

# **MALWARE ANALYSIS**

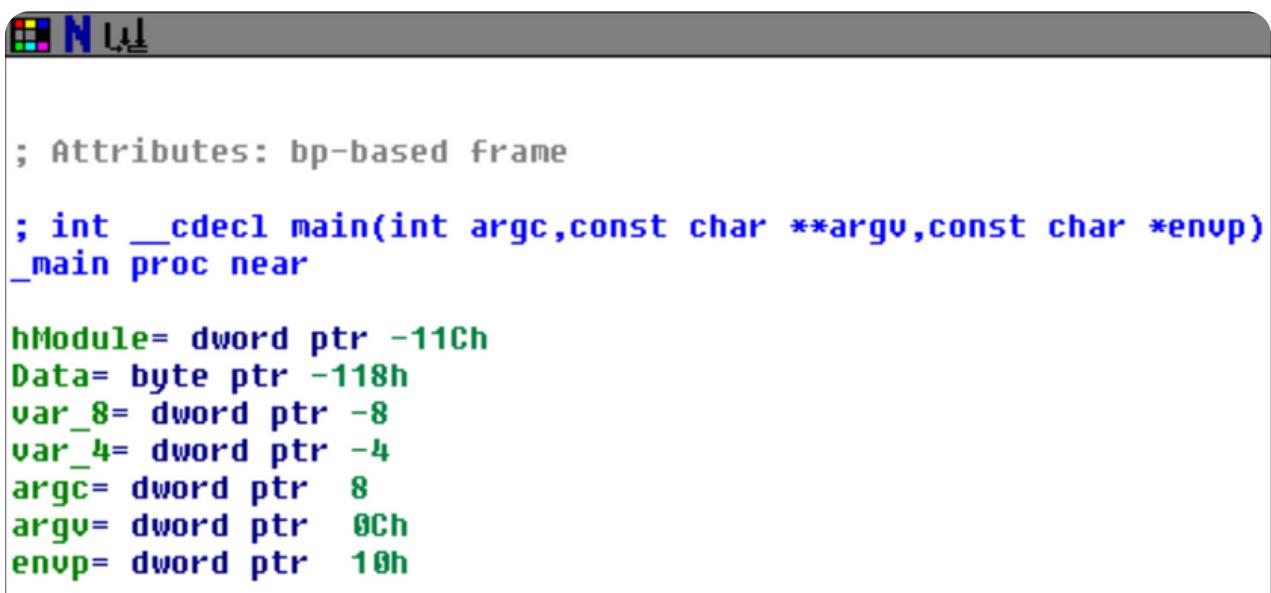
## **BUILD WEEK III**

# GIORNO 1

Per analizzare il malware in oggetto abbiamo utilizzato **IDA Pro**, un software disassembler che analizza un malware e restituisce in output il suo codice in linguaggio assembly (nel nostro caso assembly dell'architettura X86).

Il codice disassemblato verrà rappresentato da una barra con diversi colori, quello che a noi interessa è il colore blu poiché rappresenta il codice del malware.

Una volta che IDA avrà terminato le sue operazioni, sullo schermo comparirà l'entry point dell'applicazione, ovvero il punto da cui il programma comincia ad eseguire il suo codice. E' possibile vederlo grazie allo screenshot sottostante:



```
; 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
```

La **funzione main** quindi rappresenta l'**entry point** e grazie ad una veloce analisi possiamo individuare quali sono i parametri della funzione nonché le variabili dichiarate al suo interno (sempre visibili nello screenshot), di seguito le abbiamo elencate:

## Parametri

Sulla base dei dati forniti dalla traccia dell'esercizio, è stato possibile rilevare tre parametri: **argc**; **argv**; **envp**.

## Variabili

Le variabili riscontrate sono quattro e sono individuabili tramite l'offset negativo: **hModule**; **Data**; **var\_8**; **var\_4**.

Malware_Build_Week_U3.exe						
Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
000076D0	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

## Librerie

Le librerie visualizzate tramite il tool CFF Explorer sono due: **KERNEL32.dll** e **ADVAPI32.dll**

- **KERNEL32.dll**: questa libreria contiene le funzioni principali per interagire con il sistema operativo, ad esempio gestione della memoria e manipolazione file.
- **ADVAPI32.dll**: contiene le funzioni per interagire con i registri del sistema operativo.

