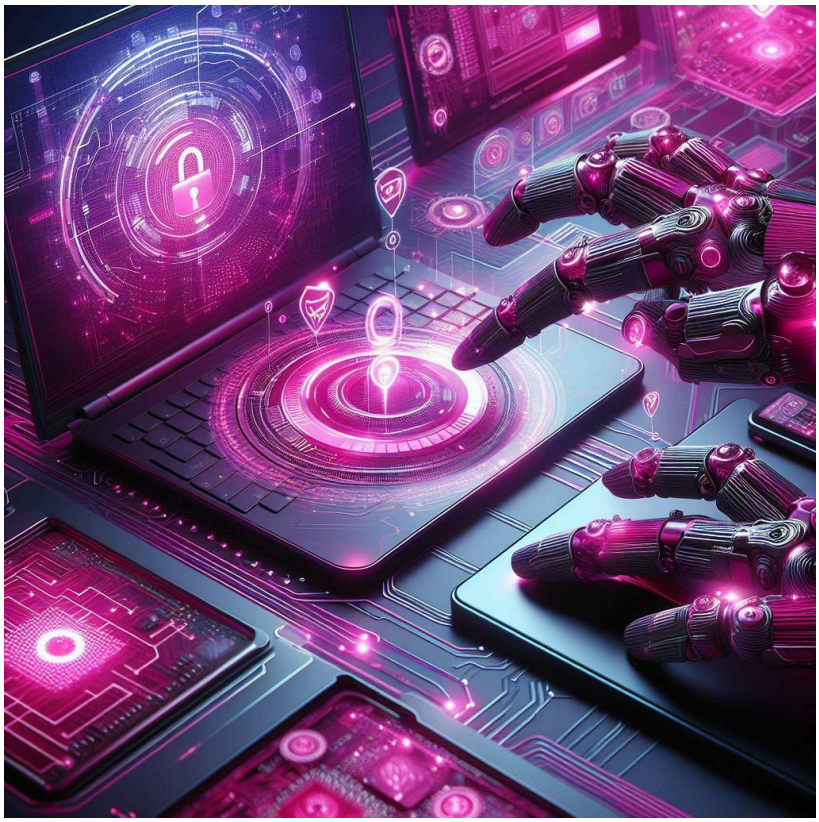
Progetto M6

22/04/2024



Rebecca Caldarella

Analisi statica

File Eseguibile: Malware_Build_Week_U3

Quanti parametri sono passati dalla funzione Main()?

Utilizzando IDA-PRO, vediamo che sono passati 3 **parametri** alla funzione Main(): argc, argvm ed envp. Capiamo che questi sono parametri in quanto si trovano ad un offset positivo.

```
; 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
```

Quante variabili sono dichiarate all'interno della funzione Main()?

All'interno della funzione Main () sono dichiarate 5 **variabili**: hModule, Data, var_117, var_8, var_4. Come si evince dallo screen precedente, le riconosciamo in quanto hanno, a differenza dei parametri, un offset negativo.

Quali sezioni sono presenti all'interno del file eseguibile? Descrivete brevemente almeno 2 di quelle identificate.

Nel file eseguibili sono presenti 4 sezioni: .text, .idata, .rdata, .data

.text: contiene le righe di codice che verranno eseguite dalla CPU quando il software viene avviato. Questa è l'unica sezione di un file eseguibile che viene eseguita dalla CPU(tutte le altre sezioni contengono dati o informazioni a supporto).

.rdata: include le informazioni sulle librerie e le funzioni importate ed esportate dall'eseguibile.

.data: contiene i dati e le variabili globali del programma eseguibile, che devono essere disponibili da qualsiasi parte del programma.

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.

Le librerie importate dal malware sono:

- **ADVAPI32**, che contiene le funzioni per interagire con i servizi e i registri del sistema operativo Microsoft. Vediamo ad esempio che questa libreria è stata utilizzata per la funzione RegSetValueExA, la quale permette di aggiungere un nuovo valore all'interno del registro e di settare i rispettivi dati.
- **KERNEL32**, che contiene le funzioni principali per poter interagire con il sistema operativo (manipolazione dei file, gestione della memoria, ..).

