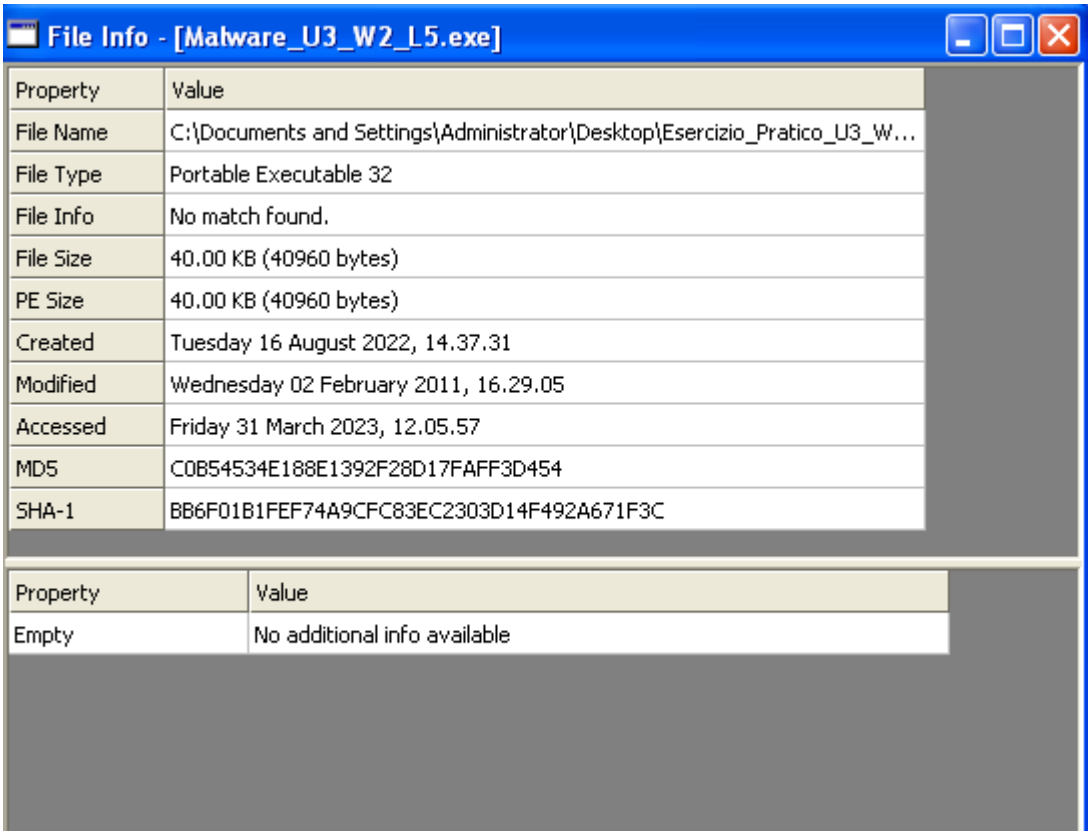
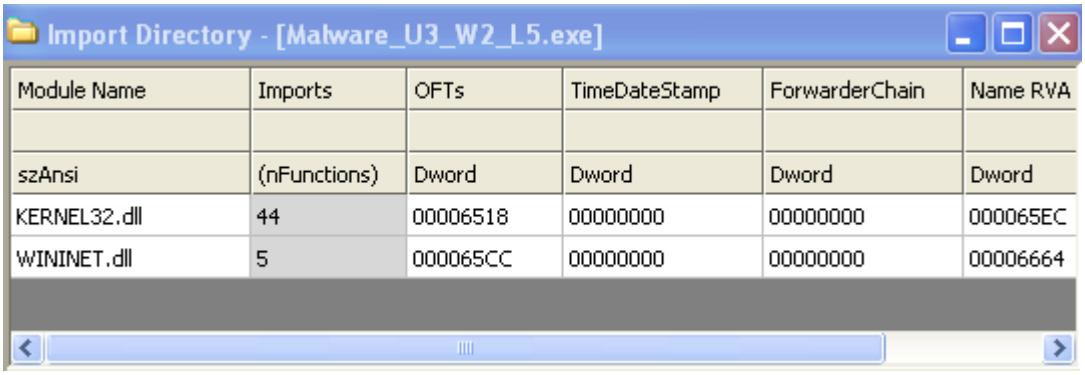
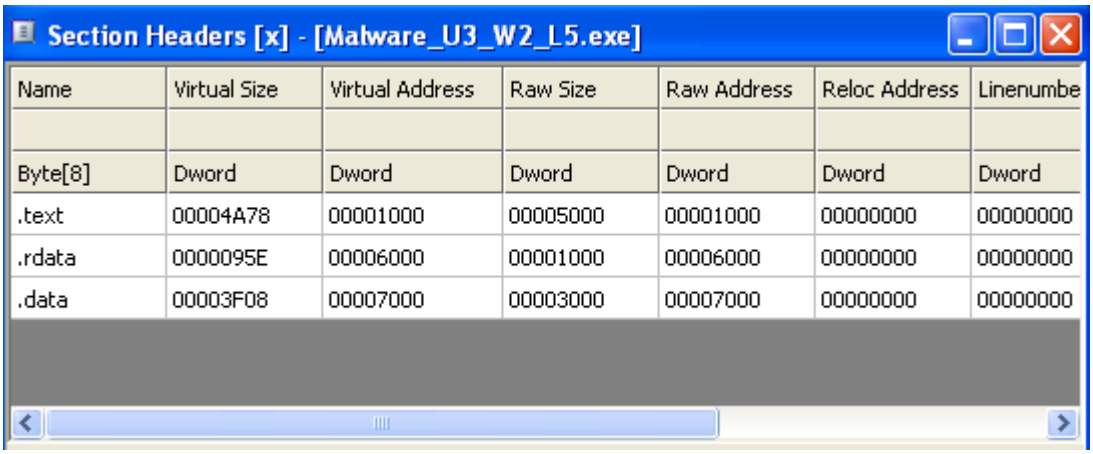
Oggi andremo ad effettuare un’analisi completa del malware scritto per le build week.

