

# TEST FINALE MODULO M6



Andrea Ferrantino

## Traccia

### Malware Analysis

Il Malware da analizzare è nella cartella Build\_Week\_Unit\_3 presente sul desktop della macchina virtuale dedicata.

#### 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.

### Malware Analysis

Con riferimento al Malware in analisi, spiegare:

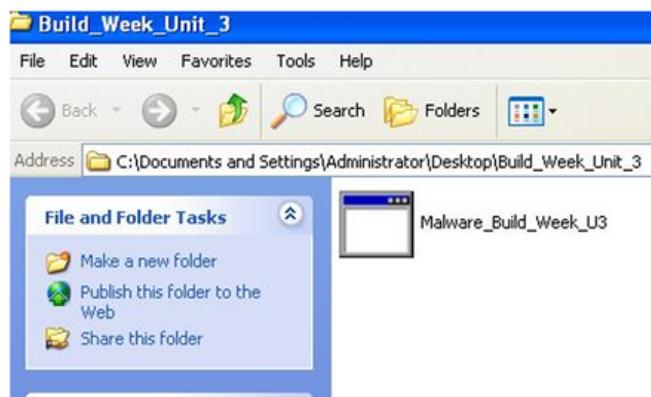
- Lo scopo della funzione chiamata alla locazione di memoria **00401021**
- Come vengono passati i parametri alla funzione alla locazione **00401021**;
- Che oggetto rappresenta il parametro alla locazione **00401017**
- Il significato delle istruzioni comprese tra gli indirizzi **00401027** e **00401029**.
- 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»?



## Malware Analysis

### Analisi dinamica

Preparate l'ambiente ed i tool per l'esecuzione del Malware (suggerimento: avviate principalmente Process Monitor ed assicurate di eliminare ogni filtro cliccando sul tasto «reset» quando richiesto in fase di avvio). Eseguite il Malware, facendo doppio click sull'icona dell'eseguibile

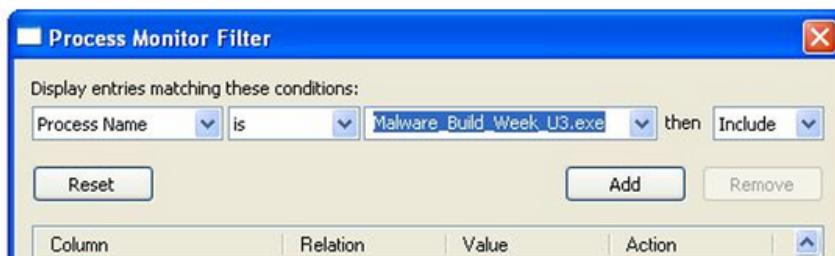


4

## Malware Analysis

- 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

Analizzate ora i risultati di Process Monitor (consiglio: utilizzate il filtro come in figura sotto per estrarre solo le modifiche apportate al sistema da parte del Malware). Fate click su «ADD» poi su «Apply» come abbiamo visto nella lezione teorica.



## 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.

# Introduzione

Abbiamo diversi tipi di analisi:

- Analisi statica: Essa utilizza tecniche e strumenti per carpire il comportamento del software malevolo senza attivarlo, fornendo informazioni sulle minacce
- Analisi dinamica: Osserva il comportamento del malware sotto copertura scoprendo quali dati modifica e come agisce sul sistema

Esse si suddividono in ulteriori 2 livelli

- Analisi basica: Esegue il malware in un ambiente sandbox per osservarne il comportamento e tentare di neutralizzarlo.
- Analisi avanzata: Identifica il comportamento del malware analizzando le sue istruzioni, essa impiega debugger per monitorare lo stato del programma durante l'esecuzione, raccogliendo informazioni dettagliate sulle sue azioni e interazioni con il sistema.