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

Tra le funzioni utilizzate dalle librerie quelle che vale la pena evidenziare per **KERNEL32.dll** sono quelle presenti nello screenshot sopra, si potrebbe ipotizzare che il malware sia un **dropper** che carica una risorsa malevola una volta sul pc target. Per la libreria **ADVAPI32.dll** invece abbiamo due funzioni (mostrate sotto) che permettono di creare e settare una chiave del registro, molto probabilmente per ottenere la **persistenza** necessaria.

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

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

## Sezioni

Le sezioni visualizzate tramite CFF Explorer sono quattro: **.text**; **.rdata**; **.data**; **.rsrc**.

- **.text**: la sezione in questione contiene il codice eseguibile dal programma, ovvero le istruzioni in Assembly che vengono eseguite dal processore nel momento in cui il programma è in esecuzione.
- **.rdata**: questa sezione contiene dati di sola lettura (read only data) i quali rimangono costanti e non vengono modificati durante l'esecuzione del programma.
- **.data**: a differenza della sezione precedente, qui i dati possono essere modificati durante l'esecuzione. Spesso sono incluse variabili globali e altri valori che possono essere cambiati dinamicamente.
- **.rsrc**: quest'ultima sezione contiene immagini, icone, stringhe e altri dati riconducibili all'interfaccia dell'utente e alle funzionalità del programma.

# GIORNO 2

```
    .text:00401009      push    eax          ; phkResult
    .text:0040100A      push    0             ; lpSecurityAttributes
    .text:0040100C      push    0F 003Fh   ; sanDesired
    .text:00401011      push    0             ; dwOptions
    .text:00401013      push    0             ; lpClass
    .text:00401015      push    0             ; Reserved
    .text:00401017      push    offset SubKey  ; "SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon"
    .text:0040101C      push    80000002h  ; hKey
    .text:00401021      call    ds:RegCreateKeyExA
    .text:00401027      test   eax, eax
    .text:00401029      jz     short loc_401032
    .text:00401028      mov    eax, 1
    .text:00401030      jmp    short loc_40107B
.loc_401032 ; -----
    .text:00401032      mov    ecx, [ebp+cbData] ; CODE XREF: sub_401000+29↑j
    .text:00401032      mov    edx, [ebp+lpData]
    .text:00401035      push   ecx          ; cbData
    .text:00401036      push   edx          ; lpData
    .text:00401039      push   1             ; dwType
    .text:0040103A      push   0             ; Reserved
    .text:0040103C
```

Nella locazione di memoria **00401021** è presente la funzione **RegCreateKeyExA**. Essa ha il compito di creare una specifica chiave di registro, se è già presente, viene aperta.

I parametri vengono passati tramite una serie di istruzioni push.

L'oggetto che rappresenta il parametro alla locazione **00401017** è la stringa **"SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon"** che corrisponde al percorso della chiave di registro di sistema che il programma sta cercando di creare utilizzando la funzione "RegCreateKeyExA". Il codice coinvolge la chiamata alla funzione "RegCreateKeyExA" che è una funzione di Windows API utilizzata per creare o aprire una chiave di registro di sistema, e questa stringa rappresenta, dunque, il nome della chiave di registro stesso.

Le istruzioni comprese tra **00401027** e **00401029** fanno riferimento ai codici mnemonici **test** e **jz**.

- **Test:** questa istruzione è utilizzata per effettuare un'operazione logica tra due operandi, tramite un confronto bit a bit. Il registro EAX viene confrontato con se stesso tramite un'operazione di AND. Il risultato non viene memorizzato ma gli effetti dell'operazioni sono riscontrabili su eventuali EFLAGS. La ZF verrà impostata ad 1 se il risultato della comparazione restituirà zero, in caso contrario la ZF avrà valore 0 se il risultato della comparazione sarà diverso da zero.
- **jz:** l'istruzione jz fa parte della famiglia dei jmp, in questo caso assume il valore di un jump condizionale. La condizione necessaria per attuare il salto è rappresentata dal valore della ZF. "Jump if Zero", il salto verrà effettuato se la ZF restituirà 1 nella comparazione precedente.

