



BUILDWEEK 3 GRUPPO 4 EPICODE

Team : Pisapia Giovanni, Carmine Caputo, Matteo Addei,
Riccardo Brunotti, Michele Macrì, Gaspare Rizzo, Pierluigi
Amorese, Riccardo Nardecchia.

DAY 1

TASK 1-2: Quanti **parametri** sono passati alla funzione Main()? Quante **variabili** sono dichiarate all'interno della funzione Main()?

Il tool **IDA** **differenzia variabili e parametri utilizzando come riferimento l'offset** (differenza rispetto ad un valore di riferimento) rispetto al puntatore EBP:

- Le **variabili** sono ad un offset negativo rispetto al registro EBP
- I **parametri** si trovano ad un offset positivo rispetto al registro EBP

Nella **funzione Main()** oggetto d'interesse abbiamo **3 parametri** e **4 variabili**

```
; Attributes: bp-based frame
; int __cdecl main(int argc,const char **argv,const char *envp)
_main proc near

hModule= dword ptr -11Ch
Data= byte ptr -118h
var_8= dword ptr -8
var_4= dword ptr -4

argc= dword ptr 8
argv= dword ptr 0Ch
envp= dword ptr 10h

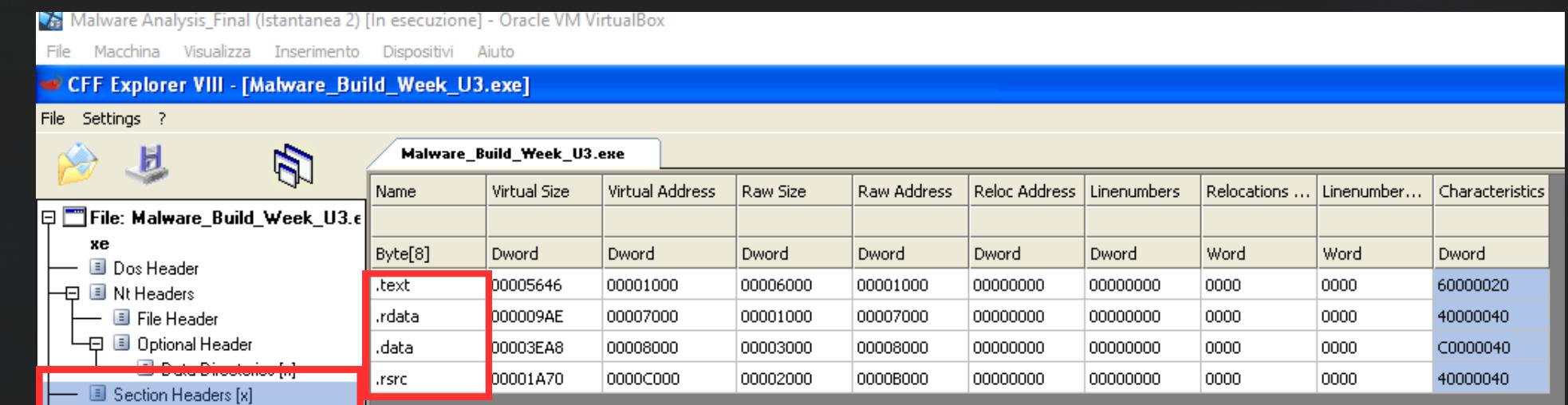
push    ebp
mov     ebp, esp
sub    esp, 11Ch
push    ebx
push    esi
push    edi
mov     [ebp+var_4], 0
push    0           ; lpModuleName
call    ds:GetModuleHandleA
mov     [ebp+hModule], eax
mov     [ebp+Data], 0
mov     ecx, 43h
xor     eax, eax
lea     edi, [ebp-117h]
rep stosd
stosb
```

DAY 1

TASK 3: Quali **sezioni** sono presenti all'interno del file eseguibile? Descrivete brevemente almeno 2 di quelle identificate

Per analizzare le sezioni di cui si compone il file eseguibile oggetto d'interesse abbiamo utilizzato il tool **CFF EXPLORER**, funzionale per **l'analisi statica basica**. Nello specifico, spostandoci nella tab **SECTION HEADERS**, si nota che il malware si compone di:

- **.text:** Questa sezione contiene le istruzioni, **ovvero le righe di codice che la CPU eseguirà quando il software viene avviato**. È la **sezione principale di un file eseguibile, poiché contiene il codice effettivo che viene eseguito per far funzionare il programma**. Tutte le altre sezioni contengono dati o informazioni di supporto per questa sezione
- **.rdata:** Questa sezione **contiene informazioni sulle librerie e le funzioni importate ed esportate dall'eseguibile**. Qui vengono memorizzate le **informazioni sui moduli esterni** che l'eseguibile utilizza, come librerie di sistema o librerie condivise, e le funzioni che vengono importate o esportate per l'utilizzo all'interno del programma
- **.data:** Questa sezione **contiene dati e variabili globali del programma eseguibile**. Le variabili definite in questa sezione **sono accessibili da qualsiasi parte del programma, poiché sono globalmente dichiarate**
- **.rsrc:** Questa sezione **include le risorse utilizzate come ad esempio icone, immagini, menu e stringhe che non sono parte dell'eseguibile stesso**.



Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations ...	Linenumber... Characteristics
Byte[8]	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword
.text	00005646	00001000	00006000	00001000	00000000	00000000	0000	60000020
.rdata	000009AE	00007000	00001000	00007000	00000000	00000000	0000	40000040
.data	00003EA8	00008000	00003000	00008000	00000000	00000000	0000	C0000040
.rsrc	00001A70	0000C000	00002000	0000B000	00000000	00000000	0000	40000040

DAY 1

TASK 4: 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.

Per vedere quali librerie importa il malware abbiam utilizzato il tool **CFF EXPLORER**, selezionando la tab IMPORT DIRECTORYper vedere e librerie importate nel malware:

- **KERNEL32.DLL:** libreria usata per le funzioni principali per interagire con il sistema operativo. Un malware potrebbe sfruttare tale libreria per manipolare i file e per accedere la gestione della memoria
- **ADVAPI32.dll:** libreria che contiene le funzioni per interagire con i servizi ed i registri del sistema operativo Microsoft, tramite la quale un malware potrebbe creare nuovi account utente, accedere al registro di sistema e crittografare o decrittografare dati sensibili;

Module Name	Imports	OFTs	TimeStamp	ForwarderChain	Name RVA	FTs (IAT)
0000769E	N/A	000074EC	000074F0	000074F4	000074F8	000074FC
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.dll	51	00007534	00000000	00000000	0000769E	0000700C
ADVAPI32.dll	2	00007528	00000000	00000000	000076D0	00007000

All'interno della libreria **KERNEL32.dll** si notano le funzioni:

- **SizeofResource, LockResource, LoadResource e FindResourceA**, APIs che permettono di localizzare all'interno della sezione “risorse” il Malware da estrarre, e successivamente da caricare in memoria per l'esecuzione o da salvare sul disco per esecuzione futura.

DATA LA PRESENZA DI TALE LIBRERIE, SI POTREBBE INIZIARE A PENSARE DI ESSER IN PRESENZA DI UN DROPPER, PROGRAMMA MALEVOLO CHE CONTIENE AL SUO INTERNO UN MALWARE.

OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
00007632	00007632	0295	SizeofResource
00007644	00007644	01D5	LockResource
00007654	00007654	01C7	LoadResource
00007622	00007622	02B8	VirtualAlloc
00007674	00007674	0124	GetModuleFileNameA
0000768A	0000768A	0126	GetModuleHandleA
00007612	00007612	00B6	FreeResource
00007664	00007664	00A3	FindResourceA
00007604	00007604	0018	CloseHandle

ALTRÉ FUNZIONI ALL' INTERNO DEL ESEGUIBILE

- **CreateFileA:** è una funzione Windows API che crea un oggetto file;
- **ReadFile:** funzione Windows API che legge dati da un file;
- **WriteFile:** funzione Windows API che scrive dati in un file;
- **GetProcAddress e LoadLibraryA:** funzionali permettono di importare le funzioni della libreria a tempo di esecuzione (runtime). Ciò significa che l'eseguibile richiama la libreria solo quando ha bisogno di utilizzare una specifica funzione, per risultare meno invasivo e meno rilevabile.

Module Name	Imports	OFTs	TimeDateStar	0000786E	0000786E	019B	HeapCreate
	N/A	000074EC	000074F0	0000787C	0000787C	02BF	VirtualFree
szAnsi	(nFunctions)	Dword	Dword	0000788A	0000788A	022F	RtlUnwind
KERNEL32.dll	51	00007534	00000000	00007896	00007896	0199	HeapAlloc
ADVAPI32.dll	2	00007528	00000000	000078A2	000078A2	01A2	HeapReAlloc
				000078B0	000078B0	027C	SetStdHandle
				000078C0	000078C0	00AA	FlushFileBuffers
				000078D4	000078D4	026A	SetFilePointer
				000078E6	000078E6	0034	CreateFileA
				000078F4	000078F4	00BF	GetCPInfo
				00007900	00007900	00B9	GetACP
				0000790A	0000790A	0131	GetOEMCP
				00007916	00007916	013E	GetProcAddress
				00007928	00007928	01C2	LoadLibraryA
				00007938	00007938	0261	SetEndOfFile
				00007948	00007948	0218	ReadFile
				00007954	00007954	01E4	MultiByteToWideChar
				0000796A	0000796A	01BF	LCMapStringA
				0000797A	0000797A	01C0	LCMapStringW
				0000798A	0000798A	0153	GetStringTypeA
				0000799C	0000799C	0156	GetStringTypeW

All'interno della libreria **ADVAPI32.dll** si notano le funzioni:

- **RegSetValueExA:** modifica il contenuto di un valore esistente e crea un nuovo valore nel caso che esso non dovesse esistere, con il tipo di dati specificato
- **RegCreateKeyEx:** crea ed apre una nuova sottochiave della chiave indicata.

Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
	N/A	00007500	00007504	00007508	0000750C	00007510
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.dll	51	00007534	00000000	00000000	0000769E	0000700C
ADVAPI32.dll	2	00007528	00000000	00000000	000076D0	00007000
OFTs	FTs (IAT)	Hint	Name			
Dword	Dword	Word	szAnsi			
000076AC	000076AC	0186	RegSetValueExA			
000076BE	000076BE	015F	RegCreateKeyExA			

DATA LA PRESENZA DI QUESTE ULTIME FUNZIONI, SI IPOTIZZA CHE IL MALWARE OGGETTO DI INTERESSE PUNTI AD OTTENERE LA **PERSISTENZA** ALL'INTERNO DELLA MACCHINA INFETTA.

DAY 1

TASK 5-6-7: Qual è 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 malware utilizza la funzione **RegCreateKeyExA**, locata all'indirizzo di memoria **00401021**, per creare una nuova sottochiave della chiave **"SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Winlogon"**.

Per trovare l'intero path basta fare doppio click su **Subkey**.

La chiave di registro **HKEY_LOCAL_MACHINE\software\Microsoft\Windows NT\CurrentVersion\Winlogon** contiene le impostazioni che controllano il processo di accesso a Windows.

```
00401000 .text:00401000 push    ebp
00401001 .text:00401001 mov     ebp, esp
00401003 .text:00401003 push    ecx
00401004 .text:00401004 push    0 ; lpdwDisposition
00401006 .text:00401006 lea     eax, [ebp+hObject]
00401009 .text:00401009 push    eax ; phkResult
0040100A .text:0040100A push    0 ; lpSecurityAttributes
0040100C .text:0040100C push    0F003Fh ; samDesired
00401011 .text:00401011 push    0 ; dwOptions
00401013 .text:00401013 push    0 ; lpClass
00401015 .text:00401015 push    0 ; Reserved
00401017 .text:00401017 push    offset SubKey ; "SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Winlogon"
0040101C .text:0040101C push    80000002h ; hKey
00401021 .text:00401021 call    ds:RegCreateKeyExA
00401027 .text:00401027 test    eax, eax
00401029 .text:00401029 jz     short loc_401032
0040102B .text:0040102B mov     eax, 1
00401030 .text:00401030 jmp    short loc_40107B
00401032 .text:00401032 ;
```

ID	Name	Type	Value	XREFs
00408040	aBinary	db	'BINAY', 0	DATA XREF: .data:lpType
00408047		align	4	
00408048	aRi	db	'RI', 0Ah, 0	DATA XREF: sub_401000:loc_401062
0040804C	; char ValueName[]			
0040804C	ValueName	db	'GinaDLL', 0	DATA XREF: sub_401000+3E
00408054	; char SubKey[]			
00408054	SubKey	db	'SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Winlogon', 0	DATA XREF: sub_401000+17
00408054				
0040808A		align	4	
0040808C	aDr	db	'DR', 0Ah, 0	DATA XREF: sub_401080+118
00408090	; char aMsgina32_dll_0[]			
00408090	aMsgina32_dll_0	db	'msgina32.dll', 0	DATA XREF: sub_401080+E6
0040809D				
004080A0	; char awb[]			
004080A0	awb	db	'wb', 0	DATA XREF: sub_401080+E1
004080A3		align	4	
004080A4	aMsgina32_dll	db	'\\msgina32.dll', 0	DATA XREF: _main+78
004080B2				
004080C0	off_4080C0	dd	offset __exit	DATA XREF: __msg_exit+1C
004080C4	dword_4080C4	dd	1	DATA XREF: _MSGANNER+F

L'istruzione **push** viene utilizzata per passare i parametri sullo stack, che verranno in seguito utilizzati dalla funzione **RegCreateKeyExA**.

```
00401000 .text:00401000 push    ebp
00401001 .text:00401001 mov     ebp, esp
00401003 .text:00401003 push    ecx
00401004 .text:00401004 push    0 ; lpdwDisposition
00401006 .text:00401006 lea     eax, [ebp+hObject]
00401009 .text:00401009 push    eax ; phkResult
0040100A .text:0040100A push    0 ; lpSecurityAttributes
0040100C .text:0040100C push    0F003Fh ; samDesired
00401011 .text:00401011 push    0 ; dwOptions
00401013 .text:00401013 push    0 ; lpClass
```

DAY 1

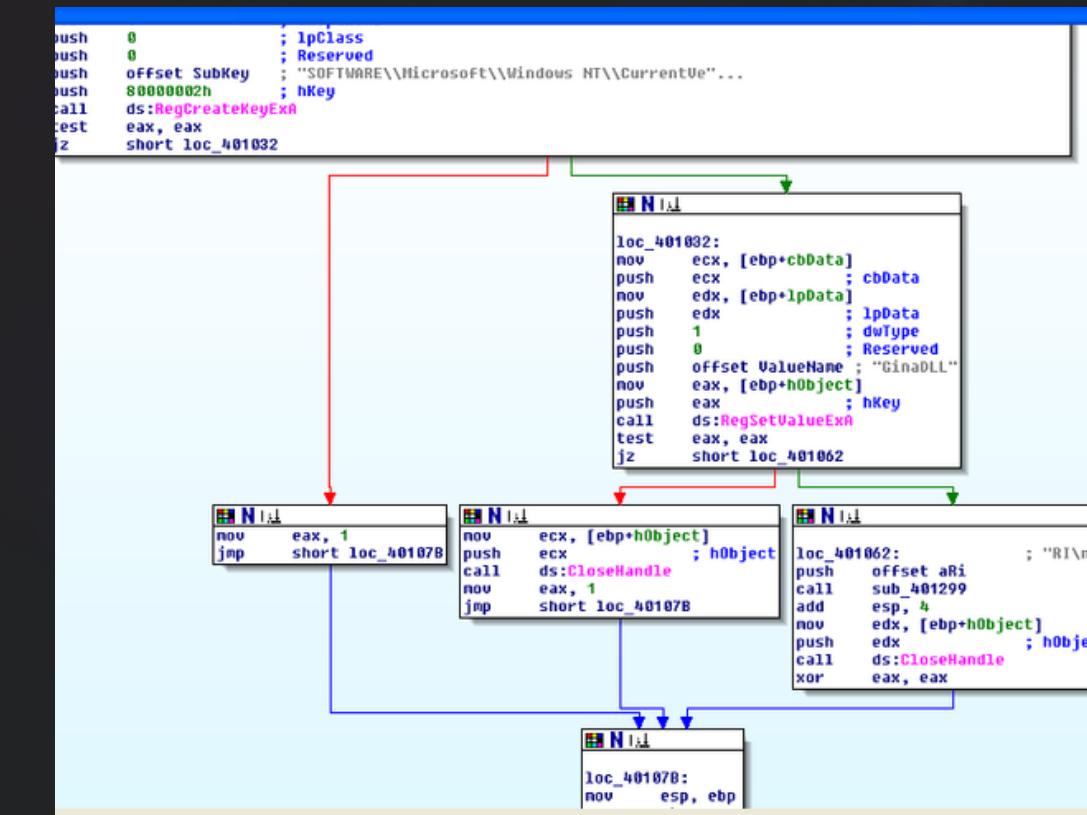
TASK 8: Il significato delle istruzioni comprese tra gli 00401027 e 00401029

La prima istruzione **test eax, eax** è un'istruzione condizionale simile all'AND logico, che rispetto ad essa non modifica gli operandi, bensì lo **ZERO FLAG (ZF)** del registro **EFLAGS**, il quale viene settato ad 1 se e soltanto **se il risultato dell'AND è 0**.

La seconda istruzione è **JZ(JUMP IF ZERO)**, istruzione questa che farà effettuare un salto condizionale al codice **qualora lo ZERO FLAG datoci dal risultato dell'istruzione precedente sia settato a 1**. Nel nostro caso, il salto verrà effettuato all'indirizzo di memoria **401032**.

```
00401000: push    ebp
00401001: mov     ebp, esp
00401002: push    ecx
00401003: push    0
00401004: push    eax, [ebp+hObject]
00401005: lea     eax, [ebp+00000000]
00401006: push    eax
00401007: push    0
00401008: push    0F003Fh
00401009: push    0
0040100A: push    0
0040100B: push    0
0040100C: push    0
0040100D: push    0
0040100E: push    0
0040100F: push    offset SubKey ; "SOFTWARE\Microsoft\Windows NT\CurrentVersion\GinaDLL"
00401010: push    80000002h
00401011: push    0
00401012: call    ds:RegCreateKeyExA
00401013: test    eax, eax
00401014: jz     short loc_401032
00401015: jmp    short loc_401078
00401016: mov     eax, 1
00401017: nov    ecx, [ebp+cbData]
00401018: push    ecx
00401019: nov    edx, [ebp+lpData]
0040101A: push    edx
0040101B: push    1
0040101C: push    0
0040101D: push    offset ValueName ; "GinaDLL"
0040101E: push    eax, [ebp+hObject]
0040101F: push    eax
00401020: call    ds:RegSetValueExA
00401021: test    eax, eax
00401022: jz     short loc_401062
00401023: nov    eax, 1
00401024: nov    edx, [ebp+hObject]
00401025: call    ds:CloseHandle
00401026: nov    eax, 1
00401027: jmp    short loc_401078
00401028: loc_401062: push    offset aRI ; "RI\n"
00401029: call    sub_401299
0040102A: add     esp, 4
0040102B: nov    edx, [ebp+hObject]
0040102C: push    edx
0040102D: call    ds:CloseHandle
0040102E: xor     eax, eax
```

Di seguito uno screenshot del diagramma di flusso che mostra quali salti effettuerà l'eseguibile:



DAY 1

TASK 9: Con riferimento all'ultimo quesito, tradurre il codice Assembly nel corrispondente **costrutto C**

```
1 int main() {  
2     int a = 0; //equivale a ZF  
3     if (a == 0) {  
4         goto loc_40107B;  
5     } else {  
6         a = 1;  
7         goto loc_401032;  
8     }  
9 }
```

DAY 1

TASK 10: Valutate ora la chiamata alla locazione **00401047**, qual è il valore del parametro “**ValueName**”?