# Procedimento

## Analisi Statica

Utilizzando IDA Pro, possiamo notare che i parametri nella funzione main() sono tre, un int e due char, e il numero delle variabili dichiarate sono cinque

The screenshot shows the IDA Pro interface with the following details:

- Functions window:** Shows a list of functions and their addresses:
  - sub\_401000
  - sub\_401080
  - sub\_main
  - sub\_401299
  - \_fclose
  - \_fwrite
  - \_fopen
  - \_fopen
  - \_strchr
  - \_start
  - \_amsg\_exit
  - \_fast\_error\_exit
  - \_stbuf
- Assembly View:** Displays the assembly code for the main function:

```
; int __cdecl main(int argc, const char **argv, const char **envp)
main proc near

hModule= dword ptr -11Ch
Data= byte ptr -118h
var_117= byte ptr -117h
var_8= dword ptr -8
var_4= dword ptr -4
argc= dword ptr 8
argv= dword ptr 0Ch
envp= dword ptr 10h

push    ebp
mov     ebp, esp
sub    esp, 11Ch
push    ebx
push    esi
```

- .text è una sezione fondamentale di un file eseguibile PE di Windows. Contiene il codice macchina effettivo che viene eseguito dal processore quando il programma viene avviato. La sezione .text può essere utilizzata per comprendere il funzionamento di un programma e per identificare potenziali vulnerabilità di sicurezza
- rdata è una sezione importante del file eseguibile che contiene dati in sola lettura utilizzati dal programma durante l'esecuzione. I dati nella sezione ".rdata" possono includere costanti di programma, tabelle di dati, risorse del programma e informazioni sul debug.

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations N...	Linenumbers ...	Characteristics
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	00003EA8	00008000	00003000	00008000	00000000	00000000	0000	0000	C0000040
.rsrc	00001A70	0000C000	00002000	0000B000	00000000	00000000	0000	0000	40000040

Le librerie importate dal malware sono la KERNEL32 e ADVAPI32;

IDA - C:\Users\user\Desktop\MALWARE\Build\_Week\_Unit\_3\Malware\_Build\_Week\_U3.exe

File Edit Search View Debugger Options Windows Help

Imports

No debugger

Functions window

Function name	Seg...
_sopen	.text
_getbuf	.text
_lseek	.text
_commit	.text
_get_osfhandle	.text
_free_osfhandle	.text
_set_osfhandle	.text
_alloc_osfhandle	.text
sub_404933	.text
sub_40486F	.text
sub_404667	.text
sub_404622	.text
sub_4045CB	.text

IDA ViewA | Hex ViewA | Structures | Enums | Imports | Exports

Address	Ordinal	Name	Library
0000000000407000		RegSetValueExA	ADVAPI32
0000000000407004		RegCreateKeyExA	ADVAPI32
000000000040700C		SizeofResource	KERNEL32
0000000000407010		LockResource	KERNEL32
0000000000407014		LoadResource	KERNEL32
0000000000407018		VirtualAlloc	KERNEL32
000000000040701C		GetModuleFileNameA	KERNEL32
0000000000407020		GetModuleHandleA	KERNEL32
0000000000407024		FreeResource	KERNEL32
0000000000407028		FindResourceA	KERNEL32
000000000040702C		CloseHandle	KERNEL32
0000000000407030		GetCommandLineA	KERNEL32
0000000000407034		GetVersion	KERNEL32
0000000000407038		ExitProcess	KERNEL32

