

BUILD WEEK 3:

Malware analysis and reverse engineering in practice

Il lavoro dei primi quattro giorni si concentra sull'analisi del file `Malware_Build_Week_U3` contenuto nella cartella `Build_Week_Unit3` presente nella macchina virtuale dedicata.

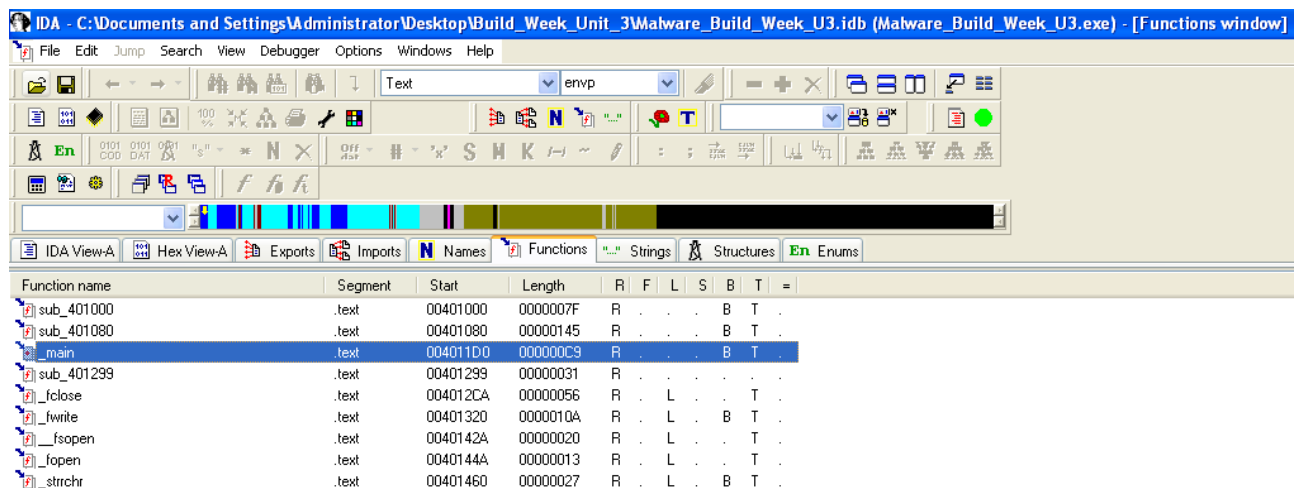
Il giorno cinque, invece, ci occuperemo dell'analisi di due malware.

GIORNO 1

PARTE 1

Per iniziare, abbiamo proceduto al caricamento del file eseguibile all'interno del software **IDA Pro**, una suite di analisi e reverse engineering.

Successivamente, ci siamo diretti al pannello "Functions" all'interno dell'interfaccia di IDA Pro, dove abbiamo individuato la funzione "**Main()**".



Una volta entrati nella funzione `Main()`, abbiamo identificato tre parametri specifici che vengono passati alla funzione quando viene chiamata, ovvero **argc**, **argv** ed **envp**.

Questi parametri rappresentano rispettivamente il numero di argomenti passati al programma, un array di stringhe che rappresentano gli argomenti stessi e un array di stringhe che rappresenta l'ambiente in cui viene eseguito il programma.

Abbiamo anche rilevato la presenza di quattro variabili particolari all'interno della funzione `Main()`. Queste variabili sono identificate come **hModule**, **Data**, **var_8** e **var_4**. Ognuna di queste variabili svolge un ruolo specifico nel contesto del programma in esecuzione, fornendo spazio di memoria per l'immagine del modulo, dati variabili e valori temporanei utilizzati durante l'esecuzione del programma.

```

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

```

Dopo un'attenta analisi della funzione Main() all'interno del codice, abbiamo proceduto al caricamento del malware su CFF Explorer, uno strumento di analisi dei file eseguibili, al fine di esaminare in dettaglio le varie sezioni che compongono il file e le librerie che vengono importate.

Nell'elenco delle sezioni presenti nella directory "Section Headers", abbiamo individuato un totale di quattro sezioni:

CFF Explorer VIII - [Malware_Build_Week_U3.exe]

File Settings ?

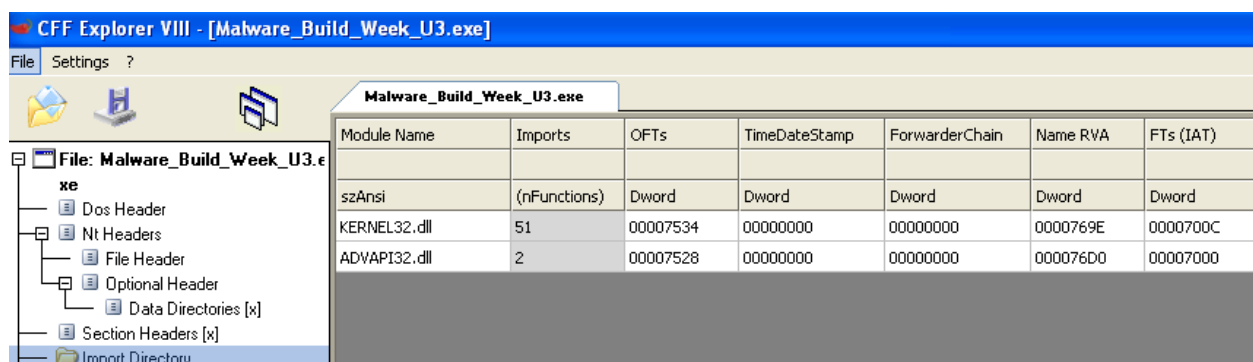
Malware_Build_Week_U3.exe

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

- **.text:** contiene il codice eseguibile del programma.
Questa sezione è di sola lettura, il che significa che il codice all'interno di essa non può essere modificato durante l'esecuzione del programma. Contiene le istruzioni macchina che vengono eseguite dal processore per eseguire le operazioni specificate dal programma.
- **.rdata:** contiene dati di sola lettura che sono utilizzati dal programma durante l'esecuzione, ad esempio stringhe di testo costanti o tabelle di lookup.
Questa sezione è anche di sola lettura, il che significa che i dati al suo interno non possono essere modificati durante l'esecuzione del programma.
- **.data:** contiene i dati inizializzati che possono essere letti e scritti dal programma durante l'esecuzione. Questi dati possono includere variabili globali, variabili statiche o altri dati che devono mantenere uno stato durante l'esecuzione del programma. A differenza della sezione .rdata, i dati possono essere modificati durante l'esecuzione del programma.
- **.rsrc:** contiene risorse aggiuntive utilizzate dal programma, come icone, immagini, stringhe localizzate e tabelle di risorse. Questi dati possono essere passati al programma durante l'esecuzione per personalizzare l'aspetto o il comportamento dell'applicazione. La sezione .rsrc non contiene codice eseguibile, ma solo dati e informazioni utilizzate dal programma.

Successivamente, abbiamo proceduto a spostarci nella sezione "Import Directory" al fine di esaminare in dettaglio le librerie importate nel file in questione.

Abbiamo individuato due librerie importate:



- **KERNEL32.dll:** contiene le funzioni principali per interagire con il sistema operativo. Ad esempio, è utilizzata per manipolare i file o gestire la memoria del sistema.

OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
00007632	00007632	0295	SizeofResource
00007644	00007644	01D5	LockResource
00007654	00007654	01C7	LoadResource
00007622	00007622	02BB	VirtualAlloc
00007674	00007674	0124	GetModuleFileNameA
0000768A	0000768A	0126	GetModuleHandleA
00007612	00007612	00B6	FreeResource
00007664	00007664	00A3	FindResourceA
00007604	00007604	001B	CloseHandle
000076DE	000076DE	00CA	GetCommandLineA
000076F0	000076F0	0174	GetVersion
000076FE	000076FE	007D	ExitProcess
0000770C	0000770C	019F	HeapFree
00007718	00007718	011A	GetLastError
00007728	00007728	02DF	WriteFile
00007734	00007734	029E	TerminateProcess
00007748	00007748	00F7	GetCurrentProcess
0000775C	0000775C	02AD	UnhandledExceptionFilter
00007778	00007778	00B2	FreeEnvironmentStringsA
00007792	00007792	00B3	FreeEnvironmentStringsW
000077AC	000077AC	02D2	WideCharToMultiByte
000077C2	000077C2	0106	GetEnvironmentStrings
000077DA	000077DA	0108	GetEnvironmentStringsW
000077F4	000077F4	026D	SetHandleCount
00007806	00007806	0152	GetStdHandle
00007816	00007816	0115	GetFileType
00007824	00007824	0150	GetStartupInfoA
00007836	00007836	0109	GetEnvironmentVariableA
00007850	00007850	0175	GetVersionExA
00007860	00007860	019D	HeapDestroy
0000786E	0000786E	019B	HeapCreate
0000787C	0000787C	02BF	VirtualFree
0000788A	0000788A	022F	RtlUnwind
00007896	00007896	0199	HeapAlloc
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

- **ADVAPI32.dll:** contiene le funzioni necessarie per interagire con i servizi e i registri del sistema operativo Microsoft.

OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
000076AC	000076AC	0186	RegSetValueExA
000076BE	000076BE	015F	RegCreateKeyExA

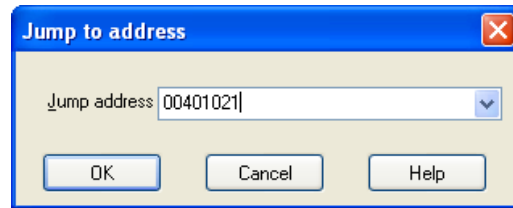
Possiamo supporre che il file in esame richiami delle funzioni specifiche per la modifica delle chiavi di registro tramite le funzioni "RegCreateKeyExA" e "RegSetValueExA", presenti nella libreria "ADVAPI32.dll".

Successivamente, il file potrebbe creare ed eseguire un nuovo file utilizzando i comandi "CreateFileA" e "SetEndOfFile" attraverso la libreria "KERNEL32.dll".

PARTE 2

La seconda parte dell'esercizio richiedeva di analizzare determinate parti del codice del malware.

Una volta tornati su **IDA pro** abbiamo come prima cosa cercato tramite "jump to address" l'indirizzo di memoria **00401021**.



L'indirizzo 00401021 corrisponde all'istruzione che richiama la funzione `RegCreateKeyExA`. Questa funzione è utilizzata per creare una nuova chiave di registro con i parametri specificati.

```
.text:00401017      push     offset SubKey      ; "SOFTWARE\\Microsoft\\Windows NT\\CurrentUe"...
.text:0040101C      push     80000002h         ; hKey
.text:00401021      call    ds:RegCreateKeyExA
.text:00401027      test    eax, eax
.text:00401029      jz      short loc_401032
.text:0040102B      mov     eax, 1
```

Le istruzioni sono passate alla variabile tramite l'istruzione **push**.

```
.text:00401009      push     eax                ; phkResult
.text:0040100A      push     0                  ; lpSecurityAttributes
.text:0040100C      push     0F003Fh           ; samDesired
.text:00401011      push     0                  ; dwOptions
.text:00401013      push     0                  ; lpClass
.text:00401015      push     0                  ; Reserved
.text:00401017      push     offset SubKey      ; "SOFTWARE\\Microsoft\\Windows NT\\CurrentUe"...
.text:0040101C      push     80000002h         ; hKey
.text:00401021      call    ds:RegCreateKeyExA
```

Abbiamo eseguito un'ulteriore ricerca al fine di individuare l'oggetto specifico situato all'indirizzo di memoria 00401017.

