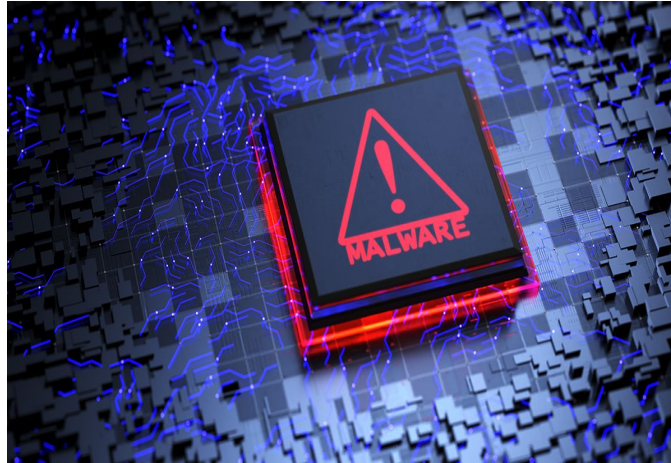


PROGETTO S10L5



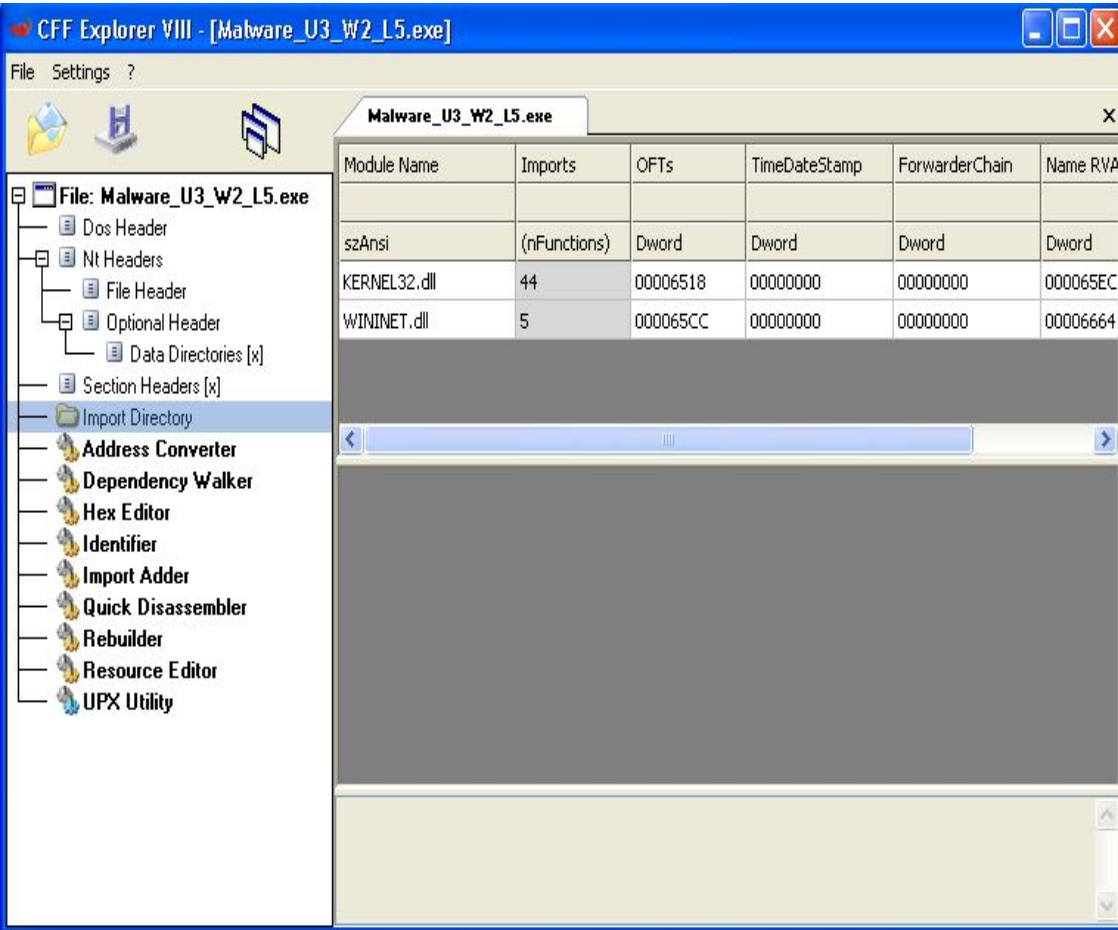
MALWARE

1.Quali librerie vengono importate dal file eseguibile?

Partiamo con il dire che conoscere le librerie e il loro scopo è una competenza fondamentale per l'analisi dei malware.