Line 85 of 117 Line 1 of 53

Output window

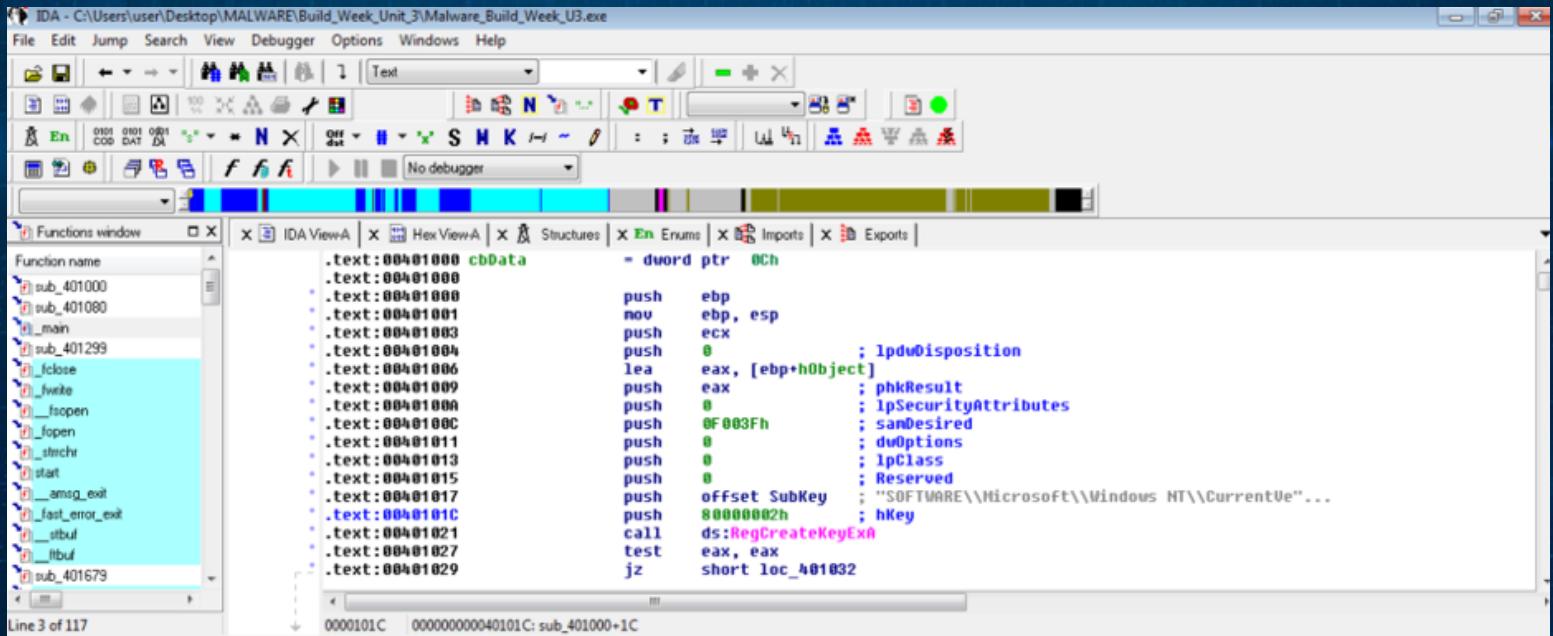
in base alle funzioni richiamate all'interno delle librerie posso ipotizzare che si tratta di un dropper, ovvero un programma malevolo che al suo interno contiene un malware.

I dropper rappresentano una tipologia di malware con caratteristiche ben precise che li distinguono da altri tipi di minacce informatiche. La loro funzione primaria è quella di distribuire e attivare altri malware all'interno di un sistema infetto. Per compiere questo compito, i dropper sfruttano diverse tecniche, tra cui l'utilizzo di specifiche API per estrarre il malware contenuto al loro interno.

# Le API chiave per l'estrazione del malware:

- FindResource(): Questa funzione permette di identificare la risorsa contenente il malware all'interno del file eseguibile del dropper
- LoadResource(): Una volta individuata la risorsa, la carica in memoria, rendendola disponibile per l'estrazione
- LockResource(): Questa funzione blocca la risorsa in memoria, impedendo che venga modificata o sovrascritta durante il processo di estrazione.
- SizeOfResource: Determina la dimensione della risorsa contenente il malware, garantendo che venga estratta la quantità corretta di dati.

La funzione chiamata all'indirizzo di memoria 00401021 sembra essere responsabile della creazione di una chiave di registro, e utilizza la funzione RegCreateKeyExA della libreria ADVAPI32.DLL.