Il valore del parametro **ValueName** è **GinaDLL**, come da screenshots effettuati sia su IDA Pro che su OllyDBG.

Gina.dll (Graphical Identification and Authentication Dynamic Link Library) è un file di libreria dinamica utilizzato nei sistemi operativi Windows per consentire funzionalità di autenticazione e identificazione grafica. Questo file era responsabile della gestione dell'interfaccia di autenticazione per gli utenti che accedevano a un computer Windows.

Gina.dll viene caricato da **Winlogon** e viene eseguito prima che il processo di accesso di Windows inizi a richiedere le credenziali dell'utente.

```
.text:00401032 loc_401032: ; CODE XREF: sub_401000+29↑j
  .text:00401032
  .text:00401033     mov    ecx, [ebp+cbData]
  .text:00401034     push   ecx          ; cbData
  .text:00401035     mov    edx, [ebp+lpData]
  .text:00401036     push   edx          ; lpData
  .text:00401037     push   edx          ; dwType
  .text:00401038     push   1             ; dwType
  .text:00401039     push   0             ; Reserved
  .text:0040103A     push   offset ValueName ; "GinaDLL"
  .text:0040103B     mov    eax, [ebp+nObject]
  .text:0040103C     push   eax          ; hKey
  .text:0040103D     call   ds:RegSetValueExA
  .text:0040103E     test  eax, eax
  .text:0040103F     jz    short loc_401062
  .text:00401040     mov    ecx, [ebp+hObject]
  .text:00401041     push   ecx          ; hObject
  .text:00401042     call   ds:CloseHandle
  .text:00401043     mov    eax, 1
  .text:00401044     jmp   short loc_40107B

...text:00401062...
00401035  . 51      PUSH ECX
00401036  . 8B55 08  MOV EDX,DWORD PTR SS:[EBP+8]
00401037  . 52      PUSH EDX
00401038  . 6A 01    PUSH 1
00401039  . 6A 00    PUSH 0
0040103A  . 68 4C804000 PUSH Malware_.0040804C
0040103B  . 8B45 FC  MOV EAX,DWORD PTR SS:[EBP-4]
0040103C  . F0      PUSH EDI
0040103D  . C3      RET
```

DAY 1

TASK 11: Nel complesso delle due funzionalità appena viste, spiegate quale funzionalità sta implementando il Malware in questa funzione.

In base alle informazioni che abbiamo inizialmente ricavato, **il malware verifica la presenza o meno della chiave di registro Gina.dll**; nel caso in cui detta chiave **sia già presente**, il malware provvederà alla modifica della stessa; viceversa, qualora la chiave non sia presente, il malware provvederà a crearla.

Attraverso tale operazione, il malware oggetto d'interesse **mira ad ottenere altresì la persistenza sulla macchina infetta**.

DAY 2

TASK 1: Qual è il valore del parametro **ResourceName** passato alla funzione **FindResourceA()**;

La funzione call **ds:FindResourceA** è locata all'indirizzo di memoria **004010C9**.

Cliccando due volte su **IpName**, veniamo riportati nella sezione **.data**, dove si nota che il valore del parametro **ResourceName** è **TGAD**

The screenshot shows the IDA Pro interface with two main panes. The top pane displays assembly code, and the bottom pane displays the data dump.

Assembly View:

```
.text:004010B8
.text:004010B8 loc_4010B8:          ; CODE XREF: sub_401080+2F↑j
    .text:004010B8     mov    eax, 1pType
    .text:004010BD     push   eax           ; 1pType
    .text:004010BE     mov    ecx, 1pName
    .text:004010C4     push   ecx           ; 1pName
    .text:004010C5     mov    edx, [ebp+hModule]
    .text:004010C8     push   edx           ; hModule
    * .text:004010C9     call   ds:FindResourceA
    .text:004010CF     mov    [ebp+hResInfo], eax
    .text:004010D2     cmp    [ebp+hResInfo], 0
    .text:004010D6     jnz    short loc_4010DF
    .text:004010D8     xor    eax, eax
    .text:004010DA     jmp    loc_4011BF
.text:004010DF :
```

Data Dump View:

Symbol	Type	Value	Comments
.data:0040802E	db	0	
.data:0040802F	db	0	
.data:00408030 ; LPCSTR 1pType	dd offset aBinary		; DATA XREF: sub_401080:loc_4010B8↑r ; "BINAY"
.data:00408034 ; LPCSTR 1pName	dd offset aTgad		; DATA XREF: sub_401080+3E↑r ; "TGAD"
.data:00408038 aTgad	db 'TGAD',0		; DATA XREF: .data:IpName↑o
.data:0040803D	align 10h		
.data:00408040 aBinary	db 'BINAY',0		; DATA XREF: .data:1pType↑o
.data:00408047	align 4		
.data:00408048 aRi	db 'RI',0Ah,0		; DATA XREF: sub_401000:loc_401062↑o
.data:0040804C ; char ValueName[]			
.data:0040804C ValueName	db 'GinaDLL',0		; DATA XREF: sub_401000+3E↑o
.data:00408054 ; char SubKey[]			

DAY 2

TASK 2: Il susseguirsi delle chiamate di funzione che effettua il Malware in questa sezione di codice. Che funzionalità sta implementando il Malware?

- **FindResourceA:** questa funzione ha il compito di individuare la posizione di una risorsa all'interno di un modulo, utilizzando il tipo e il nome specificati come criteri di ricerca. Nel nostro caso, FindResourceA restituisce la posizione del TGAD (valore del parametro `ResourceName`).

```
.text:004010B8 loc_4010B8:          ; CODE XREF: sub_401080+2F↑j
.  .text:004010B8    mov     eax, lpType      ; lpType
.  .text:004010B9    push    eax             ; lpType
.  .text:004010BD    mov     ecx, lpName      ; lpName
.  .text:004010BE    push    ecx             ; lpName
.  .text:004010C4    mov     edx, [ebp+hModule]
.  .text:004010C5    mov     [eax], edx        ; module
.  .text:004010C6    push    edx             ; module
.  .text:004010C9    call    ds:FindResourceA
.  .text:004010CF    mov     [ebp+hResInfo], eax
.  .text:004010D2    cmp     [ebp+hResInfo], 0
.  .text:004010D6    jnz    short loc_4010DF
.  .text:004010D8    xor     eax, eax
.  .text:004010DA    jmp    loc_4011BF
.  .text:004010DF    ;
```

- **LoadResource:** questa funzione restituisce un identificatore (handle) che consente di ottenere un puntatore al primo byte della risorsa specificata in memoria. Utilizzando questa funzione, si recupera l'handle per la risorsa individuata precedentemente tramite la funzione `FindResourceA`.

```
.text:004010DF loc_4010DF:          ; CODE XREF: sub_401080+3E↑j
.  .text:004010DF    mov     eax, [ebp+hResInfo]   ; hResInfo
.  .text:004010E0    push    eax             ; hResInfo
.  .text:004010E2    mov     ecx, [ebp+hModule]   ; hModule
.  .text:004010E3    push    ecx             ; hModule
.  .text:004010E6    push    edx             ; hModule
.  .text:004010E7    call    ds:LoadResource
.  .text:004010ED    mov     [ebp+hResData], eax
.  .text:004010F0    cmp     [ebp+hResData], 0
.  .text:004010F4    jnz    short loc_4010FB
.  .text:004010F6    jmp    loc_4011A5
.  .text:004010FB    ;
```

- **LockResource:** questa funzione restituisce un puntatore al primo byte della risorsa specificata in memoria, nel nostro caso un puntatore diretto alla risorsa TGAD.

```
.text:004010FB loc_4010FB:          ; CODE XREF: sub_401080+74↑j
.  .text:004010FB    mov     edx, [ebp+hResData]   ; hResData
.  .text:004010FE    push    edx             ; hResData
.  .text:004010FF    call    ds:LockResource
.  .text:00401102    mov     [ebp+var_8], eax
.  .text:00401108    cmp     [ebp+var_8], 0
.  .text:0040110C    jnz    short loc_401113
.  .text:0040110E    jmp    loc_4011A5
.  .text:00401112    ;
```

- **SizeofResource:** questa funzione recupera la dimensione, espressa in byte, della risorsa specificata, nel nostro caso TGAD.

```
.text:00401113 loc_401113:          ; CODE XREF: sub_401080+8C↑j
.  .text:00401113    mov     eax, [ebp+hResInfo]   ; hResInfo
.  .text:00401116    push    eax             ; hResInfo
.  .text:00401117    mov     ecx, [ebp+hModule]   ; hModule
.  .text:0040111A    push    ecx             ; hModule
.  .text:0040111B    call    ds:SizeofResource
.  .text:0040111B    mov     [ebp+dwSize], eax
.  .text:00401124    cmp     [ebp+dwSize], 0
.  .text:00401128    ja     short loc_40112C
.  .text:0040112A    jmp    short loc_4011A5
.  .text:0040112C    ;
```

DAY 2

TASK 3-4: È possibile identificare questa funzionalità utilizzando l'analisi statica basica? In caso di risposta affermativa, elencare le evidenze a supporto

- Come già analizzato durante il primo giorno, è possibile rintracciare tali funzionalità tramite analisi statica basica grazie al tool **CFF EXPLORER**.

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

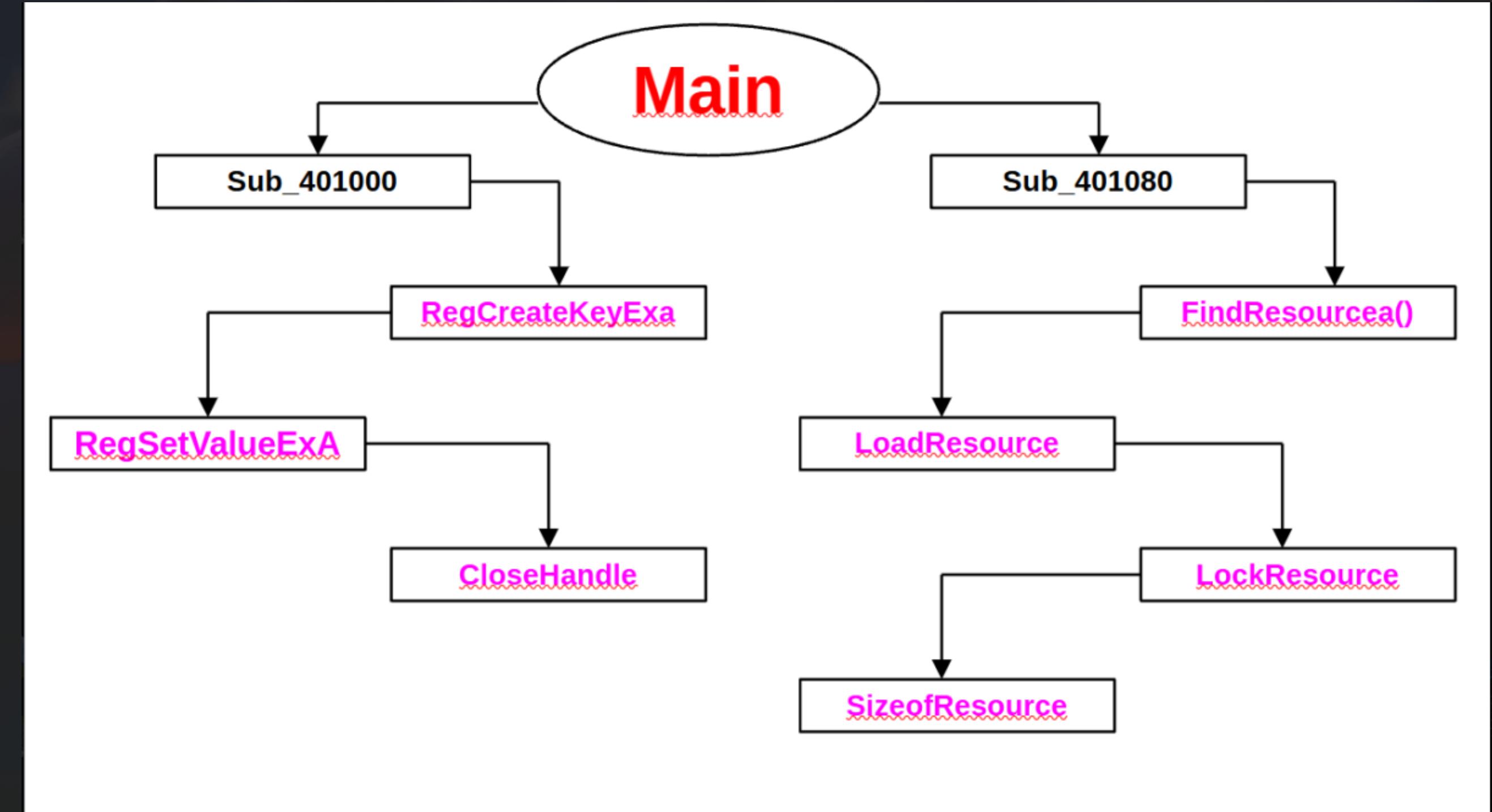
The 'Import Directory' tab is selected. The 'szAnsi' row shows imports from KERNEL32.dll. The 'FindResourceA' function is highlighted with a red box.

- Nella tab **Resource Directory**, infatti, si nota la risorsa **TGAD** richiamata dalla funzione **FindResourceA** (rettangolo rosso). Come detto in precedenza le funzioni sopra elencate sono tipiche della struttura dei Malware di tipo **DROPPER**.

The 'Resource Directory' tab is selected. A red box highlights the 'TGAD' resource entry under the first Resource Directory entry.

DAY 2

TASK 5: Disegnare un **diagramma di flusso** (inserire all'interno del box solo le informazioni circa le funzionalità principali) che comprenda le tre funzioni.



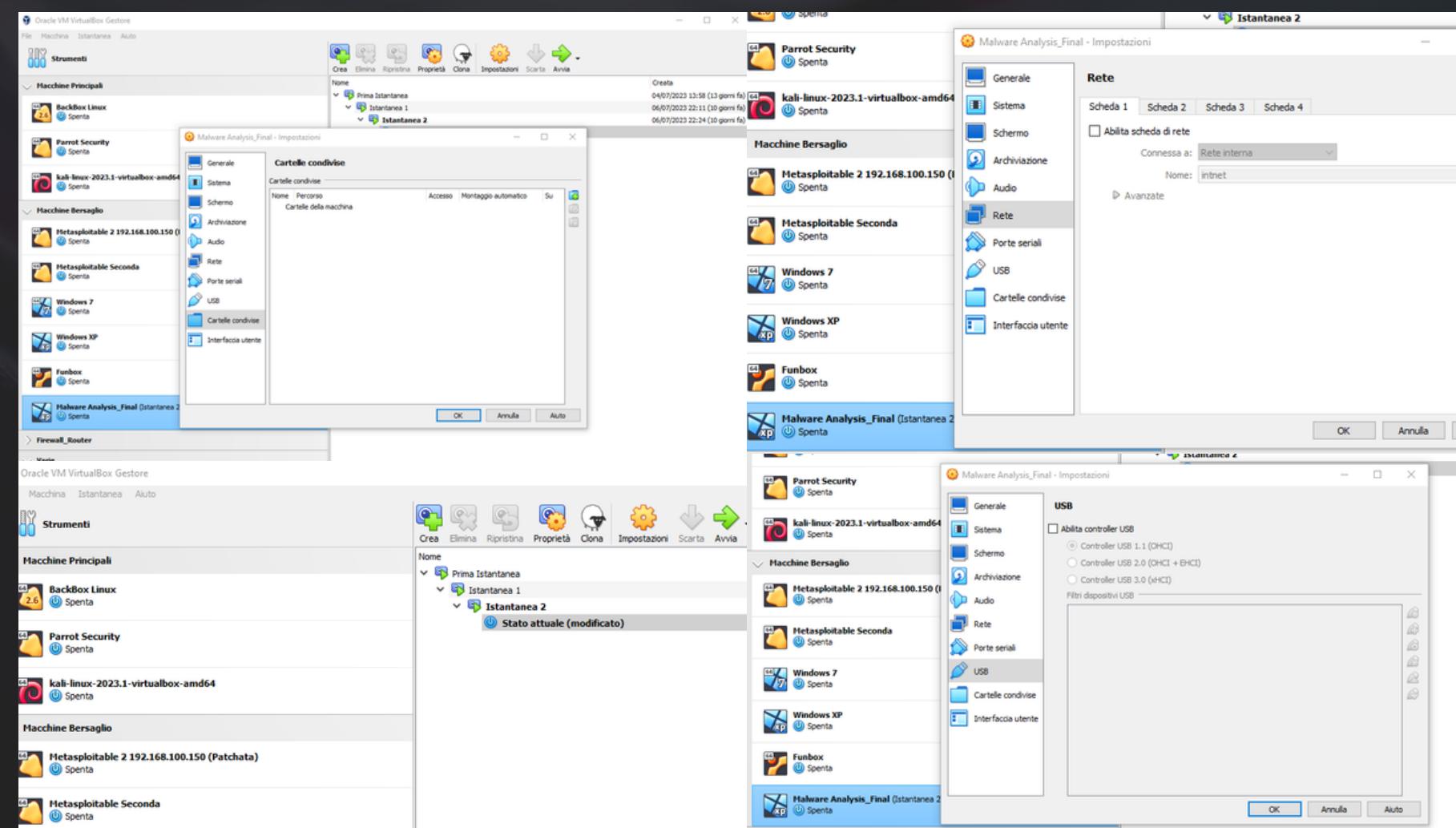
DAY 3

Effettuare un'Analisi Dinamica Basica ed analizzare il comportamento del Malware.

