CS0424
S10-L5

Traccia:

1. Quali librerie vengono importate dal file eseguibile? Fare una descrizione.
2. Quali sono le sezioni di cui si compone il file eseguibile del malware? Fare una descrizione.
3. Identificare i costrutti noti nella figura nella slide 3.
4. Ipotizzare il comportamento della funzionalità implementata.
5. Fare una tabella per spiegare il significato delle singole righe di codice.

1. Librerie importate

Per vedere le librerie importate ho iniziato con una analisi statica utilizzando CFF Explorer.

Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
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

KERNEL32.dll con le funzioni:

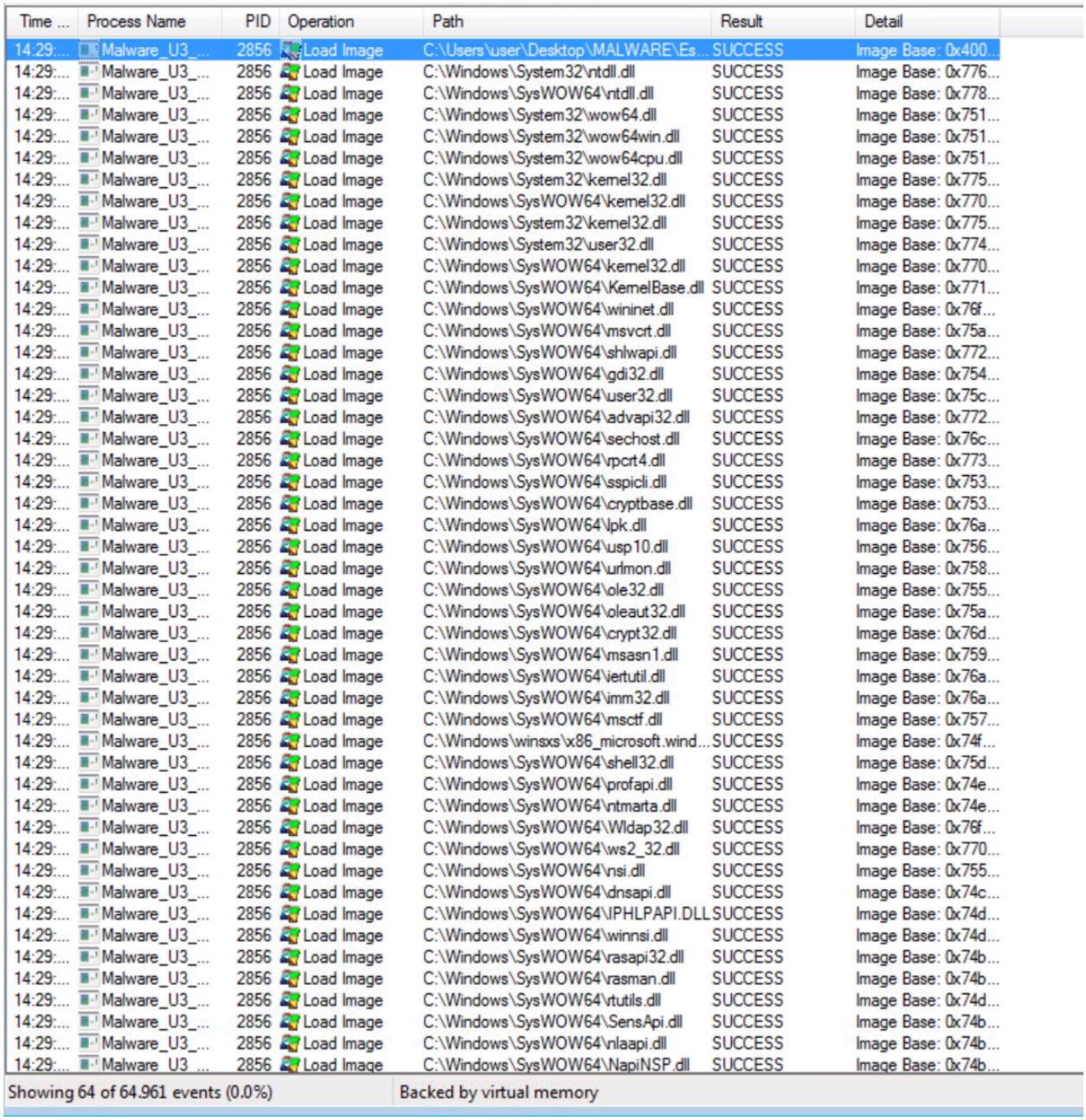
LoadLibraryA: Consente al malware di caricare di caricare librerie dinamicamente, quindi non visibili in un' analisi statica.
GetProcAddress: Permette al malware di eseguire funzioni durante il runtime.
VirtualProtect: Permette al malware di cambiare i permessi di un area della memoria e quindi potendo eseguire codice in aree non autorizzate.

