

BUILD WEEK 3

Fabiola Curcio Mazzone,
Mattia Deiana,
Mattia Pastorelli,
Stefano Pirrera,
Georges Fotsing,
Francesco Gallo



TRACCIA GIORNO 1



Con riferimento al file eseguibile Malware_Build_Week_U3, rispondere ai seguenti quesiti utilizzando i tool e le tecniche apprese nelle lezioni teoriche:

- Quanti parametri sono passati alla funzione Main()?
- Quante variabili sono dichiarate all'interno della funzione Main()?
- Quali sezioni sono presenti all'interno del file eseguibile?
- Descrivete brevemente almeno 2 di quelle identificate- Quali librerie importa il Malware? Per ognuna delle librerie importate, fate delle ipotesi sulla base della sola analisi statica delle funzionalità che il Malware potrebbe implementare. Utilizzate le funzioni che sono richiamate all'interno delle librerie per supportare le vostre ipotesi.



GIORNO 1

Per identificare quanti parametri e quali variabili sono passati per la funzione Main del nostro Malware andremo ad utilizzare IDAPro, ampiamente utilizzato dagli analisti di sicurezza, ricercatori di malware e professionisti del reverse engineering per esaminare il codice binario di software, malware e firmware.

- Con offset positivo sono accettati 3 parametri (sottolineati in blu):
- Con offset negativo sono incluse 5 variabili (sottolineate in rosso):

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

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



**IDA
Pro**



GIORNO 1

Per identificare invece le sezioni e le librerie importate dal malware utilizzeremo il tool CFF Explorer.

Una volta aperto ed eseguito il malware in analisi, potremmo vedere le sezioni e le librerie avvalendoci del menu alla sinistra:

L'eseguibile include 4 sezioni e importa 2 librerie.



| Module Name | Imports | OFTs | Time |
|--------------|--------------|----------|--------|
| szAnsi | (nFunctions) | Dword | Dword |
| KERNEL32.dll | 51 | 00007534 | 000000 |
| ADVAPI32.dll | 2 | 00007528 | 000000 |

| Name | Virtual Size | Virtual Address | Raw Size |
|---------|--------------|-----------------|----------|
| Byte[8] | Dword | Dword | Dword |
| .text | 00005646 | 00001000 | 00006000 |
| .rdata | 000009AE | 00007000 | 00001000 |
| .data | 00003EA8 | 00008000 | 00003000 |
| .rsrc | 00001A70 | 0000C000 | 00002000 |



GIORNO 2

| Name | Library |
|-----------------|----------|
| RegSetValueExA | ADVAPI32 |
| RegCreateKeyExA | ADVAPI32 |

Analizzandole più nel dettaglio con IDApro:

- **RegCreateKeyEx e RegSetValueEx** di advapi32.dll: il malware potrebbe leggere o modificare le voci del registro per configurarsi, raccogliere informazioni o nascondersi;
- **CreateFile, ReadFile, WriteFile** di kernel32.dll: il malware potrebbe intercettare o modificare i file o i dati di sistema per monitorare l'attività dell'utente o dell'intero sistema.

| |
|------------------|
| GetLastError |
| WriteFile |
| TerminateProcess |

| | |
|------------------|----------|
| RtlUnwind | KERNEL32 |
| HeapAlloc | KERNEL32 |
| HeapReAlloc | KERNEL32 |
| SetStdHandle | KERNEL32 |
| FlushFileBuffers | KERNEL32 |
| SetFilePointer | KERNEL32 |
| CreateFileA | KERNEL32 |
| GetCPIInfo | KERNEL32 |
| GetACP | KERNEL32 |
| GetOEMCP | KERNEL32 |
| GetProcAddress | KERNEL32 |
| LoadLibraryA | KERNEL32 |
| SetEndOfFile | KERNEL32 |
| ReadFile | KERNEL32 |



In particolare, la successione delle funzioni

SizeofResource, LockResource, LoadResource e

FindResource suggerisce che il malware sta

preparando un file o una risorsa da rilasciare sul sistema bersaglio, quindi si comporta come un **Dropper**.