The screenshot shows the IDA Pro interface with the assembly view selected. The assembly code for the function `sub_401000` is as follows:

```
.text:00401000 cbData        = dword ptr 0Ch
...
.push    ebp
.mov     ebp, esp
.push    ecx
.push    0          ; lpdwDisposition
.lea     eax, [ebp+hObject]
.push    eax         ; phkResult
.push    0          ; lpSecurityAttributes
.push    0F003Fh    ; sanDesired
.push    0          ; dwOptions
.push    0          ; lpClass
.push    0          ; Reserved
.push    offset SubKey    ; "SOFTWARE\Microsoft\Windows NT\CurrentVersion"
.push    80000002h    ; hKey
.call    ds:RegCreateKeyExA
.test   eax, eax
.jz     short loc_401032
```

The assembly code uses the `RegCreateKeyExA` function from the `ADVAPI32.DLL` library. The parameters passed to this function are:

- `hKEY`
- `SubKey`
- `dwFlags (dwOptions)`
- `lpSecurityAttributes`
- `Reserved`

I parametri passati dalla funzione sono:

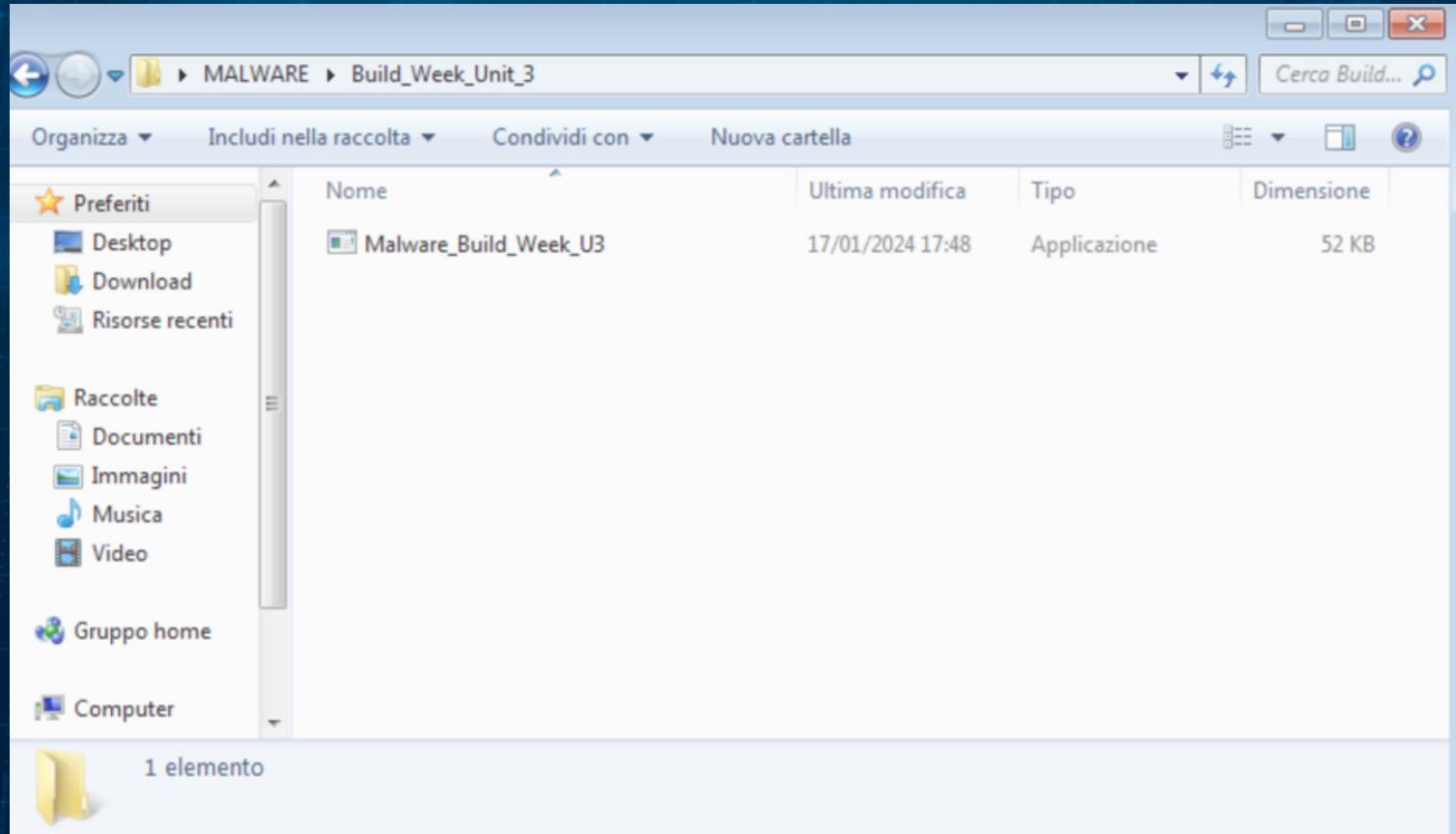
- `hKEY`
- `SubKey`
- `dwFlags (dwOptions)`
- `lpSecurityAttributes`
- `Reserved`

