

## PROGETTO MODULO 6

### Traccia: Malware Analysis

#### Analisi Statica:

In riferimento al file eseguibile **Malware\_Build\_Week\_U3**, rispondere ai quesiti utilizzando i tool e le tecniche apprese nelle lezioni teoriche:

1. Quanti **parametri** sono passati alla funzione `Main()`?
2. Quante **variabili** sono dichiarate all'interno della funzione `Main()`?
3. Quali **sezioni** sono presenti all'interno del file **eseguibile**? Descrivere brevemente almeno 2 di quelle identificate.
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. Utilizzare le funzioni che sono richiamate all'interno delle librerie per supportare le vostre ipotesi.

\*\*\*\*\*

### Analisi Statica

Per effettuare l'**Analisi Statica** ci avvaliamo del tool **IDA**.

- 1) I **Parametri** passanti alla funzione `Main()`, sono 3 ovvero:

`argc, argv, envp`.

In quanto i Parametri si trovano ad un *offset* (ovvero la differenza rispetto ad un valore di riferimento) *positivo* rispetto ad EBP.

- 2) Le **Variabili** dichiarate all'interno della funzione sono 4:

`hModule, Data, var_8, var_4`

In quanto le **Variabili** sono ad un *offset negativo* rispetto al registro EBP.

```
; 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
```

[**Parametri**  
evidenziati in  
arancione]

[**Variabili**  
evidenziati in  
celeste]

- 3) Per determinare le **Sezioni** presenti all'interno del file eseguibile, ci avvaliamo del tool **CFF Explorer**.

Come vediamo in figura sono presenti 4 sezioni:

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** —> che contiene le righe di codice che la CPU eseguirà una volta che il software sarà avviato.

Questa sezione contiene il “Code Entry Point 00001487”;

- **.rdata** —> contiene info circa le librerie e le funzioni importate ed esportate dall'eseguibile. E qui il programma ci dice che questa sezione contiene:

“Data: 00007000

Import Directory: 000074EC

Import Address Table Directory: 00007000”

- **.data** —> contiene di solito i dati / le variabili globali del programma eseguibile, che devono essere disponibili da qualsiasi parte del programma.
- **.rsrc** —> include le risorse utilizzate dall'eseguibile come icone, immagini, menu e stringhe che non sono parte dell'eseguibile stesso.

Questa sezione contiene “Resource directory : 0000C000”.






4) Le **Librerie** che importa il malware sono 2 rispettivamente:

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




- **KERNEL32.dll** —> contiene le funzioni principali per interagire con il sistema operativo. Se questa libreria viene manipolata da un malware, potrebbe essere utilizzata per eseguire azioni dannose come la creazione di file dannosi, la modifica del registro di sistema, la creazione di processi maligni.
- **ADVAPI32.dll** —> è una libreria che contiene le funzioni utilizzate per la gestione della sicurezza, la manipolazione del registro di sistema e l'interazione con i servizi Windows. Fornisce un'interfaccia per svariate operazioni di basso livello.

Un **Dropper** o un **Caricatore di Payload**, potrebbe utilizzare le funzioni di **KERNEL32.dll** per caricare dinamicamente librerie contenenti il payload dannoso, oppure utilizzare le funzioni di **ADVAPI32.dll** per configurare l'avvio automatico del malware attraverso la modifica del Registro di Sistema.

Le funzioni richiamate all'interno delle librerie sono:

 004070B4	GetProcAddress	KERNEL32	—> per ottenere l'indirizzo di una funzione all'interno di una libreria già caricata.
 004070B8	LoadLibraryA	KERNEL32	—> carica dinamicamente le librerie
 00407018	VirtualAlloc	KERNEL32	—> per allocare memoria nel processo corrente o in un altro processo.
 00407000	RegSetValueExA	ADVAPI32	—> queste funzioni sono utilizzate per modificare o creare voce nel Registro di sistema. Un malware può sfruttarle per ottenere persistenza, nascondere la sua presenza p configurare l'avvio automatico.
 00407004	RegCreateKeyExA	ADVAPI32	

Un **Ransomware**, potrebbe utilizzare le funzioni di KERNEL32.dll per crittografare file utilizzando “CreatFileA”, “WriteFile” e “ReadFile”. Potrebbe utilizzare “RegSetValueExA” per visualizzare il messaggio di richiesta di riscatto nel Registro di sistema.