1) Impostazione della macchina in ambiente sicuro

Per iniziare la nostra Analisi abbiamo per prima cosa, seguendo le istruzioni insegnateci dalle lezioni di Cybersecurity, impostato completamente offline la macchina sulla quale andremo ad avviare il Malware. Le procedure da seguire sono:

- Disabilitare le Schede di Rete
- Disabilitare le Cartelle Condivise
- Disabilitare altre Periferiche Esterne (ad esempio Porte USB)
- Creare un'istantanea della macchina

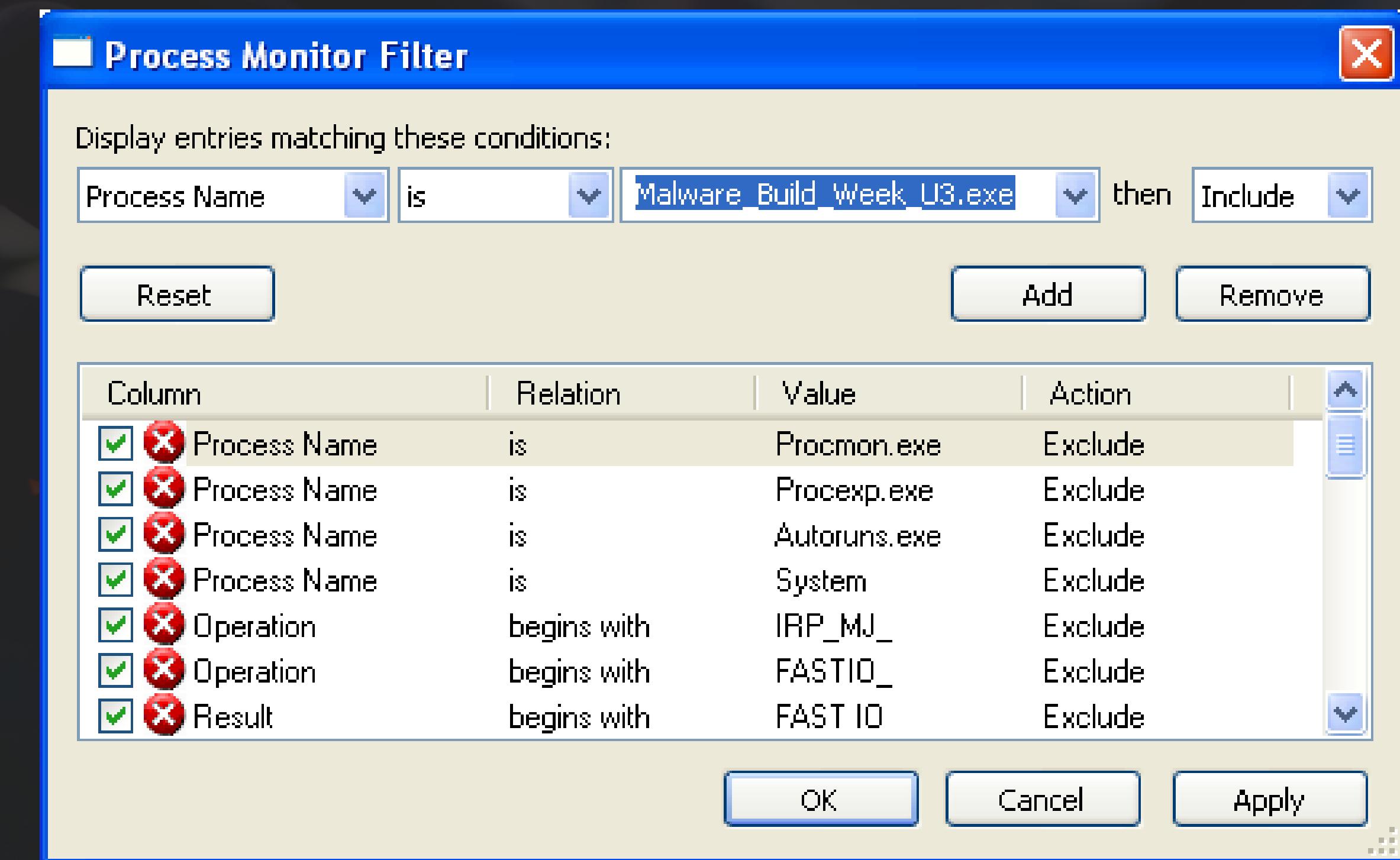


DAY 3

2) Avvio del Malware

Prima di eseguire il File Malevolo abbiamo aperto il tool **Process Monitor**, un tool che consente di monitorare in tempo reale le attività di Sistema come modifiche di **File, Registri e Processi**.

Essendo la cattura del tool in tempo reale abbiamo impostato il filtro per poter leggere le voci riguardanti soltanto il nostro **Malware**.





DAY 3

4) CHIAVI DI REGISTRO

Ci viene chiesto di Analizzare le Attività che avvengono sulle Chiavi di Registro grazie a ProcMon citato precedentemente e di spiegare quale Chiave viene Creata e il Valore che viene associato ad Essa. Per farlo abbiamo cliccato sulla Tab che riguarda appunto soltanto le Chiavi di Registro.

Time of Day	Process Name	PID	Operation	Path	Result	Detail
10:23:51.829...	Malware_Build_Week_U3...	1008	RegOpenKey	HKEY\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\Malware_Build_Week_U3.exe	NAME NOT FOUND	Desired Access: Read
10:23:51.832...	Malware_Build_Week_U3...	1008	RegOpenKey	HKEY\System\CurrentControlSet\Control\Terminal Server	SUCCESS	Desired Access: Read
10:23:51.835...	Malware_Build_Week_U3...	1008	RegQueryValue	HKEY\System\CurrentControlSet\Control\Terminal Server\TSSAppCompat	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
10:23:51.838...	Malware_Build_Week_U3...	1008	RegCloseKey	HKEY\System\CurrentControlSet\Control\Terminal Server	SUCCESS	
10:23:51.842...	Malware_Build_Week_U3...	1008	RegOpenKey	HKEY\System\CurrentControlSet\Control\Terminal Server\TSSAppCompat	SUCCESS	
10:23:51.842...	Malware_Build_Week_U3...	1008	RegQueryValue	HKEY\System\CurrentControlSet\Control\Terminal Server\TSSAppCompat	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
10:23:51.842...	Malware_Build_Week_U3...	1008	RegCloseKey	HKEY\System\CurrentControlSet\Control\Terminal Server	SUCCESS	Desired Access: Read
10:23:51.844...	Malware_Build_Week_U3...	1008	RegOpenKey	HKEY\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\Secur32.dll	NAME NOT FOUND	Desired Access: Read
10:23:51.844...	Malware_Build_Week_U3...	1008	RegOpenKey	HKEY\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\RPCRT4.dll	NAME NOT FOUND	Desired Access: Read
10:23:51.845...	Malware_Build_Week_U3...	1008	RegOpenKey	HKEY\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\ADVAPI32.dll	NAME NOT FOUND	Desired Access: Read
10:23:51.845...	Malware_Build_Week_U3...	1008	RegOpenKey	HKEY\System\CurrentControlSet\Control\Terminal Server	SUCCESS	Desired Access: Read
10:23:51.845...	Malware_Build_Week_U3...	1008	RegQueryValue	HKEY\System\CurrentControlSet\Control\Terminal Server\TSSAppCompat	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
10:23:51.846...	Malware_Build_Week_U3...	1008	RegCloseKey	HKEY\System\CurrentControlSet\Control\Terminal Server\TSSAppCompat	SUCCESS	Desired Access: Read
10:23:51.846...	Malware_Build_Week_U3...	1008	RegOpenKey	HKEY\System\CurrentControlSet\Control\Terminal Server\TSSUserEnabled	SUCCESS	Desired Access: Read
10:23:51.846...	Malware_Build_Week_U3...	1008	RegCloseKey	HKEY\System\CurrentControlSet\Control\Terminal Server	SUCCESS	Desired Access: Read
10:23:51.846...	Malware_Build_Week_U3...	1008	RegOpenKey	HKEY\Software\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS	Desired Access: Read
10:23:51.847...	Malware_Build_Week_U3...	1008	RegOpenKey	HKEY\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\LeakTrack	NAME NOT FOUND	Desired Access: Read
10:23:51.847...	Malware_Build_Week_U3...	1008	RegCloseKey	HKEY\Software\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS	Length: 144
10:23:51.847...	Malware_Build_Week_U3...	1008	RegOpenKey	HKEY\Software\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS	Desired Access: Read
10:23:51.847...	Malware_Build_Week_U3...	1008	RegQueryValue	HKEY\Software\Microsoft\Windows NT\CurrentVersion\Winlogon	NAME NOT FOUND	Desired Access: Read
10:23:51.847...	Malware_Build_Week_U3...	1008	RegCloseKey	HKEY\Software\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS	Desired Access: Read
10:23:51.847...	Malware_Build_Week_U3...	1008	RegOpenKey	HKEY\Software\Microsoft\Windows NT\CurrentVersion\Diagnostics	SUCCESS	Desired Access: Maximum Allowed
10:23:51.848...	Malware_Build_Week_U3...	1008	RegOpenKey	HKEY\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\ntdll.dll	NAME NOT FOUND	Desired Access: Read
10:23:51.848...	Malware_Build_Week_U3...	1008	RegCloseKey	HKEY\Software\Microsoft\Windows NT\CurrentVersion\Winlogon	NAME NOT FOUND	Desired Access: Read
10:23:51.848...	Malware_Build_Week_U3...	1008	RegCreateKey	HKEY\Software\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS	Desired Access: Read
10:23:51.850...	Malware_Build_Week_U3...	1008	RegSetValue	HKEY\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\GinaDLL	SUCCESS	Desired Access: Read
10:23:51.850...	Malware_Build_Week_U3...	1008	RegCloseKey	HKEY\Software\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS	Type: REG_SZ, Length: 520, Data: C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\msgina32.dll

Notiamo che viene creata la chiave **HKLM\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Winlogon** (Operation RegCreateKey) e le viene assegnato il valore **GinaDLL** (Operation RegSetValue) con Descrizione aggiuntiva nella sezione *Details*.

Per confermare la Presenza di tale valore siamo ricorsi all'utilizzo di **Regedit** (che apre l'editor di **Registro di Windows**) andando all'interno del Registro di **Winlogon** e all'utilizzo di **OllyDBG** per confermare il comportamento del **Malware**.

Name	Type	Data
(Default)	REG_SZ	(value not set)
allocatedcdroms	REG_SZ	0
allocatedasd	REG_SZ	0
allocatefloppies	REG_SZ	0
AllowMultipleTSSessions	REG_DWORD	0x00000001 (1)
AltDefaultDomainName	REG_SZ	MALWARE_TEST
AltDefaultUserName	REG_SZ	Administrator
AutoRestartShell	REG_DWORD	0x00000001 (1)
Background	REG_SZ	0 0 0
cachedlogonscount	REG_SZ	10
DebugServerCommand	REG_SZ	no
DefaultDomainName	REG_SZ	MALWARE_TEST
DefaultUserName	REG_SZ	Administrator
ForceunlockLogon	REG_DWORD	0x00000000 (0)
GinaDLL	REG_SZ	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\msgina32.dll
HibernationPreviouslyEnabled	REG_DWORD	0x00000001 (1)
LegalNoticeCaption	REG_SZ	
LegalNoticeText	REG_SZ	
LogonType	REG_DWORD	0x00000001 (1)
passwordexpirywarning	REG_DWORD	0x0000000e (14)
powerdownAfterShutdown	REG_SZ	0


```

00401000: $ 55 PUSH EBP
00401001: . 8BEC MOV EBP,ESP
00401003: . 51 PUSH ECX
00401004: . 6A 00 PUSH 0
00401005: . 8045 FC LEA EAX,DWORD PTR SS:[EBP-4]
00401009: . 6A 00 PUSH EAX
0040100C: . 28 3F 0000F00 PUSH 0F00SF
00401011: . 6A 00 PUSH 0
00401013: . 6A 00 PUSH 0
00401017: . 6A 00 PUSH 0
0040101D: . 68 54004000 PUSH Malware_.00400054
00401021: . C7 04000000 CALL DWORD PTR DS:[<&ADVAPI32.RegCreate...
00401022: . FF15 04704000 TEST EAX,EAX
00401029: . 55 PUSH EBP
0040102A: . 8BEC MOV EBP,ESP
00401030: . 51 PUSH ECX
00401031: . 6A 01 PUSH 1
00401034: . 6A 00 PUSH 0
0040103E: . 68 40C00000 PUSH Malware_.0040004C
00401044: . C7 04000000 CALL DWORD PTR DS:[<&ADVAPI32.RegSetVa...
00401047: . FF15 00704000 TEST EAX,EAX
0040104D: . 55 PUSH EBP
0040104E: . 8BEC MOV EBP,ESP
00401050: . 51 PUSH ECX
00401053: . 6A 01 PUSH 1
00401056: . 68 40C00000 PUSH Malware_.0040004C
0040105C: . C7 04000000 CALL DWORD PTR DS:[<&ADVAPI32.RegSetVa...
0040105F: . FF15 00704000 TEST EAX,EAX
00401062: . 55 PUSH EBP
00401063: . 8BEC MOV EBP,ESP
00401065: . 51 PUSH ECX
00401066: . 6A 01 PUSH 1
00401069: . 68 40C00000 PUSH Malware_.0040004C
00401075: . C7 04000000 CALL DWORD PTR DS:[<&ADVAPI32.RegSetVa...
00401078: . FF15 00704000 TEST EAX,EAX
0040107B: . 55 PUSH EBP
0040107C: . 8BEC MOV EBP,ESP
0040107E: . 51 PUSH ECX
00401081: . 6A 01 PUSH 1
00401084: . 68 40C00000 PUSH Malware_.0040004C
0040108A: . C7 04000000 CALL DWORD PTR DS:[<&ADVAPI32.RegSetVa...
0040108D: . FF15 00704000 TEST EAX,EAX
00401090: . 55 PUSH EBP
00401091: . 8BEC MOV EBP,ESP
00401093: . 51 PUSH ECX
00401094: . 6A 01 PUSH 1
00401097: . 68 40C00000 PUSH Malware_.0040004C
0040109D: . C7 04000000 CALL DWORD PTR DS:[<&ADVAPI32.RegSetVa...
004010A0: . FF15 00704000 TEST EAX,EAX
004010A3: . 55 PUSH EBP
004010A4: . 8BEC MOV EBP,ESP
004010A6: . 51 PUSH ECX
004010A9: . 6A 01 PUSH 1
004010AC: . 68 40C00000 PUSH Malware_.0040004C
004010B2: . C7 04000000 CALL DWORD PTR DS:[<&ADVAPI32.RegSetVa...
004010B5: . FF15 00704000 TEST EAX,EAX
004010B8: . 55 PUSH EBP
004010B9: . 8BEC MOV EBP,ESP
004010BC: . 51 PUSH ECX
004010BD: . 6A 01 PUSH 1
004010C0: . 68 40C00000 PUSH Malware_.0040004C
004010C6: . C7 04000000 CALL DWORD PTR DS:[<&ADVAPI32.RegSetVa...
004010C9: . FF15 00704000 TEST EAX,EAX
004010CC: . 55 PUSH EBP
004010CD: . 8BEC MOV EBP,ESP
004010CE: . 51 PUSH ECX
004010D1: . 6A 01 PUSH 1
004010D4: . 68 40C00000 PUSH Malware_.0040004C
004010DA: . C7 04000000 CALL DWORD PTR DS:[<&ADVAPI32.RegSetVa...
004010DD: . FF15 00704000 TEST EAX,EAX
004010E0: . 55 PUSH EBP
004010E1: . 8BEC MOV EBP,ESP
004010E3: . 51 PUSH ECX
004010E6: . 6A 01 PUSH 1
004010E9: . 68 40C00000 PUSH Malware_.0040004C
004010F5: . C7 04000000 CALL DWORD PTR DS:[<&ADVAPI32.RegSetVa...
004010F8: . FF15 00704000 TEST EAX,EAX
004010FB: . 55 PUSH EBP
004010FC: . 8BEC MOV EBP,ESP
004010FD: . 51 PUSH ECX
004010FD: . 6A 01 PUSH 1
004010FD: . 68 40C00000 PUSH Malware_.0040004C
004010FD: . C7 04000000 CALL DWORD PTR DS:[<&ADVAPI32.RegSetVa...
004010FD: . FF15 00704000 TEST EAX,EAX
004010FD: . 55 PUSH EBP
004010FD: . 8BEC MOV EBP,ESP
004010FD: . 51 PUSH ECX
004010FD: . 6A 01 PUSH 1
004010FD: . 68 40C00000 PUSH Malware_.0040004C
004010FD: . C7 04000000 CALL DWORD PTR DS:[<&ADVAPI32.RegSetVa...
004010FD: . FF15 00704000 TEST EAX,EAX
004010FD: . 55 PUSH EBP
004010FD: . 8BEC MOV EBP,ESP
004010FD: . 51 PUSH ECX
004010FD: . 6A 01 PUSH 1
004010FD: . 68 40C00000 PUSH Malware_.0040004C
004010FD: . C7 04000000 CALL DWORD PTR DS:[<&ADVAPI32.RegSetVa...
004010FD: . FF15 00704000 TEST EAX,EAX
004010FD: . 55 PUSH EBP
004010FD: . 8BEC MOV EBP,ESP
004010FD: . 51 PUSH ECX
004010FD: . 6A 01 PUSH 1
004010FD: . 68 40C00000 PUSH Malware_.0040004C
004010FD: . C7 04000000 CALL DWORD PTR DS:[<&ADVAPI32.RegSetVa...
004010FD: . FF15 00704000 TEST EAX,EAX
004010FD: . 55 PUSH EBP
004010FD: . 8BEC MOV EBP,ESP
004010FD: . 51 PUSH ECX
004010FD: . 6A 01 PUSH 1
004010FD: . 68 40C00000 PUSH Malware_.0040004C
004010FD: . C7 04000000 CALL DWORD PTR DS:[<&ADVAPI32.RegSetVa...
004010FD: . FF15 00704000 TEST EAX,EAX
004010FD: . 55 PUSH EBP
004010FD: . 8BEC MOV EBP,ESP
004010FD: . 51 PUSH ECX
004010FD: . 6A 01 PUSH 1
004010FD: . 68 40C00000 PUSH Malware_.0040004C
004010FD: . C7 04000000 CALL DWORD PTR DS:[<&ADVAPI32.RegSetVa...
004010FD: . FF15 00704000 TEST EAX,EAX
004010FD: . 55 PUSH EBP
004010FD: . 8BEC MOV EBP,ESP
004010FD: . 51 PUSH ECX
004010FD: . 6A 01 PUSH 1
004010FD: . 68 40C00000 PUSH Malware_.0040004C
004010FD: . C7 04000000 CALL DWORD PTR DS:[<&ADVAPI32.RegSetVa...
004010FD: . FF15 00704000 TEST EAX,EAX
004010FD: . 55 PUSH EBP
004010FD: . 8BEC MOV EBP,ESP
004010FD: . 51 PUSH ECX
004010FD: . 6A 01 PUSH 1
004010FD: . 68 40C00000 PUSH Malware_.0040004C
004010FD: . C7 04000000 CALL DWORD PTR DS:[<&ADVAPI32.RegSetVa...
004010FD: . FF15 00704000 TEST EAX,EAX
004010FD: . 55 PUSH EBP
004010FD: . 8BEC MOV EBP,ESP
004010FD: . 51 PUSH ECX
004010FD: . 6A 01 PUSH 1
004010FD: . 68 40C00000 PUSH Malware_.0040004C
004010FD: . C7 04000000 CALL DWORD PTR DS:[<&ADVAPI32.RegSetVa...
004010FD: . FF15 00704000 TEST EAX,EAX
004010FD: . 55 PUSH EBP
004010FD: . 8BEC MOV EBP,ESP
004010FD: . 51 PUSH ECX
004010FD: . 6A 01 PUSH 1
004010FD: . 68 40C00000 PUSH Malware_.0040004C
004010FD: . C7 04000000 CALL DWORD PTR DS:[<&ADVAPI32.RegSetVa...
004010FD: . FF15 00704000 TEST EAX,EAX
004010FD: . 55 PUSH EBP
004010FD: . 8BEC MOV EBP,ESP
004010FD: . 51 PUSH ECX
004010FD: . 6A 01 PUSH 1
004010FD: . 68 40C00000 PUSH Malware_.0040004C
004010FD: . C7 04000000 CALL DWORD PTR DS:[<&ADVAPI32.RegSetVa...
004010FD: . FF15 00704000 TEST EAX,EAX
004010FD: . 55 PUSH EBP
004010FD: . 8BEC MOV EBP,ESP
004010FD: . 51 PUSH ECX
004010FD: . 6A 01 PUSH 1
004010FD: . 68 40C00000 PUSH Malware_.0040004C
004010FD: . C7 04000000 CALL DWORD PTR DS:[<&ADVAPI32.RegSetVa...
004010FD: . FF15 00704000 TEST EAX,EAX
004010FD: . 55 PUSH EBP
004010FD: . 8BEC MOV EBP,ESP
004010FD: . 51 PUSH ECX
004010FD: . 6A 01 PUSH 1
0
```

DAY 3

5) FILE SYSTEM

Ulteriore richiesta che viene data dalla traccia è di controllare eventuali **Chiamate di Sistema** che sono andate a modificare il File System della macchina. Per fare questo da ProcMon abbiamo utilizzato la **Tab** apposita riguardante **i File System**.