Questo oggetto svolge un ruolo fondamentale come chiave di registro impiegata dal malware per garantire la sua persistenza nel sistema.

La creazione di questa chiave avviene mediante l'utilizzo della funzione "**RegCreateKeyExA**", e la sua collocazione precisa si trova nel percorso "**Software\\Microsoft\\Windows NT\\CurrentVersion\\WinLogon**".

```
.text:00401009      push     eax                ; phkResult
.text:0040100A      push     0                  ; lpSecurityAttributes
.text:0040100C      push     0F003Fh           ; samDesired
.text:00401011      push     0                  ; dwOptions
.text:00401013      push     0                  ; lpClass
.text:00401015      push     0                  ; Reserved
.text:00401017      push     offset SubKey      ; "SOFTWARE\\Microsoft\\Windows NT\\CurrentUe"...
.text:0040101C      push     80000002h         ; hKey
.text:00401021      call    ds:RegCreateKeyExA
```

In seguito, siamo passati all'analisi delle istruzioni comprese tra gli indirizzi **00401027** e **00401029**.

La prima istruzione presente nella locazione di memoria 00401027 è un'istruzione di **test**. Questa istruzione è di natura condizionale e assomiglia a un'operazione logica "AND".

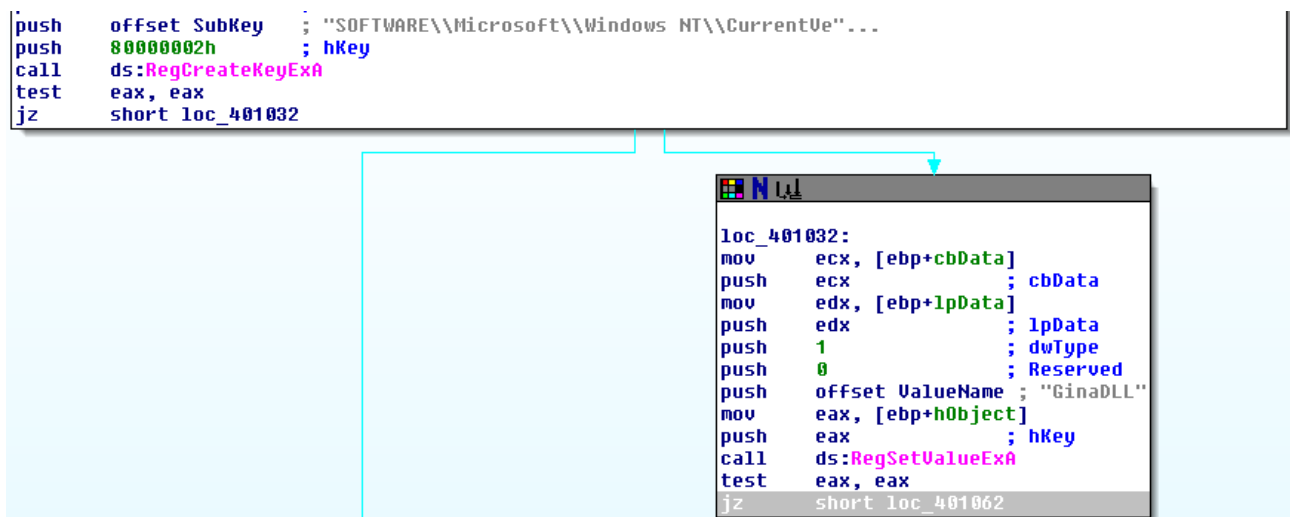
Tuttavia, a differenza dell' "AND" logico, non modifica gli operandi coinvolti ma modifica il flag **ZF** (zero-flag) che sarà impostato a 1 solo se il risultato dell'operazione è 0.

La seconda istruzione è denominata "**JZ**" ed esegue un salto condizionale solo se il flag **ZF** dell'operazione precedente è impostato a 1.

In questo caso specifico, il salto verrà eseguito verso la locazione di memoria 00401032.

```
.text:00401027 test    eax, eax
.text:00401029 jz      short loc_401032
.text:0040102B mov     eax, 1
.text:00401030 jmp     short loc_40107B
.text:00401032 ; -----
.text:00401032 loc_401032:                ; CODE XREF: sub_401000+29↑j
```

Visualizzazione dal diagramma di flusso:



Successivamente, abbiamo proceduto con la ricostruzione dettagliata del codice in linguaggio C al fine di simulare in modo accurato il funzionamento del costrutto "IF" presente nel nostro codice assembly.

Per prima cosa, abbiamo dichiarato le variabili necessarie per eseguire l'operazione di confronto tra i due operandi.

Successivamente, abbiamo impostato delle condizioni basate sul risultato ottenuto da tale operazione.

Nel caso in cui il risultato del test fosse uguale a zero, abbiamo impostato il valore della flag ZF a 1 e abbiamo effettuato un salto nel programma.

In caso contrario, abbiamo proseguito l'esecuzione del codice senza effettuare alcun salto.

```

int main()
{
    int zf; // Variabile che simula il flag della zero (ZF)
    int eax = 12; // Variabile che simula il registro eax
    int test; // Variabile che simula l'operazione di test
    test = eax - eax; // Operazione di AND logico
    if (test == 0) // Inizio del blocco IF
    {
        zf = 1; // Imposta il flag ZF a 1
    }
    if (zf == 1) // Se il flag ZF è uguale a 1, salta alla locazione "481032"
    {
        goto loc_401032;
    }
    else
    {
        goto loc_40107B; // Altrimenti, continua con il codice successivo
    }
}

loc_401032:
    printf("Salto avvenuto"); // Stampa un messaggio indicante che il salto è avvenuto
loc_40107B:
    printf("Salto non avvenuto"); // Stampa un messaggio indicante che il salto non è avvenuto
    return 0;
}

```

Infine, siamo passati a valutare la chiamata alla locazione **00401047**.

Durante questa chiamata, viene passato un valore al parametro denominato "ValueName", e tale valore è specificamente **"GinaDLL"**.

```

.text:00401032
.text:00401032 loc_401032:
.text:00401032     mov     ecx, [ebp+cbData] ; CODE XREF: sub_401000+29↑j
.text:00401035     push    ecx                ; cbData
.text:00401036     mov     edx, [ebp+lpData]
.text:00401039     push    edx                ; lpData
.text:0040103A     push    1                  ; dwType
.text:0040103C     push    0                  ; Reserved
.text:0040103E     push    offset ValueName ; "GinaDLL"
.text:00401043     mov     eax, [ebp+hObject]
.text:00401046     push    eax                ; hKey
.text:00401047     call    ds:RegSetValueExA
.text:0040104D     test    eax, eax
.text:0040104F     jz      short loc_401062
.text:00401051     mov     ecx, [ebp+hObject]
.text:00401054     push    ecx                ; hObject
.text:00401055     call    ds:CloseHandle
.text:0040105B     mov     eax, 1
.text:00401060     jmp     short loc_40107B

```

Per completezza abbiamo lanciato il file eseguibile del malware con il tool "OllyDBG" per verificare che il valore di "ValueName" fosse effettivamente quello riscontrato con IDA.

00401035	51	PUSH ECX	BufSize
00401036	8B55 08	MOV EDX,DWORD PTR SS:[EBP+8]	Buffer
00401039	52	PUSH EDX	ValueType = REG_SZ
0040103A	6A 01	PUSH 1	Reserved = 0
0040103C	6A 00	PUSH 0	ValueName = "GinaDLL"
0040103E	68 4C804000	PUSH Malware_.0040804C	
00401043	8B45 FC	MOV EAX,DWORD PTR SS:[EBP-4]	hKey
00401046	50	PUSH EAX	RegSetValueExA
00401047	FF15 00704000	CALL DWORD PTR DS:[<&ADVAPI32.RegSetVal	
0040104D	85C9	TEST EAX,EAX	

Utilizzando le informazioni fornite dal codice, siamo stati in grado di formulare un'ipotesi più dettagliata sul comportamento del malware.

Sembrerebbe che il malware stia cercando di stabilire una persistenza all'interno del registro di Windows, ovvero una capacità di sopravvivenza e di avvio automatico del malware all'avvio del sistema.

Per raggiungere questo obiettivo, il malware utilizza due funzioni specifiche.

La prima funzione coinvolta è "RegCreateKeyExA", che viene utilizzata per creare una nuova chiave di registro all'interno del sistema operativo Windows. Una chiave di registro è un'unità di archiviazione nel registro di Windows che contiene valori e impostazioni importanti per il funzionamento del sistema.

Successivamente, il malware si avvale della funzione "RegSetValueExA" per configurare la chiave di registro precedentemente creata. Questa funzione consente di specificare i dettagli e i valori da associare alla chiave di registro, fornendo al malware la possibilità di impostare i parametri iniziali e le informazioni necessarie per la sua persistenza nel sistema.

In sintesi, il malware sembra utilizzare le funzioni "RegCreateKeyExA" e "RegSetValueExA" per creare una nuova chiave di registro e configurarla con le impostazioni desiderate, al fine di ottenere una persistenza all'interno del registro di Windows.

00401004	. 6A 00	PUSH 0	pDisposition = NULL
00401006	. 8D45 FC	LEA EAX,DWORD PTR SS:[EBP-4]	pHandle
00401009	. 50	PUSH EAX	pSecurity = NULL
0040100A	. 6A 00	PUSH 0	Access = KEY_ALL_ACCESS
0040100C	. 68 3F00F00	PUSH 0F003F	Options = REG_OPTION_NON_VOLATILE
00401011	. 6A 00	PUSH 0	Class = NULL
00401013	. 6A 00	PUSH 0	Reserved = 0
00401015	. 6A 00	PUSH 0	Subkey = "SOFTWARE\Microsoft\Windows
00401017	. 68 54804000	PUSH Malware_.00408054	hKey = HKEY_LOCAL_MACHINE
0040101C	. 68 02000080	PUSH 80000080	
00401021	. FF15 04704000	CALL DWORD PTR DS:[&ADVAPI32.RegCreateKeyExA]	RegCreateKeyExA

GIORNO 2

Dopo le analisi eseguite sul Malware il primo giorno, si è proceduto con lo studio delle funzioni e dei relativi parametri presenti agli indirizzi di memoria compresi tra **00401080** e **00401128**.

Come richiesto tutte le analisi eseguite rientrano nella categoria dell'analisi statica basica, avendo utilizzato ai fini della nostra ricerca il disassembler IDA Pro e il tool CFF Explorer.

Mediante il primo è stata individuata come valore del parametro "ResourceName", passato alla funzione "FindResourceA()", la stringa "TGAD", che, come si può notare, viene "puntata" come parametro "name" da passare alla funzione nel momento in cui essa viene chiamata.

```
.data:00408034 ; LPCSTR lpName
.data:00408034 lpName
.data:00408038 aTgad      db 'TGAD',0
.data:0040803D          align 10h
.data:00408040 aBinary    db 'BINARY',0
.data:00408047          align 4
.data:00408048 aRi       db 'RI',0Ah,0
.data:0040804C ; char ValueName[]
.data:0040804C ValueName db 'GinaDLL',0
.data:00408054 ; char SubKey[]
```

DATA XREF: sub_401080+3E↑r
"TGAD"
DATA XREF: .data:lpName↑o
DATA XREF: .data:lpType↑o
DATA XREF: sub_401000:loc_401062↑o
DATA XREF: sub_401000+3E↑o

Aver individuato tale funzione ci ha immediatamente fatto pensare che l'eseguibile oggetto di analisi sia un Dropper.