- ipdwDisposition:

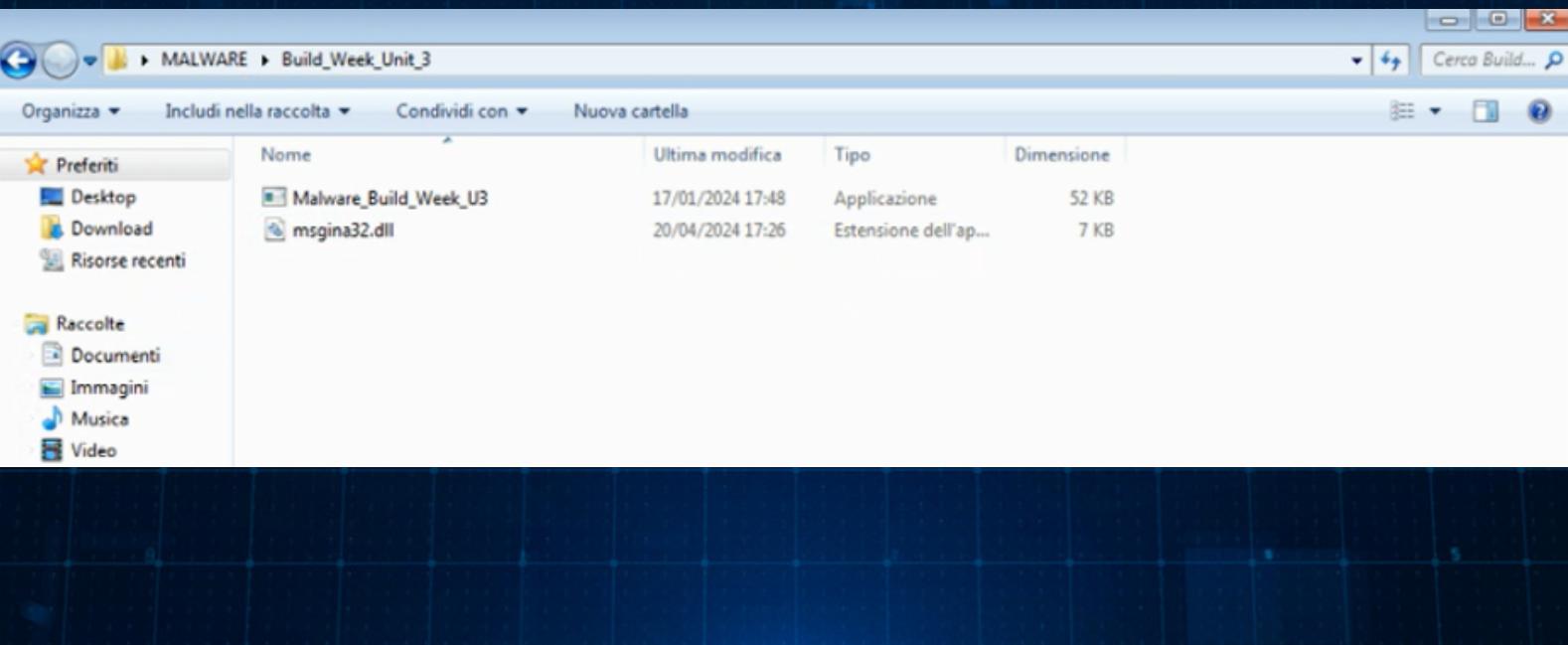
L'oggetto rappresentato dal parametro alla locazione di memoria 00401017 è molto probabilmente una stringa. Questa stringa rappresenta con tutta probabilità il percorso della chiave di registro che la funzione alla locazione 00401021 sta tentando di creare, ed è probabilmente memorizzata in memoria come stringa null-terminata, ovvero termina con un carattere nullo (\0). Le istruzioni comprese tra gli indirizzi 00401027 e 00401029 controllano il valore di ritorno della chiamata alla funzione RegCreateKeyExA e saltano a una routine di gestione degli errori se la chiamata ha fallito:

- 00401027: Questa istruzione esegue un confronto tra il valore contenuto nel registro EAX e il valore zero.
- 00401029: Questa istruzione è un'istruzione condizionale di salto.
- Istruzione successiva: Se il valore in EAX è uguale a zero, il programma salterà all'istruzione successiva situata a otto byte di distanza. Se il valore in EAX non è uguale a zero, il programma continuerà a eseguire l'istruzione successiva a questa istruzione.

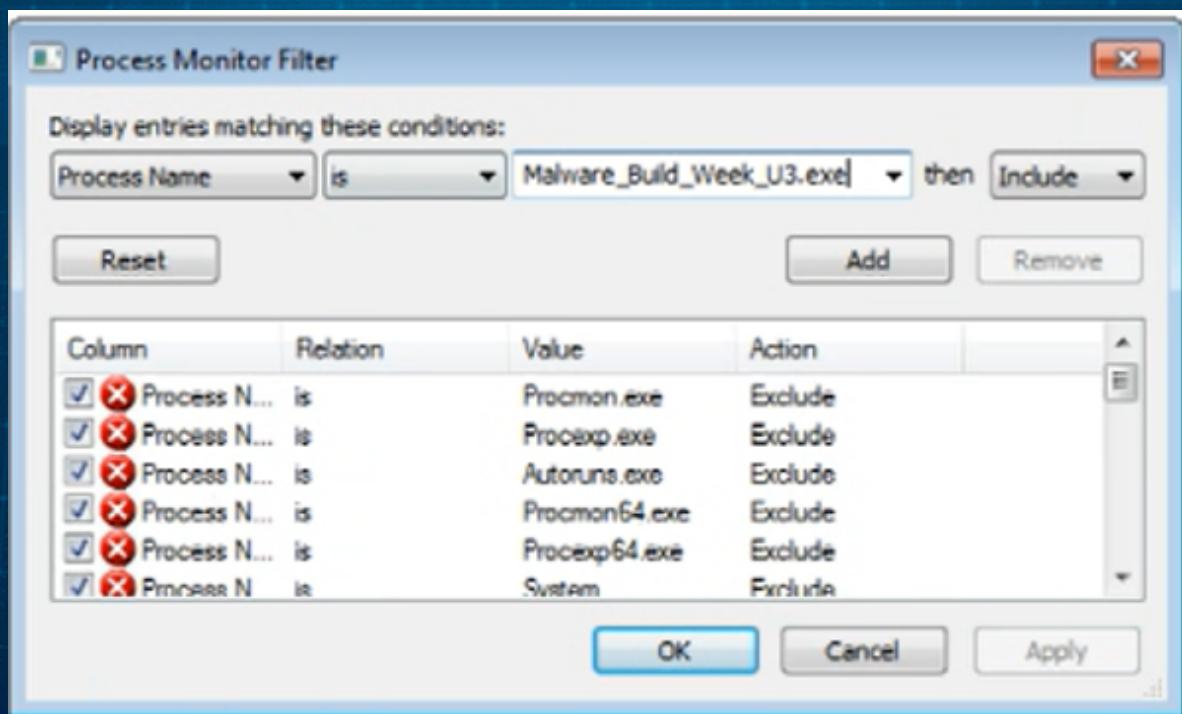
# Analisi dinamica



Dopo aver fatto il doppio click è apparsa l'estensione dell'app msgina32.dll, questo file è un componente critico del processo di accesso a Windows