Notiamo che ci sono due Operazioni di CreateFile e WriteFile che vanno a Creare e Scrivere il file **msgina32.dll** all'interno della Cartella del Malware citata nel **Punto 3**.

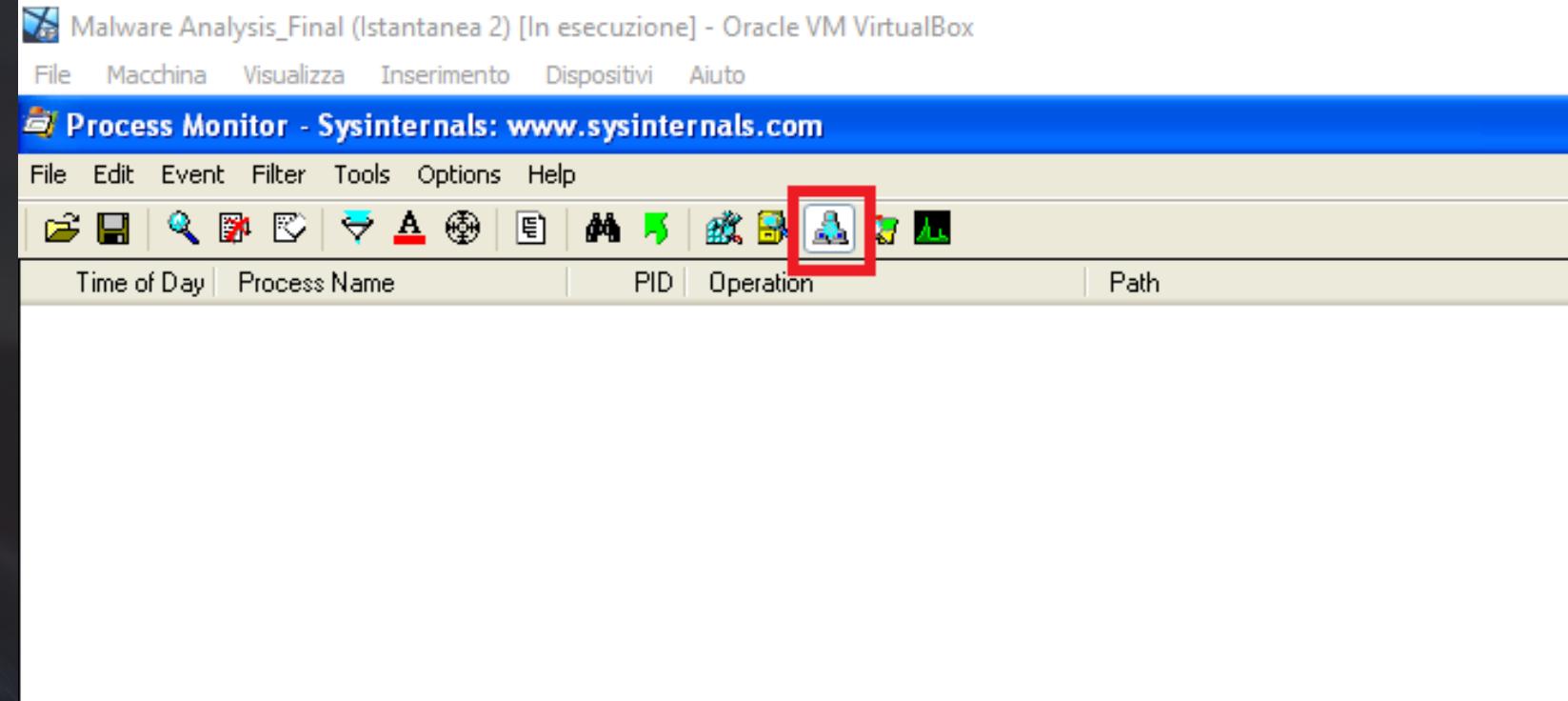
10:23:51.7933...	Malware_Build_Week_U3...	1008	CreateFile	C:\WINDOWS\system32\setupapi.dll	SUCCESS	Desired Access: Execute/Traverse, Disposition: Open, Options: Non-Directory File, Attributes: N, ShareMode: Read, Write, Delete, Alloc...
10:23:51.7946...	Malware_Build_Week_U3...	1008	CreateFile	C:\WINDOWS\system32\shlwapi.dll	SUCCESS	Desired Access: Execute/Traverse, Disposition: Open, Options: Non-Directory File, Attributes: N, ShareMode: Read, Write, Delete, Alloc...
10:23:51.7953...	Malware_Build_Week_U3...	1008	CreateFile	C:\WINDOWS\system32\apphelp.dll	SUCCESS	Desired Access: Execute/Traverse, Disposition: Open, Options: Non-Directory File, Attributes: N, ShareMode: Read, Write, Delete, Alloc...
10:23:51.8295...	Malware_Build_Week_U3...	1008	CreateFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3	SUCCESS	Desired Access: Execute/Traverse, Synchronize, Disposition: Open, Options: Directory, Synchronous IO Non-Alert, Attributes: n/a, Shar...
10:23:51.8435...	Malware_Build_Week_U3...	1008	CreateFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\msgina32.dll	SUCCESS	Desired Access: Generic Write, Read Attributes, Disposition: Overwrite, Options: Synchronous IO Non-Alert, Non-Directory File, Attribut...
10:23:51.8437...	Malware_Build_Week_U3...	1008	CreateFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3	SUCCESS	Desired Access: Synchronize, Disposition: Open, Options: Directory, Synchronous IO Non-Alert, Open For Backup, Attributes: N, Share...
10:23:51.8449...	Malware_Build_Week_U3...	1008	WriteFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\msgina32.dll	SUCCESS	Offset: 0, Length: 4.096
10:23:51.8457...	Malware_Build_Week_U3...	1008	WriteFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\msgina32.dll	SUCCESS	Offset: 4.096, Length: 2.560
10:35:58.9019...	Malware_Build_Week_U3...	1012	CreateFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\Malware_Build_Week_U3.exe	SUCCESS	Desired Access: Generic Read, Disposition: Open, Options: Synchronous IO Non-Alert, Attributes: n/a, ShareMode: Read, Write, Delete...
10:35:58.9625...	Malware_Build_Week_U3...	1012	CreateFile	C:\WINDOWS\system32\ntdll.dll	SUCCESS	Desired Access: Generic Read, Disposition: Open, Options: Synchronous IO Non-Alert, Attributes: n/a, ShareMode: Read, Write, Delete...
10:35:58.9713...	Malware_Build_Week_U3...	1012	CreateFile	C:\WINDOWS\Prefetch\MALWARE_BUILD_WEEK_U3.EXE-0E171D0F.pf	SUCCESS	Desired Access: Generic Read, Disposition: Open, Options: Synchronous IO Non-Alert, Attributes: n/a, ShareMode: None, AllocationSiz...
10:35:58.9721...	Malware_Build_Week_U3...	1012	CreateFile	C:	SUCCESS	Desired Access: Read Attributes, Write Attributes, Synchronize, Disposition: Open, Options: Synchronous IO Non-Alert, Attributes: n/a, S...

DAY 3

5) EXTRA

Per quanto riguarda questa analisi abbiamo voluto effettuare delle procedure Extra.

Per prima cosa abbiamo analizzato da ProcMon la Tab riguardante le Network Activity per dimostrare ulteriormente che il Malware non effettua alcuna operazione su Internet, motivo per cui ci risulta superfluo al momento l'utilizzo di ApateDNS e Wireshark (come emerso infatti dall'Analisi Statica all'interno dell'eseguibile non sono presenti librerie come WinInet.dll o Wsock32.dll, utilizzate per Funzioni di Rete).



DAY 3

5) EXTRA

Abbiamo inoltre utilizzato il tool Regshot per effettuare due Instantanee del Registro di Windows Prima e Dopo l'esecuzione del Malware per mostrare le modifiche che sono state effettuate.

Come possiamo vedere dalla figura sopra la **Chiave di Registro** `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\GinaDLL` viene aggiunta ed è utilizzata dal Malware per ottenere Persistenza.

DAY 3

5) EXTRA

Abbiamo utilizzato inoltre **l'Utility String** per approfondire lo Studio del Malware. Oltre alle **Stringhe** di **Funzioni** che abbiamo potuto notare nell'Analisi Statica notiamo anche la presenza di **msutil32.sys** che andremo ad approfondire ulteriormente nel Giorno 4 della Build Week.

```
GinaDLL
Software\Microsoft\Windows NT\CurrentVersion\Winlogon
MSGina.dll
UN %s DM %s PW %s OLD %s
WlxLoggedOut$AS
ErrorCode:%d ErrorMessage:%s.
%z %s - %s
msutil32.sys
0&0-080j0z0
3!313A3Q3a3q3
4#4*474B4Y4
4Z5c5
6!686b6h6n6t6z6
7"7<7J7<7
```

Infine abbiamo provato a chiudere la Macchina Windows e abbiamo notato che la Schermata di Disconnessione è stata modificata e non è più quella di Windows XP.



DAY 4

TASK 1: Cosa può succedere se il file .dll lecito viene sostituito con un file dll.malevolo che intercetta i dati inseriti?

Se il file .dll venisse sostituito con un file .dll malevolo, si potrebbe andar incontro a spiacevoli conseguenze, quali:

- 1. Controllo degli accessi:** La DLL malevola potrebbe modificare il comportamento del sistema operativo per quanto riguarda **l'accesso non autorizzato alla macchina infetta, ignorando di conseguenza le restrizioni di accesso alla stessa**, motivo per cui **un utente malintenzionato potrebbe raggiungere tranquillamente determinate risorse o aree riservate, avendo la possibilità di ricavare informazioni da file sensibili.**
- 2. Furto di credenziali:** La DLL malevola potrebbe **intercettare e registrare le credenziali di accesso degli utenti, come nomi utente e password**. Queste informazioni **potrebbero essere utilizzate per scopi di accesso non autorizzato o per scopi fraudolenti.**
- 3. Monitoraggio delle attività:** La DLL dannosa potrebbe **registrare e monitorare le attività dell'utente sul sistema, inclusi i siti web visitati, le applicazioni utilizzate** e le azioni compiute allo scopo di profilare l'utente, o **per vendere i dati raccolti a terze parti.**
- 4. Compromissione della sicurezza:** La DLL malevola potrebbe **indebolire o compromettere i meccanismi di sicurezza del sistema operativo**. Ad esempio, potrebbe **disabilitare l'antivirus o il firewall, consentendo l'accesso a malware aggiuntivi o permettendo l'attuazione di attacchi più dannosi sul sistema.**
- 5. Privilege escalation:** La DLL malevola potrebbe includere codice dannoso che viene eseguito nel contesto del processo o dell'applicazione in cui viene caricata, al fine di **compiere un processo di privilege escalation**, consentendo all'attaccante di **assumere pieno controllo del sistema o di compiere azioni dannose bypassando tutte le restrizioni del caso.**

Abbiamo deciso pertanto di analizzare lo scopo della nostra DLL malevola per verificarne il suo comportamento. A tal proposito, dopo aver notato le differenze tra la schermata precedente di login e logout di Windows e quella attuale, il nostro studio si è incentrato sull'analisi della DLL malevola **msgina32.dll con il disassembler IDA Pro**, grazie al quale abbiamo rintracciato altresì il file **msutil32.sys**, il quale viene passato come parametro alla funzione call _wfopen tramite l'istruzione **push offset aMsuti32_sys**.

Abbiamo fatto delle ricerche online, dalla quale è emerso che **il file msutil32.sys non è un file di sistema noto o legittimo nel sistema operativo Windows**, motivo per cui la nostra ricerca è virata sull'analisi dello stesso.

DAY 4

TASK 1

FORMATO TESTUALE

```
sub_10001570 proc near ; CODE XREF: WlxLoggedOutSAS+63↑p
    hMem      = dword ptr -854h
    var_850    = word ptr -850h
    var_828    = word ptr -828h
    var_800    = word ptr -800h
    dwMessageId = dword ptr 4
    arg_4      = dword ptr 8
    arg_8      = byte ptr 0Ch

    mov     ecx, [esp+arg_4]
    sub     esp, 854h
    lea     eax, [esp+854h+arg_8]
    lea     edx, [esp+854h+var_800]
    push    esi
    push    eax      ; va_list
    push    ecx      ; wchar_t *
    push    800h      ; size_t
    push    edx      ; wchar_t *
    call    _vsnwprintf
    push    offset word_10003320 ; wchar_t *
    push    offset aMsutil132_sys ; "msutil132.sys"
    call    _wfopen
    mov     esi, eax
    add     esp, 18h
    test   esi, esi
    jz     loc_1000164F
    lea     eax, [esp+858h+var_800]
    push    edi
    lea     ecx, [esp+85Ch+var_850]
    push    eax
    push    ecx      ; wchar_t *
```

FORMATO GRAFICO

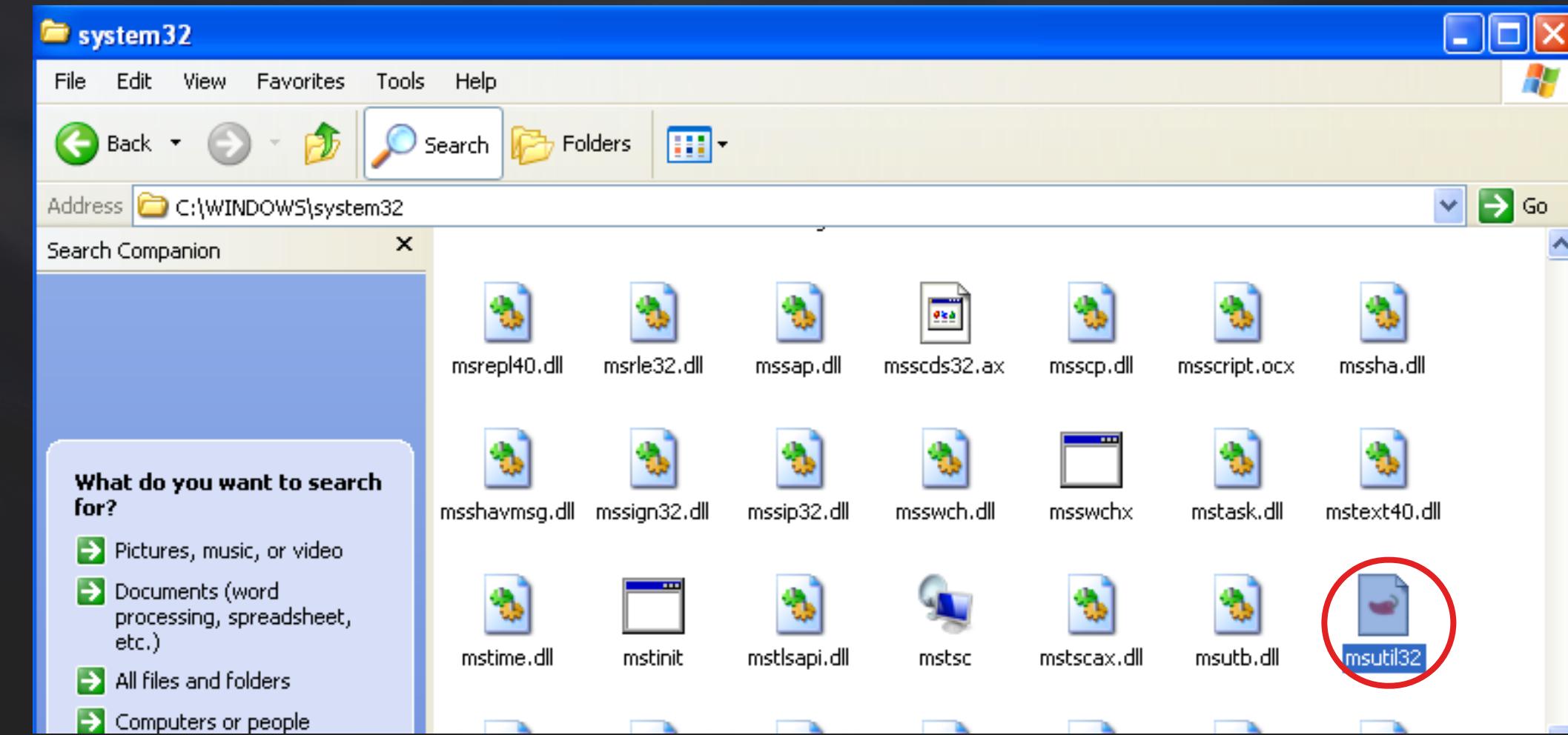
```
; int __cdecl sub_10001570(DWORD dwMessageId,wchar_t *,char)
sub_10001570 proc near
    hMem= dword ptr -854h
    var_850= word ptr -850h
    var_828= word ptr -828h
    var_800= word ptr -800h
    dwMessageId= dword ptr 4
    arg_4= dword ptr 8
    arg_8= byte ptr 0Ch

    mov     ecx, [esp+arg_4]
    sub     esp, 854h
    lea     eax, [esp+854h+arg_8]
    lea     edx, [esp+854h+var_800]
    push    esi
    push    eax      ; va_list
    push    ecx      ; wchar_t *
    push    800h      ; size_t
    push    edx      ; wchar_t *
    call    _vsnwprintf
    push    offset word_10003320 ; wchar_t *
    push    offset aMsutil132_sys ; "msutil132.sys"
    call    _wfopen
    mov     esi, eax
    add     esp, 18h
    test   esi, esi
    jz     loc_1000164F
```

DAY 4

TASK 1

- Poiché il file è **.sys**, estensione tipica dei file di sistema, ci spostiamo all'interno della **directory di sistema C:\WINDOWS\system32**, dove notiamo la presenza del suddetto file generato dal malware.



