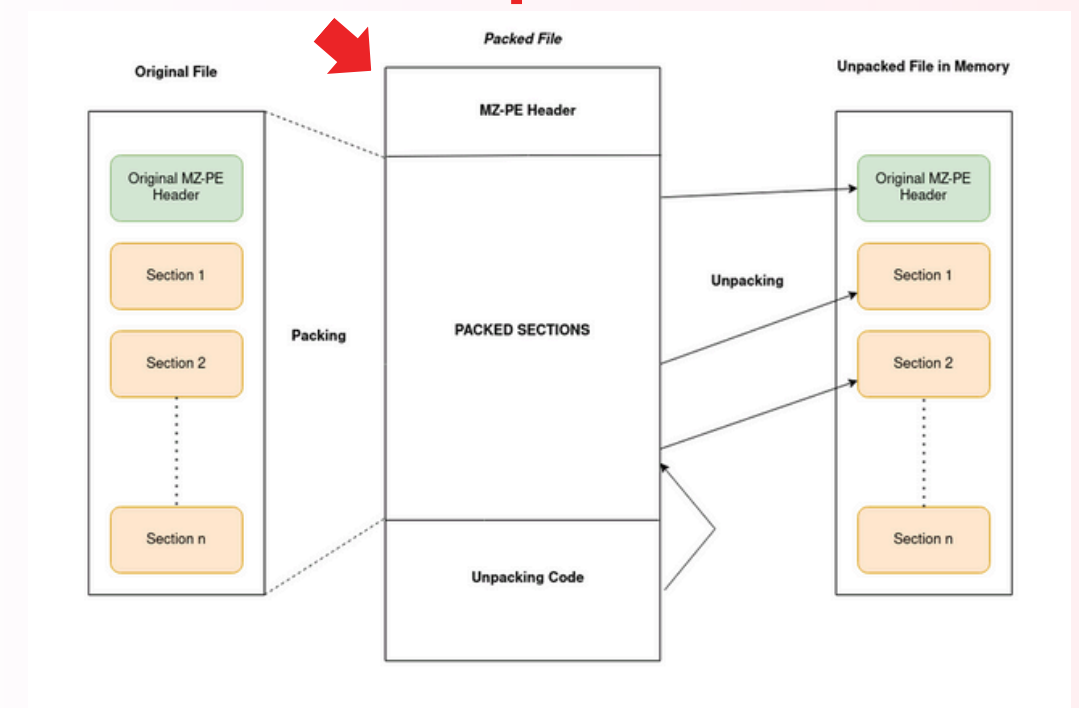
Controlliamo ora, come richiesto dalla traccia, le **sezioni di cui si compone il malware**. Da notare come CFF Explorer abbia al suo interno, oltre alle varie funzionalità di analisi e modifica di un PE, un editor esadecimale (simile al tool che ho usato prima a scopo di test: HxD).



Queste tre sezioni hanno nomi strani (**UPX**). Cercando su internet pare si tratti di un **executable packer** open source utilizzato dagli autori di malware per offuscare e comprimere il loro codice dannoso.

**Noi siamo qui**


Abbiamo quindi bisogno di **maggiori competenze di analisi** di malware per poter deoffuscare le sezioni.





# Considerazioni finali

Comunque sia, le funzioni **LoadLibrary** e **GetProcAddress** beccate prima durante l'analisi con CFF Explorer, **complicano** l'analisi statica del malware perché permettono il caricamento e l'uso dinamico delle librerie a runtime. LoadLibrary carica DLL solo quando sono effettivamente necessarie durante l'esecuzione, mentre GetProcAddress risolve gli indirizzi delle funzioni in queste librerie **al momento dell'esecuzione**. Questo significa che il malware può **nascondere** le sue dipendenze reali e le funzioni utilizzate, rendendo difficile identificare quali librerie e funzioni sono realmente impiegate solo analizzando il file eseguibile staticamente. Per scoprire queste informazioni, dovrei effettuare un'**analisi dinamica**, in cui il malware viene eseguito in un ambiente controllato per osservare quali librerie e funzioni vengono caricate e utilizzate effettivamente. Motivo per cui ci vediamo nella prossima puntata!







# **GRAZIE**

**Flavio Scognamiglio**