 00407044	WriteFile
 004070C0	ReadFile
 004070A4	CreateFileA

Un **Dropper** è un programma malevolo che contiene al suo interno un malware. Quando viene eseguito inizia ad estrarre il malware che contiene per salvarlo sul disco. Il malware è contenuto nella sezione **risorse** ( **.rss**) dell'eseguibile.  
Inoltre ha delle API caratteristiche che ritroviamo anche qui e sono:

0040700C	SizeofResource	KERNEL32
00407010	LockResource	KERNEL32
00407014	LoadResource	KERNEL32
00407024	FreeResource	KERNEL32

\*\*\*\*\*

## **MALWARE ANALYSIS:**

Con riferimento Malware in analisi, spiegare:

### **1. Lo scopo della funzione chiamata alla locazione di memoria 00401021.**

```
.text:00401021 call ds:RegCreateKeyExA
```

È utilizzata per creare o aprire una chiave nel Registro di sistema.

Quindi può essere sfruttata per creare le proprie chiavi di registro e memorizzare informazioni importanti o per configurare l'avvio automatico dell'applicazione al momento dell'avvio del sistema operativo.

### **2. Come vengono passati i parametri alla funzione alla locazione 00401021.**

```
.text:00401004 push 0 ; lpdwDisposition
.text:00401006 lea eax, [ebp+hObject] ; phkResult
.text:00401009 push eax ; lpSecurityAttributes
.text:0040100A push 0 ; samDesired
.text:0040100C push 0F003Fh ; dwOptions
.text:00401011 push 0 ; lpClass
.text:00401013 push 0 ; Reserved
.text:00401015 push offset SubKey ; "SOFTWARE\\Microsoft\\Windows NT\\CurrentVe".
.text:00401017 push 80000002h ; hKey
.text:00401021 call ds:RegCreateKeyExA
```

I parametri (evidenziati in figura) vengono passati tramite **"push"**.

Dove:

- **lpdwDisposition**: puntatore a una variabile per il risultato. Utilizzato per indicare se la chiave è stata appena creata o se sia stata già aperta perché esistente.
- **phkResult**: puntatore a un handle alla chiave creata o aperta.
- **lpSecurityAttributes**: può essere utilizzato per specificare gli attributi di sicurezza della nuova chiave.
- **samDesired**: specifica i diritti di accesso.
- **dwOptions**: opzioni di creazione o apertura della chiave.
- **lpClass**: può essere utilizzato per specificare la classe della chiave.
- **Reserved**: parametro riservato impostato di solito su 0.
- **hKey**: è la chiave padre sotto la quale verrà creata o aperta la nuova chiave.

### **3. Che oggetto rappresenta il parametro alla locazione 00401017.**

```
.text:00401017 push offset SubKey ; "SOFTWARE\\Microsoft\\Windows NT\\CurrentVe"...
```

**SubKey** rappresenta la stringa del percorso della chiave di registro.

Che è indicato nel percorso relativo alla chiave di registro

"HKEY\_LOCAL\_MACHINE\\Software\\Microsoft\\Windows NT\\CurrentVersion".

Quindi quando la funzione “RegCreateKeyExA” viene chiamata, utilizzerà questa stringa per creare o aprire la chiave specificata nel Registro di sistema Windows.

La chiave sarà situata sotto la chiave di base **HKEY\_LOCAL\_MACHINE** e avrà il percorso “Software\Microsoft\Windows NT\CurrentVersion”.

#### 4. Il significato delle istruzioni comprese tra gli indirizzi 00401027 e 00401029.

```
.text:00401021      call     ds:RegCreateKeyExA
.text:00401027      test     eax, eax
.text:00401029      jz       short loc_401032
.text:0040102B      mov      eax, 1
```

test eax, eax = confronta 2 operandi effettuando un **AND** logico tra di essi. Entrambi gli operandi sono nel registro “eax”. Quindi si sentano le flag del processore in base al risultato dell’AND. Questa istruzione viene utilizzata per verificare se un registro è zero o meno.

jz short loc\_401032 = istruzione “jz” (Jump if Zero) esegue un salto condizionale se la flag Zero è impostata, in questo modo il risultato è 0. Indicando l’indirizzo di destinazione del salto. Quindi se ZF è impostato, l’esecuzione del programma salterà all’indirizzo specificato, altrimenti continuerà con l’istruzione successiva.