- Di seguito il contenuto del file msutil32.sys, dal quale si nota che le credenziali di accesso vengono salvate al suo interno, corredate da **data, ora, nome utente (UN), nome della macchina target (DM), password attuale (PW) e password precedente (OLD)**

The screenshot shows a Notepad window titled 'msutil32 - Notepad'. The content of the file is a log of credential saves:

Date	Time	User	Machine	Current Password	Previous Password
07/17/23	16:02:43	- UN	Administrator	DM MALWARE_TEST	PW malware OLD (null)
07/17/23	22:55:03	- UN	Administrator	DM MALWARE_TEST	PW malware OLD (null)
07/17/23	23:44:21	- UN	Administrator	DM MALWARE_TEST	PW malware OLD (null)
07/18/23	09:08:13	- UN	Administrator	DM MALWARE_TEST	PW malware OLD (null)

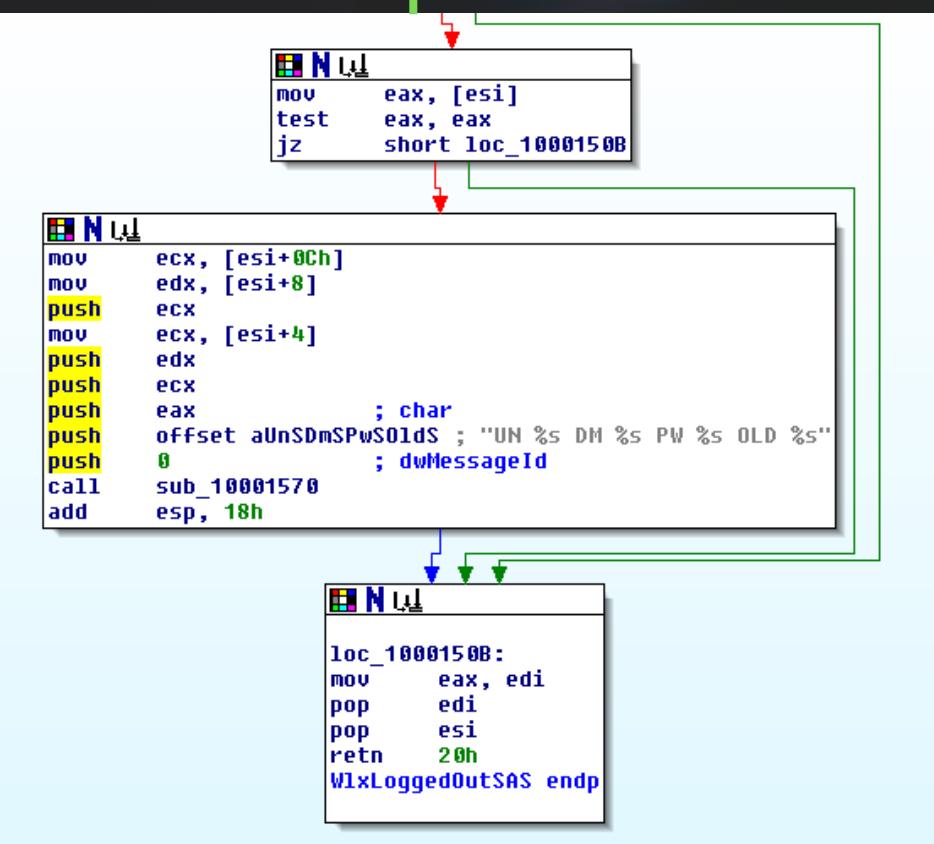
DAY 4

TASK 1

- La funzione **WlxLoggedOutSAS** fa parte della vecchia API GINA e generalmente viene chiamata quando un utente ha effettuato il logoff o l'arresto del sistema, momento in cui le informazioni di cui sopra (passw, nome utente, ecc...) vengono salvate all'interno del file **msunit32.sys**.

```
phToken= dword ptr 20h
pNprNotifyInfo= dword ptr 24h
pProfile= dword ptr 28h

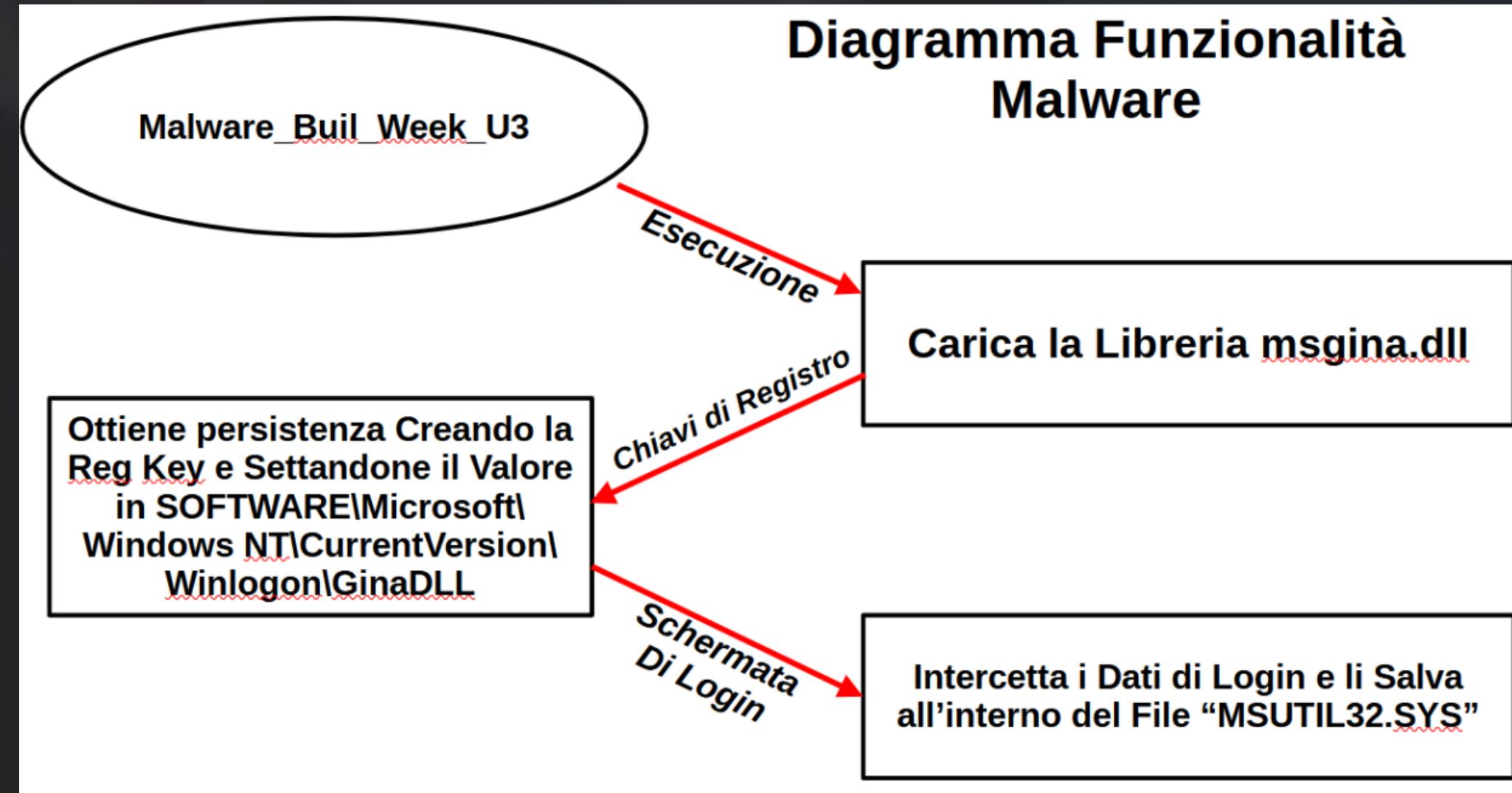
push    esi
push    edi
push    offset aWlxloggedoutsa ; "WlxLoggedOutSAS"
call    sub_10001000
push    64h
mov     edi, eax
call    ??2@YAPAXI@Z ; operator new(uint)
mov     eax, [esp+4+pProfile]
mov     esi, [esp+4+pNprNotifyInfo]
mov     ecx, [esp+4+phToken]
mov     edx, [esp+4+pdwOptions]
add    esp, 4
push    eax
mov     eax, [esp+4+pLogonSid]
push    esi
push    ecx
mov     ecx, [esp+0Ch+pAuthenticationId]
push    edx
mov     edx, [esp+10h+dwSasType]
push    eax
mov     eax, [esp+14h+pWlxContext]
push    ecx
push    edx
push    eax
call    edi
mov     edi, eax
cmp    edi, 1
jnz    short loc_1000150B
```



DAY 4

TASK 2: Sulla base della risposta sopra, delineate il profilo del Malware e delle sue funzionalità. Unire tutti i punti per creare un grafico che ne rappresenti lo scopo ad alto livello.

- Il malware analizzato è progettato per ottenere la persistenza su una macchina infetta attraverso la manipolazione della chiave di registro "**GinaDLL**". Si nota che, inizialmente, il malware verifica se la chiave di registro "**Gina.dll**" è presente in modo tale da modificarla; qualora suddetta chiave non sia presente, il malware provvederà alla creazione della stessa.
- La chiave di registro aggiunta è "**HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\GinaDLL**", la quale serve al malware per ottenere la persistenza, assicurando che venga caricata la sua libreria specifica (msgina32.dll) al momento dell'avvio del sistema.
- È stato notato che il malware **non effettua alcuna operazione su Internet** in quanto non si rileva alcuna attività di rete nell'analisi della scheda "Network Activity" del tool Process Monitor. Pertanto, l'uso di strumenti come ApateDNS e Wireshark risulta superfluo. Questa assenza di attività di rete può essere confermata anche durante la fase di analisi statica, in quanto non sono presenti librerie come WinINet.dll o WSock32.dll, che vengono tipicamente importate per abilitare le funzioni di rete.
- Inoltre, è stato rintracciato il file "**msutil32.sys**", non riconosciuto come un file di sistema legittimo di Windows, il che solleva preoccupazioni riguardo alla sua natura dannosa e alla possibilità che contenga dati sensibili rubati dal malware.
- Tale file si trova nella directory di sistema "**C:\WINDOWS\system32**" ed è stato generato dal malware stesso; al suo interno **vengono salvate le credenziali di accesso**, tra cui **data, ora, nome utente (UN), nome della macchina target (DM), password attuale (PW) e password precedente (OLD)**.
- In conclusione, il malware analizzato è **stato progettato per ottenere persistenza e svolgere azioni dannose sul sistema infetto**. La sua assenza di attività di rete suggerisce che **agisca localmente**, e il file "**msutil32.sys**" è **utilizzato per memorizzare le credenziali di accesso rubate per scopi illeciti**.





BUILDWEEK 3 GRUPPO 4

EPICODE

Team : Pisapia Giovanni, Carmine Caputo, Matteo Addei,
Riccardo Brunotti, Michele Macrì, Gaspare Rizzo, Pierluigi
Amorese, Riccardo Nardecchia.

DAY 5

TASK 1: Analizzare il file calcolatriceinnovativa50.exe e confermare che è un malware

- Come primo passo della nostra analisi decidiamo di analizzare le **stringhe** all' interno del file malevolo per acquisire qualche informazione rilevante. con il tool **STRINGS**
- Si noti la presenza di **"PADDINGXXpadding"**, la quale, all'interno di un malware **potrebbe indicare l'uso di tecniche di riempimento (padding) per mascherare o nascondere dati o codice malevoli all'interno delle risorse. Il padding è una tecnica comune utilizzata nei malware avanzati per aggiungere dati inutili o spazi vuoti all'interno delle risorse dell'eseguibile**, modificando la struttura del file in modo che risulti più difficile per gli analisti identificare il malware o estrarre il suo contenuto dannoso.
- L'uso di **"XX"** all'interno del termine potrebbe indicare un valore specifico

```
And
Xor
Lsh
Mod
x^y
Int
Not
sin
cos
log
sqrt
x^2
x^3
1/x
dms
F-E
Exp
Ave
Sum
Dat
Hex
Dec
Oct
Bin
calc.hlp
calc.chm
PAI
VS_VERSION_INFO
StringFileInfo
041904B0
CompanyName
FileDescription
Windows
FileVersion
5.1.2600.0 <xpclient.010817-1148>
InternalName
CALC
LegalCopyright
OriginalFilename
CALC.EXE
ProductName
Microsoft
Windows
ProductVersion
5.1.2600.0
VarFileInfo
Translation
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">
<assemblyIdentity
    name="Microsoft.Windows.Shell.calc"
    processorArchitecture="x86"
    version="5.1.0.0"
    type="win32"/>
<description>Windows Shell</description>
<dependency>
    <dependentAssembly>
        <assemblyIdentity
            type="win32"
            name="Microsoft.Windows.Common-Controls"
            version="6.0.0.0"
            processorArchitecture="x86"
            publicKeyToken="6595b64144ccf1df"
            language="*"
        />
    </dependentAssembly>
</dependency>
</assembly>
PPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPA
GPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPA
GPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPA
GPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPA
GPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPA
G
```

DAY 5

TASK 1: Analizzare il file calcolatriceinnovativa50.exe e confermare che è un malware

- Tramite **CFF EXPLORER**, abbiamo notato che nella sezione .text è presente **SHELL32.dll**, il che ci fa presagire che probabilmente l'eseguibile in questione potrebbe caricare una shell in background sulla macchina target.

CFF Explorer interface showing the analysis of the executable **calcolatriceinnovativa50.exe**.

The left pane displays the file structure:

- File: calcolatriceinnovativa50.exe**
- e** (Export)
 - Dos Header
 - Nt Headers
 - File Header
 - Optional Header
 - Data Directories [x]
- Section Headers [x] (selected)
- Import Directory
- Resource Directory
- Debug Directory
- Address Converter
- Dependency Walker
- Hex Editor
- Identifier
- Import Adder
- Quick Disassembler
- Rebuilder
- Resource Editor
- UPX Utility

The right pane shows the **Section Headers** table:

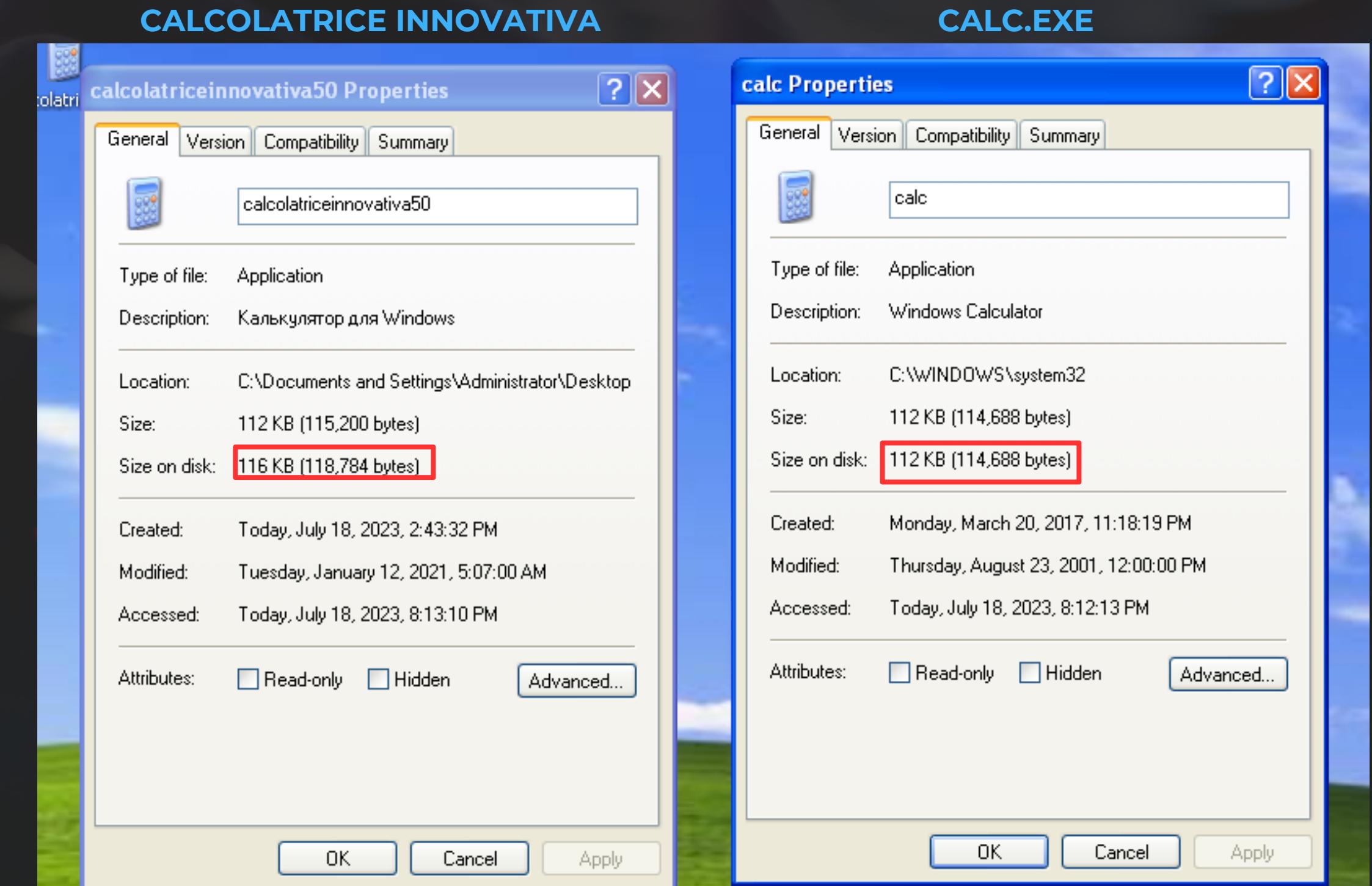
Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations ...	Linenumber...	Characteristics
000001E8	000001F0	000001F4	000001F8	000001FC	00000200	00000204	00000208	0000020A	0000020C
Byte[8]	Dword	Dword	Dword	Dword	Dword	Word	Word	Word	Dword
.text	000126B0	00001000	00012800	00000400	00000000	00000000	0000	0000	60000020
.data	0000101C	00014000	00000A00	00012C00	00000000	00000000	0000	0000	C0000040
.rsrc	00008A70	00016000	00008C00	00013600	00000000	00000000	0000	0000	40000040

The bottom section shows the **Hex Editor** view with the **Section Headers** table above it. A red box highlights the memory dump of the **.text** section, showing assembly code and ASCII characters. The assembly code includes references to **SHELL32.dll** functions like **_CxxFrameHandle** and **_CxxThrowExc**.

DAY 5

TASK 1: Analizzare il file calcolatriceinnovativa50.exe e confermare che è un malware

- Abbiamo fatto un confronto con il file .exe in questione e la calcolatrice di Windows, dal quale è emerso che il file .exe ha una dimensione maggiore rispetto alla calcolatrice di circa **4 KB**, probabilmente dovuto all'aggiunta di codice malevolo all'interno dello stesso.