Per prima cosa effettueremo un’analisi statica, quindi andremo a capire se effettivamente si tratti di un malware e, dopo di ciò, lanceremo il malware e vedremo come si comporta e le sue azioni (analisi dinamica),infine ne andremo a studiare proprio il contenuto, avendo studiato il linguaggio assembly (analisi statica avanzata).

Effettuiamo l’analisi statica:

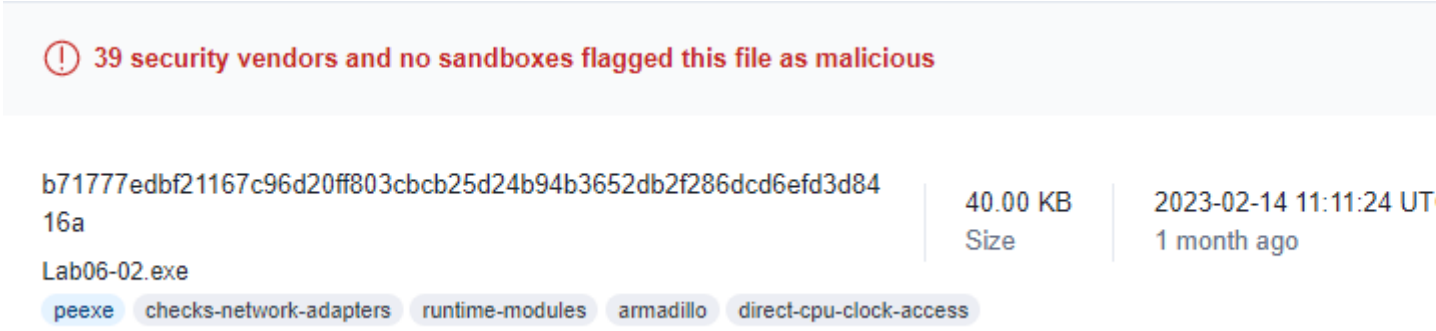






Section Headers [x] - [Malware_U3_W2_L5.exe]

