

Progetto S10-L5

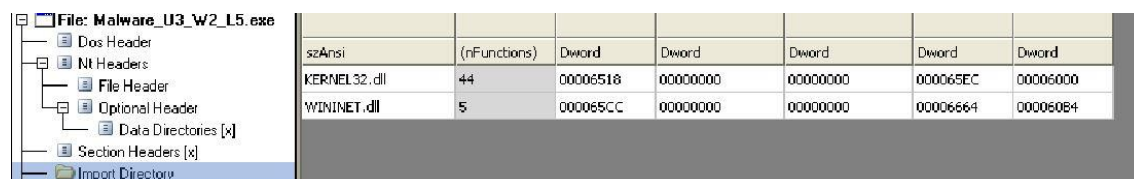
Studente: Simone Mininni

Malware Analysis

Analisi Statica

Nome File: Malware_U3_W2_L5

1) Librerie Importate dal file eseguibile



szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.dll	44	00006518	00000000	00000000	000065EC	00006000
WININET.dll	5	000065CC	00000000	00000000	00006664	000060B4

Il programma importa due librerie di windows:

- Kernel32.dll

- Contiene le funzioni core del kernel di windows.

- Wininet.dll

- Contiene funzioni per interagire con i protocolli ftp e http per accedere a risorse internet.

Funzioni del kernel:

-LoadLibraryA

-GetProcAddress

Spesso usate dai malware per offuscare il codice e per scopi di evasione importando le librerie a runtime.

Funzioni di Wininet.dll:

OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
00006640	00006640	0071	InternetOpenUrlA
0000662A	0000662A	0056	InternetCloseHandle
00006616	00006616	0077	InternetReadFile
000065FA	000065FA	0066	InternetGetConnectedState
00006654	00006654	006F	InternetOpenA

Funzioni utilizzate della libreria Wininet.dll

-InternetGetConnectedState

-InternetOpenUrlA

Quindi mi aspetto che il programma verifica lo stato della connessione internet e se è presente effettua una richiesta http.

2) Sezioni di cui si compone il file eseguibile

File: Malware_U3_W2_L5.exe									
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword
.text	00004A78	00001000	00005000	00001000	00000000	00000000	0000	0000	60000020
.rdata	0000095E	00006000	00001000	00006000	00000000	00000000	0000	0000	40000040
.data	00003F08	00007000	00003000	00007000	00000000	00000000	0000	0000	C0000040

Il programma si compone di tre sezioni:

- **.text**
 - contiene il codice(insieme di istruzioni) che verrà eseguito dalla cpu;
- **.rdata**
 - contiene informazioni riguardo le librerie importate ed esportate;
- **.data**
 - contiene le variabili globali;

Hash:

C0B54534E188E1392F28D17FAFF3D454 md5

BB6F01B1FEF74A9CFC83EC2303D14F492A671F3C sha1

Virus Total analysis:

