

Progetto Modulo 6 - Alessio Russo

Malware Analysis

Analisi statica: Con riferimento al file eseguibile Malware_Build_Week_U3, rispondere ai seguenti quesiti utilizzando i tool e le tecniche apprese nelle lezioni teoriche:

- Quanti parametri sono passati alla funzione Main()?
- Quante variabili sono dichiarate all'interno della funzione Main()?
- Quali sezioni sono presenti all'interno del file eseguibile?
- Descrivete brevemente almeno 2 di quelle identificate
- Quali librerie importa il Malware?
- Per ognuna delle librerie importate, fate delle ipotesi sulla base della sola analisi statica delle funzionalità che il Malware potrebbe implementare. Utilizzate le funzioni che sono richiamate all'interno delle librerie per supportare le vostre ipotesi.

spiegare:

- Lo scopo della funzione chiamata alla locazione di memoria 00401021
- Come vengono passati i parametri alla funzione alla locazione 00401021;
- Che oggetto rappresenta il parametro alla locazione 00401017
- Il significato delle istruzioni comprese tra gli indirizzi 00401027 e 00401029.
- Con riferimento all'ultimo quesito, tradurre il codice Assembly nel corrispondente costrutto C.
- Valutate ora la chiamata alla locazione 00401047, qual è il valore del parametro «ValueName»?

Analisi dinamica: Eseguite il Malware

- Cosa notate all'interno della cartella dove è situato l'eseguibile del Malware?
Spiegate cosa è avvenuto, unendo le evidenze che avete raccolto finora per rispondere alla domanda
- Analizzare i risultati con procmon

Malware Analysis

Filtrate includendo solamente l'attività sul registro di Windows.

- Quale chiave di registro viene creata?
- Quale valore viene associato alla chiave di registro creata?

Passate ora alla visualizzazione dell'attività sul file system.

- Quale chiamata di sistema ha modificato il contenuto della cartella dove è presente l'eseguibile del Malware?

Unite tutte le informazioni raccolte fin qui sia dall'analisi statica che dall'analisi dinamica per delineare il funzionamento del Malware.

- Parametri: notiamo in modo immediato che i parametri passati alla funzione main() sono: argc, argv, envp questi possono essere notati sia alla definizione di funzione mai in blu che rispetto alle righe di codice evidenziate in grigio, in generale per convenzione IDA Pro identifica i parametri quelli con i valori positivi come da evidenze.

```
.text:00401100 ; int __cdecl main(int argc, const char **argv, const char **envp)
.text:00401100 _main          proc near          ; CODE XREF: start+AF↓p
.text:00401100
.text:00401100 hModule      = dword ptr -11Ch
.text:00401100 Data        = byte ptr -118h
.text:00401100 var_117     = byte ptr -117h
.text:00401100 var_8       = dword ptr -8
.text:00401100 var_4       = dword ptr -4
.text:00401100 argc        = dword ptr 8
.text:00401100 argv        = dword ptr 0Ch
.text:00401100 envp        = dword ptr 10h
.text:00401100
* .text:00401100      such      sha
```

- Variabili: per quanto riguarda le variabili si nota immediatamente che queste sono 5, in genere vengono identificate da quei caratteri che hanno valore negativo come da evidenziazione in grigio

```
.text:004011D0 ; int __cdecl main(int argc, const char **argv, const char **envp)
.text:004011D0 _main          proc near          ; CODE XREF: start+AF↓p
.text:004011D0
.text:004011D0 hModule      = dword ptr -11Ch
.text:004011D0 Data        = byte ptr -118h
.text:004011D0 var_117      = byte ptr -117h
.text:004011D0 var_8        = dword ptr -8
.text:004011D0 var_4        = dword ptr -4
.text:004011D0 argc         = dword ptr 8
.text:004011D0 argv         = dword ptr 0Ch
.text:004011D0 envp         = dword ptr 10h
.text:004011D0
.text:004011D0
```