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



39 security vendors and no sandboxes flagged this file as malicious

b71777edbf21167c96d20ff803cbcb25d24b94b3652db2f286dcd6efd3d8416a

Lab06-02.exe

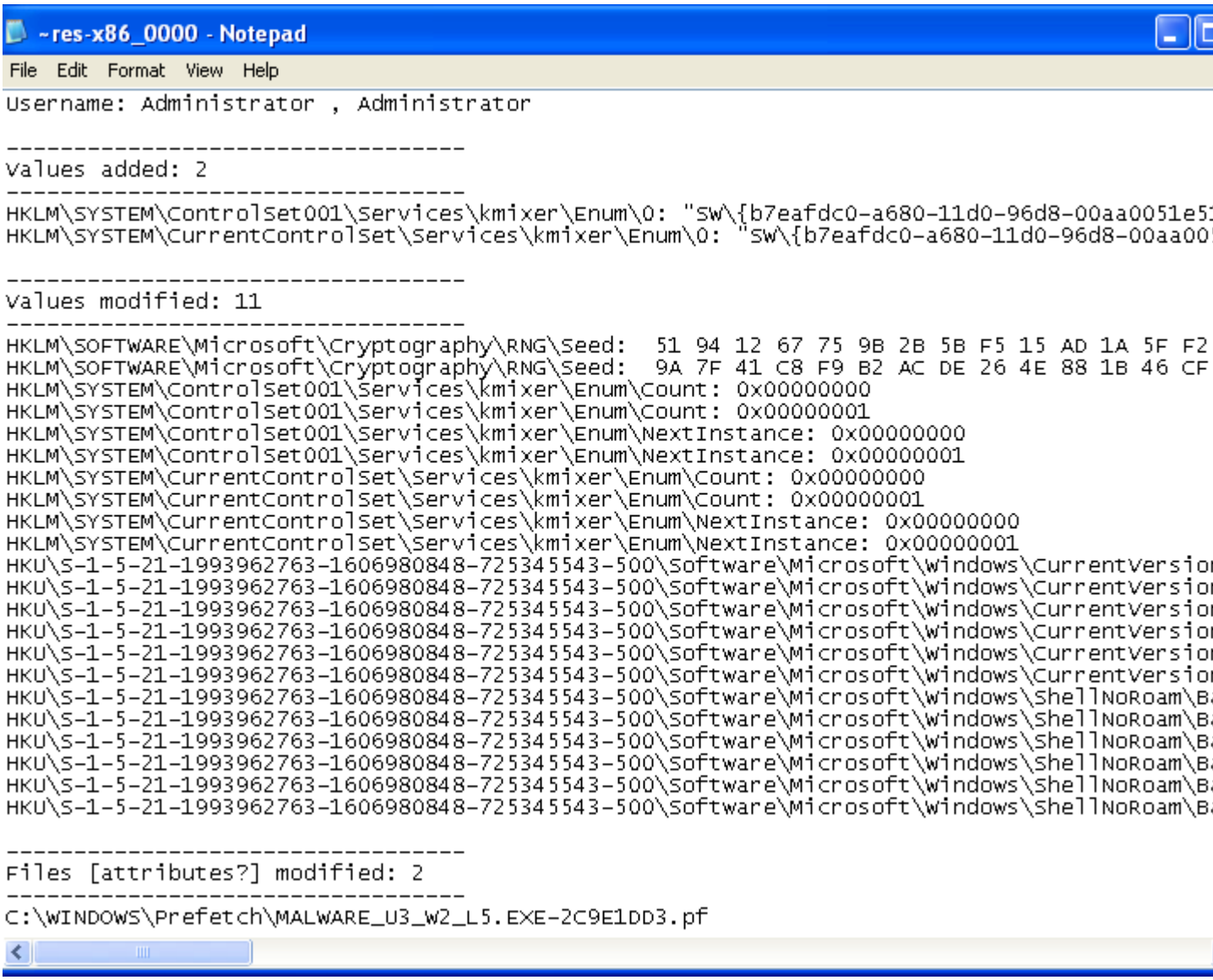
peexe checks-network-adapters runtime-modules armadillo direct-cpu-clock-access