## Avviamo Process Monitor ed analizziamo i risultati filtrando:



Possiamo notare che viene creata la chiave di registro HKLM, abbreviazione di HKEY\_LOCAL\_MACHINE, è una delle chiavi principali del registro di sistema di Windows. Contiene informazioni di configurazione cruciali per il funzionamento del sistema operativo e dei programmi installati. A differenza di altre chiavi del registro di sistema che riguardano profili utente specifici, HKLM memorizza impostazioni globali accessibili a tutti gli utenti del computer

Time	Process Name	PID	Operation	Path	Result	Detail
19:04:	Malware_Build_Week_U3.exe	2476	RegQueryValue	HKEY\SYSTEM\CurrentControlSet\Control\... NAME NOT FOUND Length: 548	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
19:04:	Malware_Build_Week_U3.exe	2476	RegQueryValue	HKEY\SYSTEM\CurrentControlSet\Control\... SUCCESS	SUCCESS	Desired Access: Maximum Allowed, Granted Access: Read
19:04:	Malware_Build_Week_U3.exe	2476	RegOpenKey	HKLM\... SUCCESS	SUCCESS	Query: Handle Tags, Handle Tag: 0x0
19:04:	Malware_Build_Week_U3.exe	2476	RegQueryKey	HKLM\... NAME NOT FOUND Desired Access: Read	SUCCESS	Desired Access: Generic Write, Read Attributes, Disposition: Overwrite, Options: Synchronous IO
19:04:	Malware_Build_Week_U3.exe	2476	RegOpenKey	C:\Users\user\Desktop\MALWARE\Bu... SUCCESS	SUCCESS	Offset: 0, Length: 4 056, Priority: Normal
19:04:	Malware_Build_Week_U3.exe	2476	CreateFile	C:\Users\user\Desktop\MALWARE\Bu... SUCCESS	SUCCESS	Offset: 4 096, Length: 2 560, Priority: Normal
19:04:	Malware_Build_Week_U3.exe	2476	WriteFile	C:\Users\user\Desktop\MALWARE\Bu... SUCCESS	SUCCESS	
19:04:	Malware_Build_Week_U3.exe	2476	CloseFile	C:\Users\user\Desktop\MALWARE\Bu... SUCCESS	SUCCESS	
19:04:	Malware_Build_Week_U3.exe	2476	RegQueryKey	HKLM\... Query: Handle Tags, Handle Tag: 0x0	SUCCESS	
19:04:	Malware_Build_Week_U3.exe	2476	RegCreateKey	HKLM\SOFTWARE\Wow6432Node\Micro... SUCCESS	SUCCESS	Desired Access: All Access, Disposition: REG_OPENED_EXISTING_KEY
19:04:	Malware_Build_Week_U3.exe	2476	RegSetInfoKey	HKLM\SOFTWARE\Wow6432Node\Micro... SUCCESS	SUCCESS	KeySetInformationClass: KeySetHandleTagsInformation, Length: 0
19:04:	Malware_Build_Week_U3.exe	2476	RegQueryKey	HKLM\SOFTWARE\Wow6432Node\Micro... SUCCESS	SUCCESS	Query: Handle Tags, Handle Tag: 0x400
19:04:	Malware_Build_Week_U3.exe	2476	RegSetValue	HKLM\SOFTWARE\Wow6432Node\Micro... ACCESS DENIED	REG_SZ, Length: 520, Data: C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\ms... Type: REG_SZ, Length: 520, Data: C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\ms...	
19:04:	Malware_Build_Week_U3.exe	2476	RegCloseKey	HKLM\SOFTWARE\Wow6432Node\Micro... SUCCESS	SUCCESS	
19:04:	Malware_Build_Week_U3.exe	2476	Thread Exit		SUCCESS	Thread ID: 3052 User Time: 0.0000000, Kernel Time: 0.0156250
19:04:	Malware_Build_Week_U3.exe	2476	QueryNameFormat	C:\Windows\System32\apiexecschema.dll	SUCCESS	Name: \Windows\System32\apiexecschema.dll
19:04:	Malware_Build_Week_U3.exe	2476	QueryNameFormat	C:\Users\user\Desktop\MALWARE\Bu... SUCCESS	SUCCESS	Name: \Users\user\Desktop\MALWARE\Build_Week_Unit_3\Malware_Build_Week_U3.exe
19:04:	Malware_Build_Week_U3.exe	2476	QueryNameFormat	C:\Windows\System32\wowl64cpu.dll	SUCCESS	Name: \Windows\System32\wowl64cpu.dll