Il prosieguo del nostro studio ha confermato tali ipotesi poiché, mediante l'utilizzo e il confronto dei dati estrapolati da entrambi i tool citati precedentemente, non è stata individuata soltanto la funzione "FindResourceA", ma anche le altre funzioni che gestiscono la locazione e la gestione di risorse contenute nello stesso eseguibile, tipiche dei Dropper, come "LoadResource", "LockResource", "SizeOfResource".

```
.text:00401088 loc_401088: mov     eax, lpType ; CODE XREF: sub_401080+2F↑j
.text:00401088      push    eax ; lpType
.text:0040108D      mov     ecx, lpName ; lpName
.text:0040108D      push    ecx ; lpName
.text:0040108E      mov     edx, [ebp+hModule]
.text:0040108E      push    edx ; hModule
.text:0040108F      call    ds:FindResourceA
.text:0040108F      mov     [ebp+hResInfo], eax
.text:00401090      cmp     [ebp+hResInfo], 0
.text:00401090      jnz     short loc_4010DF
.text:00401090      xor     eax, eax
.text:00401090      jmp     loc_4011BF
.text:004010DF loc_4010DF: mov     eax, [ebp+hResInfo] ; CODE XREF: sub_401080+56↑j
.text:004010DF      push    eax ; hResInfo
.text:004010E0      mov     ecx, [ebp+hModule]
.text:004010E0      push    ecx ; hModule
.text:004010E1      call    ds:LoadResource
.text:004010E1      mov     [ebp+hResData], eax
.text:004010E2      cmp     [ebp+hResData], 0
.text:004010E2      jnz     short loc_4010FB
.text:004010E2      jmp     loc_4011A5
.text:004010FB loc_4010FB: mov     edx, [ebp+hResData] ; CODE XREF: sub_401080+74↑j
.text:004010FB      push    edx ; hResData
.text:004010FC      call    ds:LockResource
.text:004010FC      mov     [ebp+var_0], eax
.text:004010FD      cmp     [ebp+var_0], 0
.text:004010FD      jnz     short loc_401113
.text:004010FD      jmp     loc_4011A5
.text:00401113 loc_401113: mov     eax, [ebp+hResInfo] ; CODE XREF: sub_401080+8C↑j
.text:00401113      push    eax ; hResInfo
.text:00401114      mov     ecx, [ebp+hModule]
.text:00401114      push    ecx ; hModule
.text:00401115      call    ds:SizeOfResource
.text:00401115      mov     [ebp+dwSize], eax
.text:00401116      cmp     [ebp+dwSize], 0
.text:00401116      ja      short loc_40112C
.text:00401116      jmp     short loc_4011A5
```


Ulteriore conferma di quanto ipotizzato è stata fornita da CFF Explorer, grazie al quale è stato possibile individuare la sezione , e il “resource Directory” consultabile dallo stesso tool.

Malware_Build_Week_U3.exe									
Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations ...	Linenumber...	Characteristics
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword
.text	00005646	00001000	00006000	00001000	00000000	00000000	0000	0000	60000020
.rdata	000009AE	00007000	00001000	00007000	00000000	00000000	0000	0000	40000040
.data	00003EA8	00008000	00003000	00008000	00000000	00000000	0000	0000	C0000040
resource	00001A70	0000C000	00002000	0000B000	00000000	00000000	0000	0000	40000040

Malware_Build_Week_U3.exe	
Resource Directory	
Resource Directory Entry 1, Name: BINARY	
Resource Directory	
Resource Directory Entry 1, Name: TGAD	
Resource Directory	
Resource Directory Entry 1, ID: 0	
Resource Data Entry	

Sempre da CFF, abbiamo re-individuato le funzioni tipiche della famiglia di Malware a cui crediamo quello preso in esame appartenga e che sono state già citate precedentemente.

Si noti infatti come l'eseguibile utilizza KERNEL32.dll (da cui importa ben 51 funzioni) e ADVAPI32.dll, entrambe responsabili dell'interazione dell'utente con il File System.

Malware_Build_Week_U3.exe						
Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.dll	51	00007534	00000000	00000000	0000769E	0000700C
ADVAPI32.dll	2	00007528	00000000	00000000	000076D0	00007000

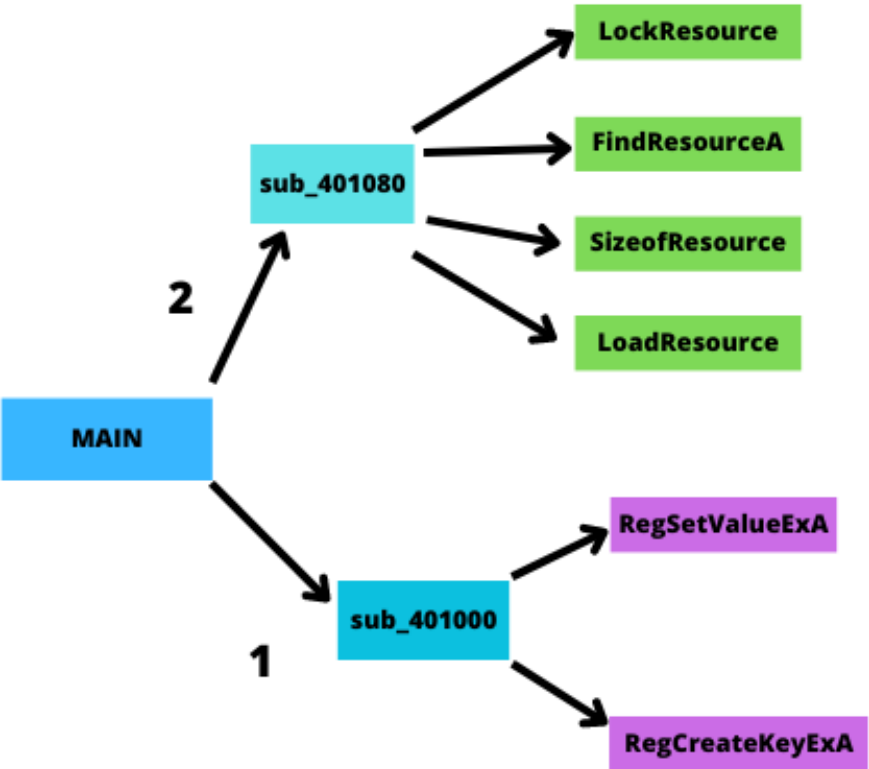
Analizzando, invece, le varie funzioni presenti nel codice del malware, notiamo che ognuna delle funzioni evidenziate gestisce una parte dell'interazione della macchina infetta con il programma da scrivere su di essa:

- **LoadResource:** recupera un handle che può essere usato per ottenere un puntatore al primo byte della risorsa specificata in memoria;
- **LockResource:** recupera un puntatore alla risorsa specificata in memoria;
- **SizeOfResource:** recupera la dimensione, in byte, della risorsa specificata;
- **FindResourceA:** determina la posizione di una risorsa con il tipo e il nome indicati nel modulo specificato.

Ai fini dell'ottenimento della Persistenza, il Malware va ad operare sui registri di Sistema utilizzando le due funzioni importate dalla libreria ADVAPI.dll. In particolare, esse sono:

- **RegCreateKeyExA:** necessaria all'eseguibile per la creazione di una nuova chiave di registro;
- **RegSetValueExA:** che viene utilizzata per andare ad impostare il valore della nuova chiave creata.

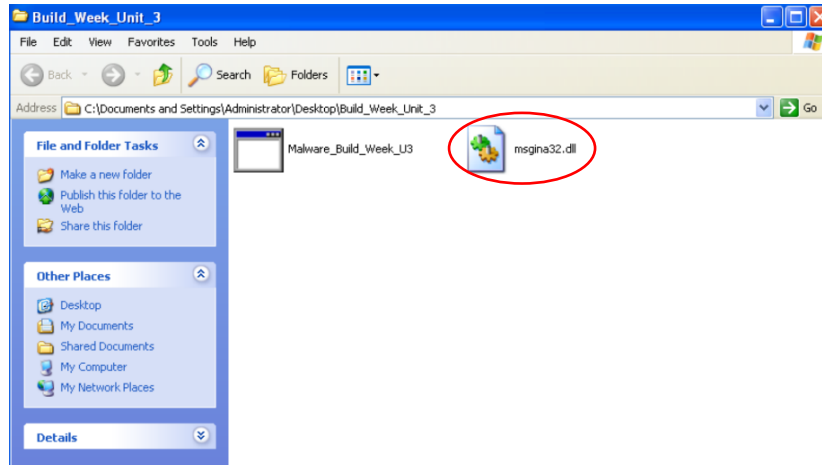
Segue una rappresentazione grafica del funzionamento della parte di codice analizzato.



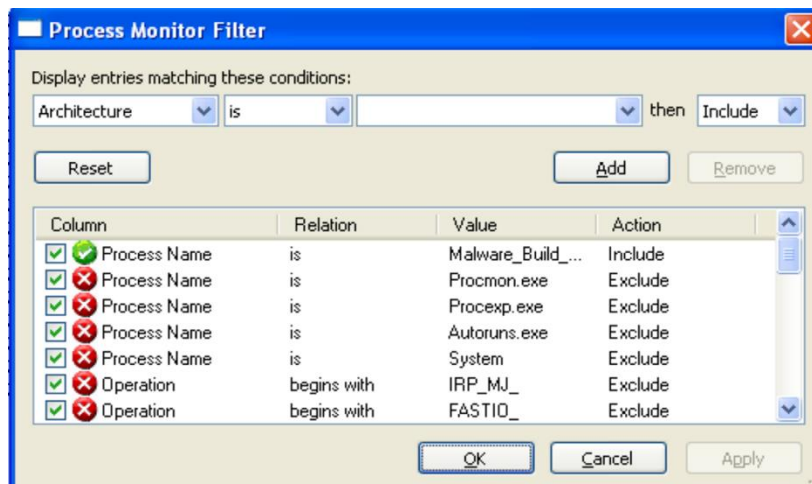
GIORNO 3

Dopo aver messo in sicurezza il laboratorio virtuale disabilitando scheda di rete, periferiche USB e cartelle condivise, sono passato all'analisi dinamica basica del malware eseguendo quindi il codice.