DAY 5

TASK 1: Analizzare il file calcolatriceinnovativa50.exe e confermare che è un malware

- Mettendo a paragone il file calc.exe con il file calcolatriceinnovativa.exe, si nota che, come visto precedentemente con STRINGS, **le sezioni .rsrc di entrambi gli eseguibili sono diverse tra loro, in quanto la dimensione del secondo risulta più grande del primo.**

calc.exe							
Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations ...
00000238	00000240	00000244	00000248	0000024C	00000250	00000254	00000258
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word
.text	000126B0	00001000	00012800	00000400	00000000	00000000	0000
.data	0000101C	00014000	00000A00	00012C00	00000000	00000000	0000
.rsrc	00008960	00016000	00008A00	00013600	00000000	00000000	0000

calcolatriceinnovativa50.exe									
Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations ...	Linenumber... Cha	Cha
00000238	00000240	00000244	00000248	0000024C	00000250	00000254	00000258	00000254	000
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dwc
.text	000126B0	00001000	00012800	00000400	00000000	00000000	0000	0000	600
.data	0000101C	00014000	00000A00	00012C00	00000000	00000000	0000	0000	C00
.rsrc	00008A70	00016000	00008C00	00013600	00000000	00000000	0000	0000	400

DAY 5

TASK 1: Analizzare il file calcolatriceinnovativa50.exe e confermare che è un malware

- Dall'analisi dell'ascii di entrambi gli eseguibili è possibile notare che la sezione .rsrc del file **calcolatriceinnovativa.exe ha dimensione maggiore in quanto contiene stringhe di PADDING**

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Ascii
000007E0	69	63	4B	65	79	54	6F	6B	65	6E	3D	22	36	35	39	35	icKeyToken="6595
000007F0	62	36	34	31	34	34	63	63	66	31	64	66	22	0D	0A	20	b64144ccf1df"...
00000800	20	20	20	20	20	20	20	20	20	20	6C	61	6E	67	75	langu
00000810	61	67	65	3D	22	2A	22	0D	0A	20	20	20	20	20	20	20	age="*".
00000820	20	2F	3E	0D	0A	20	20	20	20	3C	2F	64	65	70	65	6E	
00000830	64	65	6E	74	41	73	73	65	6D	62	6C	79	3E	0D	0A	3C	dentAssembly>...<
00000840	2F	64	65	70	65	6E	64	65	6E	63	79	3E	0D	0A	3C	2F	/dependency>...</
00000850	61	73	73	65	6D	62	6C	79	3E	0D	0A	00	00	00	00	00	assembly>.....
00000860	01	00	FF	FF	00	00	00	00	00	04	00	40	00	CA	00		I.yy.....I.@.E.
00000870	4E	00	00	80	00	00	3C	01	A4	00	FF	FF	6B	00	53	00	N..I..<I@.yyk.S.
00000880	63	00	69	00	43	00	61	00	6C	00	63	00	00	00	43	00	c.i.C.a.l.c...C.
00000890	61	00	6C	00	63	00	75	00	6C	00	61	00	74	00	6F	00	a.l.c.u.l.a.t.o.
000008A0	72	00	00	00	08	00	00	00	00	01	4D	00	53	00	20	00	r...I...M.S...
000008B0	53	00	68	00	65	00	6C	00	6C	00	20	00	44	00	6C	00	S.h.e.l.l...D.1.
000008C0	67	00	00	00	00	00	00	00	00	00	00	02	08	81	50	g.....I...P	
000008D0	05	00	01	00	31	01	0E	00	93	01	00	00	FF	FF	81	00	I..I...I...yy.
000008E0	00	00	00	00	00	00	00	00	00	00	00	03	00	01	50I...P	
000008F0	31	00	29	00	22	00	0A	00	8D	00	00	00	FF	FF	80	00	1.)"....I...yy.

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Ascii
00008A10	22	0D	0A	20	20	20	20	20	20	20	20	20	20	20	20	6C	".....1
00008A20	61	6E	67	75	61	67	65	3D	22	2A	22	0D	0A	20	20	20	angage="*".
00008A30	20	20	20	20	20	2F	3E	0D	0A	20	20	20	20	3C	2F	64	
00008A40	65	70	65	6E	64	65	6E	74	41	73	73	65	6D	62	6C	79	dependentAssembly>...</dependency>..</assembly>..P
00008A50	3E	0D	0A	3C	2F	64	65	70	65	6E	64	65	6E	63	79	3E	PADDINGXXPADDING
00008A60	0D	0A	3C	2F	61	73	73	65	6D	62	6C	79	3E	0D	0A	50	PADDINGXXPADDING
00008A70	50	41	44	44	49	4E	47	58	58	50	41	44	44	49	4E	47	PADDINGXXPADDING
00008A80	50	41	44	44	49	4E	47	58	58	50	41	44	44	49	4E	47	PADDINGXXPADDING
00008A90	50	41	44	44	49	4E	47	58	58	50	41	44	44	49	4E	47	PADDINGXXPADDING
00008AA0	50	41	44	44	49	4E	47	58	58	50	41	44	44	49	4E	47	PADDINGXXPADDING
00008AB0	50	41	44	44	49	4E	47	58	58	50	41	44	44	49	4E	47	PADDINGXXPADDING
00008AC0	50	41	44	44	49	4E	47	58	58	50	41	44	44	49	4E	47	PADDINGXXPADDING
00008AD0	50	41	44	44	49	4E	47	58	58	50	41	44	44	49	4E	47	PADDINGXXPADDING
00008AE0	50	41	44	44	49	4E	47	58	58	50	41	44	44	49	4E	47	PADDINGXXPADDING
00008AF0	50	41	44	44	49	4E	47	58	58	50	41	44	44	49	4E	47	PADDINGXXPADDING
00008B00	50	41	44	44	49	4E	47	58	58	50	41	44	44	49	4E	47	PADDINGXXPADDING
00008B10	50	41	44	44	49	4E	47	58	58	50	41	44	44	49	4E	47	PADDINGXXPADDING
00008B20	50	41	44	44	49	4E	47	58	58	50	41	44	44	49	4E	47	PADDINGXXPADDING
00008B30	50	41	44	44	49	4E	47	58	58	50	41	44	44	49	4E	47	PADDINGXXPADDING

DAY 5

TASK 1: Analizzare il file calcolatriceinnovativa50.exe e confermare che è un malware

- Abbiamo notato una differenza tra il valore del CHECKSUM del file calcolatriceinnovativa.exe e il file calc.exe. Nello specifico, il valore del CHECKSUM del file malevolo risulta pari a 0.
- Il "CHECKSUM" è un valore calcolato che serve a verificare l'integrità di un file eseguibile o di un'altra struttura dati, ed è spesso utilizzato nei file Portable Executable (PE) di Windows. Tuttavia, come nel nostro caso, un checksum con valore pari 0 può anche indicare che il file è stato danneggiato o che è stato modificato in modo malevolo, poiché un checksum valido dovrebbe essere diverso da 0 per indicare l'integrità del file.**

CALCOLATRICE INNOVATIVA

Member	Offset	Size	Value	Meaning
Magic	00000108	Word	010B	PE32
MajorLinkerVersion	0000010A	Byte	07	
MinorLinkerVersion	0000010B	Byte	00	
SizeOfCode	0000010C	Dword	00012800	
SizeOfInitializedData	00000110	Dword	00009C00	
SizeOfUninitializedData	00000114	Dword	00000000	
AddressOfEntryPoint	00000118	Dword	00012475	.text
BaseOfCode	0000011C	Dword	00001000	
BaseOfData	00000120	Dword	00014000	
ImageBase	00000124	Dword	01000000	
SectionAlignment	00000128	Dword	00001000	
FileAlignment	0000012C	Dword	00000200	
MajorOperatingSystemVersion	00000130	Word	0005	
MinorOperatingSystemVersion	00000132	Word	0001	
MajorImageVersion	00000134	Word	0005	
MinorImageVersion	00000136	Word	0001	
MajorSubsystemVersion	00000138	Word	0004	
MinorSubsystemVersion	0000013A	Word	0000	
Win32VersionValue	0000013C	Dword	00000000	
SizeOfImage	00000140	Dword	0001F000	
SizeOfHeaders	00000144	Dword	00000400	
CheckSum	00000148	Dword	0001D7FC	
Subsystem	0000014C	Word	0002	Windows GUI
DllCharacteristics	0000014E	Word	8000	Click here
SizeOfStackReserve	00000150	Dword	00040000	
SizeOfStackCommit	00000154	Dword	00001000	
SizeOfHeapReserve	00000158	Dword	00100000	
SizeOfHeapCommit	0000015C	Dword	00001000	
LoaderFlags	00000160	Dword	00000000	

CALC.EXE

Member	Offset	Size	Value	Meaning
Magic	00000108	Word	010B	PE32
MajorLinkerVersion	0000010A	Byte	07	
MinorLinkerVersion	0000010B	Byte	00	
SizeOfCode	0000010C	Dword	00012800	
SizeOfInitializedData	00000110	Dword	00009600	
SizeOfUninitializedData	00000114	Dword	00000000	
AddressOfEntryPoint	00000118	Dword	000034BA	.text
BaseOfCode	0000011C	Dword	00001000	
BaseOfData	00000120	Dword	00014000	
ImageBase	00000124	Dword	01000000	
SectionAlignment	00000128	Dword	00001000	
FileAlignment	0000012C	Dword	00000200	
MajorOperatingSystemVersion	00000130	Word	0005	
MinorOperatingSystemVersion	00000132	Word	0001	
MajorImageVersion	00000134	Word	0005	
MinorImageVersion	00000136	Word	0001	
MajorSubsystemVersion	00000138	Word	0004	
MinorSubsystemVersion	0000013A	Word	0000	
Win32VersionValue	0000013C	Dword	00000000	
SizeOfImage	00000140	Dword	0001F000	
SizeOfHeaders	00000144	Dword	00000400	
CheckSum	00000148	Dword	00000000	
Subsystem	0000014C	Word	0002	Windows GUI
DllCharacteristics	0000014E	Word	8000	Click here
SizeOfStackReserve	00000150	Dword	00040000	
SizeOfStackCommit	00000154	Dword	00001000	
SizeOfHeapReserve	00000158	Dword	00100000	
SizeOfHeapCommit	0000015C	Dword	00001000	
LoaderFlags	00000160	Dword	00000000	

DAY 5

TASK 1: Analizzare il file calcolatriceinnovativa50.exe e confermare che è un malware

- Altra differenza notata in sede di analisi è relativa al **valore dell'entry point** (punto di inizio dell'esecuzione del programma), **a conferma che il file calcolatriceinnovativa50.exe sia stato modificato rispetto al suo eseguibile originale.**

CALCOLATRICE INNOVATIVA

Member	Offset	Size	Value	Meaning
Magic	00000108	Word	010B	PE32
MajorLinkerVersion	0000010A	Byte	07	
MinorLinkerVersion	0000010B	Byte	00	
SizeOfCode	0000010C	Dword	00012800	
SizeOfInitializedData	00000110	Dword	00009C00	
SizeOfUninitializedData	00000114	Dword	00000000	
AddressOfEntryPoint	00000118	Dword	00012475	.text
BaseOfCode	0000011C	Dword	00001000	
BaseOfData	00000120	Dword	00014000	
ImageBase	00000124	Dword	01000000	
SectionAlignment	00000128	Dword	00001000	
FileAlignment	0000012C	Dword	00000200	
MajorOperatingSystemVersion	00000130	Word	0005	
MinorOperatingSystemVersion	00000132	Word	0001	
MajorImageVersion	00000134	Word	0005	
MinorImageVersion	00000136	Word	0001	
MajorSubsystemVersion	00000138	Word	0004	
MinorSubsystemVersion	0000013A	Word	0000	
Win32VersionValue	0000013C	Dword	00000000	
SizeOfImage	00000140	Dword	0001F000	
SizeOfHeaders	00000144	Dword	00000400	
CheckSum	00000148	Dword	0001D7FC	
Subsystem	0000014C	Word	0002	Windows GUI
DllCharacteristics	0000014E	Word	8000	Click here
SizeOfStackReserve	00000150	Dword	00040000	
SizeOfStackCommit	00000154	Dword	00001000	
SizeOfHeapReserve	00000158	Dword	00100000	
SizeOfHeapCommit	0000015C	Dword	00001000	
LoaderFlags	00000160	Dword	00000000	
NumberOfRvaAndSizes	00000164	Dword	00000010	

CALC.EXE

Member	Offset	Size	Value	Meaning
Magic	00000108	Word	010B	PE32
MajorLinkerVersion	0000010A	Byte	07	
MinorLinkerVersion	0000010B	Byte	00	
SizeOfCode	0000010C	Dword	00012800	
SizeOfInitializedData	00000110	Dword	00009600	
SizeOfUninitializedData	00000114	Dword	00000000	
AddressOfEntryPoint	00000118	Dword	000034BA	.text
BaseOfCode	0000011C	Dword	00001000	
BaseOfData	00000120	Dword	00014000	
ImageBase	00000124	Dword	01000000	
SectionAlignment	00000128	Dword	00001000	
FileAlignment	0000012C	Dword	00000200	
MajorOperatingSystemVersion	00000130	Word	0005	
MinorOperatingSystemVersion	00000132	Word	0001	
MajorImageVersion	00000134	Word	0005	
MinorImageVersion	00000136	Word	0001	
MajorSubsystemVersion	00000138	Word	0004	
MinorSubsystemVersion	0000013A	Word	0000	
Win32VersionValue	0000013C	Dword	00000000	
SizeOfImage	00000140	Dword	0001F000	
SizeOfHeaders	00000144	Dword	00000400	
CheckSum	00000148	Dword	00000000	
Subsystem	0000014C	Word	0002	Windows GUI
DllCharacteristics	0000014E	Word	8000	Click here
SizeOfStackReserve	00000150	Dword	00040000	
SizeOfStackCommit	00000154	Dword	00001000	
SizeOfHeapReserve	00000158	Dword	00100000	
SizeOfHeapCommit	0000015C	Dword	00001000	
LoaderFlags	00000160	Dword	00000000	
NumberOfRvaAndSizes	00000164	Dword	00000010	

DAY 5

TASK 1: Analizzare il file calcolatriceinnovativa50.exe e confermare che è un malware

- **SHELL32.dll** essenziale per la gestione dell'interfaccia utente di Windows, consentendo il controllo delle funzionalità della shell del sistema operativo.
- **msvcrt.dll** include una serie di funzioni standard del linguaggio C, contribuendo alla corretta esecuzione e gestione dei programmi scritti in C.
- **ADVAPI32.dll** è responsabile dell'interazione con le chiavi di registro del sistema operativo, consentendo l'accesso e la modifica delle informazioni cruciali per il funzionamento delle applicazioni e delle impostazioni di sistema.
- **KERNEL32.dll** è un componente fondamentale per l'interazione con il sistema operativo, facilitando operazioni come gestione dei processi, gestione della memoria e manipolazione dei file.
- **GDI32.dll** offre le funzioni necessarie per la gestione della componente grafica di Windows, inclusa la creazione di oggetti e immagini bidimensionali.
- **USER32.dll** contiene le API necessarie per la gestione e il funzionamento dell'interfaccia utente di Windows, consentendo il controllo delle finestre, dei controlli e delle interazioni con l'utente.

Module Name	Imports	OFTs	TimeStamp	ForwarderChain	Name RVA	FTs (IAT)
000125D4	N/A	00011FBC	00011FC0	00011FC4	00011FC8	00011FC
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
SHELL32.dll	1	00012CA8	FFFFFFFFF	FFFFFFFFF	00012E42	0000109
msvcrt.dll	26	00012DC8	FFFFFFFFF	FFFFFFFFF	00012F60	000011B
ADVAPI32.dll	3	00012C0C	FFFFFFFFF	FFFFFFFFF	00012FFC	0000100
KERNEL32.dll	30	00012C2C	FFFFFFFFF	FFFFFFFFF	000131D4	0000102
GDI32.dll	3	00012C1C	FFFFFFFFF	FFFFFFFFF	0001320C	0000101
USER32.dll	69	00012CB0	FFFFFFFFF	FFFFFFFFF	000136A4	000010A

DAY 5

TASK 1: Analizzare il file calcolatriceinnovativa50.exe e confermare che è un malware

Nella libreria KERNEL32.dll si notano funzioni note come:

- **LoadLibraryA**: Carica dinamicamente una libreria DLL durante l'esecuzione del programma, consentendo l'accesso a funzioni o risorse aggiuntive.
- **GetProcAddress**: Ottiene l'indirizzo di una funzione all'interno di una libreria DLL già caricata, permettendo di richiamarla durante l'esecuzione del programma.
- **CreateEventW**: Crea un oggetto "evento" di Windows per la sincronizzazione tra processi o thread, consentendo di attendere fino a quando l'evento viene segnalato.
- **CreateThread**: Crea un nuovo thread all'interno del processo in esecuzione, permettendo di svolgere attività in parallelo.
- **GetCommandLine**: Restituisce il comando con cui il programma è stato avviato, inclusi il nome dell'eseguibile e gli eventuali parametri passati all'avvio.
- **Sleep**: Sospende l'esecuzione del thread corrente per un periodo di tempo specificato, misurato in millisecondi, permettendo di aggiungere ritardi nell'esecuzione del programma.

OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
000131AE	77E79F93	0167	GetModuleHandleA
0001319E	77E805D8	022E	LoadLibraryA
0001318C	77E7A5FD	0189	GetProcAddress
0001317C	77E9A9AD	01D8	GlobalCompact
0001316E	77E736A3	01D7	GlobalAlloc
00013160	77E73803	01DE	GlobalFree
00013150	77E6E341	01E5	GlobalReAlloc
00013144	77E78D60	0393	IstricmpW
0001313C	77E61BE6	0329	Sleep
00013126	77E72A2B	0383	WriteProfileStringW
000131C2	77E6177A	019C	GetStartupInfoA
0001310A	77E6C879	01E6	GlobalSize
000130FA	77E71B14	01E9	GlobalUnlock
000130EA	77E730C1	0047	CreateEventW
000130DA	77E7AC37	0065	CreateThread

DAY 5

TASK 1: Analizzare il file calcolatriceinnovativa50.exe e confermare che è un malware

Nella libreria ADVAPI32.dll si notano le seguenti funzioni:

- **RegOpenKeyExA:** funzione utilizzata per aprire una chiave specifica nel Registro di sistema di Windows.
- **RegQueryValueExA:** funzione che consente di recuperare il valore associato a una specifica chiave nel Registro di sistema di Windows.
- **RegCloseKey:** funzione utilizzata per chiudere l'handle di una chiave del Registro di sistema precedentemente aperta con la funzione RegOpenKey.

OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
00012FEC	77DC22EA	01E1	RegOpenKeyExA
00012FD8	77DC23D7	01EB	RegQueryValueExA
00012FCA	77DC1894	01C8	RegCloseKey

DAY 5

TASK 1: Analizzare il file calcolatriceinnovativa50.exe e confermare che è un malware

- Dopo aver avviato il malware abbiamo utilizzato il tool Proc Mon per effettuare un'analisi dinamica sul file malevolo. Nello specifico **abbiamo notato che il file "crea" nuovi file e ne sovrascrive altri già presenti senza modificarli** (shell32.dll e comctl32.dll).
- Si nota altresì la creazione del file CALCOLATRICEINNOVATIVA50.EXE(pf all'interno della directory **C:\WINDOWS\Prefetch**. Nello specifico, i file ".pf" **presenti nella cartella "Prefetch" sono file di prefetch creati dal sistema operativo Windows per migliorare le prestazioni di avvio delle applicazioni**. Si potrebbe ipotizzare che il file eseguibile possa sfruttare la cartella "Prefetch" per mantenere una presenza persistente sul sistema.