  Questa libreria è stata utilizzata dal malware per la funzione CreateFileA, la quale permette di creare un file (se non esistente) o di aprirne uno che esiste già; ReadFile, funzione usata per leggere da un file (questa accetta l'handle al file da cui leggere e il numero di byte da leggere); WriteFile, che serve per scrivere su un file e accetta anch'essa come parametri l'handle del file sul quale scrivere, il buffer da scrivere e il numero di byte da scrivere.

La presenza di chiamate a funzioni come *"RegCreateKeyExA"* e *"RegSetValueExA"* suggerisce che il malware potrebbe cercare di modificare il registro di sistema di Windows.Questo potrebbe essere utilizzato per installare il malware in modo persistente, configurare le impostazioni del sistema o nascondere la sua presenza.

Inoltre la presenza della stringa *"GinaDLL"* potrebbe indicare un tentativo di sostituire il DLL *"msgina 32.dll"*, che gestisce l'autenticazione degli utenti in Windows, con una versione malevola. Questo potrebbe consentire al malware di intercettare le credenziali degli utenti durante il processo di accesso al sistema.

Con riferimento al Malware in analisi spiegare:

- **Lo scopo della funzione chiamata alla locazione di memoria 00401021:**

Call ds:RegCreateKeyExA

La funzione RegCreateKeyExA viene chiamata dalla posizione di memoria 0x00401021 all'interno della sezione .text.

La funzione RegCreateKeyExA è utilizzata per creare una nuova chiave o aprire una chiave esistente nel registro di sistema di Windows. Questa funzione richiede diversi parametri, tra cui il tipo di chiave (HKEY), il percorso della sottochiave (lpSubKey), le opzioni di accesso (samDesired), e altri ancora. Restituisce un valore di tipo LSTATUS che indica se l'operazione è stata completata con successo o se si è verificato un errore.

Dal momento che il codice chiama RegCreateKeyExA, si può ipotizzare che stia cercando di creare o aprire una chiave nel registro di sistema di Windows, probabilmente per archiviare o recuperare delle informazioni utili per il funzionamento del programma. La funzione potrebbe essere chiamata per inizializzare determinati parametri o configurazioni necessari per il corretto funzionamento dell'applicazione.

- **Come vengono passati i parametri alla funzione alla locazione 00401021**

Poichè nella funzione viene utilizzata l'istruzione assembly "push", si può dedurre che viene utilizzato lo stack di memoria.
I valori vengono inseriti nello stack nell'ordine corretto, in modo che siano disponibili per la funzione chiamata.

```
.text:00401000                push    ebp
.text:00401001                mov     ebp, esp
.text:00401003                push    ecx
.text:00401004                push    0               ; lpdwDisposition
.text:00401006                lea     eax, [ebp+hObject]
.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\\CurrentVe"...
.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
.text:00401030                jmp     short loc_40107B
```

- **Che oggetto rappresenta il parametro alla locazione 00401017**

Il parametro alla funzione di memoria 00401017 è un puntatore alla stringa *"SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\WinLogon"*, terminata da un byte nullo (0) che indica la fine della stringa.
Probabilmente la stringa rappresenta il percorso della sottochiave nel registro di sistema.

```
.data:00408054 SubKey          db 'SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon',0
.data:00408054                                          ; DATA XREF: sub_401000+17↑o
```

- **Il significato delle istruzioni comprese tra gli indirizzi 00401027 e 00401029**

1. test eax, eax

Questa istruzione esegue un operazione logica AND tra il registro eax, in sostanza controlla se eax è zero. Se è così il risultato sarà ì altrimenti sarà diverso da 0, impostando il flag ZF in base al risultato dell'operazione.

2. jz short loc_401032.

Questa istruzione salta all'indirizzo loc_401032 se il flag ZF è impostato, il che significa che il registro eax è zero.

Riassumendo controlla se il registro eax è a zero, e se è a zero salta alla locazione 401032, altrimenti procede con le successive istruzioni.

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

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

- **Valutare ora la chiamata alla locazione 00401047, qual è il valore del parametro ValueName?**

```
.data:0040804C ; char ValueName[]
.data:0040804C ValueName       db 'GinaDLL',0          ; DATA XREF: sub_401000+3E↑o
```

Il valore del parametro *ValueName* è una stringa *"GinaDLL"*, terminata da un byte nullo (0).

