

Report Malware Analysis

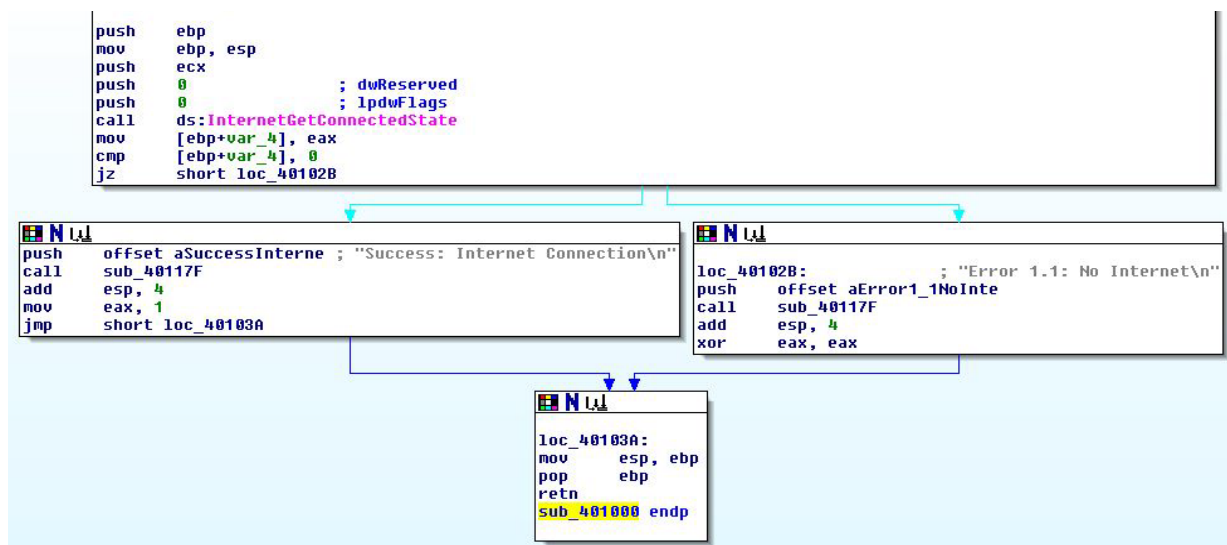
Questo progetto prevede l'esecuzione di attività di **malware analysis**. In particolare, vengono effettuate due attività distinte.

Nella prima parte viene effettuata, all'interno del **laboratorio virtuale** appositamente configurato, l'analisi di base del malware al seguente path:

C:\Documents and Settings\Administrator\Desktop\Esercizio_Pratico_U3_W2_L5\Malware_U3_W2_L5.exe

Tale attività è finalizzata a reperire le seguenti informazioni sul malware: quali **librerie** vengono importate dall'eseguibile e da quali **sezioni** è composto.

Nella seconda parte, invece, viene effettuata un'analisi più approfondita sul seguente estratto di codice **Assembly x86**:



Gli obiettivi dell'analisi sono: identificare i **costrutti** noti ed ipotizzare il **comportamento** della funzionalità implementata.

Prima parte

Per ottenere le informazioni richieste è sufficiente effettuare un'analisi **statica** di base del malware. A tal proposito, viene utilizzato il tool CFF Explorer, che restituisce i seguenti risultati:

Section Headers [x] - [Malware_U3_W2_1.5.exe]

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc. Address	LineNumbers	Relocations ...	LineNumber...	Characters
00001E00	000001E8	000001EC	000001F0	000001F4	000001F8	000001FC	00000200	00000202	00000204
Byte[0]	Dword	Dword	Dword	Dword	Dword	Dword	Dword	Dword	Dword
.text	00004A78	00001000	00005000	00001000	00000000	0000	0000	0000	60000020
.rdata	0000095E	00006000	00006000	00006000	00000000	00000000	0000	0000	40000040
.data	00003F08	00007000	00003000	00007000	00000000	00000000	0000	0000	C0000040

Import Directory - [Malware_U3_W2_1.5.exe]

Module Name	Imports	OFIs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
000065EC	N/A	000064DC	000064E0	000064E4	000064EC	000064EC
kernel32.dll	(nFunctions)	Dword	Dword	Dword	Dword	Dword
USER32.dll	44	00006518	00000000	00000000	000065EC	00006000
WININET.dll	5	000065CC	00000000	00000000	00006664	000060B4