Riassumendo queste istruzioni stanno effettuando una verifica condizionale per determinare se il valore contenuto nel registro “**eax**” è **zero**. Se fosse così allora il salto condizionale porterà esecuzione del programma all’indirizzo specificato (loc\_401032). Se “**eax**” *non fosse zero*, l’esecuzione proseguirà normalmente con l’istruzione successiva.

```
.text:00401029      jz       short loc_401032
.text:0040102B      mov      eax, 1
.text:00401030      jmp      short loc_40107B
.text:00401032      ; -----
.text:00401032      ;
.text:00401032 loc_401032:      ; CODE XREF: sub_401000+29↑j
.text:00401032      mov      ecx, [ebp+cbData]
.text:00401035      push     ecx      ; cbData
```

#### 5. Con riferimento all’ultimo quesito, tradurre il codice Assembly nel corrispondente costruito C.

```
int main ()
{
    int a;          // valore di eax
    int c;          // valore di ecx
    int d;          // valore di [ebp+cdData]

    if (a == 0) {
        c = d;
    } else {
        a = 1;
    }

    return 0;
}
```

6. Valutare la chiamata alla locazione 00401047, qual è il valore del parametro “ValueName”?

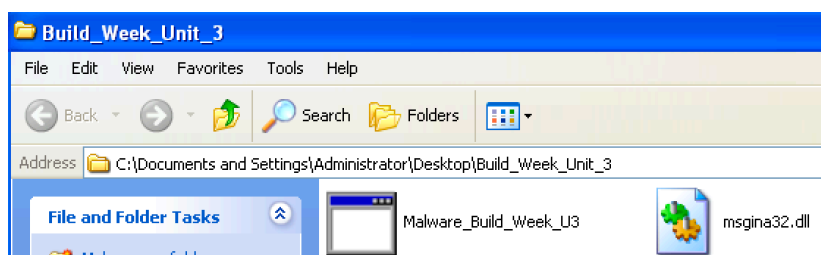
```
.text:00401032 loc_401032: ; CODE XREF: s
.text:00401032 mov     ecx, [ebp+cbData]
.text:00401035 push    ecx ; cbData
.text:00401036 mov     edx, [ebp+lpData]
.text:00401039 push    edx ; lpData
.text:0040103A push    1 ; dwType
.text:0040103C push    0 ; Reserved
.text:0040103E push    offset ValueName ; "GinaDLL"
.text:00401043 mov     eax, [ebp+hObject]
.text:00401046 push    eax ; hKey
.text:00401047 call    ds:RegSetValueExA
.text:0040104D test     eax, eax
```

RegSetValueExA è una funzione nella libreria **ADVAPI32.dll** ed è utilizzata per impostare il valore di una chiave nel Registro di sistema.

Il valore corrisponde a “GinaDLL” ( un’interfaccia personalizzabile che gestisce l’interazione tra utente e sistema operativo durante l’autenticazione).

### **Analisi Dinamica:**

- 1) Eseguire il Malware con Process Monitor, e spiegare cosa si è notato all’interno della cartella dove è situato l’eleggibile del Malware, unendo le evidenze che sono state raccolte fino ad ora.



Avviando il *Malware* si nota la creazione del file **msgina32.dll**, questo può essere segno di un tentativo di manipolare o sostituire la libreria di autenticazione Windows con una versione compromessa o malevola, appunto di “**msgina32.dll**”.

Il *Malware* potrebbe aver creato o sovrascritto il file “**msgina32.dll**” nel tentativo di sostituire la libreria di autenticazione di Windows.

Considerando la presenza del valore di chiave di registro “**GinaDLL**” il *Malware* potrebbe aver cercato di modificare la configurazione dell’interfaccia di autenticazione. Potrebbe aver cercato di specificare una *DLL* personalizzata o dannosa come componente *GINA*.