Le istruzioni **00401027** e **00401029** possono essere tradotte in linguaggio C in un modo simile a questo:

```
if (eax==0) {  
    //salto alla locazione 401032  
} else {  
    //proseguimento del codice  
}
```

```
00401027 test eax, eax  
00401029 jz short loc_401032  
0040102B mov eax, 1  
0040102D jmp short loc_40107B  
00401032 ;  
00401032 loc_401032:  
00401032 mov ecx, [ebp+cbData] ; cbData  
00401035 push ecx  
00401036 mov edx, [ebp+lpData] ; lpData  
00401039 push edx  
0040103A push 1 ; dwType  
0040103C push 0 ; Reserved  
0040103E push offset ValueName ; "GinaDLL"  
00401043 mov eax, [ebp+hObject]  
00401046 push eax ; hKey  
00401047 call ds:RegSetValueExA  
0040104D test eax, eax  
0040104F jz short loc_401062  
00401051 mov ecx, [ebp+hObject]  
00401054 push ecx ; hObject  
00401055 call ds:CloseHandle
```

La chiamata alla locazione **00401047** è una chiamata alla funzione **RegSetValueExA**. Il valore del parametro ValueName è **GinaDLL**.

```
0040103E push offset ValueName ; "GinaDLL"  
00401043 mov eax, [ebp+hObject]  
00401046 push eax ; hKey  
00401047 call ds:RegSetValueExA  
0040104D test eax, eax  
0040104F jz short loc_401062  
00401051 mov ecx, [ebp+hObject]  
00401054 push ecx ; hObject  
00401055 call ds:CloseHandle
```

In questo estratto del codice il malware sta cercando di inserire un valore associato ad una chiave di registro di sistema che è comunemente utilizzato per configurare il caricamento di DLL. Poichè GINA (Graphical Identification and Authentication) gestisce l'interfaccia di autenticazione, **probabilmente il malware sta utilizzando il valore "GinaDLL" per manipolare il comportamento del sistema operativo in maniera da influenzare il processo di autenticazione.**

# GIORNO 3

Abbiamo analizzato la subroutine compresa tra gli indirizzi di memoria **00401080** e **0401128**, in particolare la funzione **FindResourceA()**. Uno dei parametri passati a questa funzione è **lpName** (ResourceName nella traccia dell'esercizio), e utilizzando OllyDB possiamo determinarne il valore che corrisponde a "**TGAD**". Di seguito è possibile vedere uno screenshot con evidenziato il valore:



Il malware implementa delle funzionalità appartenenti alla famiglia dei dropper. Quest'ultimo è un malware che contiene all'interno un file malevolo da estrarre per essere eseguito.

Le funzioni che ci fanno capire che si tratti di un dropper sono:

- **FindResource();**
- **LoadResource();**
- **LockResource();**
- **SizeOfResource();**

Queste particolari funzioni permettono di estrarre il file malevolo.

The screenshot displays the assembly code for four related functions:

- FindResourceA()**: Located at **00401080**, this function takes parameters **lpName** (resource name) and **hModule** (handle to module). It calls **FindResource()** and returns the handle to the resource.
- LoadResource()**: Located at **004010DF**, this function takes parameters **hResInfo** (resource info) and **hModule** (handle to module). It calls **LoadResource()** and returns the handle to the resource data.
- LockResource()**: Located at **004010F0**, this function takes parameters **hResData** (resource data handle) and **dwDesiredAccess** (desired access flags). It calls **LockResource()**.
- SizeOfResource()**: Located at **004010F8**, this function takes parameters **hResInfo** (resource info) and **hModule** (handle to module). It calls **SizeOfResource()** and returns the size of the resource data.

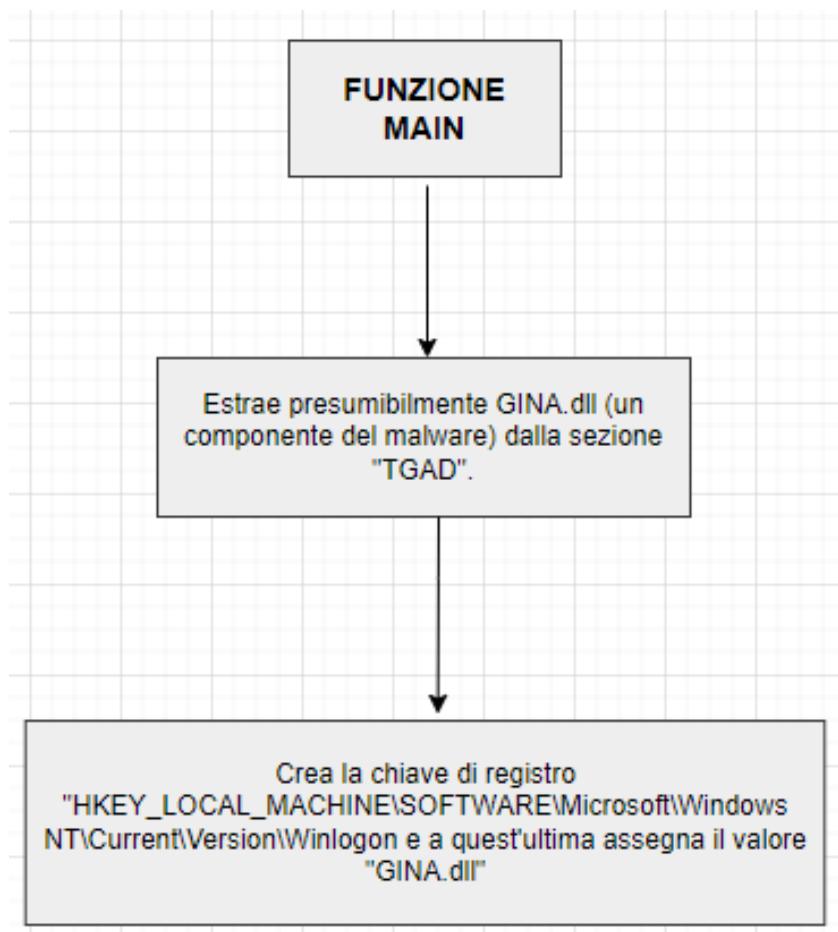
Questa funzionalità è possibile identificarla utilizzando l'analisi statica basica perché tramite il tool "CFF Explorer" si possono vedere le funzioni usate dal dropper nella libreria **Kernel32.dll**.