OFIs

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	0100	LMapStringW
000068FC	000068FC	01C8	LMapStringA
000068E6	000068E6	01E4	MultiByteToWideChar
00006670	00006670	00CA	GetCommandLineA
00006682	00006682	0174	GetVersion
00006690	00006690	007D	ExitProcess
0000669E	0000669E	029F	TerminateProcess
00006682	00006682	00F7	GetCurrentProcess
000066C6	000066C6	02AD	UnhandledExceptionFilter
000066E2	000066E2	0182	GetModuleFileName
000066F8	000066F8	00B4	FreeEnvironmentStringsA
00006712	00006712	00B3	FreeEnvironmentStringsW
0000672C	0000672C	00D2	WideCharToMultiByte
00006742	00006742	0106	GetEnvironmentStrings
0000675A	0000675A	0108	GetEnvironmentStringW
00006774	00006774	0260	SetHandleCount
00006786	00006786	0125	GetStdHandle
00006796	00006796	0155	GetFileType
000067A4	000067A4	0110	GetStartupInfo
00006786	00006786	0126	GetModuleHandleA
000067CA	000067CA	0109	GetEnvironmentVariableA
000067E4	000067E4	0175	GetVersionExA
000067F4	000067F4	019D	HeapDestroy
00006802	00006802	0190	HeapCreate
00006810	00006810	02BF	VirtualFree
0000681E	0000681E	01F8	HeapFree

This section contains:

Code Entry Point: 00001B80

Offset

0

1

2

3

4

5

6

7

8

9

A

B

C

D

E

F

Ascii

00000000	55	8B	EC	51	6A	00	6A	00	FF	15	C0	40	00	89	45	0133 0x3310	
00000001	FC	87	7D	3C	00	74	14	68	48	70	40	00	8B	EC	51	00000000	
00000002	00	83	C4	04	B8	01	00	00	EB	0F	00	00	00	00	00	00000000	
00000003	E8	44	01	00	83	C4	04	33	00	8B	EC	51	C3	CC	CC	00000000	
00000004	55	8B	EC	51	6A	00	00	6A	00	6A	00	6A	00	6A	00	00000000	
00000005	00	6A	70	00	00	FF	15	C4	60	40	00	89	45	F4	6A	00000000	
00000006	00	6A	00	6A	00	6A	00	68	C4	70	00	8B	45	F4	6A	00000000	
00000007	55	8B	EC	51	6A	00	00	6A	00	F3	70	00	75	16	68	00000000	
00000008	A8	70	40	00	E8	F6	00	00	83	C4	04	00	8D	F4	F1	00000000	
00000009	FF	15	B8	60	40	00	32	C0	E9	8F	00	00	8D	55	F8	00000000	
0000000A	FF	60	00	02	45	8D	85	F0	FD	FF	50	8D	40	00	00	00000000	
0000000B	51	FF	15	BC	60	40	00	89	45	FC	73	00	00	75	25	00000000	
0000000C	68	68	70	40	00	E8	B5	00	00	83	C4	04	B8	55	F4	00000000	
0000000D	52	FF	15	B8	60	40	00	89	45	F0	FD	FF	50	8D	40	00000000	
0000000E	00	32	C0	EB	47	0F	BE	8D	F0	FD	FF	50	8D	40	00	00000000	
0000000F	2C	0F	BE	95	F1	FD	FF	FF	83	F4	21	75	20	0F	BE	00000000	
00000010	F2	FD	FF	83	F4	21	75	14	0F	BE	8D	F0	FD	FF	FF	00000000	
00000011	83	F9	2D	75	08	8A	85	F4	FD	FF	FF	FF	0F	68	70	00000000	
00000012	00	00	E8	58	00	00	00	83	C4	04	32	C0	EB	55	D3	00000000	
00000013	55	8B	EC	51	6A	00	00	E8	C5	FE	FF	89	45	FC	73	00000000	
00000014	FC	00	75	04	33	C0	EB	33	E8	F3	FE	FF	89	45	F8	00000000	
00000015	0F	BE	45	F8	85	C0	75	04	33	C0	E8	F3	FE	FF	89	00000000	
00000016	51	68	10	71	40	00	E8	14	00	00	83	C4	04	68	60	00000000	
00000017	EA	00	00	FF	15	60	40	00	33	C0	E8	14	00	C3	53	00000000	
00000018	5E	BE	60	71	40	00	57	56	E8	48	01	00	8B	F8	B8	00000000	
00000019	44	18	60	FF	74	18	56	E8	04	02	00	00	8B	F8	B8	00000000	
0000001A	8B	DE	8B	EB	01	00	00	83	C4	18	63	00	57	56	5B	00000000	
0000001B	55	8B	EC	5A	FF	68	60	40	00	68	F8	27	40	00	64	00000000	
0000001C	A1	00	00	00	00	00	00	54	89	25	00	00	00	83	EC	10	00000000
0000001D	53	87	59	45	E9	FF	15	60	40	00	33	D2	A4	00	00	00000000	
0000001E	89	15	08	9A	40	00	8B	C8	E1	FE	00	00	00	8D	00	00000000	
0000001F	04	9A	40	00	C1	E1	08	33	C4	89	00	9A	40	00	C1	00000000	
00000020	8B	DE	8B	EB	01	00	00	E9	92	14	00	00	59	85	00	00000000	
00000021	00	75	08	6A	C1	E8	9A	00	00	59	83	65	FC	C0	E8	00000000	
00000022	5C	11	00	00	FF	15	60	40	00	A3	04	AF	40	00	E8	00000000	
00000023	A3	10	00	00	83	D8	99	40	00	E8	C3	0D	00	00	E8	00000000	
00000024	00	00	EB	7A	0A	00	00	00	A1	18	9A	40	00	A3	1C	9A	00000000
00000025	40	00	50	FF	35	10	9A	40	00	FF	35	0C	9A	40	00	E8	00000000
00000026	CC	FF	FF	FF	C4	C4	C4	89	45	E4	50	E8	7B	0A	00	00	00000000
00000027	8B	45	EC	8B	89	89	89	40	00	50	E1	51	89	43	0B	00	00000000