ADVAPI32.dll con la funzione **CreateServiceA**: Potrebbe essere utilizzato dal malware per più scopi come ad esempio far avviare il malware all' avvio della macchina per essere persistente, inoltre grazie a questa funzione il malware potrebbe anche eseguire codice malevolo e ricevere comandi da un server esterno per consentire funzioni tipiche di ad esempio uno spyware eseguire un attacco DDoS.

MSVCRT.dll con la funzione **exit**: Potrebbe essere utilizzata dal malware per terminare processi creati dal malware stesso o anche processi legittimi consentendo un migliore occultamento.

WININET.dll con la funzione **InternetOpenA**: Consente al malware di stabilire connessioni HTTP o FTP con un server remoto e di ricevere/ inviare file.

Poi ho eseguito un analisi dinamica utilizzando ProcMon



Time ...	Process Name	PID	Operation	Path	Result	Detail
14:29:...	Malware_U3_...	2856	Load Image	C:\Users\user\Desktop\MALWARE\Es...	SUCCESS	Image Base: 0x400...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\System32\ntdll.dll	SUCCESS	Image Base: 0x776...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\ntdll.dll	SUCCESS	Image Base: 0x778...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\System32\wow64.dll	SUCCESS	Image Base: 0x751...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\System32\wow64win.dll	SUCCESS	Image Base: 0x751...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\System32\wow64cpu.dll	SUCCESS	Image Base: 0x751...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\System32\kernel32.dll	SUCCESS	Image Base: 0x775...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\kernel32.dll	SUCCESS	Image Base: 0x770...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\System32\kernel32.dll	SUCCESS	Image Base: 0x775...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\System32\user32.dll	SUCCESS	Image Base: 0x774...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\kernel32.dll	SUCCESS	Image Base: 0x770...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\KernelBase.dll	SUCCESS	Image Base: 0x771...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\wininet.dll	SUCCESS	Image Base: 0x76f...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\msvcrt.dll	SUCCESS	Image Base: 0x75a...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\shlwapi.dll	SUCCESS	Image Base: 0x772...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\gdi32.dll	SUCCESS	Image Base: 0x754...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\user32.dll	SUCCESS	Image Base: 0x75c...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\advapi32.dll	SUCCESS	Image Base: 0x772...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\sechost.dll	SUCCESS	Image Base: 0x76c...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\rpcrt4.dll	SUCCESS	Image Base: 0x773...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\sspicli.dll	SUCCESS	Image Base: 0x753...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\cryptbase.dll	SUCCESS	Image Base: 0x753...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\lpk.dll	SUCCESS	Image Base: 0x76a...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\usp10.dll	SUCCESS	Image Base: 0x756...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\urlmon.dll	SUCCESS	Image Base: 0x758...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\ole32.dll	SUCCESS	Image Base: 0x755...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\oleaut32.dll	SUCCESS	Image Base: 0x75a...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\crypt32.dll	SUCCESS	Image Base: 0x76d...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\msasn1.dll	SUCCESS	Image Base: 0x759...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\iertutil.dll	SUCCESS	Image Base: 0x76a...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\imm32.dll	SUCCESS	Image Base: 0x76a...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\msctf.dll	SUCCESS	Image Base: 0x757...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\winsxs\x86_microsoft.wind...	SUCCESS	Image Base: 0x74f...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\shell32.dll	SUCCESS	Image Base: 0x75d...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\profapi.dll	SUCCESS	Image Base: 0x74e...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\ntmarta.dll	SUCCESS	Image Base: 0x74e...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\Wldap32.dll	SUCCESS	Image Base: 0x76f...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\ws2_32.dll	SUCCESS	Image Base: 0x770...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\nsi.dll	SUCCESS	Image Base: 0x755...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\dnsapi.dll	SUCCESS	Image Base: 0x74c...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\IPHLPAPI.DLL	SUCCESS	Image Base: 0x74d...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\winnsi.dll	SUCCESS	Image Base: 0x74d...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\rasapi32.dll	SUCCESS	Image Base: 0x74b...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\rasman.dll	SUCCESS	Image Base: 0x74b...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\rtutils.dll	SUCCESS	Image Base: 0x74d...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\SensApi.dll	SUCCESS	Image Base: 0x74b...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\nlaapi.dll	SUCCESS	Image Base: 0x74b...
14:29:...	Malware_U3_...	2856	Load Image	C:\Windows\SysWOW64\NapiNSP.dll	SUCCESS	Image Base: 0x74b...

