Progetto Malware Analysis   
**Andrea Volterra**

Analisi Statica  
  
Immagine che contiene testo, elettronica, schermata, schermo

Descrizione generata automaticamente

1) Nella funzione *main()* vengono passati tre parametri:

* **argc** (argument count): È un parametro in posizione 0 (primo parametro) e viene rappresentato da dword ptr 8, quindi occupa 4 byte.
* **argv** (argument vector): È un parametro in posizione 1 (secondo parametro) e viene rappresentato da dword ptr 0Ch, quindi occupa 4 byte.
* **envp** (environment pointer): È un parametro in posizione 2 (terzo parametro) e viene rappresentato da dword ptr 10h, quindi occupa 4 byte.

-----------------------------------------------------------------------------------------------------------------------------------------------

2) sono dichiarate 4 variabili nello stack frame della funzione:

* **hModule**: È una variabile di tipo dword (4 byte) che viene dichiarata come -11Ch rispetto all'indirizzo della base dello stack (ebp). Quindi occupa 4 byte di spazio nello stack.
* **Data**: È una variabile di tipo byte (1 byte) che viene dichiarata come -118h rispetto all'indirizzo della base dello stack (ebp). Quindi occupa 1 byte di spazio nello stack.
* **var\_8**: È una variabile di tipo dword (4 byte) che viene dichiarata come -8 rispetto all'indirizzo della base dello stack (ebp). Quindi occupa 4 byte di spazio nello stack.
* **var\_4**: È una variabile di tipo dword (4 byte) che viene dichiarata come -4 rispetto all'indirizzo della base dello stack (ebp). Quindi occupa 4 byte di spazio nello stack.

-----------------------------------------------------------------------------------------------------------------------------------------  
  
3)Immagine che contiene testo, schermata, numero, Carattere

Descrizione generata automaticamente

Sono presenti quattro sezioni, visibili con cff explorer

* **.rsrc**: include le risorse utilizzati dall’eseguibile che non vengono considerate parte di esso, come icone, immagini, menu e stringhe.
* **.rdata**: solitamente contiene le informazioni da importare ed esportare. Può inoltre salvare dai dati read-only (ossia che si può solo leggere) usati dal programma;

----------------------------------------------------------------------------------------------------------------------------------------------

4) Vengono importate le librerie **Kernel32.dll** e **ADVAPI32.dll**

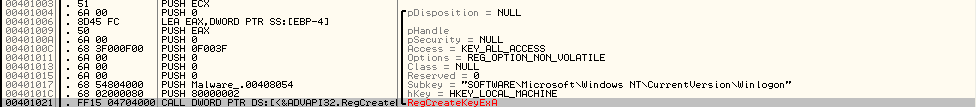
Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

La libreria **Kernel32.dll** può far presupporre che il malware utilizzi funzioni per la gestione della memoria o funzioni per interagire con il sistema operativo

**ADVAPI32.dll** permette invece al malware di entrare nelle chiavi di registro

-----------------------------------------------------------------------------------------------------------------------------------------------

5) 

Lo scopo della funzione all’indirizzo di memoria 00401021 è creare la chiave di registro “*SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon*” e viene passata tramite i push che caricano i parametri



All’indirizzo 00401017 troviamo la chiave di registro che porta all’avvio automatico la dll fallata.



Il malware qui verifica se è già stato avviato. In caso negativo fa il salto all’indirizzo *loc\_401032* altrimenti si chiude, come visibile nel diagramma di flusso a seguito

Immagine che contiene testo, schermata, software, diagramma

Descrizione generata automaticamente

Di seguito la traduzione del costrutto if in C

if (eax == 0){  
 funct\_401032();   
}  
else{  
 eax = 1;  
 funct\_40107B();

}

*Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente*

All’indirizzo di memoria 00401047 il parametro ValueName ha valore “*GinaDLL”*

*-----------------------------------------------------------------------------------------------------------------------------------------------*

**Analisi Dinamica**  
  
6) Immagine che contiene testo, schermata, Carattere, software

Descrizione generata automaticamente

all’interno della cartella dove era situato inizialmente il malware si è creato il file msgina32.dll, che è probabilmente una versione corrotta della DLL GINA la quale, citando le dispense di Microsoft:  
“The purpose of a GINA DLL is to provide customizable user identification and authentication procedures.”  
Ovvero: “Ha lo scopo di fornire procedure di identificazione e autenticazione dell'utente personalizzabili”

-------------------------------------------------------------------------------------------------------------------------------------------------------------

7) 

Viene creata la chiave di registro Winlogon, e gli viene dato il valore msgina32.dll che abbiamo trovato nella cartella del malware



La chiamata che modifica il contenuto della cartella è la CreateFile

**Conclusioni:** Il malware è probabilmente un **dropper** in quanto utilizza la sezione .rsrc e contiene al suo interno un logger che copia le credenziali di accesso