40.00 KB
Size

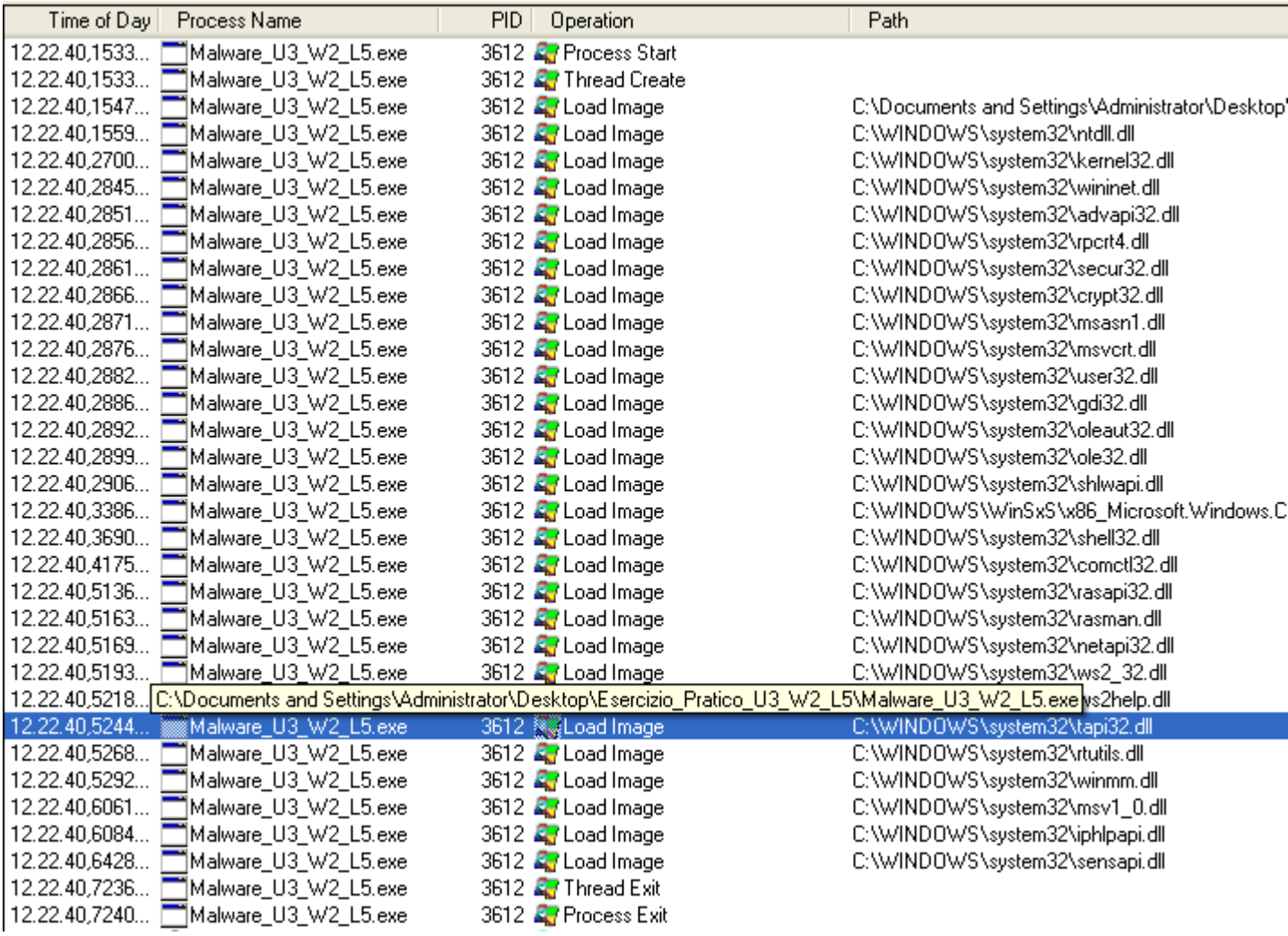
2023-02-14 11:11:24 UT
1 month ago

Come possiamo notare, prima abbiamo ricavato l'hash del file presunto malevolo, poi abbiamo controllato su virus total se fosse un malware gia noto: il risultato è positivo, in effetti è un malware noto e si tratta di un trojano. Appurato ciò ricaviamo le informazioni richieste dall'esercizio: le librerie chiamate: KERNEL32.dll, per interagire con il sitema operativo e la libreria WININET.dll, che viene usata principalmente per effettuare richieste https, http (Praticamente per avere accesso ad internet e magari rimandando la macchina "attaccata" ad un dominio o sottodominio malevolo), e le sezioni di cui è composto il malware: .rtext (con le righe di codice effettive del malware), .rdata (include le informazioni sulle librerie esportate), .data(contiene dati e variabili del programma).