Showing 64 of 64.961 events (0.0%) Backed by virtual memory

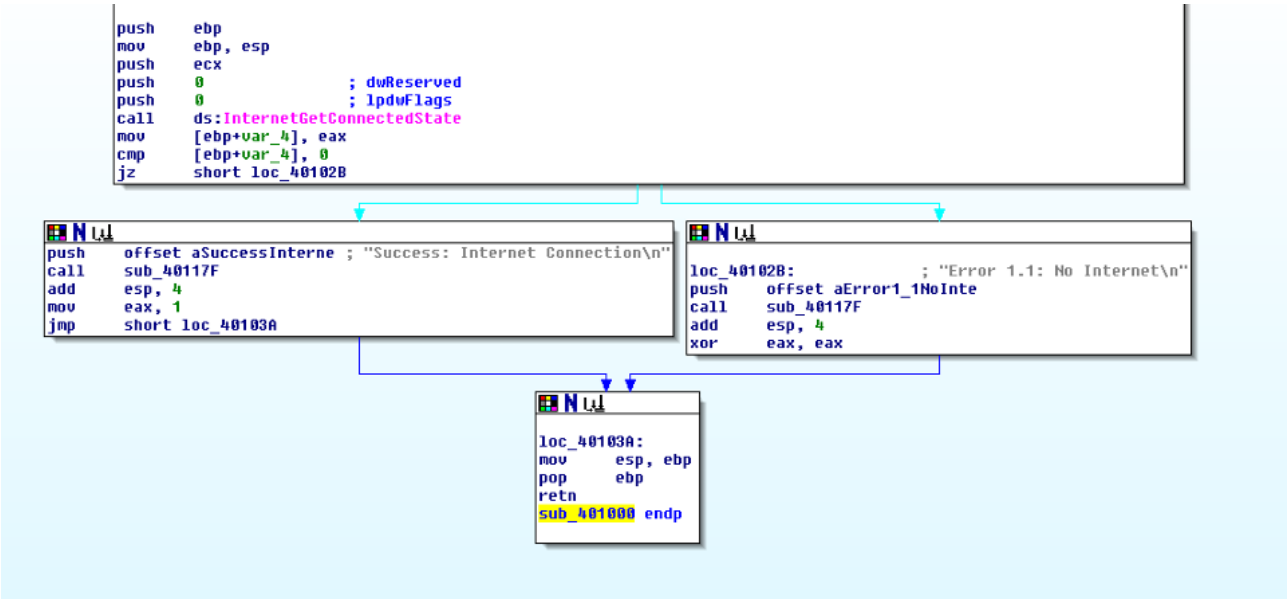
Trovando 64 librerie caricate dal malware