L’importazione delle librerie **KERNEL32.dll** e **ADVAPI32.dll**, insieme alle funzioni sopra citate nell’analisi, fa pensare che il *Malware* ha eseguito operazioni avanzate, incluse la manipolazione del registro di sistema, la creazione o sovrascrittura di file e il caricamento dinamico di librerie. Quindi si potrebbe dire che il *Malware* cerca di ottenere un controllo più profondo e persistente sul sistema. Tentando di manipolare il sistema operativo, in particolare nella gestione dell’interfaccia di autenticazione Windows.

## 2. Analizzare i risultati di Process Monitor.

Dall'analisi dei risultati si notano le operazioni, elencate sotto, riferite al percorso che abbiamo individuato in precedenza (Software\Microsoft\Windows NT\CurrentVersion) e il file (msgina32.dll) individuato in precedenza all'avvio del malware:

- RegOpenKey → Apre una chiave di Registro esistente. Viene utilizzata per ottenere l'accesso a una chiave di registro specifica, in modo da poter leggere o modificare i valori.
- RegQueryValue → Si ottiene il Valore di una chiave di registro specifica. Quindi anche la lettura la lettura di informazioni, di configurazioni o di percorsi di file.
- RegCloseKey → utilizzata per chiudere una chiave di registro aperta, per garantire la corretta gestione delle risorse.
- CreateFile → Crea o Apre un file.
- WriteFile → Scrive dati in un file →

I Malware utilizzano queste funzioni appunto per Creare, Aprire o Scrivere in file, inclusi quelli che possono essere usati per la loro esecuzione o per la persistenza del sistema.

2012	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\Malware_Build_Week_U3.exe	NAME N...	Desired Access: Read
2012	CreateFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3	SUCCESS	Desired Access: Execute/Traverse, Synchron...
2012	FileSystemControl	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3	SUCCESS	Control: FSCTL_IS_VOLUME_MOUNTED
2012	QueryOpen	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\Malware_Build_Week_U3.exe.Local	NAME N...	
2012	RegOpenKey	HKLM\System\CurrentControlSet\Control\Terminal Server	SUCCESS	Desired Access: Read
2012	RegQueryValue	HKLM\System\CurrentControlSet\Control\Terminal Server\TSAppCompat	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
2012	RegCloseKey	HKLM\System\CurrentControlSet\Control\Terminal Server	SUCCESS	
2012	RegOpenKey	HKLM\System\CurrentControlSet\Control\Terminal Server	SUCCESS	Desired Access: Read
2012	RegQueryValue	HKLM\System\CurrentControlSet\Control\Terminal Server\TSAppCompat	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
2012	RegCloseKey	HKLM\System\CurrentControlSet\Control\Terminal Server	SUCCESS	
2012	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\Secur32.dll	NAME N...	Desired Access: Read
2012	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\RPCRT4.dll	NAME N...	Desired Access: Read
2012	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\ADVAPI32.dll	NAME N...	Desired Access: Read
2012	RegOpenKey	HKLM\System\CurrentControlSet\Control\Terminal Server	SUCCESS	Desired Access: Read
2012	RegQueryValue	HKLM\System\CurrentControlSet\Control\Terminal Server\TSAppCompat	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
2012	RegQueryValue	HKLM\System\CurrentControlSet\Control\Terminal Server\TSUserEnabled	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
2012	RegCloseKey	HKLM\System\CurrentControlSet\Control\Terminal Server	SUCCESS	
2012	RegOpenKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS	Desired Access: Read
2012	RegQueryValue	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\LeakTrack	NAME N...	Length: 144
2012	RegCloseKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS	
2012	RegOpenKey	HKLM	SUCCESS	Desired Access: Maximum Allowed
2012	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Diagnostics	NAME N...	Desired Access: Read
2012	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\ntdll.dll	NAME N...	Desired Access: Read
2012	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\kernel32.dll	NAME N...	Desired Access: Read
2012	CreateFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\msgina32.dll	SUCCESS	Desired Access: Generic Write, Read Attrib...
2012	CreateFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3	SUCCESS	Desired Access: Synchronize, Disposition: C...
2012	CloseFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3	SUCCESS	
2012	IRP_MJ_CLOSE	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3	SUCCESS	
2012	WriteFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\msgina32.dll	SUCCESS	Offset: 0, Length: 4,096
2012	WriteFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\msgina32.dll	FAST IO ...	Offset: 4,096, Length: 2,560
2012	WriteFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\msgina32.dll	SUCCESS	Offset: 4,096, Length: 2,560
2012	CloseFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\msgina32.dll	SUCCESS	
2012	IRP_MJ_CLOSE	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\msgina32.dll	SUCCESS	
2012	RegCreateKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS	Desired Access: All Access
2012	RegSetValue	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\GinaDLL	SUCCESS	Type: REG_SZ, Length: 520, Data: C:\Doc...
2012	SetEndOfFileInformationFile	C:\WINDOWS\system32\config\software.LOG	SUCCESS	EndOfFile: 8,192
2012	SetEndOfFileInformationFile	C:\WINDOWS\system32\config\software.LOG	SUCCESS	EndOfFile: 8,192
2012	SetEndOfFileInformationFile	C:\WINDOWS\system32\config\software.LOG	SUCCESS	EndOfFile: 16,384
2012	SetEndOfFileInformationFile	C:\WINDOWS\system32\config\software.LOG	SUCCESS	EndOfFile: 20,480
2012	RegCloseKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS	
2012	CloseFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3	SUCCESS	
2012	IRP_MJ_CLOSE	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3	SUCCESS	