39
/ 71

Community Score

39 security vendors and no sandboxes flagged this file as malicious

b71777edbf21167c96d20ff803cbcb25d24b94b3652db2f286dcd6efd3d8416a

Size40.00 KB

Last Analysis Date5 months ago

Lab06-02.exe

peexechecks-network-adaptersruntime-modulesarmadillo direct-cpu-clock-access

DETECTION

DETAILS

RELATIONS

BEHAVIOR

COMMUNITY7

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

Popular threat labeltrojan.r002c0pdm21

Threat categoriestrojan

Family labelsr002c0pdm21

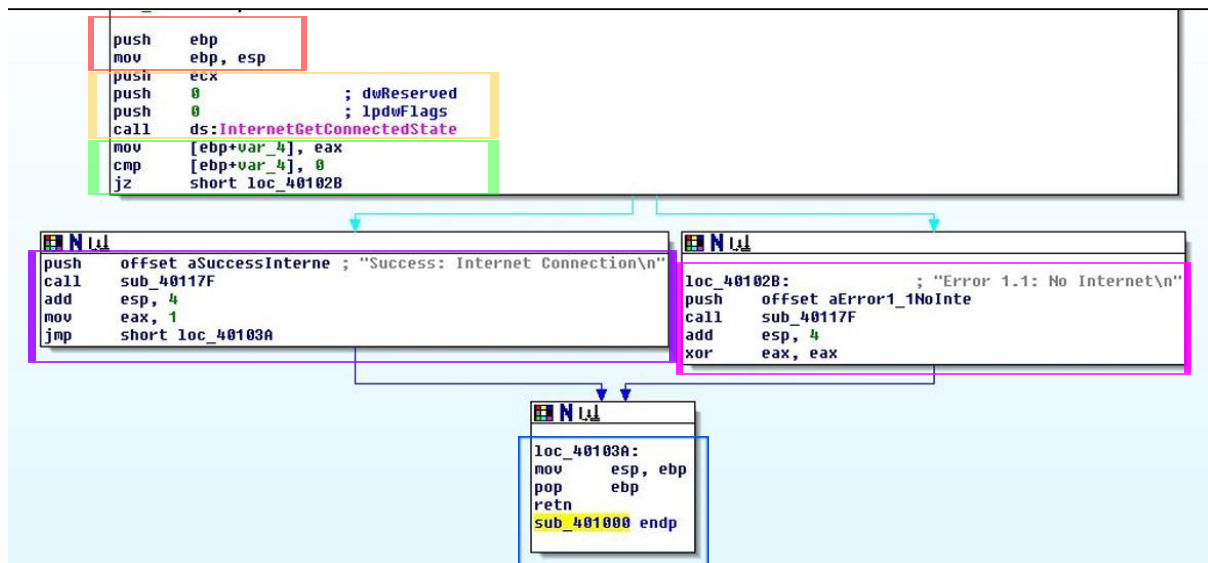
Security vendors' analysis

Do you want to aut

Alibaba	Trojan:Win32/Generic.be125c32	Antiy-AVL	Trojan:Win32.BTSGeneric
Avast	Win32:Trojan-gen	AVG	Win32:Trojan-gen
CrowdStrike Falcon	Win/malicious_confidence_100% (W)	Cybereason	Malicious.1fef74
Cylance	Unsafe	Cynet	Malicious (score: 100)
DeepInstinct	MALICIOUS	DrWeb	Trojan.MulDrop7.63090

Il file eseguibile viene etichettato come malevolo da 39 vendor ed è di tipo Trojan

3) Identificazione dei costrutti noti



La porzione di codice assembly può essere schematizzata in 4 costrutti noti:

- ☐ Creazione dello stack
- ☐ Chiamata di funzione("InternetGetConnectedState")
- ☐ Costrutto IF-ELSE
 - ☐ Corpo dell' if che viene eseguito se la condizione è verificata
 - ☐ Corpo dell' else, eseguito se la condizione non è verificata.
- ☐ Distruzione dello stack e ritorno alla funzione chiamante.

4) Possibile comportamento della funzionalità implementata

La porzione di codice definita si pone l'obiettivo di verificare lo stato della connessione della macchina corrente chiamando la funzione di windows "InternetGetConnectedState", alla quale vengono passati due parametri:

Syntax

```
C++

BOOL InternetGetConnectedState(
    [out] LPDWORD lpdwFlags,
    [in] DWORD dwReserved
);
```

In C restituirà un valore 1 o 0(True or False) e verrà confrontato in un if statement. Se è 0 verrà stampato un messaggio di errore "Error 1.1: No Internet\n", altrimenti viene stampato un messaggio di conferma "Success: Internet Connection\n". Infine la funzione ritornerà un valore alla funzione chiamante.

Link documentazione ufficiale windows:

<https://learn.microsoft.com/en-us/windows/win32/api/wininet/nf-wininet-internetgetconnectedstate>

Possibile rappresentazione in C del codice assembly:

```
int func(){  
  
    int stato = InternetGetConnectedState(0,0);  
  
    if(stato != 0){  
  
        printf("Success: Internet Connection\n");  
  
        return 1;  
    }  
    else{  
  
        printf("Error 1.1: No Internet\n");  
  
        return 0;  
    }  
}
```

5) Descrizione riga per riga del codice assembly

Costruzione stack:

push ebp

- Salvo nello stack il registro ebp che punta alla base dello stack;
-

mov ebp, esp

- Copio il valore di esp, registro che punta alla cima dello stack, in ebp;
-

Chiamata di funzione:

push ecx

- Salva il registro ecx nello stack;
-

push 0 ; dwReserved

- Salva la variabile riservata = 0 nello stack;
-

push 0 ; lpdwFlags

- Salva la variabile flag = 0 nello stack;
-

call ds:InternetGetConnectedState

- Chiama la funzione passando i parametri prima salvati;
-

Costrutto If-Else:

mov [ebp+var_4], eax

- Copia il valore del registro eax(valore di ritorno della funzione chiamata prima) nella variabile var_4;
-

cmp [ebp+var_4], 0

- Confronta il valore della variabile var_4 con 0, andando ad aggiornare il registro EFLAGS, in particolare i campi ZeroFlag e CarryFlag, in base al risultato della sottrazione tra il valore destinazione e sorgente;
-

jz short loc_40102B

- Dopo il confronto, se i due valori sono uguali, si salta alla locazione 40102B;
-

push offset aSuccessInterne ; "Success: Internet Connection\n"

- Si inserisce nello stack il messaggio che poi deve essere stampato;
-

call sub_40117F

- Si chiama una funzione di subroutine, probabilmente "printf";
-

add esp, 4

- Si ripulisce lo stack;
-

mov eax, 1

- Viene salvato nel registro eax il valore di ritorno 1;
-

jmp short loc_40103A

- Si salta alla locazione 40103A;
-

loc_0040102B:

push offset aError1_1NoInte ; "Error 1.1: No Internet\n"

- Salva il messaggio da stampare nello stack;
-

call sub_40117F

- chiama la funzione subroutine, probabilmente sempre printf;
-

add esp, 4

- Ripulisce lo stack dopo la printf;
-

xor eax, eax

- Assegna il valore 0 al registro eax;
-

Eliminazione dello stack:

loc_40103A:

mov esp, ebp

- Copia il valore di ebp in esp;
-

pop ebp

- Elimina il registro ebp dallo stack;
-

retn

- ritorna alla funzione chiamante il valore di eax;
-

sub_401000 endp

- Fine della funzione;
-