Malware Analysis

Il Malware da analizzare è nella cartella Build_Week_Unit_3 presente sul desktop della macchina virtuale dedicata.

Analisi statica

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.

Traccia 1 e 2
-Quanti parametri sono passati alla funzione Main()?
-Quante variabili sono dichiarate all'interno della funzione Main()?

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):

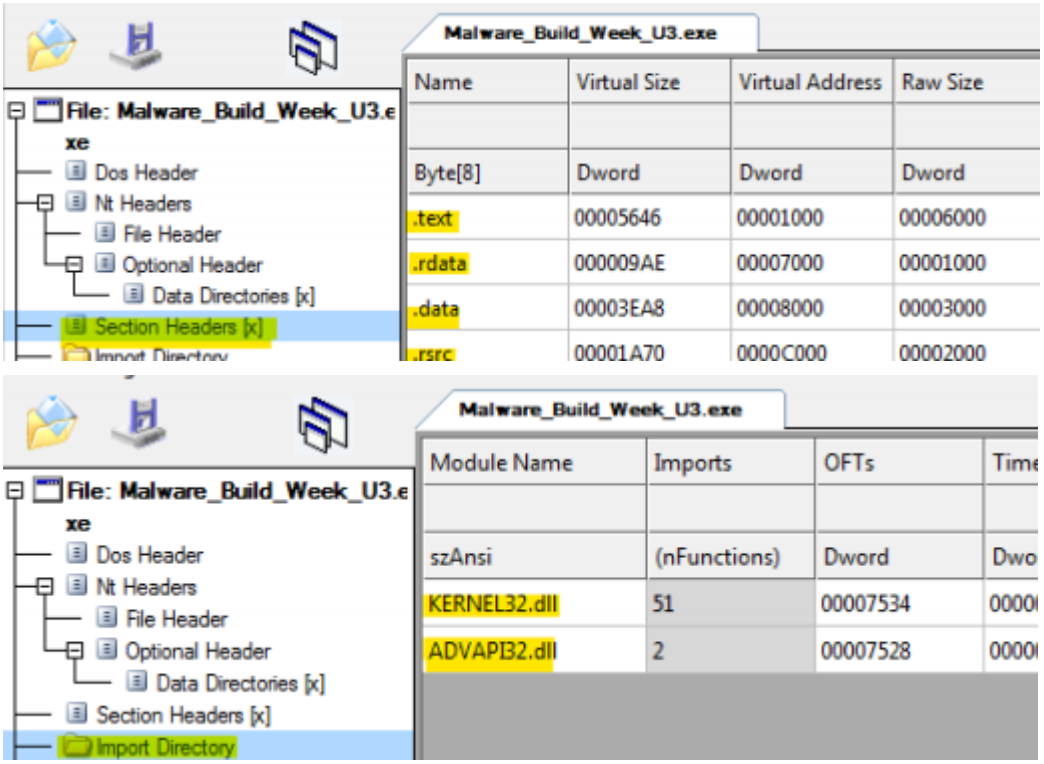
```
; int __cdecl main(int argc, const char **argv, co
_main proc near

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
```

Traccia 3

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

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:



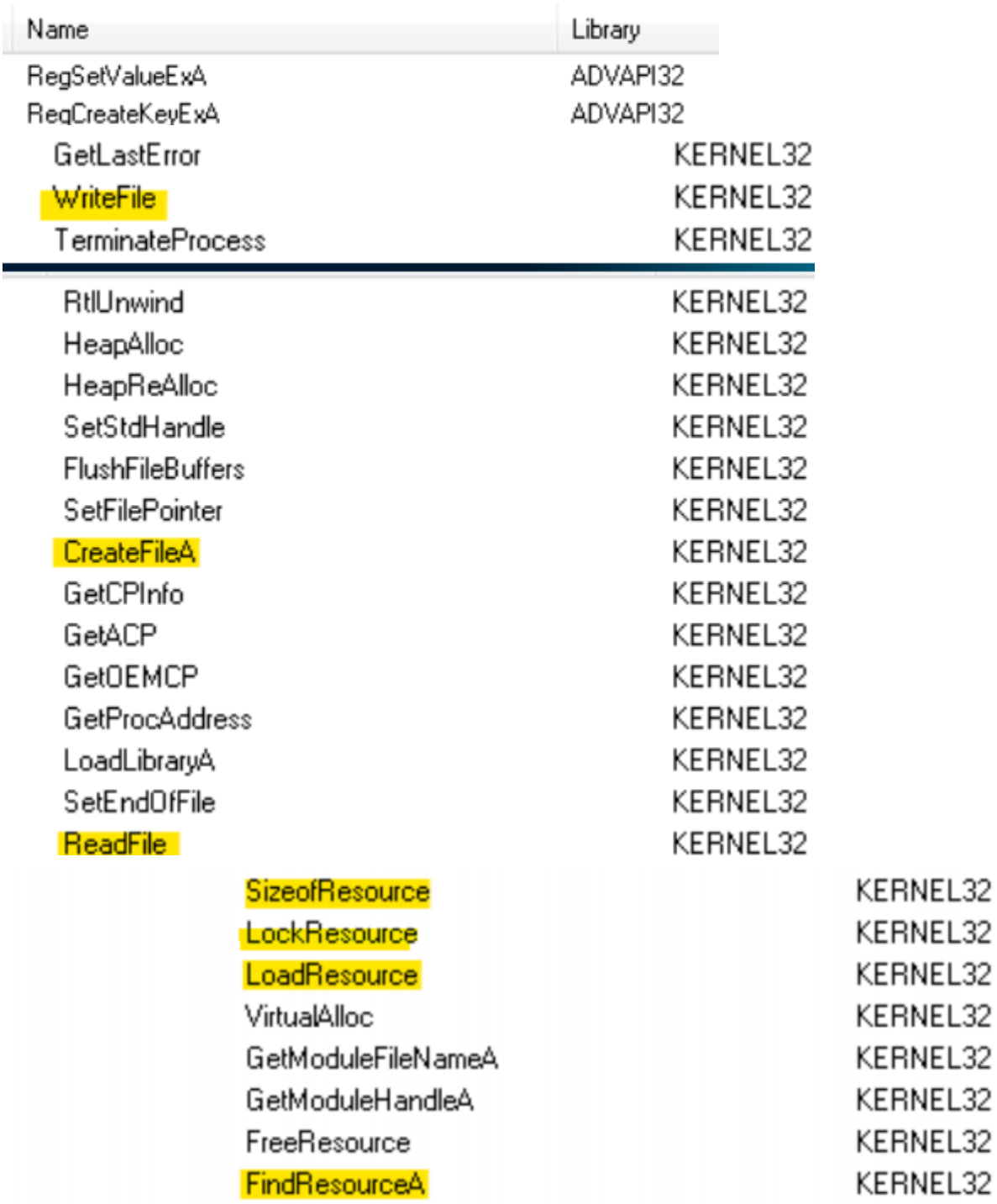
L'eseguibile include 4 sezioni e importa 2 librerie.