Module Name	Imports	OFTs	TimeDateStamp	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

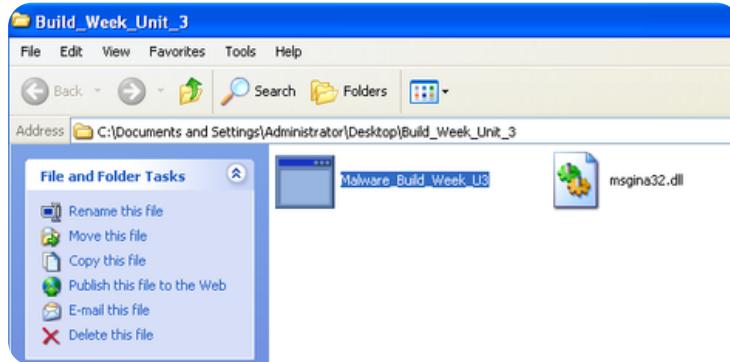
OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
00007632	00007632	0295	SizeofResource
00007644	00007644	01D5	LockResource
00007654	00007654	01C7	LoadResource
00007622	00007622	02B8	VirtualAlloc
00007674	00007674	0124	GetModuleFileNameA
0000768A	0000768A	0126	GetModuleHandleA
00007612	00007612	00B6	FreeResource
00007664	00007664	00A3	FindResourceA
00007604	00007604	0018	CloseHandle
000076DE	000076DE	00CA	GetCommandLineA
000076F0	000076F0	0174	GetVersion
000076FE	000076FE	007D	ExitProcess

Di seguito le quattro funzioni rappresentate da un diagramma di flusso:



# GIORNO 4

Una volta eseguito il malware, ritroveremo sulla stessa cartella dell'eseguibile un file dll nominato **msgina32.dll**. Il file creato potrebbe corrispondere ad un rifacimento malevolo della libreria **GINA.dll**, componente lecito di Windows.



La chiave di registro creata è:

**HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon.**

Il valore associato alla chiave di registro è **GinaDLL**.

```
237:13.11651... [Malware_Build_Week_U3....] 1452 [RegCreateKey  
237:13.11659... [Malware_Build_Week_U3....] 1452 [RegSetValue  
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon  
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\GinaDLL
```

Le chiamate alla funzione **CreateFile** sono le responsabili della creazione del file **msgina32.dll** nella cartella Malware\_Build\_Week\_U3.

```
237:13.10783... [Malware_Build_Week_U3....] 1452 [CreateFile  
37:13.10822... [Malware_Build_Week_U3....] 1452 [Createfile  
37:13.10982... [Malware_Build_Week_U3....] 1452 [CloseFile  
37:13.11225... [Malware_Build_Week_U3....] 1452 [ReadFile  
37:13.11403... [Malware_Build_Week_U3....] 1452 [WriteFile  
37:13.11523... [Malware_Build_Week_U3....] 1452 [WriteFile  
37:13.11576... [Malware_Build_Week_U3....] 1452 [CloseFile  
C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\msgina32.dll  
C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3  
C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3  
C:  
C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\msgina32.dll  
C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\msgina32.dll  
C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\msgina32.dll
```

Tramite le esecuzioni delle tecniche di analisi statica e dinamica, abbiamo delineato il funzionamento del Malware.