- Le sezioni utilizzate dal malware possiamo verificarle sia con IDA pro che con CFF Explorer:
 - o IDA Pro: effettuando la visualizzazione sulla segmentation possiamo notare che le sezioni utilizzate dal malware sono:
 - .text: contiene le istruzioni (le righe di codice) che la CPU eseguirà una volta che il software sarà avviato
 - .idata: contiene le informazioni sull'importazione di funzioni da altre librerie dinamiche (DLL)
 - .rdata: include generalmente le informazioni circa le librerie e le funzioni importate ed esportate dall'eseguibile
 - .data: contiene tipicamente i dati / le variabili globali del programma eseguibile, che devono essere disponibili da qualsiasi parte del programma

Name	Start	End
.text	00401000	00407000
.idata	00407000	004070DC
.rdata	004070DC	00408000
.data	00408000	0040C000

- o Stessa cosa viene evidenziata attraverso CFF Explorer, il quale a differenza di IDA Pro mette in evidenza l'utilizzo di .rsrc: include le risorse utilizzate dall'eseguibile come ad esempio icone, immagini, menu e stringhe

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations N...	Linenumbers ...	Characteristics
000001D8	000001E0	000001E4	000001E8	000001EC	000001F0	000001F4	000001F8	000001FA	000001FC
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword
.text	00005646	00001000	00006000	00001000	00000000	00000000	0000	0000	60000020
.rdata	000009AE	00007000	00001000	00007000	00000000	00000000	0000	0000	40000040
.data	00003EAB	00008000	00003000	00008000	00000000	00000000	0000	0000	C0000040
.rsrc	00001A70	0000C000	00002000	0000B000	00000000	00000000	0000	0000	40000040

- Import Librerie: come sopra possiamo vedere il risultato sia attraverso IDA Pro che attraverso CFF Explorer: Entrambi i software mettono in evidenza l'utilizzo di due librerie quali:
 - o advapi32.dll: fornisce una serie di funzioni per l'accesso ai servizi di sicurezza avanzati e per la gestione delle autorizzazioni di sistema
 - o kernel32.dll: fornisce una serie di funzionalità di basso livello necessarie per il corretto funzionamento del sistema operativo e delle applicazioni

- Librerie importate con CFF Explorer

CFF Explorer VIII - [Malware_Build_Week_U3.exe]

File Settings ?

Malware_Build_Week_U3.exe

Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.dll	51	00007534	00000000	00000000	0000769E	0000700C
ADVAPI32.dll	2	00007528	00000000	00000000	000076D0	00007000

File: Malware_Build_Week_U3.exe

- File Header
- Optional Header
- Data Directories [x]
- Section Headers [x]
- Import Directory

- Librerie importate con IDA Pro: Da queste possiamo notare anche che il malware in questione è un dropper rispetto alle operazioni quali:
 - o CreateFileA: utilizzata per creare o aprire un file
 - o SizeofResource: utilizzata per recuperare la dimensione, in byte, di una risorsa all'interno di un file eseguibile o di una DLL
 - o LockResources: utilizzata per ottenere un puntatore al contenuto di una risorsa all'interno di un file eseguibile o di una DLL

Program Segmentation | IDA View-A | Hex View-A | Structures | Enums | Imports | Exports

Name	Start	End	Address	Ordinal	Name	Library
.text	00401000	00407000	00407000		RegSetValueExA	ADVAPI32
.idata	00407000	004070DC	00407004		RegCreateKeyExA	ADVAPI32
.rdata	004070DC	00408000	00407074		GetStartupInfoA	KERNEL32
.data	00408000	0040C000	00407078		GetEnvironmentVariableA	KERNEL32
			0040707C		GetVersionExA	KERNEL32
			00407080		HeapDestroy	KERNEL32
			00407084		HeapCreate	KERNEL32
			00407088		VirtualFree	KERNEL32
			0040708C		RtlUnwind	KERNEL32
			00407090		HeapAlloc	KERNEL32
			00407094		HeapReAlloc	KERNEL32
			00407098		SetStdHandle	KERNEL32
			0040709C		FlushFileBuffers	KERNEL32
			004070A0		SetFilePointer	KERNEL32
			004070A4		CreateFileA	KERNEL32
			004070A8		GetCPInfo	KERNEL32
			004070...		GetACP	KERNEL32
			004070B0		GetOEMCP	KERNEL32
			004070B4		GetProcAddress	KERNEL32
			004070B8		LoadLibraryA	KERNEL32
			004070...		SetEndOfFile	KERNEL32
			004070C0		ReadFile	KERNEL32
			004070C4		MultiByteToWideChar	KERNEL32
			004070C8		LCMapStringA	KERNEL32
			004070...		LCMapStringW	KERNEL32
			004070...		GetStringTypeA	KERNEL32
			004070...		GetStringTypeW	KERNEL32
			004070DC		SizeofResource	KERNEL32
			00407010		LockResource	KERNEL32