19:04:	Malware_Build_Week_U3.exe	2476	QueryNameFormat	C:\Windows\System32\wow64win.dll	SUCCESS	Name: \Windows\System32\wow64win.dll
19:04:	Malware_Build_Week_U3.exe	2476	QueryNameFormat	C:\Windows\System32\wow64.dll	SUCCESS	Name: \Windows\System32\wow64.dll
19:04:	Malware_Build_Week_U3.exe	2476	QueryNameFormat	C:\Windows\SysWOW64\cryptbase.dll	SUCCESS	Name: \Windows\SysWOW64\cryptbase.dll
19:04:	Malware_Build_Week_U3.exe	2476	QueryNameFormat	C:\Windows\SysWOW64\asppcl.dll	SUCCESS	Name: \Windows\SysWOW64\asppcl.dll
19:04:	Malware_Build_Week_U3.exe	2476	QueryNameFormat	C:\Windows\SysWOW64\advapi32.dll	SUCCESS	Name: \Windows\SysWOW64\advapi32.dll
19:04:	Malware_Build_Week_U3.exe	2476	QueryNameFormat	C:\Windows\SysWOW64\sechost.dll	SUCCESS	Name: \Windows\SysWOW64\sechost.dll
19:04:	Malware_Build_Week_U3.exe	2476	QueryNameFormat	C:\Windows\SysWOW64\pcp04.dll	SUCCESS	Name: \Windows\SysWOW64\pcp04.dll
19:04:	Malware_Build_Week_U3.exe	2476	QueryNameFormat	C:\Windows\SysWOW64\kernel32.dll	SUCCESS	Name: \Windows\SysWOW64\kernel32.dll
19:04:	Malware_Build_Week_U3.exe	2476	QueryNameFormat	C:\Windows\SysWOW64\msvcr32.dll	SUCCESS	Name: \Windows\SysWOW64\msvcr32.dll
19:04:	Malware_Build_Week_U3.exe	2476	QueryNameFormat	C:\Windows\SysWOW64\KernBase.dll	SUCCESS	Name: \Windows\SysWOW64\KernBase.dll
19:04:	Malware_Build_Week_U3.exe	2476	QueryNameFormat	C:\Windows\System32\rtld.dll	SUCCESS	Name: \Windows\System32\rtld.dll
19:04:	Malware_Build_Week_U3.exe	2476	QueryNameFormat	C:\Windows\SysWOW64\rtld.dll	SUCCESS	Name: \Windows\SysWOW64\rtld.dll
19:04:	Malware_Build_Week_U3.exe	2476	Process Exit		SUCCESS	Exit Status: 0 User Time: 0.0000000 seconds, Kernel Time: 0.0156250 seconds, Private Pages: 0

Viene fatta una chiamata di sistema ( CreateFile ) che crea la msgina.dll nella cartella del malware e a seguire vediamo la write file che inserisce il contenuto malevolo e poi la close file

2604	RegQueryKey	HKLM
2604	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Diagnos... C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll
2604	CreateFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll
2604	WriteFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll
2604	WriteFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll
2604	CloseFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll
2604	RegQueryKey	HKLM
2604	RegCreateKey	HKLM\Software\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon
2604	RegSetInfoKey	HKLM\Software\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon
2604	RegQueryKey	HKLM\Software\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon
2604	RegSetValue	HKLM\Software\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon\GinaDLL
2604	RegCloseKey	HKLM\Software\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon
2604	Thread Exit	

# Conclusione

Il comportamento globale del malware è quello della creazione e apertura della chiave di registro Winlogon con inserimento nella stessa il valore che punta alla dll malevola creata dopo la sua esecuzione. Essendo la dll in questione fondamentale per l'autenticazione degli utenti su windows, possiamo presupporre che lo scopo del software malevolo sia quello di registrare le autenticazioni da parte degli utenti al sistema per poi impossessarsene.