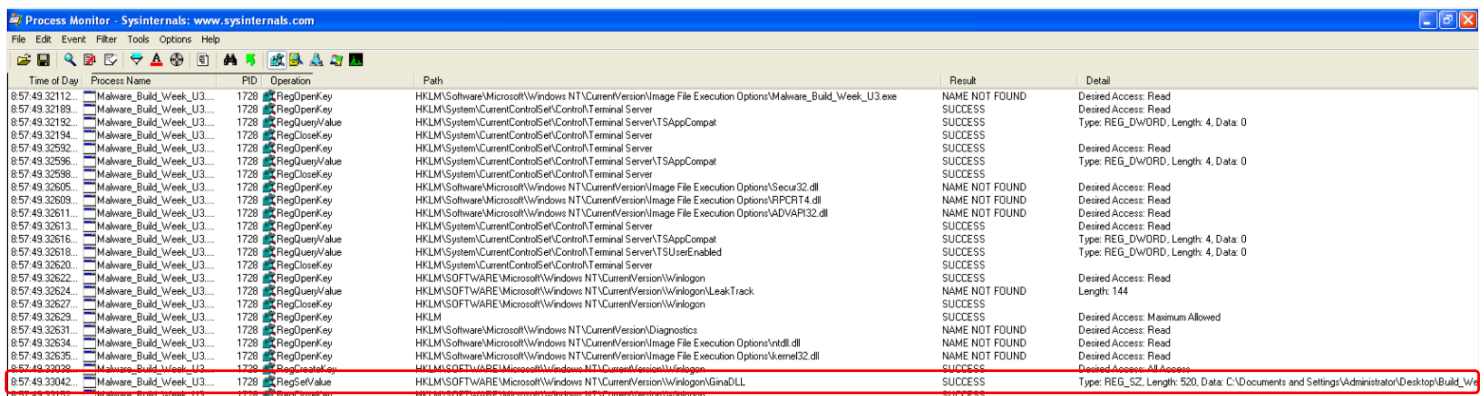
Una volta avviato il malware, si nota che si è creata una libreria nella stessa cartella in cui è stato posizionato, come mostrato nell'immagine sopra.



Successivamente, abbiamo avviato **"Process Monitor"**, uno strumento che consente l'analisi dell'eseguibile una volta avviato, e impostato il filtro **"process name"** per cercare tutte le chiavi di registro aggiunte o modificate dal virus.



Questo ci ha permesso di monitorare l'attività del malware e identificare le modifiche effettuate alle chiavi di registro nel sistema.



Con l'operazione **RegSetValue**, è stato aggiunto un nuovo valore alla chiave evidenziata nell'immagine. L'operazione RegSetValue consente di impostare o modificare un valore all'interno del Registro di sistema di Windows.

Nell'immagine, il malware ha utilizzato questa operazione per aggiungere un nuovo valore a una specifica chiave di registro nel sistema.

Questa modifica potrebbe avere implicazioni per il funzionamento del sistema o per le impostazioni di sicurezza.

Per confermare l'esistenza della nuova chiave di registro, abbiamo effettuato una verifica utilizzando **Regedit**.

Aperta l'applicazione, abbiamo esaminato le chiavi di registro per individuare quella creata dal malware.

Questa verifica ci ha permesso di confermare la presenza della nuova chiave di registro nel sistema.

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	

Successivamente, abbiamo applicato un ulteriore filtro per visualizzare il tipo di operazione sul file system, andando ad analizzare la chiamata di sistema che ha modificato il contenuto della cartella in cui è presente l'eseguibile del malware.

Questa operazione ci ha permesso di identificare l'operazione specifica che ha alterato i file o le cartelle nella directory in questione.

8:57:49.31072...	Malware_Build_Week_U3...	1728	CreateFile	C:\WINDOWS\system32\wtsapi32.dll	SUCCESS	Desired Access: Execute/Traverse, Disposition: Open, Options: Non-Directory File, Attributes: f
8:57:49.31105...	Malware_Build_Week_U3...	1728	CreateFile	C:\WINDOWS\system32\setupapi.dll	SUCCESS	Desired Access: Execute/Traverse, Disposition: Open, Options: Non-Directory File, Attributes: f
8:57:49.31141...	Malware_Build_Week_U3...	1728	CreateFile	C:\WINDOWS\system32\shlwapi.dll	SUCCESS	Desired Access: Execute/Traverse, Disposition: Open, Options: Non-Directory File, Attributes: f
8:57:49.31176...	Malware_Build_Week_U3...	1728	CreateFile	C:\WINDOWS\system32\apphelp.dll	SUCCESS	Desired Access: Execute/Traverse, Disposition: Open, Options: Non-Directory File, Attributes: f
8:57:49.32130...	Malware_Build_Week_U3...	1728	CreateFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3	SUCCESS	Desired Access: Execute/Traverse, Synchronize, Disposition: Open, Options: Directory, Synch
8:57:49.32694...	Malware_Build_Week_U3...	1728	CreateFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\msgina32.dll	SUCCESS	Desired Access: Generic Write, Read Attributes, Disposition: OverwriteIf, Options: Synchronou

Attraverso l'operazione CreateFile, il malware ha creato un nuovo file, come evidenziato nell'immagine sopra. L'operazione CreateFile è una chiamata di sistema che consente di creare un nuovo file nel sistema operativo. Nell'immagine, è possibile vedere che il malware ha eseguito questa operazione per creare un nuovo file all'interno della cartella indicata. Questo nuovo file potrebbe contenere codice dannoso o essere utilizzato per scopi malevoli.

Event Properties	
Event	Process Stack
Date:	7/17/2023 8:57:49.3269424 AM
Thread:	1128
Class:	File System
Operation:	CreateFile
Result:	SUCCESS
Path:	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\msgina32.dll
Duration:	0.0014174
Desired Access:	Generic Write, Read Attributes
Disposition:	OverwriteIf
Options:	Synchronous IO Non-Alert, Non-Directory File
Attributes:	N
ShareMode:	Read, Write
AllocationSize:	0
OpenResult:	Created

GIORNO 4

GINA.dll (Acronimo di “Graphical Identification ‘N’ Authentication”) è un componente lecito di Windows utilizzato per la gestione di impostazioni e credenziali di accesso.

Nello specifico, le principali funzioni del componente sono:

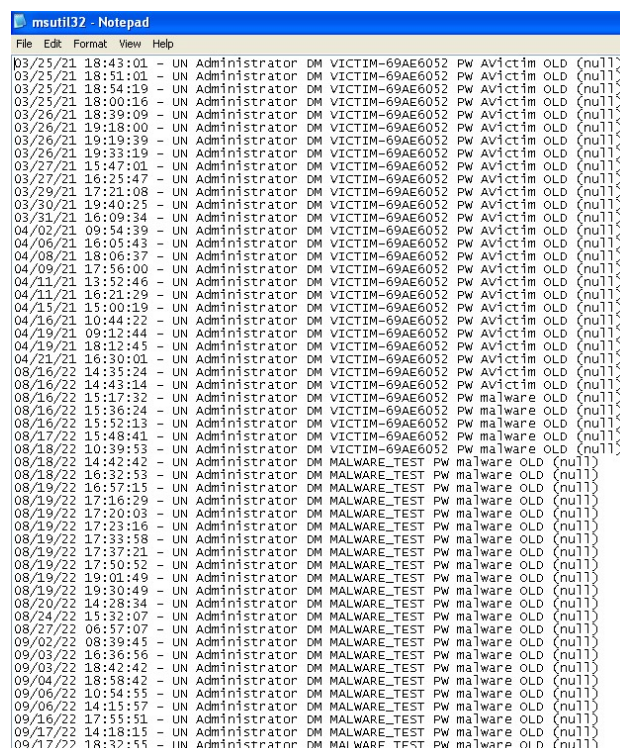
- Monitoraggio della firma di accesso condiviso: GINA è responsabile del riconoscimento di una *sequenza di attenzione sicura* (SAS), del monitoraggio degli eventi sas e della notifica di Winlogon (responsabile, tra le altre cose, del providing dei servizi di rete) quando si è verificata una firma di accesso condiviso;
- Elaborazione della firma di Accesso Condiviso;
- Attivazione della Shell.

In parole più semplici, l'uso più comune di GINA consiste nel comunicare con un dispositivo esterno, come ad esempio un *lettore di smart card*. È essenziale impostare il parametro di avvio per il driver di dispositivo sul sistema (Winnt.h: SERVICE_SYSTEM_START) per assicurarsi che il driver venga caricato dal momento in cui viene richiamata GINA.

Lo scopo, dunque, di una dll GINA è fornire procedure personalizzabili di identificazione e autenticazione dell'utente. La dll GINA predefinita esegue questa operazione delegando il monitoraggio degli eventi di firma di accesso condiviso a Winlogon, come già anticipato, che riceve ed elabora le *sequenze di attenzione protette*.

Come notato durante le precedenti fasi di analisi, il Malware va a sostituire la versione lecita di GINA.dll con una versione “contraffatta”. Tale processo risulterà essere estremamente pericoloso per l’utente della macchina infetta in quanto tale programma si comporterà a tutti gli effetti come uno **Spyware**, andando a sottrarre all’utente che incappa nel Malware una certa quantità di dati sensibili, come le credenziali di autenticazione.