2. Sezioni del file eseguibile

Utilizzando ancora CFF Explorer si possono vedere 3 sezioni compresse in UPX

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address
Byte[8]	Dword	Dword	Dword	Dword	Dword
UPX0	00004000	00001000	00000000	00000400	00000000
UPX1	00001000	00005000	00000600	00000400	00000000
UPX2	00001000	00006000	00000200	00000A00	00000000

3. Identificare i costrutti



Nella scheda in alto viene inizializzato lo stack nelle prime 2 righe, vengono inseriti i valori per la funzione con “push” nelle righe 3-5 e viene chiamata la funzione alla riga 6. Alla riga 7 è presente un ciclo IF/ELSE.

La scheda a sinistra è il blocco IF, avviene se la condizione è vera. Carica un messaggio di successo sullo stack, effettua una chiamata a una funzione(probabilmente per salvare o mostrare il messaggio), porta il valore di eax a 1 e salta alle istruzioni della scheda in basso.

La scheda a destra (blocco ELSE) contiene quasi le stesse istruzioni del blocco IF, ma il messaggio caricato è un messaggio di errore e e il valore di eax viene portato a 0 e come il blocco IF, continua con le istruzioni della scheda in basso.

Nella scheda in basso viene copiato il valore di **ebp** in **esp,** viene azzerato ebp e viene restituito il controllo alla funzione chiamante.

4. Comportamento

Questa funzionalità verifica tramite la funzione “InternetGetConnectedState” se è presente o meno una connessione a internet o meno e risponde di conseguenza con uno dei due stati (Connessione riuscita o Errore), e termina ritornando al chiamante.

5. Significato delle singole righe di codice.

Codice Assembly	Significato
push ebp	Salva il valore del registro ebp sullo stack.
mov ebp, esp	Imposta ebp al valore di esp per creare un nuovo stack.
push ecx	Salva il valore del registro ecx sullo stack.
push 0	Imposta 0 come argomento “dwReserved” per la funzione InternetGetConnectedState.
push 0	Imposta 0 come argomento “lpdwFlags” per la funzione InternetGetConnectedState.
call ds:InternetGetConnectedState	Chiama la funzione InternetGetConnectedState
mov [ebp+var_4], eax	Memorizza il risultato della funzione InternetGetConnectedState in una variabile locale.
cmp [ebp+var_4], 0	Confronta il valore della variabile locale con 0.
jz short loc_401028	Se il valore=0, salta all'etichetta loc_401028 (indicando che non c'è connessione Internet).
push offset aSuccessInterne	Carica la stringa "Success: Internet Connection\n" sullo stack.
call sub_40117F	Chiama la funzione sub_40117F (per salvare o visualizzare il messaggio della stringa).
add esp, 4	Aumenta di 4 il valore di esp
mov eax, 1	Imposta eax a 1
jmp short loc_40103A	Salta all'etichetta loc_40103A
loc_401028:	Etichetta che indica l'inizio del blocco ELSE (Se non c’è connessione, valore=0)
push offset aError1_1NoInte	Carica la stringa "Error 1.1: No Internet\n" sullo stack.
call sub_40117F	Chiama la funzione sub_40117F (per salvare o visualizzare il messaggio della stringa).
add esp, 4	Ripristina lo stack rimuovendo l'argomento pushato.
xor eax, eax	Imposta eax a 0.
loc_40103A:	Etichetta che indica la locazione a cui deve saltare il blocco IF
mov esp, ebp	Copia il valore di ebp in esp
pop ebp	Azzera ebp
retn	Restituisce il controllo alla funzione chiamante.
sub_401000 endp	Direttiva che indica la fine della funzione sub_401000.