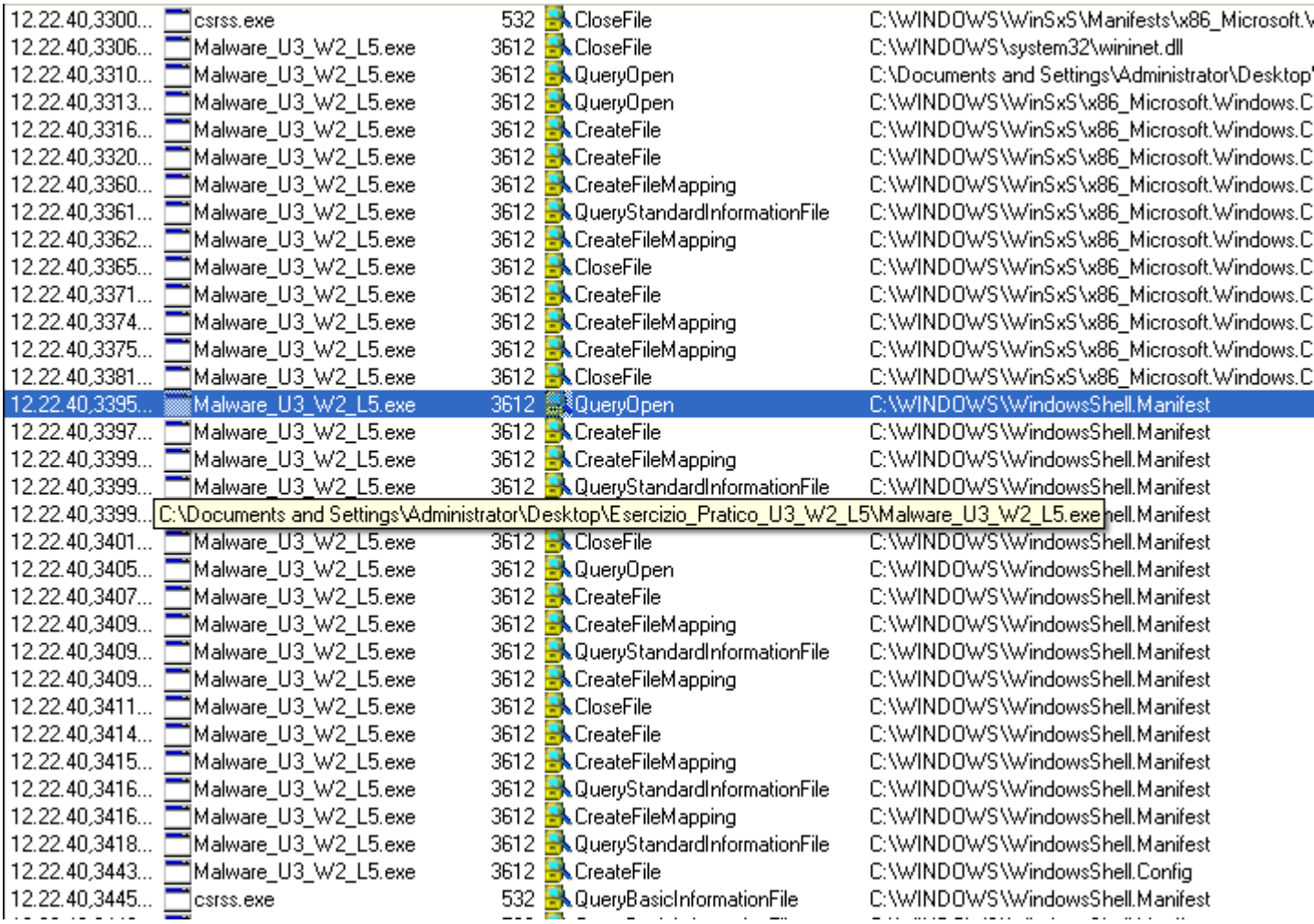
Ora effettueremo un'analisi dinamica, quindi lanceremo il malware e ne studieremo il comportamento:



```
Username: Administrator , Administrator

----------------------------------
Values added: 2
----------------------------------
HKLM\SYSTEM\ControlSet001\Services\kmixer\Enum\0: "SW\{b7eafdc0-a680-11d0-96d8-00aa0051e5
HKLM\SYSTEM\CurrentControlSet\Services\kmixer\Enum\0: "SW\{b7eafdc0-a680-11d0-96d8-00aa00

----------------------------------
Values modified: 11
----------------------------------
HKLM\SOFTWARE\Microsoft\Cryptography\RNG\Seed:  51 94 12 67 75 9B 2B 5B F5 15 AD 1A 5F F2
HKLM\SOFTWARE\Microsoft\Cryptography\RNG\Seed:  9A 7F 41 C8 F9 B2 AC DE 26 4E 88 1B 46 CF
HKLM\SYSTEM\ControlSet001\Services\kmixer\Enum\Count: 0x00000000
HKLM\SYSTEM\ControlSet001\Services\kmixer\Enum\Count: 0x00000001
HKLM\SYSTEM\ControlSet001\Services\kmixer\Enum\NextInstance: 0x00000000
HKLM\SYSTEM\ControlSet001\Services\kmixer\Enum\NextInstance: 0x00000001
HKLM\SYSTEM\CurrentControlSet\Services\kmixer\Enum\Count: 0x00000000
HKLM\SYSTEM\CurrentControlSet\Services\kmixer\Enum\Count: 0x00000001
HKLM\SYSTEM\CurrentControlSet\Services\kmixer\Enum\NextInstance: 0x00000000
HKLM\SYSTEM\CurrentControlSet\Services\kmixer\Enum\NextInstance: 0x00000001
HKU\S-1-5-21-1993962763-1606980848-725345543-500\Software\Microsoft\Windows\CurrentVersio
HKU\S-1-5-21-1993962763-1606980848-725345543-500\Software\Microsoft\Windows\CurrentVersio
HKU\S-1-5-21-1993962763-1606980848-725345543-500\Software\Microsoft\Windows\CurrentVersio
HKU\S-1-5-21-1993962763-1606980848-725345543-500\Software\Microsoft\Windows\CurrentVersio
HKU\S-1-5-21-1993962763-1606980848-725345543-500\Software\Microsoft\Windows\CurrentVersio
HKU\S-1-5-21-1993962763-1606980848-725345543-500\Software\Microsoft\Windows\CurrentVersio
HKU\S-1-5-21-1993962763-1606980848-725345543-500\Software\Microsoft\Windows\ShellNoRoam\B
HKU\S-1-5-21-1993962763-1606980848-725345543-500\Software\Microsoft\Windows\ShellNoRoam\B
HKU\S-1-5-21-1993962763-1606980848-725345543-500\Software\Microsoft\Windows\ShellNoRoam\B
HKU\S-1-5-21-1993962763-1606980848-725345543-500\Software\Microsoft\Windows\ShellNoRoam\B
HKU\S-1-5-21-1993962763-1606980848-725345543-500\Software\Microsoft\Windows\ShellNoRoam\B
HKU\S-1-5-21-1993962763-1606980848-725345543-500\Software\Microsoft\Windows\ShellNoRoam\B

----------------------------------
Files [attributes?] modified: 2
----------------------------------
C:\WINDOWS\Prefetch\MALWARE_U3_W2_L5.EXE-2C9E1DD3.pf
```