Si è inoltre osservato, difatti, che la dll installata dal dropper crea nel percorso di sistema “WINDOWS32” un nuovo file, chiamato “**MSutils32.sys**”, che va a tenere traccia dei Log dell’utente inconsapevole sulla macchina infetta.



```
msutils32 - Notepad
File Edit Format View Help
03/25/21 18:43:01 - UN Administrator DM VICTIM-69AE6052 PW Avictim OLD (null)
03/25/21 18:51:01 - UN Administrator DM VICTIM-69AE6052 PW Avictim OLD (null)
03/25/21 18:54:19 - UN Administrator DM VICTIM-69AE6052 PW Avictim OLD (null)
03/25/21 18:00:16 - UN Administrator DM VICTIM-69AE6052 PW Avictim OLD (null)
03/26/21 18:39:09 - UN Administrator DM VICTIM-69AE6052 PW Avictim OLD (null)
03/26/21 19:18:00 - UN Administrator DM VICTIM-69AE6052 PW Avictim OLD (null)
03/26/21 19:19:39 - UN Administrator DM VICTIM-69AE6052 PW Avictim OLD (null)
03/26/21 19:33:19 - UN Administrator DM VICTIM-69AE6052 PW Avictim OLD (null)
03/27/21 15:47:01 - UN Administrator DM VICTIM-69AE6052 PW Avictim OLD (null)
03/27/21 16:25:47 - UN Administrator DM VICTIM-69AE6052 PW Avictim OLD (null)
03/29/21 17:21:08 - UN Administrator DM VICTIM-69AE6052 PW Avictim OLD (null)
03/30/21 19:40:25 - UN Administrator DM VICTIM-69AE6052 PW Avictim OLD (null)
03/31/21 16:09:34 - UN Administrator DM VICTIM-69AE6052 PW Avictim OLD (null)
04/02/21 09:54:39 - UN Administrator DM VICTIM-69AE6052 PW Avictim OLD (null)
04/06/21 18:05:43 - UN Administrator DM VICTIM-69AE6052 PW Avictim OLD (null)
04/08/21 18:06:37 - UN Administrator DM VICTIM-69AE6052 PW Avictim OLD (null)
04/09/21 17:56:00 - UN Administrator DM VICTIM-69AE6052 PW Avictim OLD (null)
04/11/21 13:52:46 - UN Administrator DM VICTIM-69AE6052 PW Avictim OLD (null)
04/11/21 16:21:29 - UN Administrator DM VICTIM-69AE6052 PW Avictim OLD (null)
04/15/21 15:00:19 - UN Administrator DM VICTIM-69AE6052 PW Avictim OLD (null)
04/16/21 10:44:22 - UN Administrator DM VICTIM-69AE6052 PW Avictim OLD (null)
04/19/21 09:12:44 - UN Administrator DM VICTIM-69AE6052 PW Avictim OLD (null)
04/19/21 18:12:45 - UN Administrator DM VICTIM-69AE6052 PW Avictim OLD (null)
04/21/21 16:30:01 - UN Administrator DM VICTIM-69AE6052 PW Avictim OLD (null)
08/16/22 14:35:34 - UN Administrator DM VICTIM-69AE6052 PW Avictim OLD (null)
08/16/22 14:43:14 - UN Administrator DM VICTIM-69AE6052 PW Avictim OLD (null)
08/16/22 15:17:32 - UN Administrator DM VICTIM-69AE6052 PW malware OLD (null)
08/16/22 15:36:24 - UN Administrator DM VICTIM-69AE6052 PW malware OLD (null)
08/16/22 15:52:13 - UN Administrator DM VICTIM-69AE6052 PW malware OLD (null)
08/17/22 15:48:41 - UN Administrator DM VICTIM-69AE6052 PW malware OLD (null)
08/18/22 10:39:53 - UN Administrator DM VICTIM-69AE6052 PW malware OLD (null)
08/18/22 14:42:42 - UN Administrator DM MALWARE_TEST PW malware OLD (null)
08/18/22 16:32:53 - UN Administrator DM MALWARE_TEST PW malware OLD (null)
08/19/22 16:57:15 - UN Administrator DM MALWARE_TEST PW malware OLD (null)
08/19/22 17:16:29 - UN Administrator DM MALWARE_TEST PW malware OLD (null)
08/19/22 17:20:03 - UN Administrator DM MALWARE_TEST PW malware OLD (null)
08/19/22 17:23:16 - UN Administrator DM MALWARE_TEST PW malware OLD (null)
08/19/22 17:33:58 - UN Administrator DM MALWARE_TEST PW malware OLD (null)
08/19/22 17:37:21 - UN Administrator DM MALWARE_TEST PW malware OLD (null)
08/19/22 17:50:52 - UN Administrator DM MALWARE_TEST PW malware OLD (null)
08/19/22 19:01:49 - UN Administrator DM MALWARE_TEST PW malware OLD (null)
08/19/22 19:30:49 - UN Administrator DM MALWARE_TEST PW malware OLD (null)
08/20/22 14:28:34 - UN Administrator DM MALWARE_TEST PW malware OLD (null)
08/24/22 15:32:07 - UN Administrator DM MALWARE_TEST PW malware OLD (null)
08/27/22 06:57:07 - UN Administrator DM MALWARE_TEST PW malware OLD (null)
09/02/22 08:39:45 - UN Administrator DM MALWARE_TEST PW malware OLD (null)
09/03/22 16:36:56 - UN Administrator DM MALWARE_TEST PW malware OLD (null)
09/03/22 18:42:42 - UN Administrator DM MALWARE_TEST PW malware OLD (null)
09/04/22 18:58:42 - UN Administrator DM MALWARE_TEST PW malware OLD (null)
09/06/22 10:54:55 - UN Administrator DM MALWARE_TEST PW malware OLD (null)
09/06/22 14:15:57 - UN Administrator DM MALWARE_TEST PW malware OLD (null)
09/16/22 17:55:51 - UN Administrator DM MALWARE_TEST PW malware OLD (null)
09/17/22 14:18:15 - UN Administrator DM MALWARE_TEST PW malware OLD (null)
09/17/22 18:32:55 - UN Administrator DM MALWARE_TEST PW malware OLD (null)
```


Tali Log possono essere costantemente verificati dall'utente malevolo da remoto, sfruttando la connettività che GINA.dll mette a disposizione mediante la componente lecita di Windows WinLogOn che di fatto andrà a gestire la dll installata come quella legittima e dall'import di eventuali ulteriori funzioni di rete in fase di RunTime ("LoadLibrary" e "GetProcAddress" possono essere facilmente notate usando CFF Explorer).

Eseguendo ulteriori analisi con **Process Monitor**, **CFF** e **IDA**, è stato possibile notare l'interazione tra il File System e la nuova componente fraudolenta, come evidenziato anche nelle primissime analisi svolte sull'eseguibile.

9:28:23.90414...	Explorer.EXE	192	QueryOpen	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\Malware_Build_Week_U3.exe	SUCCESS
9:28:23.90421...	Explorer.EXE	192	QueryOpen	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\Malware_Build_Week_U3.exe	SUCCESS
9:28:23.90434...	Explorer.EXE	192	QueryOpen	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\Malware_Build_Week_U3.exe	SUCCESS
9:28:23.90441...	Explorer.EXE	192	CreateFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\Malware_Build_Week_U3.exe	SUCCESS
9:28:23.90445...	Explorer.EXE	192	QueryBasicInformationFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\Malware_Build_Week_U3.exe	SUCCESS
9:28:23.90449...	Explorer.EXE	192	SetBasicInformationFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\Malware_Build_Week_U3.exe	SUCCESS
9:28:23.90459...	Explorer.EXE	192	ReadFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\Malware_Build_Week_U3.exe	SUCCESS
9:28:23.90466...	Explorer.EXE	192	QueryStandardInformationFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\Malware_Build_Week_U3.exe	SUCCESS
9:28:23.90611...	Malware_Build_Week_U3...	1944	CreateFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\msgina32.dll	SUCCESS
9:28:23.90617...	Malware_Build_Week_U3...	1944	CreateFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3	SUCCESS
9:28:23.90621...	Malware_Build_Week_U3...	1944	CloseFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3	NOTIFY_ENUM_DIR
9:28:23.90643...	Explorer.EXE	192	NotifyChangeDirectory	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3	NOTIFY_ENUM_DIR
9:28:23.90648...	Explorer.EXE	192	NotifyChangeDirectory	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3	SUCCESS
9:28:23.90655...	Malware_Build_Week_U3...	1944	WriteFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\msgina32.dll	SUCCESS
9:28:23.90938...	Malware_Build_Week_U3...	1944	WriteFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\msgina32.dll	SUCCESS
9:28:23.90954...	Malware_Build_Week_U3...	1944	CloseFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\msgina32.dll	SUCCESS
9:28:23.90973...	Explorer.EXE	192	NotifyChangeDirectory	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3	SUCCESS
9:28:23.90977...	Explorer.EXE	192	NotifyChangeDirectory	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3	SUCCESS
9:28:23.91280...	Malware_Build_Week_U3...	1944	CloseFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3	SUCCESS

```

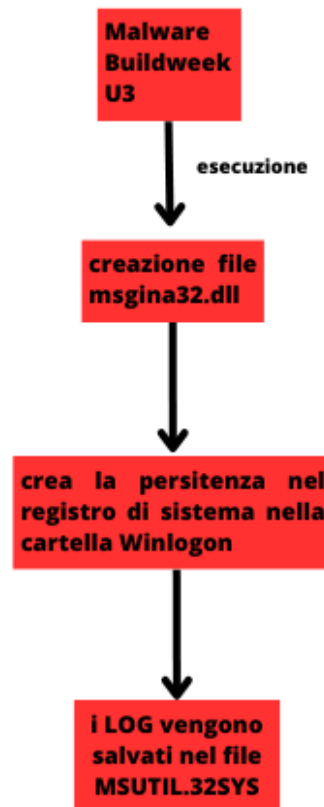
; int __cdecl sub_10001570(DWORD dwMessageId, wchar_t *, char)
sub_10001570 proc near

hMem= dword ptr -854h
var_850= word ptr -850h
var_820= 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 aHsutil32_sys ; "hsutil32.sys"
call    _wfopen
mov     esi, eax
add     esp, 10h
test    esi, esi
jz      loc_1000164F

```

Di seguito una rappresentazione grafica semplificata del flusso di funzionamento dell'intero pacchetto di Malware.



CONCLUSIONI

A seguito delle molteplici analisi effettuate sul file eseguibile e in virtù di tutte le evidenze emerse, siamo riusciti a ricostruire il comportamento del Malware, rappresentato nel diagramma precedentemente presentato.

Risulta chiaro come il primo eseguibile lanciato vada a recuperare nella sua sezione "resources", un secondo file, il già nominato "Gina.dll".

A seguito della sua installazione, le chiavi di registro di Windows verranno modificate affinché il sistema vada ad interagire con questa nuova componente ed un file chiamato "**MSutils32.sys**" verrà creato nella directory di sistema "SYSTEM32" al fine di registrare i log dell'utente e, con molta probabilità, inviarli all'utente remoto.

Questo sarà possibile mediante una connessione creata tramite l'importazione delle funzionalità di condivisione di rete messe a disposizione dalla componente WinLogOn e/o importando le funzioni di connettività di rete in fase di RunTime. A sostegno di questa ipotesi, come già accennato, si può notare la presenza delle funzioni "LoadLibrary" e "GetProcAddress" individuate durante le prime analisi.

Si può dunque affermare con ragionevole certezza che l'eseguibile preso in analisi sia un **Dropper**, un particolare tipo di Malware che installa sulla macchina infetta ulteriori file o Malware. In questo specifico caso lo scopo è quello di acquisire da remoto le credenziali degli utenti, compreso l'utente amministratore, per eseguire con successo una procedura di Privilege Escalation sulla macchina target e ottenendone il completo controllo.

GIORNO 5

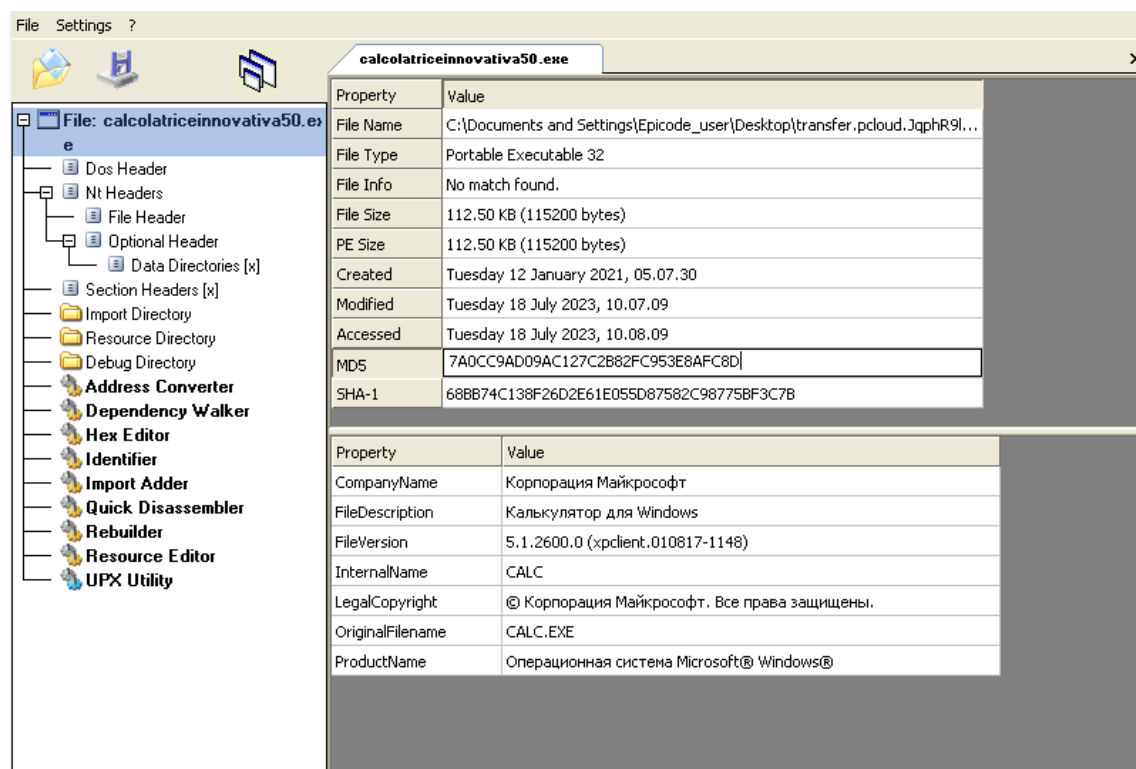
PARTE 1

Per poter eseguire le analisi degli ultimi due eseguibili, abbiamo dapprima trasferito i file scaricati da Mega a PCloud per renderli accessibili alla nostra macchina XP.