SizeofResource

KERNEL32

LockResource

KERNEL32

LoadResource

KERNEL32

VirtualAlloc

KERNEL32

GetModuleFileNameA

KERNEL32

GetModuleHandleA

KERNEL32

FreeResource

KERNEL32

FindResourceA

KERNEL32

TRACCIA GIORNO 2



Con riferimento al Malwarein analisi, spiegare:

- Lo scopo della funzione chiamata alla locazione di memoria 00401021
- Come vengono passati i parametri alla funzione alla locazione 00401021;
- Che oggetto rappresenta il parametro alla locazione 00401017
- Il significato delle istruzioni comprese tra gli indirizzi 00401027 e 00401029. (se serve, valutate anche un'altra o altre due righe assembly)
- Con riferimento all'ultimo quesito, tradurre il codice Assembly nel corrispondente costrutto C .
- Valutate ora la chiamata alla locazione 00401047, qual è il valore del parametro «ValueName»? Nel complesso delle due funzionalità appena viste, spiegate quale funzionalità sta implementando il Malwarein questa sezione.

2

GIORNO 2

```
+ .text:00401013
+ .text:00401015
+ .text:00401017
+ .text:0040101C
+ .text:00401021
+ .text:00401027
+ .text:00401029
- .text:00401030
```

```
push    $0          ; lpClass
push    $0          ; Reserved
push    offset SubKey ; "SOFTWARE"
push    80000002h   ; hKey
call    ds:RegCreateKeyExA
test   eax, eax
jz     short loc_401032
```

Sempre avvalendoci di IDAPro, individuiamo che alla locazione di memoria 00401021 è presente la funzione **RegCreateKeyExA**. che, come già visto in precedenza, è utilizzata per creare una nuova chiave o aprire una chiave esistente nel Registro di sistema.

I parametri sono passati alla funzione sullo stack, utilizzando l'istruzione "**push**". Come possiamo vedere, nello specifico, alla locazione 00401017 il valore della chiave viene passato alla funzione.

2

GIORNO 2

test

eax, eax

jz
short loc_401032

Il costrutto compreso tra gli indirizzi 00401027 e 00401029 rappresenta un "**salto condizionale**", ovvero un'istruzione in linguaggio di programmazione o in linguaggio assembly che consente al programma di prendere decisioni e di eseguire istruzioni diverse a seconda delle condizioni specificate.

Nello specifico, l'istruzione "**test eax, eax**" seguita dall'istruzione "**jz**" viene utilizzata per controllare se il parametro EAX è uguale a zero.



In codice C potremmo tradurlo così:

```
if (eax == 0) {  
    goto loc_401032  
}
```

2

GIORNO 2

```
+ .text:0040103C
+ .text:0040103E
+ .text:00401043
+ .text:00401046
+ .text:00401047 |
+ .text:0040104D
+ .text:0040104F
```

```
push    0                                ; Reserved
push    offset ValueName ; "GinaDLL"
mov     eax, [ebp+hObject]
push    eax                                ; hKey
call    ds:RegSetValueExA
test   eax, eax
jz     short loc_401062
```

Il valore del parametro **ValueName**, alla locazione 00401047, viene utilizzata in continuità per settare il valore della chiave di registro appena creata.

GinaDLL in questo caso rappresenta tale valore; quest'ultima è associata all'autenticazione nel sistema operativo Windows, usata per gestire/modificare il processo di login. L'utilizzo da parte di un malware è segnale di un attacco che mira a compromettere l'autenticazione del sistema e ad ottenere accesso non autorizzato alle risorse del sistema o alle informazioni dell'utente.

TRACCIA GIORNO 3



Riprendete l'analisi del codice, analizzando le routine tra le locazioni di memoria 00401080 e 00401128:

- Qual è il valore del parametro «`«ResourceName»` passato alla funzione `FindResourceA()`;
- Il susseguirsi delle chiamate di funzione che effettua il Malware in questa sezione di codice l'abbiamo visto durante le lezioni teoriche. Che funzionalità sta implementando il Malware?
- È possibile identificare questa funzionalità utilizzando l'analisi statica basica ? (dal giorno 1 in pratica)
- In caso di risposta affermativa, elencare le evidenze a supporto.