Possiamo notare l'utilizzo di alcuni registri specifici come RegSetValueExA, RegCreateKeyExA entrambi registri della libreria advapi32.dll

RegSetValueExA: viene utilizzata per impostare il valore di un dato all'interno del registro di sistema - advapi32.dll

Program Segmentation | IDA View-A | Hex View-A | Structures | Enums | Imports | Exports

Name	Start	End	Address	Ordinal	Name	Library
.text	00401000	00407000	00407000		RegSetValueExA	ADVAPI32
.idata	00407000	004070DC	00407004		RegCreateKeyExA	ADVAPI32
.rdata	004070DC	00408000	00407074		GetStartupInfoA	KERNEL32

RegCreateKeyExA: funzione utilizzata per creare una nuova chiave o aprire una chiave esistente del registro di sistema di Windows - advapi32.dll

Name	Start	End	Address	Ordinal	Name	Library
.text	00401000	00407000	00407000		RegSetValueExA	ADVAPI32
.idata	00407000	004070DC	00407004		RegCreateKeyExA	ADVAPI32
...	004070DC	00400000	00407004		...	KERNEL32

Per quanto riguarda la libreria kernel32.dll è immediato notare le operazioni quali CreateFileA, WriteFile. CreateFileA: utilizzata per creare o aprire file - kernel32.dll

00407030	SetCurrentDir	KERNEL32
0040709C	FlushFileBuffers	KERNEL32
004070A0	SetFilePointer	KERNEL32
004070A4	CreateFileA	KERNEL32
004070A8	GetCPLInfo	KERNEL32
004070...	GetACP	KERNEL32

WriteFile: Questa funzione viene utilizzata per scrivere dati in un file - kernel32.dll

00407038	ExitProcess	KERNEL32
0040703C	HeapFree	KERNEL32
00407040	GetLastError	KERNEL32
00407044	WriteFile	KERNEL32
00407048	TerminateProcess	KERNEL32

- Lo scopo della funzione chiamata alla locazione di memoria 00401021:

The screenshot shows the IDA Pro interface. On the left, the 'Functions window' lists functions like sub_401000, sub_401080, main, sub_401299, fclose, and fwrite. The main window displays assembly code for the 'main' function. The code includes several 'push' instructions for parameters: 0, 0, 0, offset SubKey, 'SOFTWARE\H...', and 80000002h. At address 00401021, there is a 'call ds:RegCreateKeyExA' instruction. A 'Jump to address' dialog box is open, showing the address 00401021 in the 'Jump address' field. The dialog has 'OK', 'Cancel', and 'Help' buttons.

La funzione RegCreateKey è una funzione utilizzata per creare una nuova chiave o aprire una chiave esistente del registro di sistema di Windows. Questa funzione viene comunemente utilizzata per interagire con il registro di sistema al fine di memorizzare e recuperare informazioni di configurazione, impostazioni dell'applicazione, per l'installazione di software

- **Come vengono passati i parametri alla funzione alla locazione 00401021**

I parametri vengono passati tramite le push evidenziate: offset Subkey e 80000002h (hKey):

- o hKey: handle della chiave di registro principale con cui si desidera interagire, ricordando che un handle è un oggetto che punta a processi, file o risorse che sono state aperte o create all'interno del sistema operativo

Notiamo anche che in base al percorso del file il malware ha acquistato la persistenza all'interno del sistema operativo. La persistenza consiste nell'aggiungere il malware alle voci di avvio del sistema in modo che venga eseguito automaticamente ogni volta che il sistema operativo viene avviato.