Dopo aver preso le usuali precauzioni per rendere sicuro l'ambiente di lavoro (scheda di rete disabilitata, condivisione file disabilitata, periferiche USB disattivate), si è proceduto con l'analisi del primo programma in maniera statica.

Tale analisi è stata in primo luogo eseguita con il tool **CFF Explorer** per verificare quali import e sezioni fossero contenuti nell'eseguibile una volta esaminati i dati principali dello stesso.

La prima cosa che salta all'occhio è che il copyright e altri dati del file "Calcolatrice" sono scritti in cirillico.



Passando, invece, alle sezioni di cui si compone il malware, si individuano:

- **.text**: contiene il codice vero e proprio del programma;
- **.data**: contiene le variabili globali e le funzioni importate;
- **.rsrc**: contiene le risorse aggiuntive dell'eseguibile.

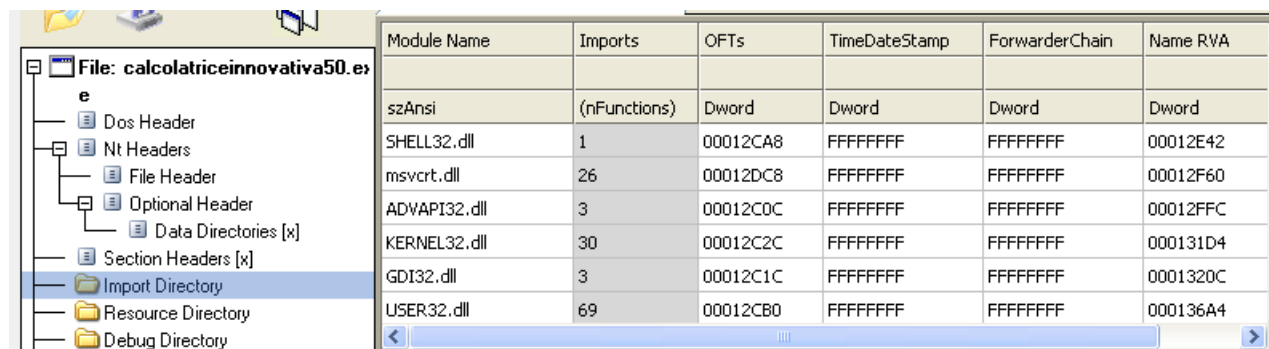
Quest'ultima sezione ci porta a pensare che si potrebbe trattare di un Dropper, pur non avendo ancora nessuna prova della nostra ipotesi, o di un Trojan, che potrebbe sfruttare risorse "ingannevoli", in questo caso come l'immagine di una calcolatrice, per spingere l'utente ad avviare il programma.

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address
Byte[8]	Dword	Dword	Dword	Dword	Dword
.text	000126B0	00001000	00012800	00000400	00000000
.data	0000101C	00014000	00000A00	00012C00	00000000
.rsrc	00008A70	00016000	00008C00	00013600	00000000

Successivamente passiamo allo studio delle librerie importate e delle funzioni utilizzate.

Notiamo immediatamente che il programma in questione risulta essere particolarmente “corposo” da questo punto di vista, infatti presenta ben sei librerie:

- **SHELL32.dll**: serve alla gestione di shell Windows;
- **msvcrt.dll**: contiene, fra l’altro, le funzioni standard del linguaggio C;
- **ADVAPI32.dll**: : responsabile dell’interazione con le chiavi di registro del sistema operativo;
- **KERNEL32.dll**: responsabile dell’interazione con il sistema operativo;
- **GDI32.dll**: contiene le funzioni necessarie all’interazione con la componente grafica del sistema e coadiuva Windows nella creazione di oggetti e immagini a due dimensioni;
- **USER32.dll**: contenente le API necessarie alla gestione e al funzionamento dell’Interfaccia Utente.



Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword
SHELL32.dll	1	00012CA8	FFFFFFFF	FFFFFFFF	00012E42
msvcrt.dll	26	00012DC8	FFFFFFFF	FFFFFFFF	00012F60
ADVAPI32.dll	3	00012C0C	FFFFFFFF	FFFFFFFF	00012FFC
KERNEL32.dll	30	00012C2C	FFFFFFFF	FFFFFFFF	000131D4
GDI32.dll	3	00012C1C	FFFFFFFF	FFFFFFFF	0001320C
USER32.dll	69	00012CB0	FFFFFFFF	FFFFFFFF	000136A4

Analizzando queste librerie nello specifico, notiamo una serie di funzioni ormai familiari, ma estremamente indicative ai fini dell’analisi.

In particolare, all’interno della libreria KERNEL32, si notano:

- **LoadLibraryA**
- **GetProcAddress**
- **CreateEventW**
- **CreateThread**
- **GetCommandLineW**
- **GlobalLock**

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

000130EA	77E730C1	0047	CreateEventW
000130DA	77E7AC37	0065	CreateThread
000130CC	77E74A69	02A9	ResetEvent

00013118	77E7166F	01E2	GlobalLock
0001300A	77E7C9DB	00FE	GetCommandLineW
0001301C	77E73679	0399	lstrcpwW
0001304A	77E641D5	0194	GetProfileIntW

All'interno della libreria ADVAPI32.dll, si notano le funzioni necessarie alla modifica delle chiavi di registro con il relativo ottenimento della persistenza:

- **RegOpenKeyExA**
- **RegQueryValueExA**
- **RegCloseKey**

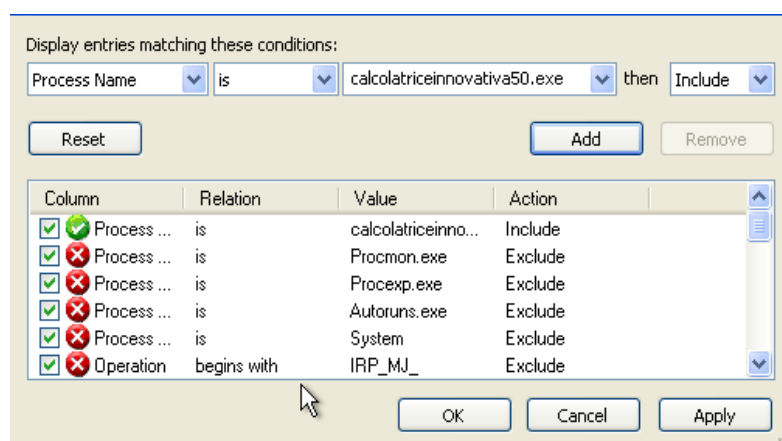
OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
00012FEC	77DC22EA	01E1	RegOpenKeyExA
00012FD8	77DC23D7	01EB	RegQueryValueExA
00012FCA	77DC189A	01C8	RegCloseKey

Analizzando, invece, la libreria GDI32.dll inizia a concretizzarsi l'ipotesi che si tratti di un Trojan in quanto le funzioni individuate all'interno potrebbero interagire con la sezione .rsrc per la creazione di un'icona ingannevole sul Desktop della vittima:

- **SetBkColor**
- **SetTextColor**
- **SetBkMode**

OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
000131E2	77C61E2E	0213	SetBkColor
000131F0	77C61D83	023A	SetTextColor
00013200	77C61EFF	0214	SetBkMode

Successivamente, abbiamo proseguito con l'analisi dinamica basica mediante l'osservazione del comportamento dell'eseguibile con il tool **"Process Monitor"**, impostando un apposito filtro per il monitoraggio esclusivo del malware in questione:



Dai risultati ottenuti, si è potuto notare come l’eseguibile modifichi le chiavi di registro per ottenere la persistenza (screen 3-4-5-6) e come vada ad interagire per la stessa ragione con le “Root Key” di Windows (screen 7).

Il malware, inoltre, va a creare dei file all’interno del percorso di sistema SYSTEM32 per l’ottenimento di una Shell remota (screen 1-3-4) successivamente utilizzabile mediante l’interazione con la componente WinLogOn (Screen 5).

Seguono gli screen che mostrano il comportamento dell’eseguibile:

Screen 1

10.53...	calcolatriceinno...	1804	CloseFile	C:\WINDOWS\system32\shell32.dll	SUCCESS	
10.53...	calcolatriceinno...	1804	CloseFile	C:\WINDOWS\system32\advapi32.dll	SUCCESS	
10.53...	calcolatriceinno...	1804	CloseFile	C:\WINDOWS\system32\rpcrt4.dll	SUCCESS	
10.53...	calcolatriceinno...	1804	CloseFile	C:\WINDOWS\system32\secur32.dll	SUCCESS	
10.53...	calcolatriceinno...	1804	CloseFile	C:\WINDOWS\system32\gdi32.dll	SUCCESS	
10.53...	calcolatriceinno...	1804	CloseFile	C:\WINDOWS\system32\user32.dll	SUCCESS	
10.53...	calcolatriceinno...	1804	CloseFile	C:\WINDOWS\system32\msvcrt.dll	SUCCESS	
10.53...	calcolatriceinno...	1804	CloseFile	C:\WINDOWS\system32\shlwapi.dll	SUCCESS	
10.53...	calcolatriceinno...	1804	CloseFile	C:\Documents and Settings\Epicode_u...	SUCCESS	
10.53...	calcolatriceinno...	1804	CloseFile	C:\WINDOWS\system32\ctype.nls	SUCCESS	
10.53...	calcolatriceinno...	1804	CloseFile	C:\WINDOWS\WinSxS\x86_Microsoft...	SUCCESS	
10.53...	calcolatriceinno...	1804	CloseFile	C:\WINDOWS\WindowsShell.Manifest	SUCCESS	
10.53...	calcolatriceinno...	1804	CreateFile	C:\WINDOWS\system32\ntdll.dll	SUCCESS	Desired Access: E...
10.53...	calcolatriceinno...	1804	CreateFileMap...	C:\WINDOWS\system32\ntdll.dll	SUCCESS	SyncType: SyncTy...
10.53...	calcolatriceinno...	1804	CreateFileMap...	C:\WINDOWS\system32\ntdll.dll	SUCCESS	SyncType: SyncTy...
10.53...	calcolatriceinno...	1804	CreateFile	C:\WINDOWS\system32\kernel32.dll	SUCCESS	Desired Access: E...
10.53...	calcolatriceinno...	1804	CreateFileMap...	C:\WINDOWS\system32\kernel32.dll	SUCCESS	SyncType: SyncTy...
10.53...	calcolatriceinno...	1804	CreateFileMap...	C:\WINDOWS\system32\kernel32.dll	SUCCESS	SyncType: SyncTy...
10.53...	calcolatriceinno...	1804	CreateFile	C:\WINDOWS\system32\shell32.dll	SUCCESS	Desired Access: E...
10.53...	calcolatriceinno...	1804	CreateFileMap...	C:\WINDOWS\system32\shell32.dll	SUCCESS	SyncType: SyncTy...
10.53...	calcolatriceinno...	1804	CreateFileMap...	C:\WINDOWS\system32\shell32.dll	SUCCESS	SyncType: SyncTy...
10.53...	calcolatriceinno...	1804	CreateFile	C:\WINDOWS\system32\advapi32.dll	SUCCESS	Desired Access: E...
10.53...	calcolatriceinno...	1804	CreateFileMap...	C:\WINDOWS\system32\advapi32.dll	SUCCESS	SyncType: SyncTy...
10.53...	calcolatriceinno...	1804	CreateFileMapping	C:\WINDOWS\system32\advapi32.dll	SUCCESS	SyncType: SyncTy...
10.53...	calcolatriceinno...	1804	CreateFile	C:\WINDOWS\system32\rpcrt4.dll	SUCCESS	Desired Access: E...
10.53...	calcolatriceinno...	1804	CreateFileMap...	C:\WINDOWS\system32\rpcrt4.dll	SUCCESS	SyncType: SyncTy...
10.53...	calcolatriceinno...	1804	CreateFileMap...	C:\WINDOWS\system32\rpcrt4.dll	SUCCESS	SyncType: SyncTy...
10.53...	calcolatriceinno...	1804	CreateFile	C:\WINDOWS\system32\secur32.dll	SUCCESS	Desired Access: E...
10.53...	calcolatriceinno...	1804	CreateFileMap...	C:\WINDOWS\system32\secur32.dll	SUCCESS	SyncType: SyncTy...
10.53...	calcolatriceinno...	1804	CreateFileMap...	C:\WINDOWS\system32\secur32.dll	SUCCESS	SyncType: SyncTy...