In questo caso utilizzeremo il software “CFF Explorer” per esplorare e analizzare il file eseguibile,e in particolare ci soffermeremo ad analizzare la cartella “Import Directory” ,una parte della struttura del file che contiene informazioni sugli importati (funzioni o librerie) che il programma utilizza.

- **Kernel32.dll:** contiene le funzioni principali per interagire con il sistema operativo, ad esempio: manipolazione dei file, la gestione della memoria.
- **Wininet.dll:** contiene le funzioni per l'implementazione di alcuni protocolli di rete come HTTP, FTP, NTP.



Approfondimenti

Si poteva ricavare le librerie anche con l’uso del servizio online gratuito “VirusTotal” (normalmente utilizzata come prima fonte di informazione nell’analizzare un malware)

39

/ 71

39 security vendors and no sandboxes flagged this file as malicious

ReanalyzeSimilarMore

b71777edbf2167c76cd20ff803cbcb25d24b94b3652db2f28dcd05ef3d8416a

Size40.00 KB

Last Analysis Date5 months ago

EXE

Lab06-02.exe

peviewchecks-network-adaptersruntime-modulesarmadillodirect-cpu-clock-access

Community Score

DETECTION

DETAILS

RELATIONS

BEHAVIOR

COMMUNITY

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat labeltrojan:ro02c0pdm21

Threat categoriestrojan

Family labelsro02c0pdm21

Security vendors' analysis

Do you want to automate checks?

Alibaba	Trojan:Win32/Generic.be125c32	Antly-AVL	Trojan:Win32/BTS/Generic
Avast	Win32:Trojan-gen	AVG	Win32:Trojan-gen
CrowdStrike Falcon	Win/malicious_confidence_100% (W)	Cybereason	Malicious.tfe74
Cylance	Unsafe	Cynet	Malicious (score: 100)
DeepInstinct	MALICIOUS	DrWeb	Trojan.MulDrop.63090
Elastic	Malicious (high Confidence)	ESET-NOD32	Win32/Agent.WOO
Fortinet	W32/Agent.WOO!tr	GData	Win32:Trojan.Agent.D29CIW
Google	Detected	Gridinsoft (no cloud)	Ransom.Win32.Wacatac.oelst
Ikarus	Trojan:Win32/Agent	Lionic	Trojan:Win32/Generic.4tc
Malwarebytes	Generic:Trojan.Malicious.DD5	MAX	Malware (a:Score=97)
MaxSecure	Trojan.Malware.300983.sugen	McAfee	GenericRXXAA-AA/C0B54534E188
McAfee-GW-Edition	Artemis!Trojan	Microsoft	Trojan:Win32/Ymacco.AAB7
NANO-Antivirus	Trojan:Win32/Agent.dveqk	Rising	Trojan.Agent!B.B1E:5W5kRu0p5wdF
Sangfor Engine Zero	Trojan:Win32/Agent.Wlo	Symantec	ML.Attribute.HighConfidence
TACHYON	Trojan:Win32/Agent.40960.ESE	Tencent	Malware.Win32/Generic.115cdf77
Trellix (FireEye)	Generic.mg.c0b54534e188a139	TrendMicro	TROJ_GEN.R002C0PDM21

Portable Executable Info ⓘ

Compiler Products

[C++] VS98 (6.0) SP6 build 8804 count=1

id: 14, version: 7299 count=14

[C] VS98 (6.0) SP6 build 8804 count=45

[...] Unmarked objects (old) count=7

id: 19, version: 8034 count=3

[...] Unmarked objects count=45

id: 21, version: 9782 count=1

Header

Target Machine Intel 386 or later processors and compatible processors

Compilation Timestamp 2011-02-02 21:29:05 UTC

Entry Point 4528

Contained Sections 3

Sections

Name	Virtual Address	Virtual Size	Raw Size	Entropy	MD5	Chi2
.text	4096	19064	20480	6.37	4b8aaeb128744c00b1f9b29dd120616e	196535.5
.rdata	24576	2398	4096	3.66	e5e39acc53e64c50fa5a35693a911478	304856
.data	28672	16136	12288	0.7	305514f6ece00473b7f18bc023f57e15	2765274