Entrambe le funzionalità principali del Malware viste finora sono richiamate all'interno della funzione `Main()`. Disegnare un diagramma di flusso (inserite all'interno dei box solo le informazioni circa le funzionalità principali) che comprenda le 3 funzioni.



GIORNO 3

| | |
|-----------------|---|
| • 33C8 | MOV ECX,ECX |
| • ~E9 07010000 | JMP Malware_.004011BF |
| > A1 30804000 | MOV EAX,DWORD PTR DS:[408030] |
| • 50 | PUSH EAX |
| • 8B00 34804000 | MOV ECX,DWORD PTR DS:[408034] |
| • 51 | PUSH ECX |
| • 8B55 08 | MOV EDX,DWORD PTR SS:[EBP+8] |
| • 52 | PUSH EDX |
| • FF15 28704000 | CALL DWORD PTR DS:[<&KERNEL32.FindResourceA>] |
| • 8945 EC | MOV DWORD PTR SS:[EBP-14],EAX |
| • 8370 EC 00 | CMP DWORD PTR SS:[EBP-14],0 |

ResourceType => "BINARY"
Malware_.00408038
ResourceName => "TGAD"

[hModule
FindResourceA]

Concentrandosi sulla chiamata alla funzione **FindResourceA**, è stato possibile determinare il valore del parametro "**ResourceName**".

Come mostrato nell'immagine, il registro ECX viene utilizzato per passare il parametro "ResourceName" alla funzione FindResourceA. Tracciando il contenuto di ECX, abbiamo individuato che il malware cerca una risorsa denominata "**TGAD**". L'analisi di questo parametro è cruciale per comprendere le specifiche azioni del malware, in quanto il nome della risorsa può fornire indizi su cosa il malware stia cercando di caricare o manipolare all'interno del sistema infetto.



GIORNO 3

```
0 mov      ecx, lpName
1 push    ecx          ; lpName
0 mov      edx, [ebp+hModule]
1 push    edx          ; hModule
0 call    ds:FindResourceA
1 mov      [ebp+hResInfo], eax
0 cmp      [ebp+hResInfo], 0
1 jnz     short loc 4010DF
```

```
, CODE XREF: sub_4

mov    eax, [ebp+hResInfo]
push   eax          ; hResInfo
mov    ecx, [ebp+hModule]
push   ecx          ; hModule
call   ds:LoadResource
mov    [ebp+hResData], eax
cmp    [ebp+hResData], 0
jnz    short loc_4010FB
jmp    loc_kB110E
```

```
; CODE XREF: S

mov    edx, [ebp+hResData]
push   edx          ; hResData
call   ds:LockResource
mov    [ebp+Str], eax
cmp    [ebp+Str], 0
jnz    short loc_401113
jmp    loc_4011A5
```

```
mov    eax, [ebp+hResInfo]
push   eax          ; hResInfo
mov    ecx, [ebp+hModule]
push   ecx          ; hModule
call   ds:SizeofResource
mov    [ebp+Count], eax
cmp    [ebp+Count], 0
ja     short loc 40112C
```

Cerca di una Risorsa:

malware inizia localizzando una risorsa specifica con FindResourceA.

Caricamento della Risorsa:

opodichè carica una risorsa da un file
seguibile in memoria con LoadResource.

locco della Risorsa:

Successivamente, si assicura l'esclusivo
accesso alla risorsa con LockResource.

Determinazione della Dimensione:

In fine, determina la dimensione della sorsa con `SizeofResource`.

GIORNO 3



malware esegue le seguenti operazioni:

- Identifica e blocca risorse interne per l'uso esclusivo per accedere a codici o dati nascosti.

Il report da VirusTotal conferma che il file è stato identificato come malware da 52 fornitori di sicurezza su 72, con nessun sandbox che lo segnala come innocuo.

La sezione .rsrc (risorse), con dimensioni e entropia notevoli, potrebbe contenere le risorse bloccate e utilizzate dal malware, come visto nel codice assembler.