Screen 2

calatriceinno...	2524	RegOpenKey	HKLM\Software\Microsoft\Windows\CurrentVersion\SideBySide\AssemblyStorageRoots	NAME NO
calatriceinno...	2524	QueryOpen	C:\Documents and Settings\Epicode_user\Desktop\transfer.pcloud.JqphR9la\calcolatriceinnovativa50.exe\cal...	NAME NO
calatriceinno...	2524	QueryOpen	C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_3...	SUCCESS
calatriceinno...	2524	CreateFile	C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_3...	SUCCESS
calatriceinno...	2524	Thread Exit		SUCCESS
calatriceinno...	2524	Process Exit		SUCCESS

Screen 3

CreateFile	C:\WINDOWS\system32\SHELL32.dll.124.Manifest	NAME NO
RegOpenKey	HKCU	SUCCESS
RegOpenKey	HKCU\Software\Policies\Microsoft\Control Panel\Desktop	NAME NO
RegOpenKey	HKCU\Control Panel\Desktop	SUCCESS
RegQueryValue	HKCU\Control Panel\Desktop\MultiUILanguageId	NAME NO
RegCloseKey	HKCU\Control Panel\Desktop	SUCCESS
RegCloseKey	HKCU	SUCCESS
CreateFile	C:\WINDOWS\system32\SHELL32.dll.124.Config	NAME NO
CloseFile	C:\WINDOWS\system32\shell32.dll	SUCCESS
RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager	SUCCESS
RegQueryValue	HKLM\System\CurrentControlSet\Control\Session Manager\SafeDllSearchMode	NAME NO
RegCloseKey	HKLM\System\CurrentControlSet\Control\Session Manager	SUCCESS
RegOpenKey	HKLM\Software\Microsoft\Windows\CurrentVersion\SideBySide\AssemblyStorageRoots	NAME NO
QueryOpen	C:\Documents and Settings\Epicode_user\Desktop\transfer.pcloud.JqphR9la\calcolatriceinnovativa50.exe\cal...	NAME NO

Screen 4

2524	CreateFileMap...	C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_3...	SUCCE
2524	RegOpenKey	HKLM\System\CurrentControlSet\Control\SafeBoot\Option	NAME I
2524	RegOpenKey	HKLM\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers	SUCCE
2524	RegQueryValue	HKLM\SOFTWARE\Policies\Microsoft\Windows\Safer\CodeIdentifiers\TransparentEnabled	SUCCE
2524	RegCloseKey	HKLM\SOFTWARE\Policies\Microsoft\Windows\Safer\CodeIdentifiers	SUCCE
2524	RegOpenKey	HKCU\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers	NAME I
2524	CloseFile	C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_3...	SUCCE
2524	Load Image	C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_3...	SUCCE
2524	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\comctl32.dll	NAME I
2524	RegOpenKey	HKCU	SUCCE
2524	RegOpenKey	HKCU\Software\Policies\Microsoft\Control Panel\Desktop	NAME I
2524	RegOpenKey	HKCU\Control Panel\Desktop	SUCCE
2524	RegQueryValue	HKCU\Control Panel\Desktop\MultiUILanguageId	NAME I
2524	RegCloseKey	HKCU\Control Panel\Desktop	SUCCE
2524	RegCloseKey	HKCU	SUCCE
2524	QueryOpen	C:\WINDOWS\WindowsShell.Manifest	SUCCE
2524	CreateFile	C:\WINDOWS\WindowsShell.Manifest	SUCCE
2524	CreateFileMap...	C:\WINDOWS\WindowsShell.Manifest	SUCCE
2524	QueryStandardl...	C:\WINDOWS\WindowsShell.Manifest	SUCCE

Screen 5

472	RegOpenKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS	Desired Access: R..
472	RegQueryValue	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Le...	NAME NOT FOUND	Length: 144
472	RegCloseKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS	

Screen 6

RegOpenKey	HKLM\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers
RegQueryValue	HKLM\SOFTWARE\Policies\Microsoft\Windows\Safer\CodeIdentifiers\TransparentEnabled
RegCloseKey	HKLM\SOFTWARE\Policies\Microsoft\Windows\Safer\CodeIdentifiers
RegOpenKey	HKCU\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers

Screen 7

CreateFileMap...	C:\WINDOWS\system32\sysprep
CreateFile	C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83\comctl32.dll
CreateFileMap...	C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83\comctl32.dll
QueryStandardl...	C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83\comctl32.dll
CreateFileMap...	C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83\comctl32.dll
CreateFile	C:\WINDOWS\WINDOWS\SHELL.MANIFEST

In seguito a queste analisi, si giunge alla conclusione che l'eseguibile analizzato sia una **Trojan** che, presentandosi all'utente come calcolatrice, va invece ad avviare una shell remota che garantisce all'utente malintenzionato una **backdoor** sul sistema infetto.

Come controllo finale di quanto ipotizzato, si procede con l'analisi dell'hash tramite **VirusTotal**.

Lo strumento, infatti, riporta che ben cinquantatré vendors su settantuno identificano il programma come un **Trojan/Backdoor**, andando quindi a confermare le nostre ipotesi.

53

71

63 security vendors and no sandboxes flagged this file as malicious

c7f8e8f17dcd7de447cc6bd99952be9c78112542030d49797683e7df6ad3e7

CALC.EXE

peexe

detect-debug-environment

checks-user-input

Size

112.50 KB

Last Analysis Date

18 hours ago

EXE

Community Score

DETECTION

DETAILS

RELATIONS

BEHAVIOR

COMMUNITY

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

Popular threat label

trojan.swroot/cryptz

Threat categories

trojan

Family labels

swroot cryptz marte

Security vendors' analysis

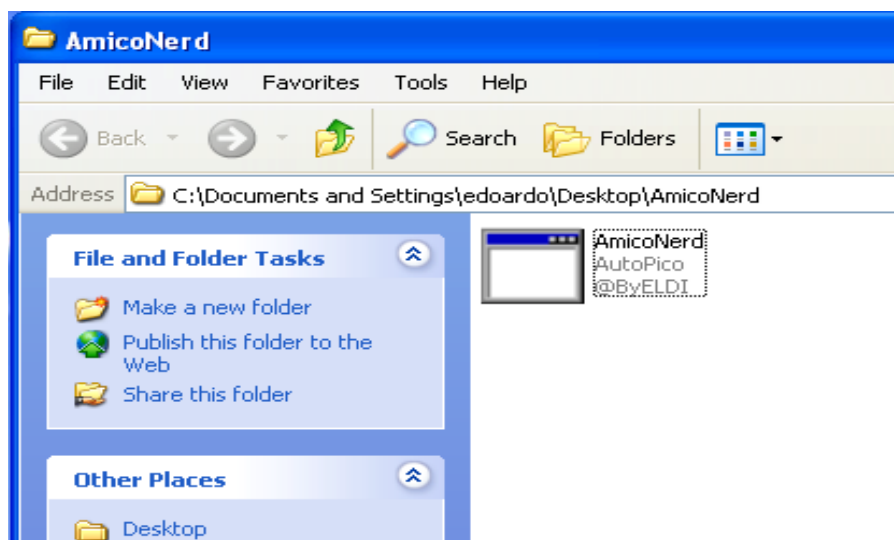
Do you want to automate checks?

AhnLab-V3	Backdoor/Win32.Bitrose.C64906	ALYac	Trojan.CryptZ.Marte.1.Gen
Arcabit	Trojan.CryptZ.Marte.1.Gen	Avast	Win32.SwPatch [Wim]
AVG	Win32.SwPatch [Wim]	Avira (no cloud)	TR/Patched.Gen2
BitDefender	Trojan.CryptZ.Marte.1.Gen	BitDefenderTheta	Gen.NN.Zexaf.36318.hm0@aOQzbzfc
Bkav Pro	W32.AIDetect/Malware	ClamAV	Win.Trojan.MSShellcode-6360730-0

PARTE 2

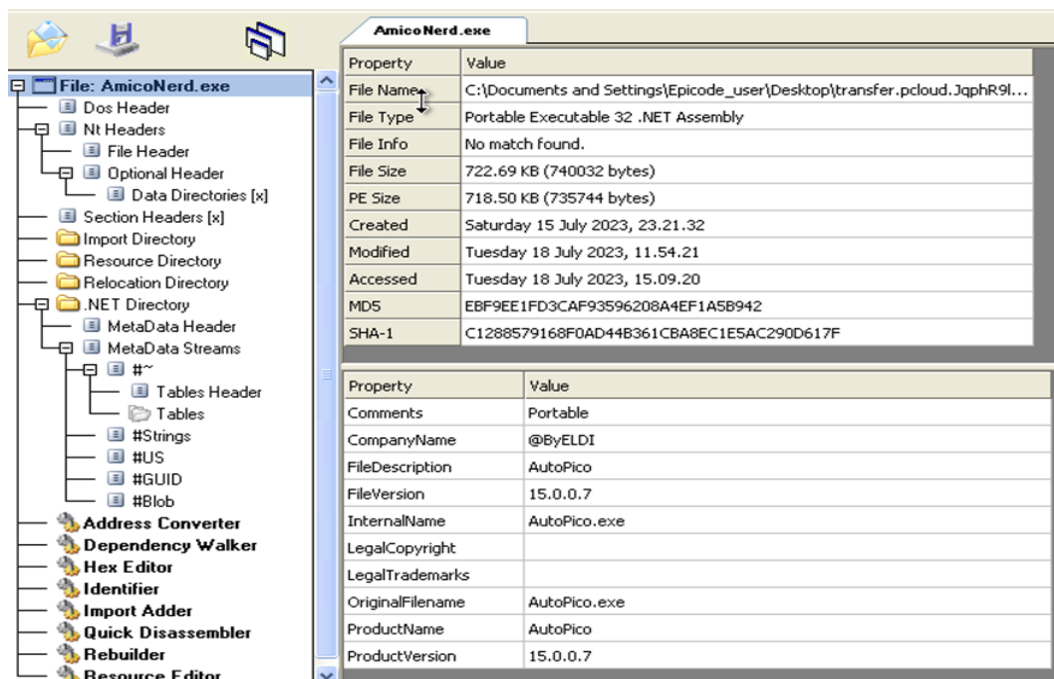
Per quanto riguarda il secondo programma oggetto di analisi, dobbiamo convincere il dipendente "sveglio" che il file avviato da un amico su un pc aziendale si tratta di un file malevolo.