Time of Day	Process Name	PID	Operation	Path	Result
10:55:11.5341...	calcolatriceinnovativa50.exe	1692	CreateFile	C:\WINDOWS\Prefetch\CALCOLATRICEINNOVATIVA50.EXE-06E7891A.pf	NAME NOT FOUND
10:55:11.5348...	calcolatriceinnovativa50.exe	1692	CreateFile	C:\WINDOWS\system32\shell32.dll	SUCCESS
10:55:11.5867...	calcolatriceinnovativa50.exe	1692	CreateFile	C:\WINDOWS\system32\SHELL32.dll.124.Manifest	SUCCESS
10:55:11.5872...	calcolatriceinnovativa50.exe	1692	CreateFile	C:\WINDOWS\system32\SHELL32.dll.124.Config	NAME NOT FOUND
10:55:11.5884...	calcolatriceinnovativa50.exe	1692	CreateFile	C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83	NAME NOT FOUND
10:55:11.6157...	calcolatriceinnovativa50.exe	1692	CreateFile	C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83\comctl32.dll	SUCCESS
10:55:11.6161...	calcolatriceinnovativa50.exe	1692	CreateFile	C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83\comctl32.dll	SUCCESS
10:55:11.6180...	calcolatriceinnovativa50.exe	1692	CreateFile	C:\WINDOWS\WindowsShell.Manifest	SUCCESS
10:55:11.6348...	calcolatriceinnovativa50.exe	1692	CreateFile	C:\WINDOWS\WindowsShell.Manifest	SUCCESS
10:55:11.6379...	calcolatriceinnovativa50.exe	1692	CreateFile	C:\WINDOWS\WindowsShell.Manifest	SUCCESS
10:55:11.6384...	calcolatriceinnovativa50.exe	1692	CreateFile	C:\WINDOWS\WindowsShell.Config	SUCCESS
10:55:11.6388...	calcolatriceinnovativa50.exe	1692	CreateFile	C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83	NAME NOT FOUND
10:55:11.6539...	calcolatriceinnovativa50.exe	1692	CreateFile	C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83	SUCCESS

DAY 5

TASK 1: Analizzare il file calcolatriceinnovativa50.exe e confermare che è un malware

- Abbiamo messo a confronto anche la **sezione “thread e processi”** del file calc.exe e del file calcolatriceinnovativa.exe, **dalla quale non si notano differenze di sorta**, anzi, da come da screenshot sottostante il file calc.exe svolge addirittura più operazioni rispetto al file malevolo.

Time of Day	Process Name	PID	Operation	Path
2:00:37.91239...	calc.exe	1532	Process Start	
2:00:37.91239...	calc.exe	1532	Thread Create	
2:00:37.94543...	calc.exe	1532	Load Image	C:\WINDOWS\system32\calc.exe
2:00:37.94579...	calc.exe	1532	Load Image	C:\WINDOWS\system32\ntdll.dll
2:00:38.07981...	calc.exe	1532	Load Image	C:\WINDOWS\system32\kernel32.dll
2:00:38.08115...	calc.exe	1532	Load Image	C:\WINDOWS\system32\shell32.dll
2:00:38.08152...	calc.exe	1532	Load Image	C:\WINDOWS\system32\advapi32.dll
2:00:38.08199...	calc.exe	1532	Load Image	C:\WINDOWS\system32\rpcrt4.dll
2:00:38.08249...	calc.exe	1532	Load Image	C:\WINDOWS\system32\secur32.dll
2:00:38.08285...	calc.exe	1532	Load Image	C:\WINDOWS\system32\gdi32.dll
2:00:38.08332...	calc.exe	1532	Load Image	C:\WINDOWS\system32\user32.dll
2:00:38.08372...	calc.exe	1532	Load Image	C:\WINDOWS\system32\msvcrt.dll
2:00:38.08431...	calc.exe	1532	Load Image	C:\WINDOWS\system32\shlwapi.dll
2:00:38.08872...	calc.exe	1532	Load Image	C:\WINDOWS\system32\shimeng.dll
2:00:38.09672...	calc.exe	1532	Load Image	C:\WINDOWS\AppPatch\AcGeneral.dll
2:00:38.09872...	calc.exe	1532	Load Image	C:\WINDOWS\system32\winmm.dll
2:00:38.09910...	calc.exe	1532	Load Image	C:\WINDOWS\system32\ole32.dll
2:00:38.09956...	calc.exe	1532	Load Image	C:\WINDOWS\system32\oleaut32.dll
2:00:38.10157...	calc.exe	1532	Load Image	C:\WINDOWS\system32\msacm32.dll
2:00:38.10216...	calc.exe	1532	Load Image	C:\WINDOWS\system32\version.dll
2:00:38.10255...	calc.exe	1532	Load Image	C:\WINDOWS\system32\userenv.dll
2:00:38.10473...	calc.exe	1532	Load Image	C:\WINDOWS\system32\uxtheme.dll
2:00:38.16399...	calc.exe	1532	Load Image	C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83\comctl32.dll
2:04:58.83663...	calc.exe	1532	Thread Exit	
2:04:58.83695...	calc.exe	1532	Process Exit	
2:05:09.78835...	calcolatriceinnovativa50.exe	1284	Thread Create	
2:05:09.81624...	calcolatriceinnovativa50.exe	1284	Load Image	C:\Documents and Settings\Administrator\Desktop\calcolatriceinnovativa50.exe
2:05:09.81660...	calcolatriceinnovativa50.exe	1284	Load Image	C:\WINDOWS\system32\ntdll.dll
2:05:09.81866...	calcolatriceinnovativa50.exe	1284	Load Image	C:\WINDOWS\system32\kernel32.dll
2:05:09.82079...	calcolatriceinnovativa50.exe	1284	Load Image	C:\WINDOWS\system32\shell32.dll
2:05:09.82116...	calcolatriceinnovativa50.exe	1284	Load Image	C:\WINDOWS\system32\advapi32.dll
2:05:09.82156...	calcolatriceinnovativa50.exe	1284	Load Image	C:\WINDOWS\system32\rpcrt4.dll
2:05:09.82193...	calcolatriceinnovativa50.exe	1284	Load Image	C:\WINDOWS\system32\secur32.dll
2:05:09.82231...	calcolatriceinnovativa50.exe	1284	Load Image	C:\WINDOWS\system32\gdi32.dll
2:05:09.82272...	calcolatriceinnovativa50.exe	1284	Load Image	C:\WINDOWS\system32\user32.dll
2:05:09.82311...	calcolatriceinnovativa50.exe	1284	Load Image	C:\WINDOWS\system32\msvcrt.dll
2:05:09.82363...	calcolatriceinnovativa50.exe	1284	Load Image	C:\WINDOWS\system32\shlwapi.dll
2:05:09.93336...	calcolatriceinnovativa50.exe	1284	Load Image	C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83\comctl32.dll
2:05:09.97030...	calcolatriceinnovativa50.exe	1284	Thread Exit	
2:05:09.97087...	calcolatriceinnovativa50.exe	1284	Process Exit	

DAY 5

TASK 1: Analizzare il file calcolatriceinnovativa50.exe e confermare che è un malware

- Infine abbiamo estrapolato **l'hash** del file calcolatriceinnovativa50.exe, **che da altra macchina abbiamo inserito su Virus Total**, dove **53 vendor su 71 hanno segnalato il file come malevolo**.

e447cc6b8d99952be9c781f2542030d49797683e7df6adf3e7

① 53 security vendors and no sandboxes flagged this file as malicious

c7f8e8f17dc7de447cc6b8d99952be9c781f2542030d49797683e7df6adf3e7
CALC.EXE

peexe detect-debug-environment checks-user-input

Community Score 53 / 71

② Reanalyze ⚙️ Similar files

Size 112.50 KB Last Analysis Date 16 hours ago

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 1

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

Popular threat label ① trojan.swort/cryptz Threat categories trojan Family labels swort cryptz marte

Security vendors' analysis ① Do you want to add this analysis to your dashboard?

Vendor	Detection	Engine	Family
AhnLab-V3	① Backdoor/Win32.Bifrose.C64906	ALYac	① Trojan.CryptZ.Marte.1.Gen
Arcabit	① Trojan.CryptZ.Marte.1.Gen	Avast	① Win32:SwPatch [Wrm]
AVG	① Win32:SwPatch [Wrm]	Avira (no cloud)	① TR/Patched.Gen2
BitDefender	① Trojan.CryptZ.Marte.1.Gen	BitDefenderTheta	① Gen:NN.Zexaf.36318.hm0@aOQzbzf
Bkav Pro	① W32.AIDetectMalware	ClamAV	① Win.Trojan.MSShellcode-6360730-0
CrowdStrike Falcon	① Win/malicious_confidence_100% (W)	Cybereason	① Malicious.d09ac1
Cylance	① Unsafe	Cynet	① Malicious (score: 100)
Cyren	① W32/Swort.B.gen!Eldorado	DeepInstinct	① MALICIOUS
DrWeb	① Trojan.Swort.1	Elastic	① Malicious (high Confidence)
Emsisoft	① Trojan.CryptZ.Marte.1.Gen (B)	eScan	① Trojan.CryptZ.Marte.1.Gen
ESET-NOD32	① A Variant Of Win32/Rozena.DT	F-Secure	① Trojan.TR/Patched.Gen2

DAY 5

TASK 1: Analizzare il file calcolatriceinnovativa50.exe e confermare che è un malware

- **IPOTESI SUL FUNZIONAMENTO DEL MALWARE:**

- Alla fine dell'analisi si può ipotizzare il file **calc.exe** sia stato copiato e modificato nella sua versione malevola “calcolatriceinnovativa50.exe”, poiché, come da verifiche, si è notato che **quest'ultimo ha una dimensione superiore rispetto all'originale, motivo per cui si potrebbe pensare che all'interno dello stesso sia stato implementando del codice malevolo.**
- A tal proposito, è doveroso segnalare che, probabilmente, la parte di codice malevolo inserita sia stata offuscata dalla tecnica **PADDING**, la quale viene utilizzata proprio per complicare il lavoro dell'analista di sicurezza durante la fase di malware analysis.
- Nello specifico, rispetto all'originale, risulta modificata la sezione **.rsrc**, dove è stato aggiunto il codice malevolo. Si notano differenze anche nel **CHECKSUM**, il quale, nel file calcolatriceinnovativa50.exe **ha valore pari a 0 (probabile indice che il file è stato danneggiato o modificato in modo malevolo**, poiché un **checksum valido dovrebbe essere diverso da 0 per indicare l'integrità del file).**
- Altra considerazione riguarda il valore **dell'entry point** che, a causa dell'aggiunta di codice, risulta modificato.
- A giudicare dal nome e dall'icona, entrambi simili al file eseguibile originale, è possibile ipotizzare che il nostro file malevolo sia un **Trojan Evader, ovvero un malware che si nasconde e mimetizza all'interno di un programma che apparentemente sembra lecito ed innocuo.**
- Infine, **i due file eseguibili hanno in comune tutte le funzioni utilizzate**, compresa la funzione **ShellAboutW**, la quale **carica una shell sul sistema operativo**. Si ipotizza che **questo comportamento potrebbe risultare dannoso** per la macchina infetta, poiché **il programma malevolo, che ricordiamo essere offuscato dalla tecnica di padding, tramite suddetta shell potrebbe essere una minaccia alla sicurezza della macchina bersaglio.**

DAY 5

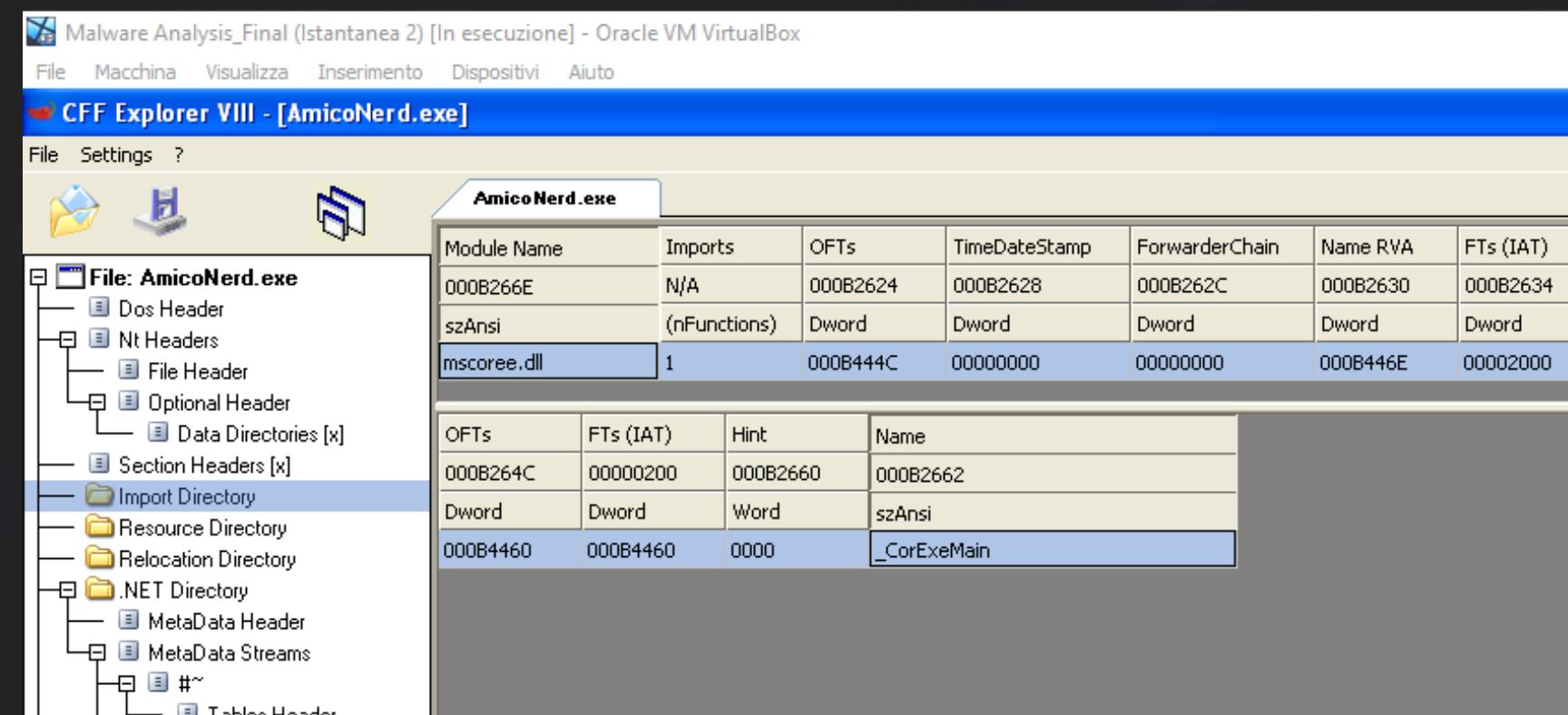
TASK 2: Convincere il dipendente che il file AmicoNerd è malevolo. Dopo l'analisi completa, pulire le tracce/gli effetti del malware

- Abbiamo avviato il tool **CFF EXPLORER** grazie al quale abbiamo visionato l'unica libreria presente all'interno del malware **mscoree.dll**, libreria utilizzata di collegamento dinamico (DLL) utilizzata nel framework .NET di Microsoft per l'esecuzione di programmi scritti in diversi linguaggi di programmazione, come C#, Visual Basic, ecc...
- È una componente essenziale per il corretto funzionamento delle applicazioni .NET ed è parte integrante dell'installazione del framework .NET sul sistema.
- **Un malware potrebbe sfruttare tale legittima libreria per nascondere la propria presenza e perseguire attività dannose senza attirare l'attenzione dell'utente o degli strumenti di sicurezza**

DAY 5

TASK 2: Convincere il dipendente che il file AmicoNerd è malevolo. Dopo l'analisi completa, pulire le tracce/gli effetti del malware

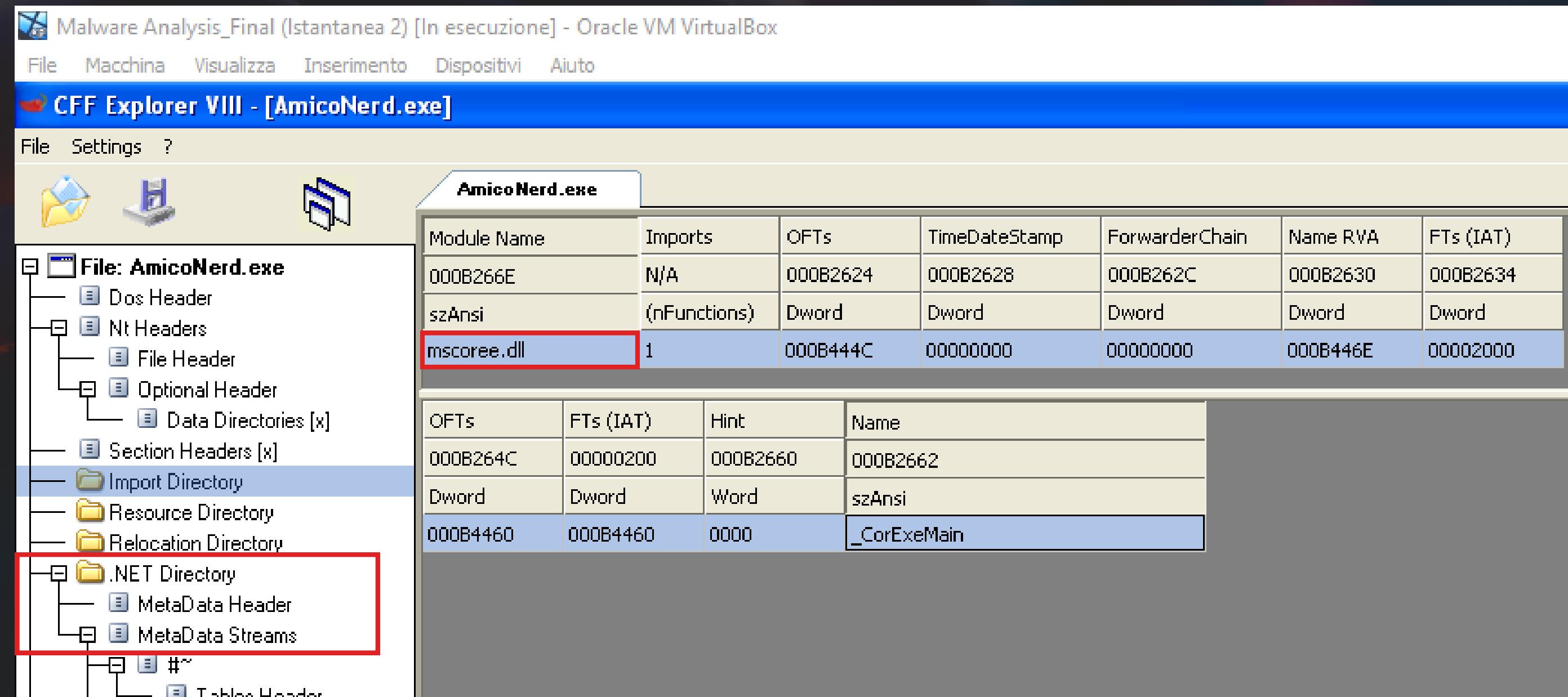
- Abbiamo avviato il tool **CFF EXPLORER** grazie al quale abbiamo visionato l'unica libreria presente all'interno del malware **mscoree.dll**, libreria utilizzata di collegamento dinamico (DLL) utilizzata nel framework .NET di Microsoft per l'esecuzione di programmi scritti in diversi linguaggi di programmazione, come C#, Visual Basic, ecc...
- È una componente essenziale per il corretto funzionamento delle applicazioni .NET ed è parte integrante dell'installazione del framework .NET sul sistema.
- **Un malware potrebbe sfruttare tale legittima libreria per nascondere la propria presenza e perseguire attività dannose senza attirare l'attenzione dell'utente o degli strumenti di sicurezza**



DAY 5

TASK 2: Convincere il dipendente che il file AmicoNerd è malevolo. Dopo l'analisi completa, pulire le tracce/gli effetti del malware

- Si nota altresì la presenza della **.NET directory** che può servire, per l'appunto, **a nascondere o modificare il comportamento malevolo del programma, ad esempio usando tecniche di offuscamento, crittografia o polimorfismo**, tecniche rendono più difficile l'analisi e il riconoscimento del malware da parte degli antivirus e dei ricercatori di sicurezza.