The screenshot shows the VirusTotal analysis interface for a file named 'Lab11-01.exe'. The file has a SHA-256 hash of '57d8d248a8741176348b5d12dcf29f34c8f48ede0ca13c30d12e5ba0384056d7'. The analysis summary indicates 52 security vendors flagged it as malicious, while no sandboxes did. The file is 52.00 KB in size and was last analyzed 8 days ago. The file type is identified as EXE. Below the file details, there are tabs for DETECTION, DETAILS, RELATIONS, BEHAVIOR, and COMMUNITY (10). A call-to-action at the bottom encourages joining the community for more insights and API keys.

52 / 72

52 security vendors and no sandboxes flagged this file as malicious

Reanalyze Similar More

57d8d248a8741176348b5d12dcf29f34c8f48ede0ca13c30d12e5ba0384056d7

Lab11-01.exe

Size 52.00 KB Last Analysis Date 8 days ago

EXE

Community Score

peexe spreader armadillo checks-user-input

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 10

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

| Sections | | | | |
|----------|-----------------|--------------|----------|---------|
| Name | Virtual Address | Virtual Size | Raw Size | Entropy |
| .text | 4096 | 22086 | 24576 | 6.23 |
| .rdata | 28672 | 2478 | 4096 | 3.77 |
| .data | 32768 | 16040 | 12288 | 0.6 |
| .rsrc | 49152 | 6768 | 8192 | 4.15 |



GIORNO 3

MAIN()

Estrae dalla risorsa denominata TGAD
una componente del Malware
(Gina.DLL).

Crea una nuova sottochiave all'interno
del registro di sistema e ne imposta il
valore (Gina.DLL).

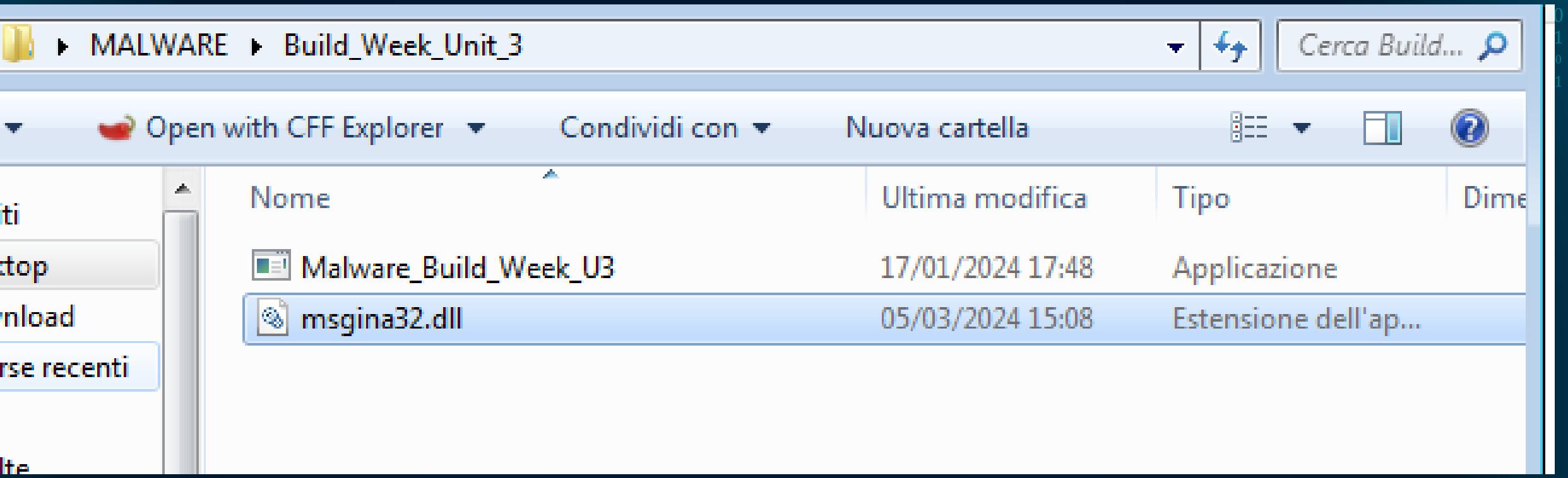


TRACCIA GIORNO 4

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

4

GIORNO 4



Eseguendo il Malware e analizzandolo con Process Monitor (procmon) la prima cosa che notiamo è che, all'interno della cartella dell'eseguibile, è stato creato un file .dll ovvero "msgina32.dll". Quest'ultimo, creato in seguito al cambio di valore del registro (Gina.DLL), è responsabile dell'autenticazione su sistemi Windows quindi potrebbe essere utilizzata per intercettare le credenziali di login sul dispositivo vittima.