Iniziamo dunque mettendo il pc in sicurezza creando una sessione istantanea che ci consentirà di agire in totale sicurezza e disabilitando scheda di rete e porte usb.



Messo in sicurezza l'ambiente di lavoro, apriamo la cartella contenente il file e notiamo, in primo luogo, che il file eseguibile AmicoNerd in questione si tratta di un file **AutoPico** che sta ad indicare un eseguibile non necessario per windows, ma che per i programmi antivirus risulta malevolo.

Iniziamo, ora, l'analisi del malware aprendo il file tramite il tool **CFF Explorer**.

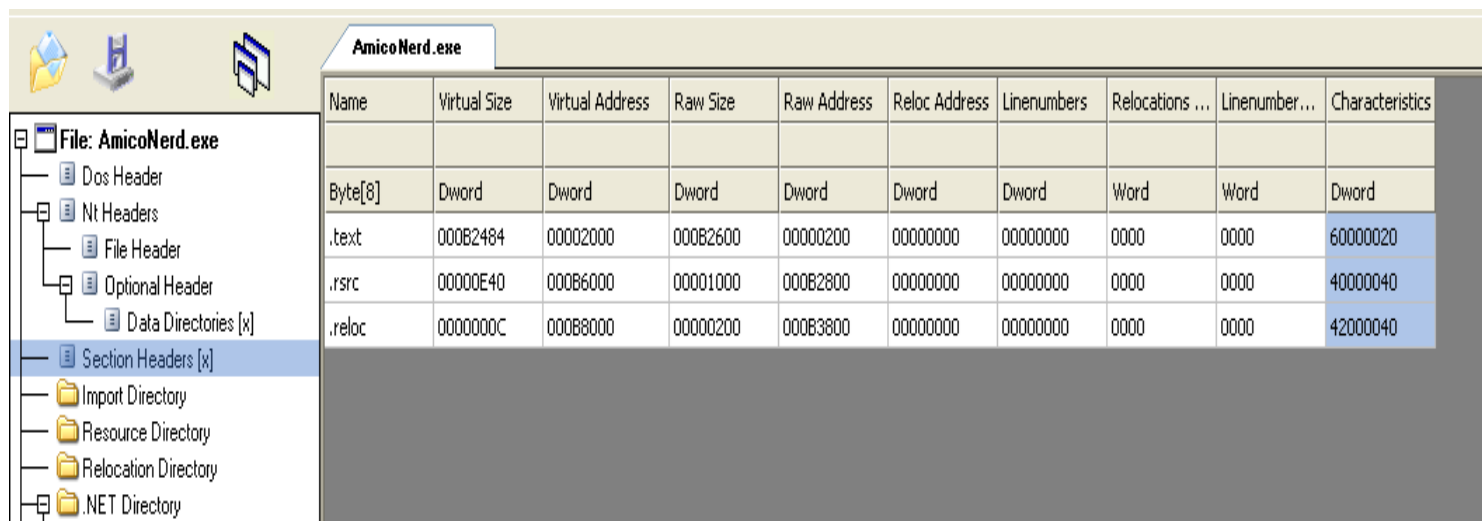


Nella prima schermata possiamo notare alcune informazioni generali, tra cui il nome del file originale (AutoPico.exe), la descrizione del file e l'hash MD5 che utilizzeremo in seguito per ampliare la nostra analisi.

Successivamente ci spostiamo nella voce del menù “section headers” per visualizzare le varie sezioni di cui è composto il malware:

- .text
- .rsrc
- .reloc

Delle prime due abbiamo già ampiamente parlato nei giorni scorsi, **.reloc**, invece, contiene le informazioni necessarie per la rilocazione delle posizioni assolute dei dati e del codice all'interno dell'eseguibile durante il processo di caricamento dell'immagine in memoria.

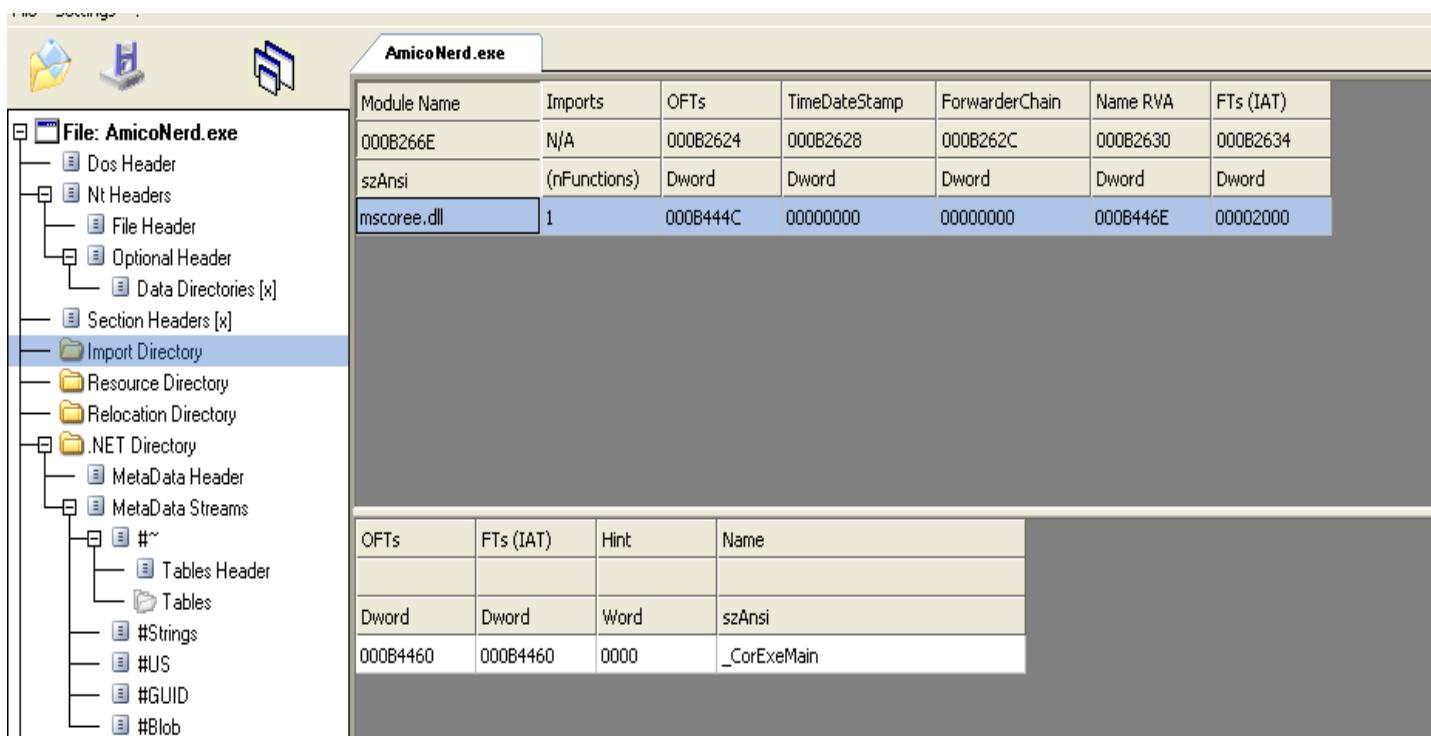


The screenshot shows the PE header view of AmicoNerd.exe. The left pane lists the file structure, with 'Section Headers [x]' selected. The right pane displays a table of section headers.

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations ...	Linenumber...	Characteristics
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

Spostandoci in seguito nel menù “import directory” troviamo la libreria **mscorlib.dll** che è una libreria di collegamento dinamico di sistema di Microsoft utilizzata per gestire l'integrazione e l'avvio dell'infrastruttura di esecuzione comune del runtime .NET Framework all'interno di applicazioni Windows. Questa libreria è fondamentale per il funzionamento delle applicazioni basate su .NET Framework.

Dalla stessa schermata si può notare inoltre la funzione **_CoreExeMain** importata da tale libreria. Questa funzione viene utilizzata internamente dal Common Language Runtime (CLR) di .NET Framework per avviare l'esecuzione di un'immagine eseguibile .NET e gestire il punto di ingresso dell'applicazione.



The screenshot shows the PE header view of AmicoNerd.exe with the 'Import Directory' selected in the left pane. The right pane displays two tables: the main import table and a detailed view of the imported function.

Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
000B266E	N/A	000B2624	000B2628	000B262C	000B2630	000B2634
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
mscorlib.dll	1	000B444C	00000000	00000000	000B446E	00002000

OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
000B4460	000B4460	0000	_CorExeMain

A questo punto proviamo ad eseguire il file e notiamo, all'interno della cartella dove è contenuto, che viene creata una ulteriore cartella chiamata "LOG" dentro la quale viene creato un file di testo con le azioni che il malware compie.

Si può infatti vedere che esso tenta di connettersi ad internet ma ottiene un codice di errore poiché la macchina sulla quale stiamo facendo l'analisi ha la scheda di rete disabilitata.

Notiamo, inoltre, un sito web al quale il malware fa riferimento.

Questo fatto, unito all'analisi eseguita precedentemente, ci fa pensare che il file sia malevolo.

Per avere un'ulteriore conferma, esaminiamo il file con **VirusTotal** tramite l'hash MD5 trovato in precedenza.

Si può notare, infatti, che 53 security vendors segnalano tale file come malevole.

Sempre dall'analisi di VirusTotal, inoltre, è possibile capire che il malware rientra nella categoria di minaccia **hacktool** che è un termine generico per identificare software o strumenti utilizzati per scopi leciti o illeciti di hacking. Sebbene alcuni di essi siano strumenti legittimi utilizzati per la sicurezza informatica, altri sono parte del malware e vengono utilizzati per compiere azioni illegali e dannose. Gli hacktool malevoli possono essere utilizzati per attacchi di hacking, furto di dati sensibili, spionaggio, diffusione di virus o ransomware e altre azioni dannose.

Infine, si nota che il malware è etichettato anche come **autokms** o **kmsactivator** che rappresentano un programma utilizzato per la creazione di licenze false e generazione di chiavi di attivazione per Office e/o Windows.

Popular threat label 🔔 hacktool.rpchook/autokms		Threat categories	Family labels
		hacktool trojan pua	rpchook autokms kmsactivator
Security vendors' analysis 🔔 Do you want to automate checks?			
Acronis (Static ML)	🔔 Suspicious	AhnLab-V3	🔔 HackTool/Win.AutoKMS.C948312
ALYac	🔔 Application.Hacktool.KMSActivator.AQ	Antiy-AVL	🔔 RiskWare[NetTool]/Win64.RPCHook
Arcabit	🔔 Application.KMS	Avast	🔔 Win32.MiscX-gen [PUP]
AVG	🔔 Win32:MiscX-gen [PUP]	BitDefender	🔔 Application.Hacktool.KMSActivator.AQ