Time of Day	Process Name	PID	Operation	Path
12.22.40,1533...	Malware_U3_W2_L5.exe	3612	Process Start	
12.22.40,1533...	Malware_U3_W2_L5.exe	3612	Thread Create	
12.22.40,1547...	Malware_U3_W2_L5.exe	3612	Load Image	C:\Documents and Settings\Administrator\Desktop'
12.22.40,1559...	Malware_U3_W2_L5.exe	3612	Load Image	C:\WINDOWS\system32\ntdll.dll
12.22.40,2700...	Malware_U3_W2_L5.exe	3612	Load Image	C:\WINDOWS\system32\kernel32.dll
12.22.40,2845...	Malware_U3_W2_L5.exe	3612	Load Image	C:\WINDOWS\system32\wininet.dll
12.22.40,2851...	Malware_U3_W2_L5.exe	3612	Load Image	C:\WINDOWS\system32\advapi32.dll
12.22.40,2856...	Malware_U3_W2_L5.exe	3612	Load Image	C:\WINDOWS\system32\rpcrt4.dll
12.22.40,2861...	Malware_U3_W2_L5.exe	3612	Load Image	C:\WINDOWS\system32\secur32.dll
12.22.40,2866...	Malware_U3_W2_L5.exe	3612	Load Image	C:\WINDOWS\system32\crypt32.dll
12.22.40,2871...	Malware_U3_W2_L5.exe	3612	Load Image	C:\WINDOWS\system32\msasn1.dll
12.22.40,2876...	Malware_U3_W2_L5.exe	3612	Load Image	C:\WINDOWS\system32\msvcrt.dll
12.22.40,2882...	Malware_U3_W2_L5.exe	3612	Load Image	C:\WINDOWS\system32\user32.dll
12.22.40,2886...	Malware_U3_W2_L5.exe	3612	Load Image	C:\WINDOWS\system32\gdi32.dll
12.22.40,2892...	Malware_U3_W2_L5.exe	3612	Load Image	C:\WINDOWS\system32\oleaut32.dll
12.22.40,2899...	Malware_U3_W2_L5.exe	3612	Load Image	C:\WINDOWS\system32\ole32.dll
12.22.40,2906...	Malware_U3_W2_L5.exe	3612	Load Image	C:\WINDOWS\system32\shlwapi.dll
12.22.40,3386...	Malware_U3_W2_L5.exe	3612	Load Image	C:\WINDOWS\WinSxS\x86_Microsoft.Windows.C
12.22.40,3690...	Malware_U3_W2_L5.exe	3612	Load Image	C:\WINDOWS\system32\shell32.dll
12.22.40,4175...	Malware_U3_W2_L5.exe	3612	Load Image	C:\WINDOWS\system32\comctl32.dll
12.22.40,5136...	Malware_U3_W2_L5.exe	3612	Load Image	C:\WINDOWS\system32\rasapi32.dll
12.22.40,5163...	Malware_U3_W2_L5.exe	3612	Load Image	C:\WINDOWS\system32\rasman.dll
12.22.40,5169...	Malware_U3_W2_L5.exe	3612	Load Image	C:\WINDOWS\system32\netapi32.dll
12.22.40,5193...	Malware_U3_W2_L5.exe	3612	Load Image	C:\WINDOWS\system32\ws2_32.dll
12.22.40,5218...	C:\Documents and Settings\Administrator\Desktop\Esercizio_Pratico_U3_W2_L5\Malware_U3_W2_L5.exe			ws2help.dll
12.22.40,5244...	Malware_U3_W2_L5.exe	3612	Load Image	C:\WINDOWS\system32\tapi32.dll
12.22.40,5268...	Malware_U3_W2_L5.exe	3612	Load Image	C:\WINDOWS\system32\rtutils.dll
12.22.40,5292...	Malware_U3_W2_L5.exe	3612	Load Image	C:\WINDOWS\system32\winmm.dll
12.22.40,6061...	Malware_U3_W2_L5.exe	3612	Load Image	C:\WINDOWS\system32\msv1_0.dll
12.22.40,6084...	Malware_U3_W2_L5.exe	3612	Load Image	C:\WINDOWS\system32\iphlpapi.dll
12.22.40,6428...	Malware_U3_W2_L5.exe	3612	Load Image	C:\WINDOWS\system32\sensapi.dll
12.22.40,7236...	Malware_U3_W2_L5.exe	3612	Thread Exit	
12.22.40,7240...	Malware_U3_W2_L5.exe	3612	Process Exit	