GIORNO 4

REGISTRO WINDOWS:

| | | | |
|----|---------------|--|---|
| 08 | RegOpenKey | HKLM\Software\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Diagnos | NAME NOT FOUND Desired Access: Read |
| 08 | RegQueryKey | HKLM | SUCCESS Query: HandleTags, HandleTags: 0x0 |
| 08 | RegCreateKey | HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon | SUCCESS Desired Access: All Access, Disposition: REG_OPENED_EXISTING_KEY |
| 08 | RegSetInfoKey | HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon | SUCCESS KeySetInformationClass: KeySetHandleTagsInformation, Length: 0 |
| 08 | RegQueryKey | HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon | SUCCESS Query: HandleTags, HandleTags: 0x400 |
| 08 | RegSetValue | HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon\GinaDLL | ACCESS DENIED Type: REG_SZ, Length: 520, Data: C:\Users\user\Desktop\MALWARE\Build_W... |
| 08 | RegCloseKey | HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon | SUCCESS |
| 08 | RegCloseKey | HKLM\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION\Image File Execution Options | SUCCESS |
| 08 | RegCloseKey | HKLM\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION\Image File Execution Options | SUCCESS |
| 08 | RegCloseKey | HKLM\System\CurrentControlSet\Control\Nls\Sorting\Versions | SUCCESS |
| 08 | RegCloseKey | HKLM | SUCCESS |

Viene creata la chiave di registro

Quando viene creata una chiave di registro, viene aggiunta una nuova voce alla struttura gerarchica del registro di sistema di Windows. Questo permette di memorizzare informazioni di configurazione e altre impostazioni importanti per il sistema operativo, le applicazioni e i dispositivi hardware. Le chiavi di registro possono essere create manualmente o automaticamente da programmi durante l'installazione o l'esecuzione. Una volta create, possono essere lette, scritte e modificate dalle applicazioni e dai processi con i permessi corretti. È importante gestire il registro di sistema con cura per evitare modifiche dannose.

4 GIORNO

FILE SYSTEM:

| | | | |
|--|------------|--|--|
| | CloseFile | C:\Windows\SysWOW64\sechost.dll | SUCCESS |
| | CreateFile | C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll | SUCCESS Desired Access: Generic Write, Read Attributes, Disposition: OverwriteIf, Options: ... |
| | WriteFile | C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll | SUCCESS Offset: 0, Length: 4.096, Priority: Normal |
| | WriteFile | C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll | SUCCESS Offset: 4.096, Length: 2.560, Priority: Normal |
| | CloseFile | C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll | SUCCESS |

Filtrando per vedere i risultati del File System ci rendiamo conto che le chiamate di sistema Create, Write e Close hanno modificato il contenuto della cartella dell'eseguibile con la creazione del file msgina32.dll.



GIORNO 4

In conclusione della quarta giornata abbiamo compreso che:

- Importa due librerie e quattro sezioni (.text, .data ecc...)
 - Assegnazione valore a chiave di registro (Gina.DLL);
 - Il Malware cerca una risorsa denominata TGAD (FindResource), LoadResource, LockResource, SizeOfResource.
 - Il Malware crea una chiave di registro e ne associa un valore;
 - A livello di file system crea un file denominato msgina32.dll all'interno della cartella dell'eseguibile;