Dallo screenshot in alto si evince che il malware è un file Portable Executable composto dalle sezioni **.text**, **.rdata** e **.data**. Inoltre importa le librerie **KERNEL32.dll** e **WININET.dll**.

Una scansione con Virustotal conferma la pericolosità del file PE:

b71777edb121167c96d20f803cbcb25d24b94b3652db21286dcd6fd3d8416a

39

/ 69

39 security vendors and no sandboxes flagged this file as malicious

b71777edb121167c96d20f803cbcb25d24b94b3652db21286dcd6fd3d8416a

Lab06-02.exe

40.00 KB

Size

2023-02-14 11:11:24 UTC

1 month ago

EXE

peexe

checks-network-adapters

runtime-modules

armadillo

direct-cpu-clock-access

Community Score

DETECTION

DETAILS

RELATIONS

BEHAVIOR

COMMUNITY 6

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

Popular threat label

trojan.r002c0pdm21

Threat categories

trojan

Family labels

r002c0pdm21

Security vendors' analysis

Do you want to automate checks?

Alibaba	Trojan/Win32/Generic.be125c32	Antiy-AVL	Trojan/Win32/BTSGeneric
Avast	Win32:Trojan-gen	AVG	Win32:Trojan-gen
Avira (no cloud)	HEUR/AGEN.1240704	CrowdStrike Falcon	Win/malicious_confidence_100% (W)
Cybereason	Malicious.1fe74	Cylance	Unsafe
Cynet	Malicious (score: 100)	DrWeb	Trojan/MulDrop7.63090

Anche da una sommaria analisi **dinamica** emerge che si tratta di un malware. In particolare, dai seguenti screenshot si evince la creazione da parte del malware di alcuni thread e attività sul filesystem, oltre che numerose chiavi di registro:

[illegible]

Tra le attività più pericolose poste in essere dal malware in esame ritroviamo: la creazione di processi e file, recupero di informazioni sul dispositivo, modifica di certificati, controllo sulla presenza di una connessione ad Internet. Inoltre, attraverso l'immissione dell'hash del file in servizi online di malware analysis (VirusTotal e any.run) vengono confermati i sospetti, a cui vengono aggiunti anche altri dettagli, come le richieste HTTP, che non erano state catturate nell'attività precedente:

b71777edbf21167c96d20ff803cbcb25d24b04b3652db2f286dc96fd3d8416a

Activity Summary

Download Artifacts Full Reports Help

Network Communication

HTTP Requests

- + http://practicalmalwareanalysis.com/?post_type=feedback&p=374
- + http://practicalmalwareanalysis.com/cc.htm
- + http://www.practicalmalwareanalysis.com
- + http://www.practicalmalwareanalysis.com/cc.htm
- + https://practicalmalwareanalysis.com
- + https://www.practicalmalwareanalysis.com