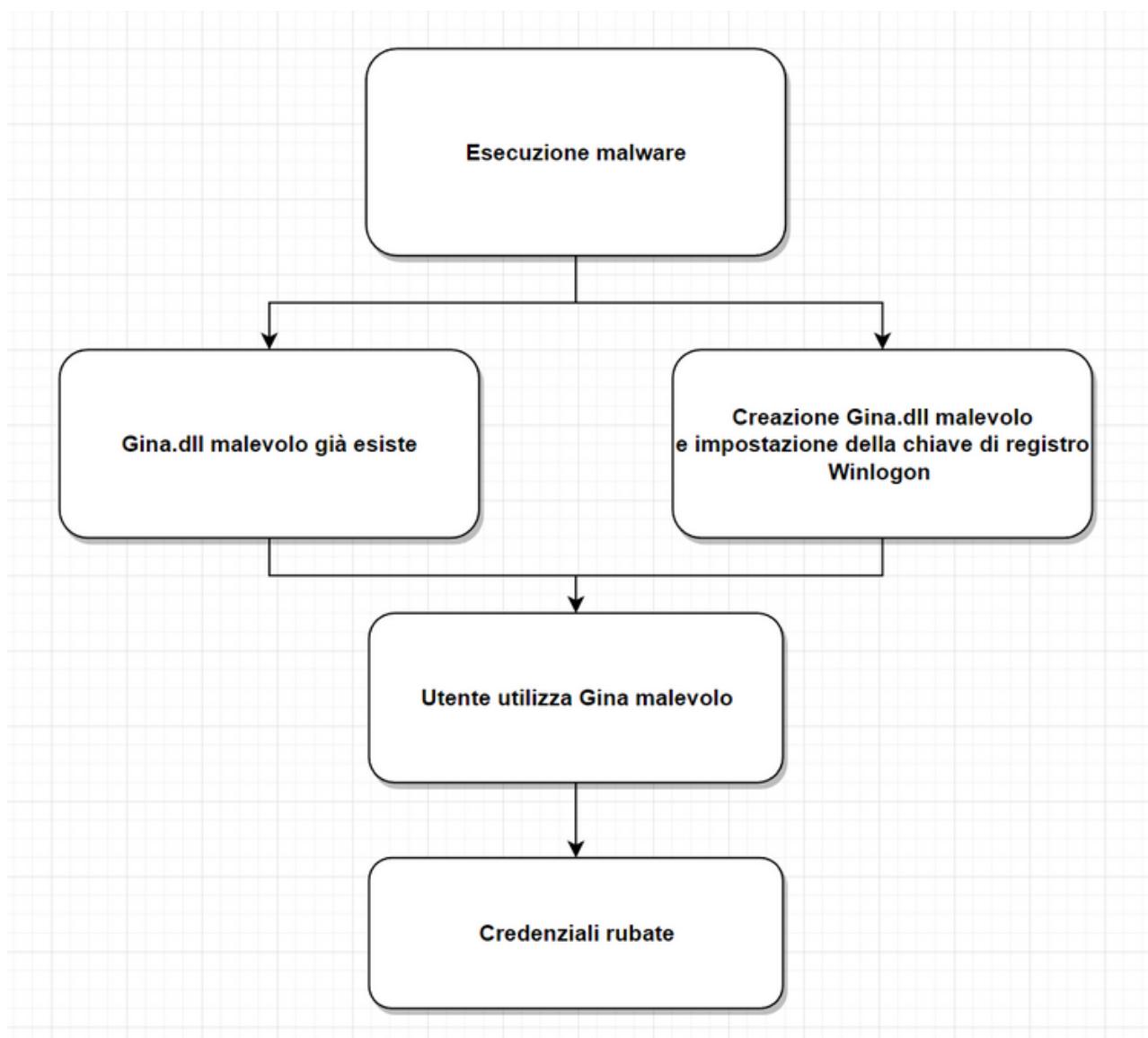
A seguito dell'installazione del file sulla macchina vittima e la conseguente esecuzione da parte dell'utente, il Malware opererà a livello software per creare una chiave di registro e settare quel valore come **GINA.DLL**. Quest'ultimo verrà sostituito al componente **GINA** originale.

# Giorno 5

**Cosa può succedere se il file .dll lecito viene sostituito con un file .dllmalevolo, che intercetta i dati inseriti?**

Rimpiazzare un file **dll lecito** con un file **dll malevolo** che intercetti i dati inseriti, avrebbe come conseguenza la possibilità, da parte dell'attaccante, di ricevere i dati per il login degli utenti. Ciò che è stato analizzato nel punto 2 si ricollega a questa funzione. Sostituendo la componente lecita GINA con un componente GINA fittizio, è possibile entrare in possesso dei dati di login.