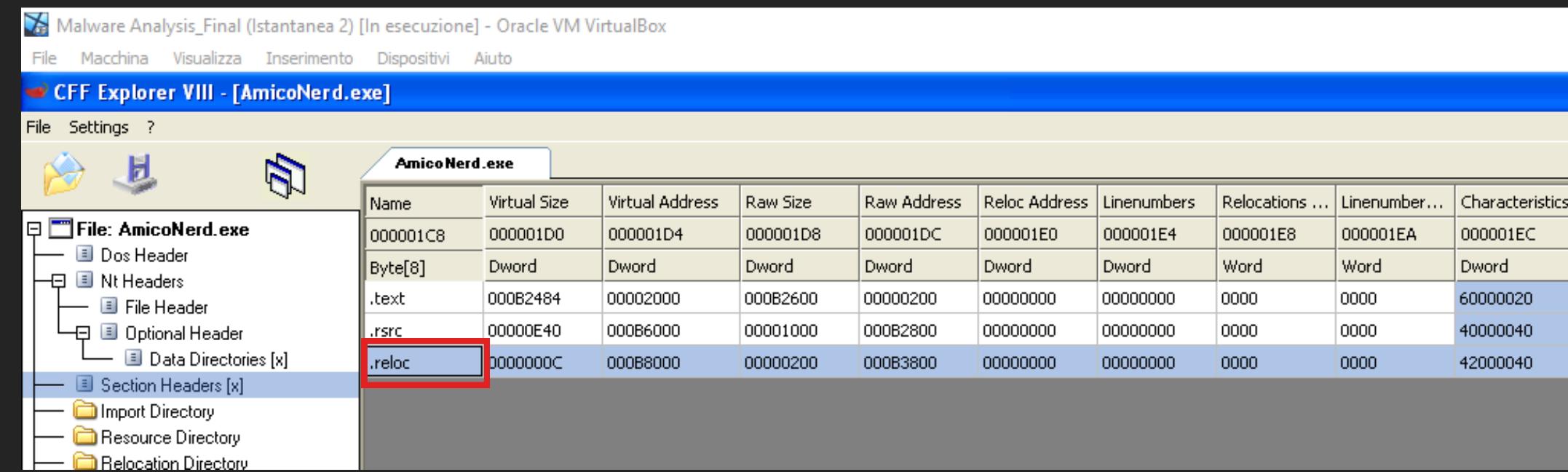
DAY 5

TASK 2: Convincere il dipendente che il file AmicoNerd è malevolo. Dopo l'analisi completa, pulire le tracce/gli effetti del malware

Il malware è composta da 3 sezioni:

- **.text:** Questa sezione contiene le istruzioni, ovvero le righe di codice che la CPU eseguirà quando il software viene avviato. È la sezione principale di un file eseguibile, poiché contiene il codice effettivo che viene eseguito per far funzionare il programma. Tutte le altre sezioni contengono dati o informazioni di supporto per questa sezione.
- **.rsrc:** Questa sezione include le risorse utilizzate come ad esempio icone, immagini, menu e stringhe che non sono parte dell'eseguibile stesso.
- **.reloc:** Nel contesto degli assembly .NET e dei file eseguibili Windows con formato PE questa sezione è parte integrante del formato PE e serve a consentire l'esecuzione del codice e dei dati in diverse posizioni di memoria, in modo da rendere l'eseguibile indipendente dall'indirizzo di caricamento effettivo.

Tuttavia, la sezione .reloc può anche essere usata dai malware per nascondere il loro codice dannoso o per eludere le tecniche di analisi statica.

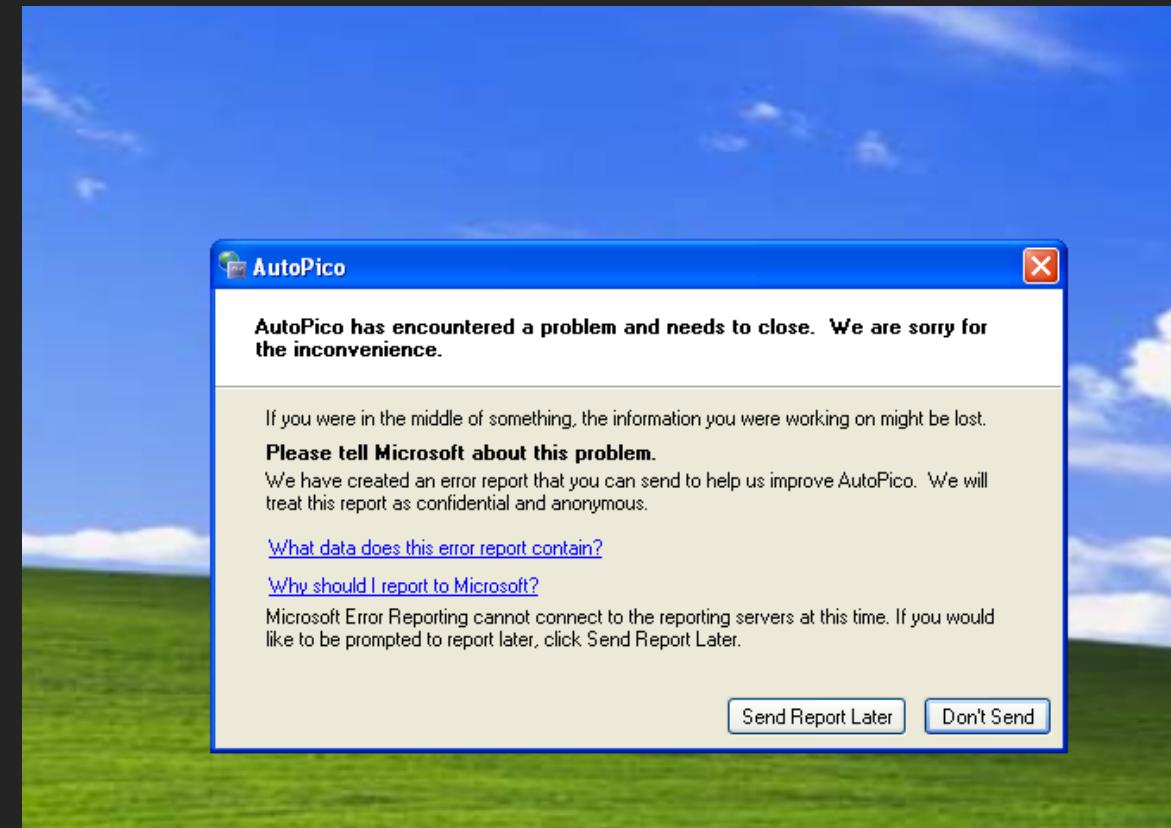


Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations ...	Linenumber...	Characteristics
000001C8	000001D0	000001D4	000001D8	000001DC	000001E0	000001E4	000001E8	000001EA	000001EC
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword
.text	000B2484	00002000	000B2600	00000200	00000000	00000000	0000	0000	60000020
.rsrc	00000E40	000B6000	00001000	000B2800	00000000	00000000	0000	0000	40000040
.reloc	0000000C	000B8000	00000200	000B3800	00000000	00000000	0000	0000	42000040

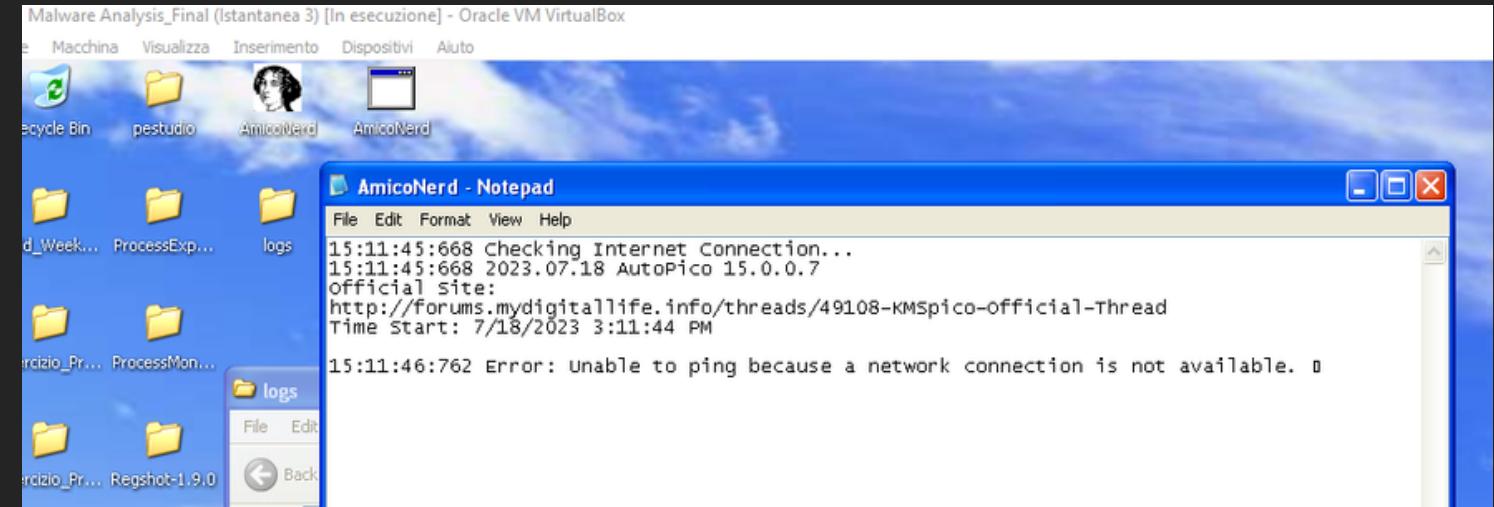
DAY 5

TASK 2: Convincere il dipendente che il file AmicoNerd è malevolo. Dopo l'analisi completa, pulire le tracce/gli effetti del malware

- In seguito abbiamo avviato il file AmicoNerd, e dopo qualche secondo si è aperta una finestra che ci ha comunicato che “AutoPico” ha riscontrato un problema.



- Poco dopo l'esecuzione del malware abbiamo notato che sul Desktop è stata creata una cartella denominata “logs”, al cui interno è presente un file a sua volta denominato AmicoNerd, nel quale si può notare come vi sia un errore riguardante l'inaccessibilità ad Internet (in quanto ovviamente si sta lavorando offline con scheda di rete disabilitata).
- Dato il comportamento iniziale del malware oggetto d'interesse, si potrebbe ipotizzare che lo stesso possa essere un **dropper** o una **backdoor**.



DAY 5

TASK 2: Convincere il dipendente che il file AmicoNerd è malevolo. Dopo l'analisi completa, pulire le tracce/gli effetti del malware

- Analizzando più a fondo la .NET Directory si nota la presenza dell'activator di [AutoPico](#). Da ricerche è emerso che [KMSPico](#) è uno strumento non legittimo sviluppato da criminali informatici per attivare illegalmente il sistema operativo Windows o Microsoft Office senza pagare per le licenze originali.
- Tuttavia, questo strumento è rischioso perché può contenere malware o infettare il computer con programmi dannosi come [adware](#), [browser hijacker](#) o minatori di criptovaluta.
- Le applicazioni indesiderate causano problemi al computer, rallentano le prestazioni, visualizzano pubblicità fastidiose, possono rubare informazioni personali e compromettere la privacy. È importante evitare l'utilizzo di [KMSPico](#) e invece attivare Windows e Microsoft Office solo con chiavi di licenza originali fornite da Microsoft. In caso di infezioni da applicazioni indesiderate, è necessario utilizzare un software antivirus/antispyware affidabile per rimuoverle dal sistema.
- Questo tipo di software piratato è noto per tentare di connettersi a server esterni per scaricare le chiavi di licenza contraffatte o utilizzare server KMS pirata per attivare illegalmente il software.

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Ascii
00007C50	00	3B	01	00	36	54	69	70	6F	20	64	65	20	61	63	74	: I.6Tipo.de.act
00007C60	69	76	61	63	69	C3	B3	6E	20	63	6F	6E	66	69	67	75	ivaciÃ³n.configu
00007C70	72	61	64	61	20	70	61	72	61	20	65	6C	20	63	6C	69	rada.para.el.cli
00007C80	65	6E	74	65	20	64	65	20	56	4C	2E	00	00	6C	01	00	ente.de.VL...1.
00007C90	67	46	72	65	63	75	65	6E	63	69	61	20	64	65	20	63	gFrecuencia.de.c
00007CA0	6F	6E	74	61	63	74	6F	20	65	6E	20	6D	69	6E	75	74	contacto.en.minut
00007CB0	6F	73	20	64	65	20	75	6E	20	63	6C	69	65	6E	74	65	os.de.un.cliente
00007CC0	20	63	6F	20	65	6C	20	68	6F	73	74	20	4B	4D	53	.con.el.host.KMS	
00007CD0	20	75	6E	61	20	76	65	7A	20	71	75	65	20	74	65	6E	.una.vez.que.ten
00007CE0	67	61	20	6C	69	63	65	6E	63	69	61	20	65	6C	20	70	ga.licencia.el.p
00007CF0	72	6F	64	75	63	74	6F	2E	00	00	05	28	00	12	82	18	roducto...!(..)
00007D00	72	01	00	6D	47	55	49	44	20	71	75	65	20	69	64	65	r GUID.que.ide
00007D10	6E	74	69	66	69	63	61	20	75	6E	20	63	6C	69	65	6E	ntifica.un.clien
00007D20	74	65	20	4B	4D	53	20	61	20	75	6E	20	68	6F	73	74	te.KMS.a.un.host
00007D30	20	4B	4D	53	2E	20	45	6C	20	63	6C	69	65	6E	74	65	.KMS..El.cliente
00007D40	20	6C	6F	20	69	6E	63	6C	75	79	65	20	65	6E	20	6C	.lo.incluye.en.l
00007D50	61	73	20	73	6F	6C	69	63	74	75	64	65	73	20	71	as.solicitudes.q	
00007D60	75	65	20	65	6E	76	C3	AD	61	20	61	6C	20	4B	4D	53	ue.envA-a.al.KM
00007D70	2E	00	00	4A	01	00	45	41	75	74	6F	50	69	63	6F	2E	..J.EAutoPico.
00007D80	41	63	74	69	76	61	64	6F	72	2E	57	4D	49	2E	53	6F	Activador.WMI.So
00007D90	66	74	77	61	72	65	4C	69	63	65	6E	73	69	6E	67	53	ftwareLicensingS
00007DA0	65	72	76	69	63	65	2B	57	4D	49	56	61	6C	75	65	54	ervice+WMIValueT
00007DB0	79	70	65	43	6F	6E	76	65	72	74	65	72	00	00	52	01	wpeConverter.R
00007DC0	00	4D	49	6E	64	69	63	61	20	73	69	20	4B	4D	53	20	MIndica.si.KMS.
00007DD0	65	73	74	C3	A1	20	68	61	62	69	6C	69	74	61	64	6F	estAI.habilitado
00007DE0	20	65	6E	20	65	6C	20	65	71	75	69	70	6F	3A	20	30	.en.el.equipo.0
00007DF0	20	73	69	20	6E	20	6C	6F	20	65	73	74	C3	A1	2C	si.no.lo.estAI.	
00007E00	20	31	20	73	69	20	6C	6F	20	65	73	74	C3	A1	2E	00	1.si.lo.estAI..
00007E10	00	80	B2	01	00	80	AC	4E	C3	BA	6D	65	72	6F	20	64	..I..-NÃºmero.d
00007E20	65	20	63	6C	69	65	6E	74	65	73	20	4B	4D	53	20	61	e.clientes.KMS.a
00007E30	63	74	69	76	6F	73	20	61	63	74	75	61	6C	6D	65	6E	ctivos.actualmen
00007E40	74	65	20	65	6E	20	65	6C	20	68	6F	73	74	20	4B	4D	te.en.el.host.KM
00007E50	53	2E	20	2D	31	20	69	6E	64	69	63	61	20	71	75	65	S.-1.indica.que
00007E60	20	65	6C	20	65	71	75	69	70	6F	20	6E	6F	20	65	73	el.equipo.no.es
00007E70	74	C3	A1	20	68	61	62	69	6C	69	74	61	64	6F	20	63	tAI.habilitado.c
00007E80	6F	6D	6E	20	6B	52	20	67	71	75	65	20	67	6F	20	65	KMS

DAY 5

TASK 2: Convincere il dipendente che il file AmicoNerd è malevolo. Dopo l'analisi completa, pulire le tracce/gli effetti del malware

- Infine abbiamo inserito l'hash del file AmicoNerd.exe su Virus Total, dove 53 vendors su 71 hanno segnalato il file come malevolo, identificandolo come un **hacktool/trojan**.

The screenshot shows the VirusTotal analysis page for the file `c6603d416dfc48894eda35d9a9a8523bdf9823e215ab926783ce6848aa8a62c4`. The main summary indicates that 53 security vendors out of 71 flagged the file as malicious. The file is identified as `AutoPico.exe` and is categorized as an EXE file, with a size of 722.69 KB and a last analysis date of 14 days ago. The threat categories listed include `peexe`, `assembly`, `overlay`, `revoked-cert`, `runtime-modules`, `invalid-signature`, `signed`, `detect-debug-environment`, `checks-network-adapters`, `long-sleeps`, `direct-cpu-clock-access`, and `via-tor`. The `calls-wmi` category is also present.

The analysis table lists 14 security vendors and their findings:

Vendor	Result	Details	Family
Acronis (Static ML)	Suspicious	AhnLab-V3	HackTool/Win.AutoKMS.C948312
ALYac	Application.Hacktool.KMSActivator.AQ	AntiAVL	RiskWare[NetTool]/Win64.RPCHook
Arcabit	Application.KMS	Avast	Win32:MiscX-gen [PUP]
AVG	Win32:MiscX-gen [PUP]	BitDefender	Application.Hacktool.KMSActivator.AQ
BitDefenderTheta	Gen:NN.ZemsilF.36270.Tm1@a8vJERd	ClamAV	Win.Tool.Kmsactivator-9811695-0
CrowdStrike Falcon	Win/grayware_confidence_100% (W)	Cybereason	Malicious.fd3caf
Cylance	Unsafe	Cynet	Malicious (score: 100)
Cyren	W32/S-eb8730b5!Eldorado	DeepInstinct	MALICIOUS
Elastic	Malicious (high Confidence)	Emsisoft	Application.HackTool (A)
eScan	Application.Hacktool.KMSActivator.AQ	ESET-NOD32	A Variant Of MSIL/HackTool.IdleKMS.E ...

DAY 5

TASK 2: Convincere il dipendente che il file AmicoNerd è malevolo. Dopo l'analisi completa, pulire le tracce/gli effetti del malware

- Dopo aver analizzato i malware in questione, per pulire le tracce/gli effetti del malware, abbiamo ripristinato l'istantanea che avevamo precedentemente creato, prima di avviare i malware, come confermato da screenshot seguente in cui si nota la normale schermata di shut down, così com'era prima della creazione di **Gina.DLL**.