__amsg_exit	.text:00401013	push	0	; lpClass
__fast_error_exit	.text:00401015	push	0	; Reserved
__stbuf	.text:00401017	push	offset SubKey	; "SOFTWARE\\Microsoft\\Windows NT\\CurrentVe..."
__ftbuf	.text:0040101C	push	8000002h	; hKey
sub_401679	.text:00401021	call	ds:RegCreateKeyExA	
write_char	.text:00401027	test	eax, eax	

- Che oggetto rappresenta il parametro alla locazione 00401017

__amsg_exit	.text:00401013	push	0	; lpClass
__fast_error_exit	.text:00401015	push	0	; Reserved
__stbuf	.text:00401017	push	offset SubKey	; "SOFTWARE\\Microsoft\\Windows NT\\CurrentVe..."
__ftbuf	.text:0040101C	push	8000002h	; hKey
sub_401679	.text:00401021	call	ds:RegCreateKeyExA	
write_char	.text:00401027	test	eax, eax	

Il parametro alla locazione **00401017** rappresenta l'offset dell'etichetta SubKey.

SubKey rappresenta il percorso relativo o assoluto della chiave del registro che il malware ha creato. Questo percorso specifica la posizione della chiave all'interno del registro di sistema, quindi il valore passato come parametro alla locazione 00401017 è l'indirizzo di memoria in cui è memorizzato il percorso della chiave del registro che si desidera creare o aprire.

__stbuf	.text:00401017	push	offset SubKey	; "SOFTWARE\\Microsoft\\Windows NT\\CurrentVe..."
__ftbuf	.text:0040101C	push	8000002h	; hKey
sub_401679	.text:00401021	call	ds:RegCreateKeyExA	
write_char	.text:00401027	test	eax, eax	
write_multi_char	.text:00401029	iz	short loc_401032	
write_string	.text:0040102B	mov	eax, 1	
	.text:0040102B	inc	eax	

Queste istruzioni controllano se la chiamata alla funzione **RegCreateKeyExA** ha avuto successo o meno, controllando se il valore restituito è zero. Se la chiamata ha avuto successo il programma continua normalmente. Se la chiamata ha fallito (valore restituito zero) il programma salta a loc_401032 per gestire l'errore.

- Con riferimento all'ultimo quesito, tradurre il codice Assembly nel corrispondente costruito C.

```
if (eax == 0) {
    goto loc_401032;
}
```

- Valutate ora la chiamata alla locazione 00401047, qual è il valore del parametro «ValueName»?

__write_char	.text:0040103E	push	0	; reserved
__write_multi_char	.text:0040103E	push	offset ValueName	; "GinaDLL"
__write_string	.text:00401043	mov	eax, [ebp+hObject]	
__get_int_arg	.text:00401046	push	eax	; hKey
__get_int64_arg	.text:00401047	call	ds:RegSetValueExA	
__get_int64_arg	.text:0040104D	test	eax, eax	
__get_int64_arg	.text:0040104F	iz	short loc_4010A2	

- ValueName: 0040103E = GinaDLL

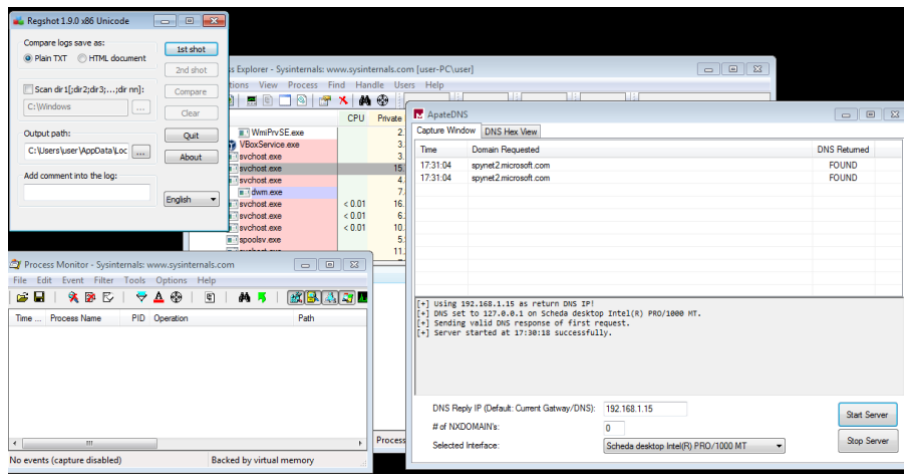
sub_401679	.text:0040103H	push	1	; dwType
__write_char	.text:0040103C	push	0	; Reserved
__write_multi_char	.text:0040103E	push	offset ValueName	; "GinaDLL"
__write_string	.text:00401043	mov	eax, [ebp+hObject]	
__get_int_arg	.text:00401046	push	eax	; hKey
__get_int64_arg	.text:00401047	call	ds:RegSetValueExA	
__get_int64_arg	.text:0040104D	test	eax, eax	