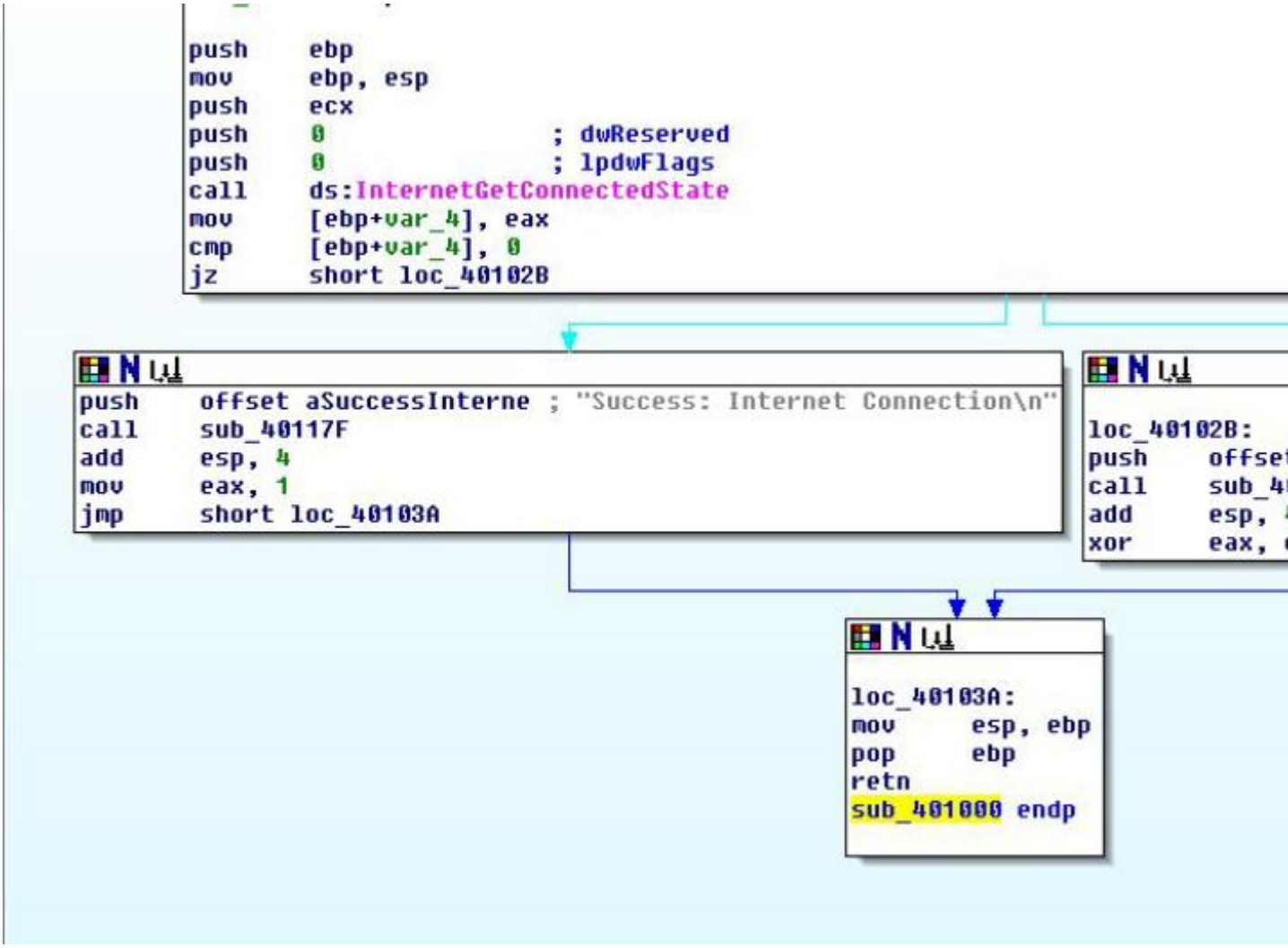
Per prima cosa abbiamo effettuato due screeenshot con REGSHOT, per vedere se il malware, una volta lanciato, cambiasse ed andasse ad intaccare le chiavi di registro della macchina attaccata. Il risultato è stato positivo (negativo per noi). Le chiavi di registro cambiate dal momento del lancio del malware sono in effetti 11.