## Grafico del Malware



# BONUS 1

Il presente report è stato creato con l'obiettivo di analizzare due tipi di malware rilevati:

- **Agent Tesla**
- **Rhadamanthys**

Allo stesso tempo, è essenziale descrivere le criticità riscontrate da tali malware e portare a conoscenza l'azienda circa i rischi e gli eventuali danni che essi possono provocare al fine di evitare una compromissione dei dispositivi presenti nell'azienda stessa e/o dei sistemi ad essi associati.

L'analisi è stata effettuata con Anyrun, uno strumento ampiamente utilizzato nel campo della sicurezza informatica che ci consente di rilevare, analizzare, monitorare e ricercare le minacce informatiche in tempo reale per ottenere informazioni dettagliate sul comportamento di potenziali rischi a livello informatico.

## Agent Tesla

### Cos'è Agent Tesla?

Questo malware è un **keylogger**, cioè un software malevolo che ruba i dati, come password/username, digitati direttamente dalla tastiera. Oltre a rubare i dati dalla tastiera può anche effettuare screenshot e registrare video dello schermo del PC.

### Come viene distribuito?

**Bisogna fare attenzione quando si controlla la posta elettronica** perché questo virus viene distribuito tramite una campagna di spam, appena si clicca il link all'interno dell'email di spam parte un download automatico.

## Problematiche del Malware

Pure essendo un malware creato nel 2014 rimane tuttora un grande problema nello sfruttare le vulnerabilità di Windows a causa delle sue continue evoluzioni

## Rhadamanthys

### Cos'è Rhadamanthys?

Rhadamanthys è un malware che prende di mira i sistemi Windows e svolge la funzione di **Advanced Info-Stealer**, ovvero mira a rubare i dati degli utenti. Può essere usato per estrarre password dell'utente ed estrapolare dati sensibili. Inoltre, risulta pericoloso per le piattaforme di criptovalute, utilizzato per rubare credenziali e wallet.

## Come viene distribuito?

La sua distribuzione è basata sul dirottare annunci pubblicitari di Google, rimpiazzando il link originale con un link malevolo. Utilizza inoltre, tecniche di spam, come la creazione di un finto documento Adobe Acrobat, e tecniche di phishing, andando a creare un sito malevolo che ricalchi altri siti legittimi.

## Problematiche del Malware

Essendo uno Stealer, il Malware cerca di operare nel modo più silenzioso possibile, cercando di evitare la rilevazione.

## Mitigazione

Di fatto nessuna misura di sicurezza rende in maniera assoluta invulnerabili i sistemi, tuttavia la combinazione di alcune pratiche può ridurre significativamente il rischio di contrarre malware.

Per cercare di evitare che questi infettino un PC è consigliabile implementare:

- un antivirus e aggiornarlo spesso
- aggiornare i sistemi operativi
- eseguire backup regolarmente
- utilizzare un firewall per monitorare e controllare costantemente il traffico di rete
- soluzioni di sicurezza per le email, come filtri anti-spam e anti-phishing, per ridurre il rischio di ricevere email malevole

# BONUS 2

## Calcolatriceinnovativa.exe

Per confermare che il file scaricato sia un Malware abbiamo utilizzato diversi strumenti. Il primo programma utilizzato è CFF Explorer, dopo una prima panoramica dell'eseguibile siamo andati ad analizzare le librerie del file. Si può notare la presenza di librerie che possono essere trovate comunemente all'interno di un Malware: **KERNEL32.dll; ADVAPI32.dll; SHELL32.dll e msrvct.dll**. Non è possibile stabilire la funzione del Malware solamente da questi aspetti ma è possibile iniziare a farsi un'idea sulle sue funzioni.

Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
000125D4	N/A	00011FBC	00011FC0	00011FC4	00011FC8	00011FCC
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
SHELL32.dll	1	00012CA8	FFFFFFFFF	FFFFFFFFF	00012E42	0000109C
msvcrct.dll	26	00012DC8	FFFFFFFFF	FFFFFFFFF	00012F60	000011BC
ADVAPI32.dll	3	00012C0C	FFFFFFFFF	FFFFFFFFF	00012FFC	00001000
KERNEL32.dll	30	00012C2C	FFFFFFFFF	FFFFFFFFF	000131D4	00001020
GDI32.dll	3	00012C1C	FFFFFFFFF	FFFFFFFFF	0001320C	00001010
USER32.dll	69	00012CB0	FFFFFFFFF	FFFFFFFFF	000136A4	000010A4

11.46...	calcolatriceinnovo...	2332	QueryOpen	C:\WINDOWS\WindowsShellManifest	SUCCESS	CreationTime: 15/07/2022 14.05.53, LastAc...
11.46...	calcolatriceinnovo...	2332	CreateFile	C:\WINDOWS\WindowsShellManifest	SUCCESS	DesiredAccess: Execute/Traverse, Synchron...
11.46...	calcolatriceinnovo...	2332	CreateFileMapp...	C:\WINDOWS\WindowsShellManifest	SUCCESS	SyncType: SyncTypeCreateSection, PagePi...
11.46...	calcolatriceinnovo...	2332	QueryStandardI...	C:\WINDOWS\WindowsShellManifest	SUCCESS	AllocationSize: 4,096, EndOfFile: 749, Numb...
11.46...	calcolatriceinnovo...	2332	CreateFileMapp...	C:\WINDOWS\WindowsShellManifest	SUCCESS	SyncType: SyncTypeOther
11.46...	calcolatriceinnovo...	2332	CloseFile	C:\WINDOWS\WindowsShellManifest	SUCCESS	
11.46...	calcolatriceinnovo...	2332	QueryOpen	C:\WINDOWS\WindowsShellManifest	SUCCESS	CreationTime: 15/07/2022 14.05.53, LastAc...
11.46...	calcolatriceinnovo...	2332	CreateFile	C:\WINDOWS\WindowsShellManifest	SUCCESS	DesiredAccess: Generic Read, Disposition: ...
11.46...	calcolatriceinnovo...	2332	CreateFileMapp...	C:\WINDOWS\WindowsShellManifest	SUCCESS	SyncType: SyncTypeCreateSection, PagePi...
11.46...	calcolatriceinnovo...	2332	QueryStandardI...	C:\WINDOWS\WindowsShellManifest	SUCCESS	AllocationSize: 4,096, EndOfFile: 749, Numb...
11.46...	calcolatriceinnovo...	2332	CreateFileMapp...	C:\WINDOWS\WindowsShellManifest	SUCCESS	SyncType: SyncTypeOther
11.46...	calcolatriceinnovo...	2332	CloseFile	C:\WINDOWS\WindowsShellManifest	SUCCESS	
11.46...	calcolatriceinnovo...	2332	CreateFile	C:\WINDOWS\WindowsShellManifest	SUCCESS	DesiredAccess: Generic Read/Execute, Dis...
11.46...	calcolatriceinnovo...	2332	CreateFileMapp...	C:\WINDOWS\WindowsShellManifest	SUCCESS	SyncType: SyncTypeCreateSection, PagePi...

Shell.Manifest è un file di manifesto associato a Windows Shell, cioè la parte del sistema operativo Windows responsabile dell'interfaccia utente grafica e di funzionalità di gestione dei file.

Il file di manifesto fornisce istruzioni su come l'applicazione dovrebbe interagire con il sistema operativo e con altre applicazioni.

Per indagare più approfonditamente, come è possibile osservare dagli screen di seguito, abbiamo deciso di utilizzare VirusTotal, per avere una rapida conferma se questo possa essere effettivamente un file malevolo.

Nel primo screenshot possiamo vedere un'attività malevola di tipo Trojan mentre nel secondo screenshot, in cui abbiamo visualizzato il report, viene rilevata una possibilità del 64% che il file possa essere malevolo.

The screenshot shows the VirusTotal analysis interface. At the top, it displays a circular icon with the number '53' and a progress bar. Below this, the file information is shown: file name 'b8ed129eb56c68cec1661206c313c6eab2e20e4b9223336f7edf661c9956e81a', file type 'CALC.EXE', and file size '112.50 KB'. The 'Community Score' is also visible. On the right, there are buttons for 'Reanalyze', 'Similar', and 'More'. Below the file info, there are tabs for 'DETECTION', 'DETAILS', 'RELATIONS', 'BEHAVIOR', and 'COMMUNITY'. The 'DETECTION' tab is selected, showing a message to 'Join the VT Community' and a table of security vendors' analysis. The table includes rows for Alibaba, Anti-AVL, Avast, and Avira (no cloud), each with their respective threat labels and vendor names. A blue button labeled 'Automate Checks' is located on the right side of the table. The 'DETAILS' tab is also visible below the table. The 'File Info' section on the right provides detailed technical information about the file, including its name, type, size, SHA1, MD5, SHA512, entropy, submission path, and preview. The preview shows the first few lines of the executable code.