DNS Resolutions

- + 132.155.190.20 in-addr.arpa
- + 150.32.88.40 in-addr.arpa
- + 16.155.190.20 in-addr.arpa
- + 24.78.0.192 in-addr.arpa
- + 25.78.0.192 in-addr.arpa
- + 29.91.21.72 in-addr.arpa
- + 48.193.43.104 in-addr.arpa
- + login.live.com
- + practicalmalwareanalysis.com
- + prda.aadg.msidentity.com

ANY.RUN

Lab06-02.exe

Win7 32 bit Complete

MD5: C0B54534E188E1392F28D17FAFF3D454

Start: 04.04.2020, 16:52 Total time: 60 s

Get sample IOC MalConf Restart

Text report Process graph ATT&CK matrix Export

Processes

PID	Process name	PE	734	3k	154
572	Lab06-02.exe	PE			

Process details ID 572 Malicious

Start: +0ms Indicators: *

Command line

"C:\Users\admin\AppData\Local\Temp\Lab06-02.exe"

More Info

Danger 1

Changes settings of System certificates

Warning 2

Adds / modifies Windows certificates

Reads Internet Cache Settings

Warning [572] Lab06-02.exe Adds / modifies Windows certificates

Try community version for free! Register now

Seconda parte

Viene, innanzi tutto, effettuata un'analisi del codice riga per riga:

push ebp -salva il valore del registro ebp nello stack

mov ebp, esp -imposta il registro ebp con il valore corrente dello stack

push ecx -salva il valore del registro ecx nello stack

push 0 ; dwReserved -salva il valore 0 nello stack come dwReserved

push 0 ; lpdwFlags -salva il valore 0 nello stack come lpdwFlags

call ds: InternetGetConnectedState -chiama la funzione InternetGetConnectedState dal Data segment

mov [ebp+var_4], eax -salva il valore di ritorno della funzione in [ebp+var_4]

<code>cmp [ebp+var_4], 0</code>	-confronta [ebp+var_4] con 0
<code>jz short loc_40102B</code>	-salta a loc_h6102B se [ebp+var_4] è zero
<code>offset aSuccessInterne ; "Success: Internet Connection\n"</code>	-stringa di output
<code>sub_40117F</code>	-chiama la funzione sub_40117F
<code>add esp, 4</code>	-aggiunge 4 byte allo stack
<code>mov eax, 1</code>	-imposta il registro eax a 1
<code>short loc_40103A</code>	-salta a loc_40103A
<code>loc_40102B: ; "Error 1.1: No Internet\n"</code>	-label di destinazione
<code>push offset aError1_1NoInte</code>	-pone l'indirizzo della stringa "Error 1.1: No Internet\n" nello stack
<code>call sub_40117F</code>	-chiama la funzione sub_40117F
<code>add esp, 4</code>	-aggiunge 4 byte allo stack
<code>xor eax, eax</code>	-esegue l'operazione xor con il registro eax
<code>loc_40103A:</code>	-label di destinazione
<code>mov esp, ebp</code>	-ripristina lo stack pointer originale
<code>pop ebp</code>	-ripristina il valore di ebp dallo stack
<code>retn</code>	-ritorna al chiamante
<code>sub_401000 endp</code>	-fine della funzione

Volendo effettuare un'analisi più approfondita, vengono esaminati i possibili **costrutti**:

<code>push ebp</code>	Inizializzazione dello stack per la funzione.
<code>mov ebp, esp</code>	

<code>push ecx</code>	Chiamata di funzione e salvataggio del valore di ritorno nella variabile <code>var_4</code> .
<code>push 0 ; dwReserved</code>	
<code>push 0 ; lpdwFlags</code>	
<code>call ds:InternetGetConnectedState</code>	
<code>mov [ebp+var_4], eax</code>	

<code>cmp [ebp+var_4], 0</code>	Confronto e salto condizionato, da cui si può dedurre l'utilizzo di if-else.
<code>jz short loc_40102B</code>	

<code>loc_40102B: ; "Error 1.1: No Internet\n"</code>	IF: se la funzione ritorna il valore 0, viene richiamata una funzione che, presumibilmente, restituisce in output
<code>push offset aError1_1NoInte</code>	
<code>call sub_40117F</code>	
<code>add esp, 4</code>	
<code>xor eax, eax</code>	

la stringa contenente il messaggio d'errore.

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

ELSE: se la funzione ritorna un valore diverso da 0, viene richiamata una funzione che, presumibilmente, restituisce in output la stringa contenente il messaggio sullo stato della connessione. Esegue infine un salto incondizionato.

```
loc_40103A:  
mov     esp, ebp  
pop     ebp  
retn  
sub_401000 endp
```

Epilogo della funzione e cancellazione dello stack.