Poi con procmon abbiamo analizzato il comportamento del malware sul fyle system e sui processi della macchina:

abbiamo notato che una volta lanciato, il malware crea diversi file all'interno di diverse cartelle, basta notare il path e nell'area riguardante i processi, l'azione più effettuata è "load image", che viene usato spesso dai malware per caricare il malware e le librerie che gli occorrono.

Ora andremo ad effettuare l'ultimo passaggio, l'analisi avanzata statica che, come abbiamo detto, analizza il contenuto del malware ed il linguaggio con cui è scritto, ovvero l'assembly.

il malware in questione da analizzare è il seguente:



Per prima cosa si crea lo stack, con le prime due azioni: si aggiunge allo stack l'indirizzo di memoria "ebp", che funge da 'bottom', quindi funge sempre da 'ultimo piatto' nella pila di piatti che, metaforicamente, contraddistingue lo stack, poi si copia il top (esp) nel bottom con il comando "mov".

Poi c'è la chiamata delle funzioni, in questo caso si chiama la funzione "internetgetconnecstate", che verifica con un if (che vedremo subito dopo), se si ha avuto accesso ad internet.

Il parametro dopo è, come abbiamo spiegato poco prima, un ciclo IF:

prima si copia il valore del registro eax nella variabile locale ebp + var_4, poi con il comando cmp(compare) si comparano i due valori (della variabile locale con 0) e, con il comando jz (che come sappiamo salta alla locazione di memoria (in questo caso loc_40102B) se il ZERO FLAG è 1 (e sappiamo che ZERO FLAG assume valore 1 se 'destinazione'='sorgente') salta alla locazione di memoria loc_40102B al verificarsi dell'IF, in questo caso se lo ZERO FLAG è 1, quindi, tradotto, se il valore della variabile locale è uguale a 0.

La parte dopo della funzione ci presenta le due condizioni dell'IF. Partiremo ad analizzare la parte di codice nella quale si prende in considerazione l'ipotesi che si abbia avuto accesso ad internet:

prima salva il valore nello del risultato dell'IF, quindi che si ha avuto successo con la connessione,nello stack, con il comando push.

Poi chiama il contenuto presente nell'indirizzo di memoria sub_40117F, poi aggiunge il valore di 4 al registro esp e copia il contenuto dal valore 1 nel registro eax, infine con il comando jmp salta alla locazione di memoria LOC4_0103° che analizzeremo in seguito.

Ora analizziamo la parte di codice dove si prende in considerazione l'altra eventualità dell'IF, ovvero che non ci connettiamo ad internet:

per prima cosa salva il valore dell'if nel caso non si raggiunga la connessione con il comando 'push', poi chiama il contenuto delll'indirizzo di memoria salvato in sub_40117F, aggiunge il valore 4 al registro esp ed infine inizializza a 0 il valore del registro eax con l'operatore logico xor, in quanto sappiamo che l'operatore logico xor inizializza sempre a 0 se i 2 parametri sono uguali, in questo caso eax=eax.

Poi, dopo aver analizzato le due casistiche seguite alll'IF, copia il contenuto del registro ebp (ultimo piatto) in esp (primo piatto) e poi lo rimuove dallo stack con il comando pop.

Per concludere, con la parte di codice studiata oggi, il malware identifica se la macchina si connette ad internet e, con l'if, studia le due casistiche in cui riusciamo e non riusciamo a farlo, poi rimuove il contenuto dallo stack.