- **RegSetValueEx**: questa funzione permette invece di aggiungere un nuovo valore all'interno del registro e di settare i rispettivi dati: in questo caso possiamo notare dalle push precedenti che aggiungerà **ValueName; GinaDLL**

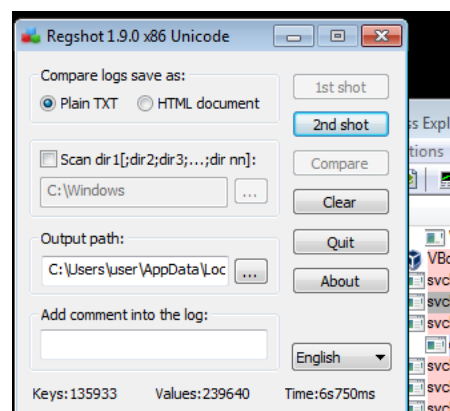
ANALISI DINAMICA

Un potenziale approccio da adottare potrebbe essere:

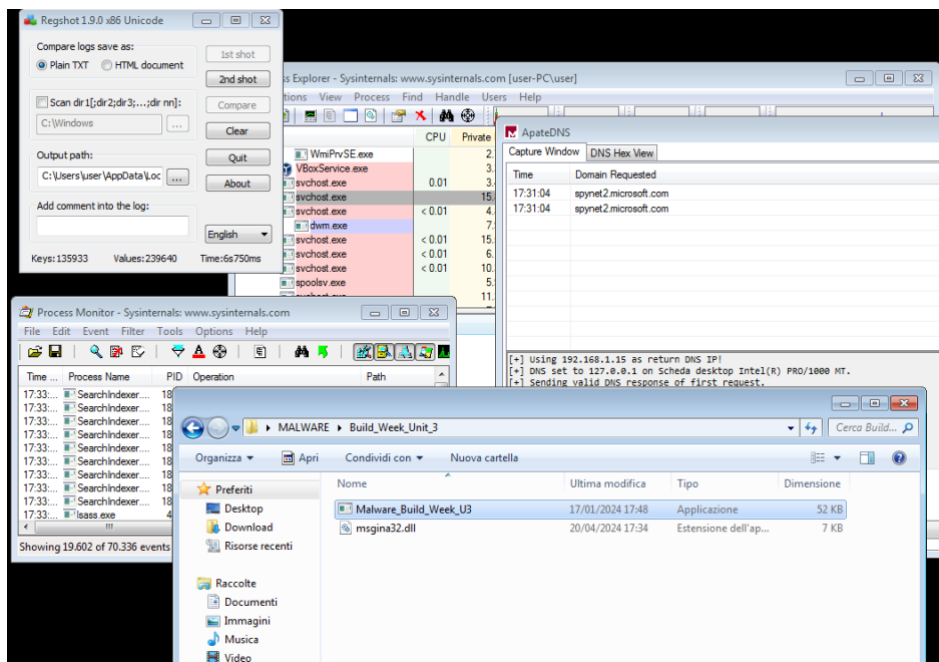
- Process Explorer (ci consente di capire quali sono i processi che la macchina sta utilizzando)
- ApateDNS (Serve per capire se qualcosa viene richiamato sulla parte DNS)
- RegShot (Serve a capire la situazione prima e post avvio malware)
- Procmon