Dai dati forniti nel report, possiamo ipotizzare che il malware in questione sia una **backdoor**. Possiamo vedere nelle figure qui sotto l'utilizzo del tool "YARA" che riconosce un payload di Metasploit e l'indirizzo IP a cui il malware cerca di connettersi.

### Remote Access Functionality (1)

4.0 Yara detected Metasploit Payload

#### Network Info

IP Info (1)

IP

192.168.1.80 malicious

Country

unknown

#### Fonte:

[#mitre-attack](https://vtbehaviour.commandstorage.googleapis.com/b8ed129eb56c68cec1661206c313c6eab2e20e4b9223336f7edf661c9956e81a_Zenbox.html?GoogleAccessId=758681729565-rc7fgq07icj8c9dm2gi34a4cckv235v1@developer.gserviceaccount.com&Expires=1702635430&Signature=KkoWoH7CjXLQS1h8n19cDjjWFkjm3eYI2SyfqQCv1X4XgW9ZFGV4Gznx2r7AcFDD4Ze1zbk6NZjEj%2BnQJS3ObbC16lpPbFkkbsW4okgUpUoVtJeokoYAFUSo5wYiF6zizXD3S0xeeP1g6AMzqjbTV9CK4%2Bi4M9jOq83Cff9YnlxD4LqVt%2BYjM4%2B%2BTRIH9auxWSdwmDkSmgzbkJBaxQPIS4xJpUSZURpQSv6voQcINEMxvhqv%2FsAWVYyY7YY%2BqS3%2Bt0RXGPqrtd3syJdxX6j%2BDgjxVFSOMRKatp51h3vHjKUu55mVk99tYaUGqpO9lbtsu3dvH4mTOVzeFFueyQ%3D%3D&response-content-type=text%2Fhtml)

# Amiconerd.exe

AmicoNerd.exe						
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
mscoree.dll	1	000B444C	00000000	00000000	000B446E	00002000

## Libreria

La libreria **mscoree.dll** fa parte del framework Microsoft.NET. Offre la possibilità di collegare informazioni, sistemi, persone e dispositivi attraverso il software. Essa è un **Microsoft Runtime Execution Engine**, ovvero contiene le funzioni fondamentali del framework Microsoft.NET.

The screenshot shows the VirusTotal analysis page for the file AmicoNerd.exe. At the top, there's a summary card with a red circular icon containing '52 / 72' and a green progress bar. It states '52 security vendors and 1 sandbox flagged this file as malicious'. Below this are file details: 'c6603d416dfc48894eda35d9a9a8523bdf9823e215ab926783ce6848aa8a62c4' (AutoPico.exe), Size 722.69 KB, Last Analysis Date 21 days ago, and a EXE file type icon. A list of threat categories is shown below: pexe, assembly, overlay, revoked-cert, runtime-modules, invalid-signature, signed, detect-debug-environment, checks-network-adapters, long-sleeps, direct-cpu-clock-access, via-tor, and calls-wmi. Below the summary card is a navigation bar with tabs: DETECTION (selected), DETAILS, RELATIONS, BEHAVIOR, and COMMUNITY (20+). A call-to-action 'Join the VT Community' is present. The 'DETECTION' tab displays a table of security vendors' analysis results:

Security vendor	Threat label	Threat category	Family labels
AhnLab-V3	HackTool/Win.AutoKMS.C948312	ALYac	Application.Hacktool.KMSActivator.AQ
Antiy-AVL	RiskWare[NetTool]/Win64.RPCHook	Arcabit	Application.Hacktool.KMSActivator.AQ
Avast	Win32:MiscX-gen [PUP]	AVG	Win32:MiscX-gen [PUP]
BitDefender	Application.Hacktool.KMSActivator.AQ	BitDefenderTheta	Gen:NN.ZemsilF.36792.Tm1@e8vJERd
Microsoft	HackTool/MSIL/AutoKms	NANO-Antivirus	Riskware.Win32.ProcPatcher.eilohf
Panda	HackingTool/AutoKMS	QuickHeal	Trojan.YakbeexMSIL.ZZ4

Controllando sempre con Virustotal abbiamo notato che l'eseguibile è un AutoKMS, cioè un software tipicamente associato all'attivazione illegale di prodotti Microsoft, come Windows e Office.

L'utilizzo di questi strumenti soprattutto da parte delle aziende comporta gravi conseguenze di tipo legale e possono rappresentare dei rischi per la sicurezza del sistema perché possono contenere malware.