Analisi dinamica

Cosa notate all'interno della cartella dove è situato l'eseguibile del Malware? Spiegate cosa è avvenuto, unendo le evidenze che avete raccolto finora per rispondere alla domanda.

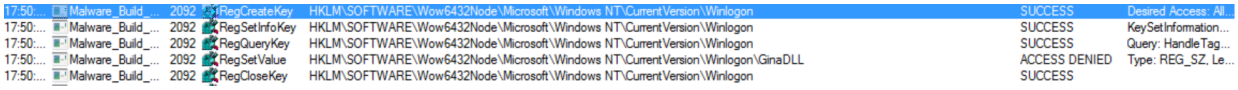
Avviando il malware viene creato il file DLL *msgina32*, si sono pertanto confermate le ipotesi precedenti. E' probabile che questo file DLL venga utilizzato per essere sostituito a quello originale.

Analizzate ora i risultati di Process Monitor.

Filtrate includendo solamente l'attività del registro di Windows.

- **Quale chiave di registro viene creata?**

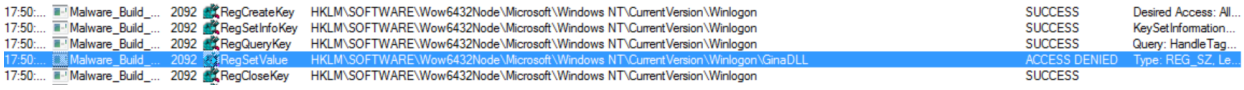
L'istruzione per inserire una chiave di registro nel sistema è "RegCreateKey". Durante l'analisi con Process Monitor è risultato l'inserimento ***Winlogon***. Tale chiave è stata inserita nel registro *HKLM*, ovvero dove sono contenuti i record e le configurazioni della macchina.



17:50:...	Malware_Build_...	2092	RegCreateKey	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS	Desired Access: All...
17:50:...	Malware_Build_...	2092	RegSetInfoKey	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS	KeySetInformation...
17:50:...	Malware_Build_...	2092	RegQueryKey	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS	Query: HandleTag...
17:50:...	Malware_Build_...	2092	RegSetValue	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon\GinaDLL	ACCESS DENIED	Type: REG_SZ, Le...
17:50:...	Malware_Build_...	2092	RegCloseKey	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS	

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

Durante l'esecuzione del malware, viene associato alla chiave precedentemente creata *Winlogon*, il valore ***GinaDLL***, corrispondente al file creato dal malware.



17:50:...	Malware_Build_...	2092	RegCreateKey	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS	Desired Access: All...
17:50:...	Malware_Build_...	2092	RegSetInfoKey	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS	KeySetInformation...
17:50:...	Malware_Build_...	2092	RegQueryKey	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS	Query: HandleTag...
17:50:...	Malware_Build_...	2092	RegSetValue	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon\GinaDLL	ACCESS DENIED	Type: REG_SZ, Le...
17:50:...	Malware_Build_...	2092	RegCloseKey	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS	

Passate ora alla visualizzazione dell'attività sul file system.

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

Le chiamate di sistema che hanno interessato la modifica del contenuto della directory del malware sono:

CreateFile, funzione utilizzata per creare un file se non esiste, o aprirne uno se già creato precedentemente.

WriteFile, che, come intuibile dal nome, scrive su un file.

CloseFile, che chiude il file

Unite tutte le informazioni raccolte fin qui sia dell'analisi statica che dell'analisi dinamica per delineare il funzionamento del malware

Dai risultati delle analisi sopra descritte, si evince come il malware si proponga di sostituire il file DLL di autenticazione di Windows, con una versione malevola (*msgina32.dll*).
Pertanto utilizza il registro di sistema per garantire che la versione malevola venga caricata al posto di quella legittima.
Poichè il contenuto del file .DLL non è stato analizzato, si può solo ipotizzare che venga utilizzato per raccogliere le credenziali dell'utente e/o favorire l'infiltrazione di utenti malintenzionati.