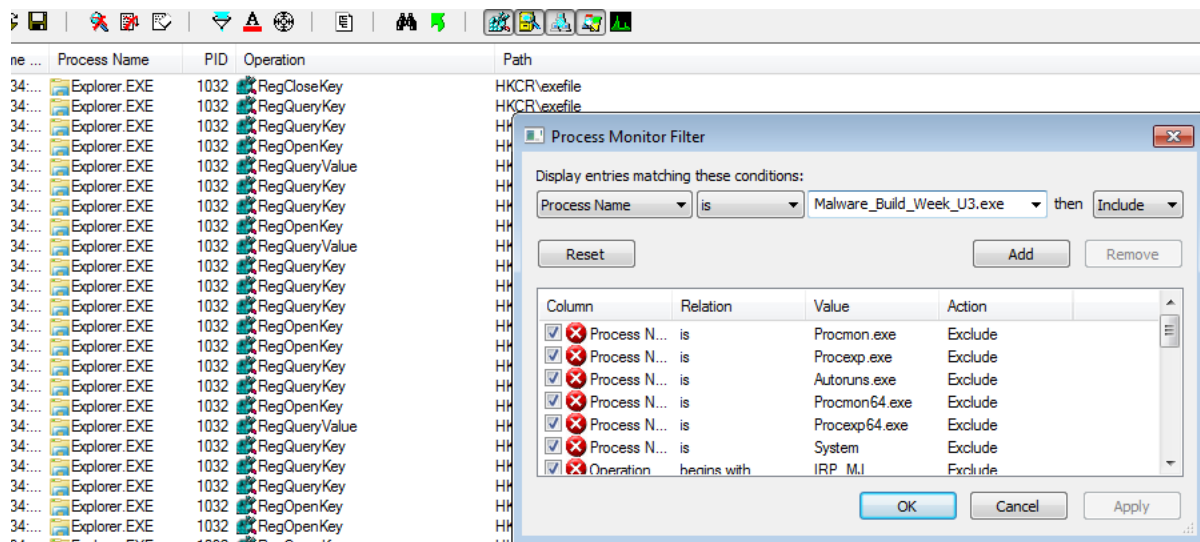


- **Regshot:** questo tool ci permette di effettuare una istantanea delle chiavi di registro per consentirci di visualizzare le chiavi di registro prima e dopo l'esecuzione del malware in modo da notare le modifiche effettuate da quest'ultimo



- Dopo aver effettuato un primo screen con Regshot procedo all'esecuzione del malware, prima avvio la cattura con procmon il quale ci consente monitorare in tempo reale l'attività del sistema operativo Windows, inclusi i processi in esecuzione, le operazioni sui file e le attività del registro di sistema





- Effettuata la cattura, dopo aver stoppato la registrazione delle attività, filtro il processo di cui ho interesse verificare le operazioni eseguite
- Filtrando per nome processo posso successivamente applicare altri filtri su operazione che ha effettuato sui registri:
in questo caso ricade l'attenzione sulle voci selezionate in blu in basso, nello specifico per le operazioni di:
 - o RegCreateKey: utilizzata per creare una nuova sottochiave o aprire una sottochiave esistente
 - o RegSetInfoKey: utilizzata per impostare le informazioni di configurazione per una chiave specifica nel registro di sistema
 - o RegQueryKey: utilizzata per recuperare informazioni sulle chiavi di registro, come il numero di sottochiavi e il numero di valori associati a una chiave specificata
 - o RegSetValue: utilizzata per impostare il valore di una chiave del registro di sistema dove vediamo il valore del registro creato GinaDLL

17:34...	Malware_Build_Week_U3.exe	2624	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Diagnostics	NAME NOT FOUND Desired Access: R...
17:34...	Malware_Build_Week_U3.exe	2624	RegQueryKey	HKLM	SUCCESS Query: HandleTag...
17:34...	Malware_Build_Week_U3.exe	2624	RegCreateKey	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS Desired Access: A...
17:34...	Malware_Build_Week_U3.exe	2624	RegSetInfoKey	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS KeySetInformation
17:34...	Malware_Build_Week_U3.exe	2624	RegQueryKey	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS Query: HandleTag...
17:34...	Malware_Build_Week_U3.exe	2624	RegSetValue	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon\GinaDLL	ACCESS DENIED Type: REG_SZ Le...
17:34...	Malware_Build_Week_U3.exe	2624	RegCloseKey	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS
17:34...	Malware_Build_Week_U3.exe	2624	RegOpenKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Uninstall\Windows File Execution Options	SUCCESS