## 3. Filtrare includendo solamente l'attività sul registro di Windows.

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

RegCreateKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS	Desired Access: All Access
RegSetValue	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\GinaDLL	SUCCESS	Type: REG_SZ, Length: 520, [

Viene creata la chiave di registro:

"HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon" utilizzando RegCreateKey.

Che può essere utilizzata per impostare configurazioni relative all'accesso e all'avvio del sistema.

Infatti nei dettagli vediamo che è stato richiesto l'Accesso Completo.

Viene poi impostato un valore nella chiave di registro "HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\GinaDLL" utilizzando RegSetValue.

Passando alla visualizzazione dell'attività sul file system.

1) Quale chiamata di sistema ha modificato il contenuto della cartella dove è presente l'eseguibile del malware?

La chiamata che ha modificato la cartella dell'eseguibile del malware è "CreateFile" evidenziata sotto.

2012	CreateFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\msgina32.dll	SUCCESS
2012	CreateFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3	SUCCESS
2012	CloseFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3	SUCCESS
2012	IRP_MJ_CLOSE	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3	SUCCESS
2012	WriteFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\msgina32.dll	SUCCESS
2012	WriteFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\msgina32.dll	FAST IO ...
2012	WriteFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\msgina32.dll	SUCCESS
2012	CloseFile	C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\msgina32.dll	SUCCESS

Il primo "CreateFile"  
Crea il contenuto —>

Date: 11/23/2023 3:48:47.0086310 PM  
Thread: 240  
Class: File System  
Operation: CreateFile  
Result: SUCCESS  
Path: C:\Documents and Settings\Administrator\Desktop\Build\_Week\_Unit\_3\msgina32.dll  
Duration: 0.0200557

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

Il secondo "CreateFile"  
Apri il contenuto — >

Date: 11/23/2023 3:48:47.0124785 PM  
Thread: 240  
Class: File System  
Operation: CreateFile  
Result: SUCCESS  
Path: C:\Documents and Settings\Administrator\Desktop\Build\_Week\_Unit\_3  
Duration: 0.0004506

Desired Access: Synchronize  
Disposition: Open  
Options: Directory, Synchronous IO Non-Alert, Open For Backup  
Attributes: N  
ShareMode: Read, Write  
AllocationSize: n/a  
OpenResult: Opened

E poi con i "WriteFile" scrive con successo.

Unire tutte le informazioni raccolte per delineare il funzionamento del Malware.

Il comportamento del Malware suggerisce che stia cercando di influire sull'autenticazione di Windows, attraverso la creazione del file "msgina32.dll" in modo da sostituire la libreria di autenticazione e manipolare la chiave di registro "GinaDLL", quindi intercettare le credenziali di accesso o eseguire azioni dannose durante il processo di autenticazione.

Creando la chiave di registro "GinaDLL" il Malware cerca di ottenere persistenza, in modo che venga eseguito automaticamente durante il processo di avvio del sistema.

In quanto **GINA** (*Graphical Identification and Authentication*) è un componente di Windows che gestisce l'interfaccia di autenticazione.