Imports

+ KERNEL32.dll

+ WININET.dll

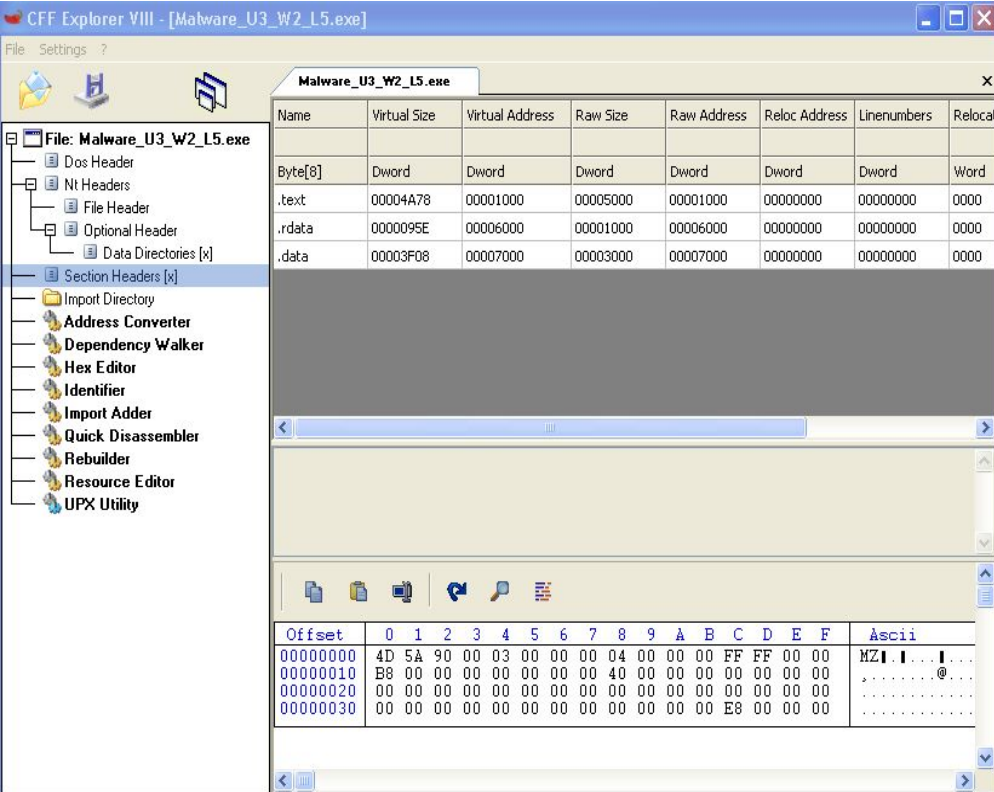
2. Quali sono le sezioni di cui si compone il file eseguibile del malware?

L'header del formato PE fornisce molte altre informazioni importanti oltre alle funzioni e librerie importate ed esportate, come ad esempio le sezioni di cui si compone il software.

Le "section headers" (intestazioni di sezione) sono una parte fondamentale di molti formati di file eseguibili.

Queste intestazioni forniscono informazioni dettagliate sulla disposizione e sulla struttura delle diverse sezioni presenti nel file eseguibile. Ogni sezione può contenere diversi tipi di dati, come il codice eseguibile, i dati in sola lettura, i dati in sola scrittura, la tabella di importazione, la tabella di esportazione e così via.

- **.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.