- Possiamo ancora effettuare un filtro sulle attività effettuate su FileSystem: in questo caso ricade l'attenzione sulle voci selezionate in blu in basso, nello specifico per le operazioni:
 - o CreateFile: utilizzata per creare o aprire un file o un dispositivo
 - o WriteFile: utilizzata per scrivere dati in un file
 - o CloseFile(coperto): chiudere un handle aperto precedentemente tramite CreateFile o altre funzioni

Il file creato sarà msgina32.dll

17:34...	Malware_Build_Week_U3.exe	2624	CloseFile	C:\Windows\SysWow64\eechost.dll	SUCCESS
17:34...	Malware_Build_Week_U3.exe	2624	CreateFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll	SUCCESS Desired Access: G...
17:34...	Malware_Build_Week_U3.exe	2624	WriteFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll	SUCCESS Offset: 0, Length: 4...
17:34...	Malware_Build_Week_U3.exe	2624	WriteFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll	SUCCESS Offset: 4,096, Leng...
17:34...	Malware_Build_Week_U3.exe	2624	WriteFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll	SUCCESS
17:34...	Malware_Build_Week_U3.exe	2624	QueryNameInformationFile	C:\Windows\System32\api-ms-win-base.dll	SUCCESS Name: \Windows\...
17:34...	Malware_Build_Week_U3.exe	2624	QueryNameInformationFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\Malware_Build_Week_U3.exe	SUCCESS Name: \Users\user...
17:34...	Malware_Build_Week_U3.exe	2624	QueryNameInformationFile	C:\Windows\System32\wow64cpu.dll	SUCCESS Name: \Windows\...

- Analisi delle modifiche sulle chiavi di registro con Regshot

```
Regshot 1.9.0 x86 unicode
Comments:
Datetime: 2024/4/20 15:32:35 , 2024/4/20 15:48:01
Computer: USER-PC , USER-PC
Username: user , user

-----
Keys deleted: 1
-----
HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Microsoft\windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012

-----
Keys added: 2
-----
HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Microsoft\windows\CurrentVersion\Explorer\RecentDocs\,dll
HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Microsoft\windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012

-----
Values deleted: 5
-----
HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Microsoft\windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012
HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Microsoft\windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012
HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Microsoft\windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012
HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Microsoft\windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012
HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Microsoft\windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012

-----
Values added: 14
-----
HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Microsoft\windows\CurrentVersion\Explorer\FileExts\,dll\OpenWithList\,a: "CFF Explorer.e
HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Microsoft\windows\CurrentVersion\Explorer\FileExts\,dll\OpenWithList\,a: "a"
HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Microsoft\windows\CurrentVersion\Explorer\RecentDocs\17: 6D 00 73 00 67 00 69 00 6E 00
HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Microsoft\windows\CurrentVersion\Explorer\RecentDocs\18: 42 00 75 00 69 00 6C 00 64 00
HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Microsoft\windows\CurrentVersion\Explorer\RecentDocs\Folder\2: 42 00 75 00 69 00 6C 00
HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Microsoft\windows\CurrentVersion\Explorer\RecentDocs\,dll\0: 6D 00 73 00 67 00 69 00 6E
HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Microsoft\windows\CurrentVersion\Explorer\RecentDocs\,dll\,a: 00 00 00 00 FF FF
HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Microsoft\windows\CurrentVersion\Explorer\UserAssist\{CEBFF5CD-ACE2-4F4F-9178-9926F4174
HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Microsoft\windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012
HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Microsoft\windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012
HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Microsoft\windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012
HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Microsoft\windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012
HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Microsoft\windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012
HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Microsoft\windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012
HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\SysInternals\Process Monitor\PropertySheetDialog: 2C 00 00 00 00 00 00 01 00 00 00
-----
Values modified: 32
```

Concludendo possiamo dire che il malware con le operazioni sui registri e nello specifico evidenziate in precedenza e rispetto al suo comportamento nell'analisi dinamica acquista la persistenza come evidenziato alla funzione 401021. Inoltre, analizzando l'import delle librerie possiamo dire che si tratta di un dropper ossia un malware che si esegue per estrarre il malware contenuto al suo interno per salvarlo sul disco (msgina32.dll).