TRACCIA GIORNO 5

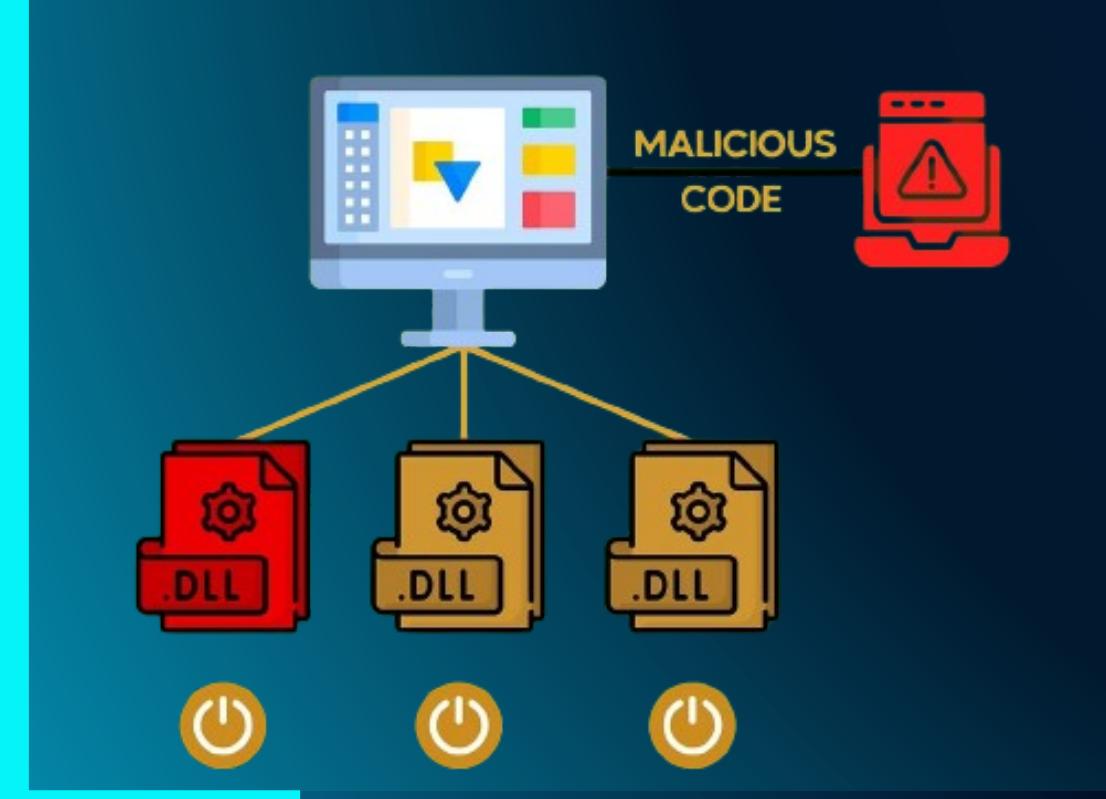
- Preparate l'ambiente ed i tool per l'esecuzione del Malware (suggerimento: avviate principalmente Process Monitor ed assicurate di eliminare ogni filtro cliccando sul tasto «reset» quando richiesto in fase di avvio). Eseguite il Malware, facendo doppio click sull'icona dell'eseguibile

5

GIORNO 5



Il malware ha la capacità di intercettare e raccogliere le credenziali degli utenti, in pratica il malware stesso si sostituisce a una libreria dinamica di collegamento (Dynamic Link Library o DLL) legittima, per eseguire codice malevolo all'insaputa dell'utente o dell'amministratore del sistema.



In sintesi, sostituire un file .dll legittimo con un file .dll malevolo potrebbe mettere a rischio la sicurezza del sistema e delle informazioni sensibili degli utenti, consentendo agli attaccanti di ottenere accesso non autorizzato e di compromettere l'integrità del sistema.

5

GIORNO 5

PROFILO DEL MALWARE E FUNZIONALITÀ:

MALWARE

ESTRAZIONE
GINA.DLL
MALEVOLO



CREAZIONE CHIAVE DI REGISTRO
E ASSEGNAZIONE VALORE
GINA.DLL

SISTEMA OPERATIVO CONTROLLA
IL REGISTRO E AVVIA IL
COMPONENTE PER FORNIRE
INTERFACCIA DI LOGIN ALL'UTENTE

INTERFACCIA LOGIN

UTENTE CHE SI
AUTENTICA



CREDENZIALI E
PASSWORD
RUBATE.



GRAZIE PER L'ATTENZIONE