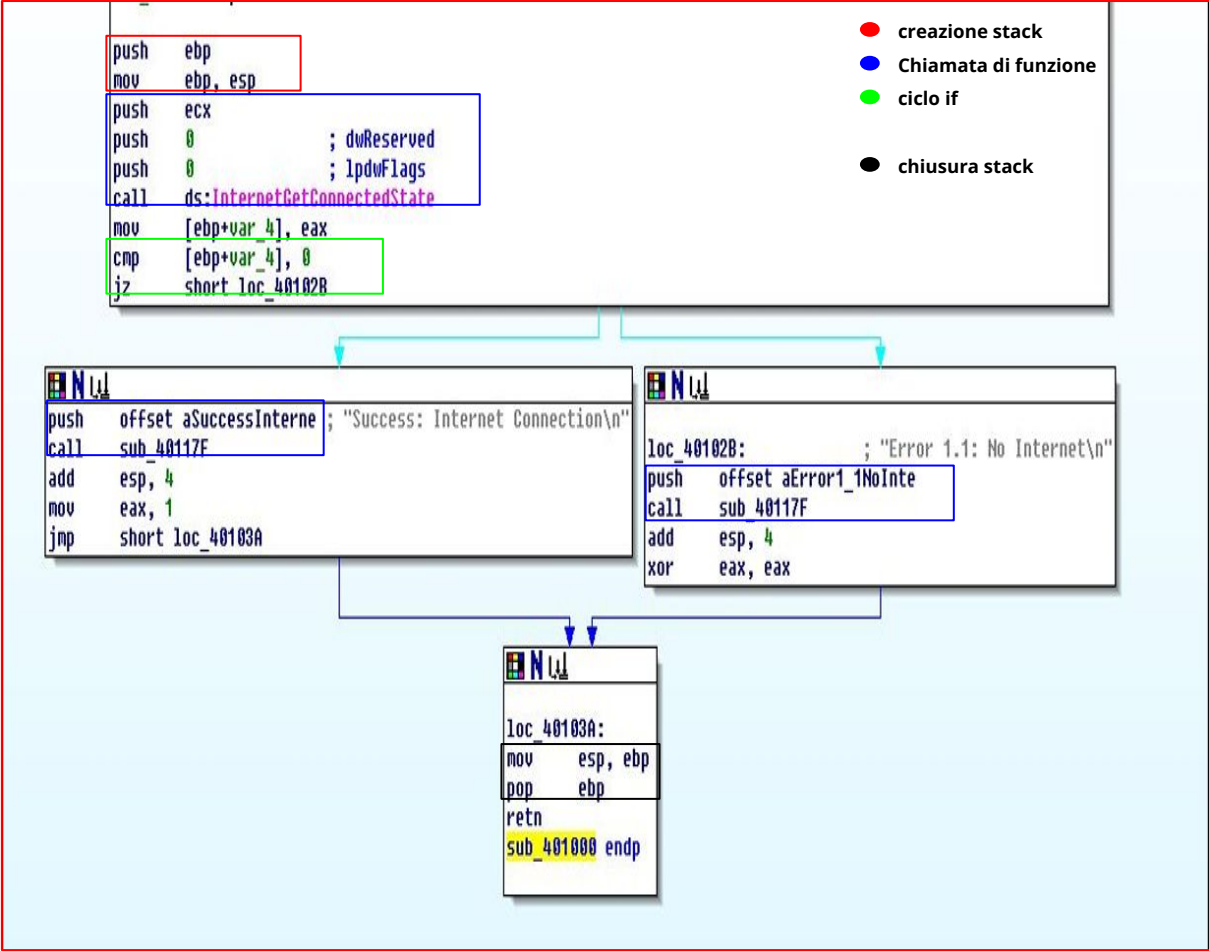


Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocat
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word
.text	00004A78	00001000	00005000	00001000	00000000	00000000	0000
.rdata	0000095E	00006000	00001000	00006000	00000000	00000000	0000
.data	00003F08	00007000	00003000	00007000	00000000	00000000	0000

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Ascii
00000000	4D	5A	90	00	03	00	00	00	04	00	00	00	FF	FF	00	00	MZ
00000010	B8	00	00	00	00	00	00	00	40	00	00	00	00	00	00	00	.
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	@
00000030	00	00	00	00	00	00	00	00	00	00	00	00	E8	00	00	00	.

3. Identificare i costrutti noti (creazione dello stack, eventuali cicli, costrutti)

- 1) Push ebp
- 2) Mov ebp, esp
- 3) Push ecx
- 4) Push 0 ; dwReserved
- 5) Push 0 ; lpdwFlags
- 6) Call ds: InternetGetConnectedState
- 7) Mov [ebp+var_4], eax
- 8) Cmp [ebp+var_4], 0
- 9) Jz short loc_40102B
- 10) Push offset aSuccessInterne ; "Success: Internet Connection\n"
- 11) Call sub_40117F
- 12) Add esp, 4
- 13) Mov eax, 1
- 14) Jmp short loc_40103A
- 15) Loc_40102B:
- 16) Push offset aError1_1NoInte
- 17) Call sub_40117F
- 18) Add esp, 4
- 19) Xor eax, eax
- 20) Loc 40103°:
- 21) Mov esp, ebp
- 22) Pop ebp
- 23) Retn
- 24) Sub_401000 endp



4. Ipotesizzare il comportamento della funzionalità implementata

In seguito all'analisi del codice assembly in questione, è stato evidenziato che il programma verifica la disponibilità di una connessione Internet sulla macchina di destinazione mediante l'utilizzo della funzione `InternetGetConnectedState`. A seconda dell'esito di tale verifica, il programma presenta un riscontro positivo in caso di connessione attiva e un riscontro negativo in caso di assenza di connessione.

Merece menzione il fatto che, in situazioni di mancanza di connessione, il programma riporta il valore del registro **eax** a zero. Tale procedura suggerisce che il presunto malware potrebbe non essere in grado di sfruttare appieno le proprie funzionalità in assenza di una connessione Internet.

Quindi si presume che ,tale malware , abbia bisogno della connessione internet per svolgere determinate operazioni, come l'invio di file e dati sensibili a server controllati dall'attaccante, la connessione a domini compromessi per il download di ulteriori malware, creazione di una backdoor in modo da stabilire una connessione continua tra attaccante e vittima.

Considerando tali elementi, è ragionevole supporre che il malware contenente la porzione di codice analizzata possa manifestare le caratteristiche di una backdoor ,di un trojan o di un downloader.



B_A - Pixabay

Dott. Alex Doddis