Traccia 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. Utilizzate le funzioni che sono richiamate all'interno delle librerie per supportare le vostre ipotesi.

Analizzandole più nel dettaglio con IDApro:

- RegCreateKeyEx e RegSetValueEx diadvapi32.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.
-



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.

Malware Analysis

Con riferimento al Malware in 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.
- 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»?

Traccia 5-6-7
-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

```
.text:00401013    push    0                   ; lpClass
.text:00401015    push    0                   ; Reserved
.text:00401017    push    offset SubKey       ; "SOFTWARE
.text:0040101C    push    80000002h           ; hKey
.text:00401021    call    ds:RegCreateKeyExA
.text:00401027    test    eax, eax
.text:00401029    jz      short loc_401032
```

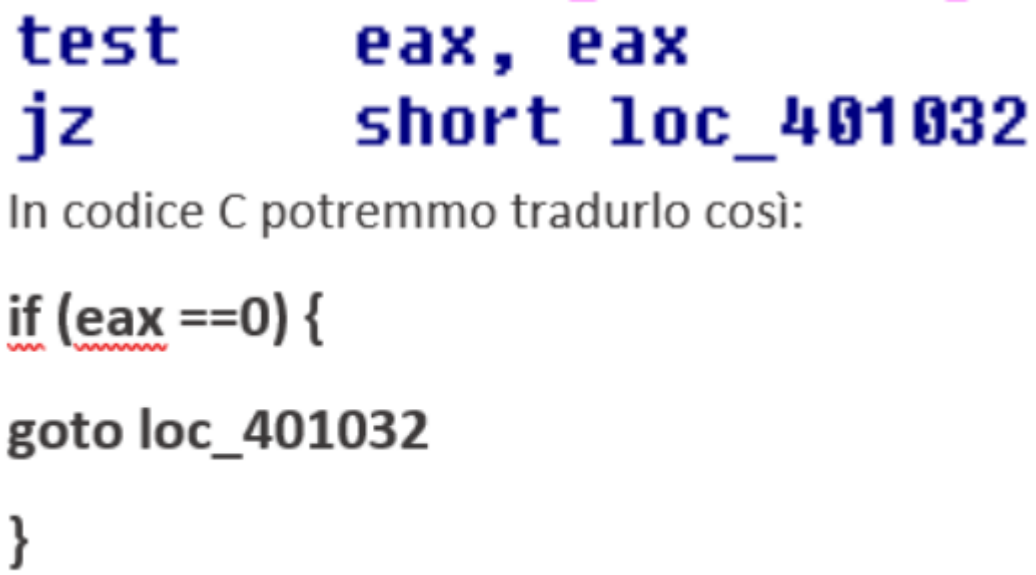
Sempre avvalendoci di IDAPro, individuiamo che alla locazione di memoria 00401021 è presente la funzione RegCreateKeyExA. che, come già visto in precedenza, è utilizzata per creare una nuova chiave o aprire una chiave esistente nel Registro di sistema.
I parametri sono passati alla funzione sullo stack, utilizzando l'istruzione "push".
Come possiamo vedere, nello specifico, alla locazione 00401017 il valore della chiave viene passato alla funzione.

Traccia 8-9

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

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

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



Traccia 10

-Valutate ora la chiamata alla locazione 00401047, qual è il valore del parametro «ValueName»?

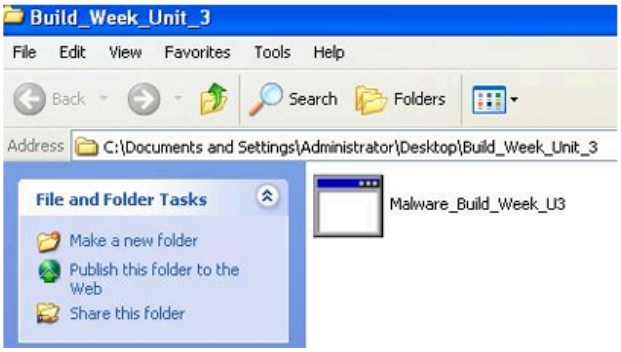
Il valore del parametro ValueName, alla locazione 00401047, viene utilizzata in continuità per settare il valore della chiave di registro appena creata. GinaDLL in questo caso rappresenta tale valore; quest'ultima è associata all'autenticazione nel sistema operativo Windows, usata per gestire/modificare 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.

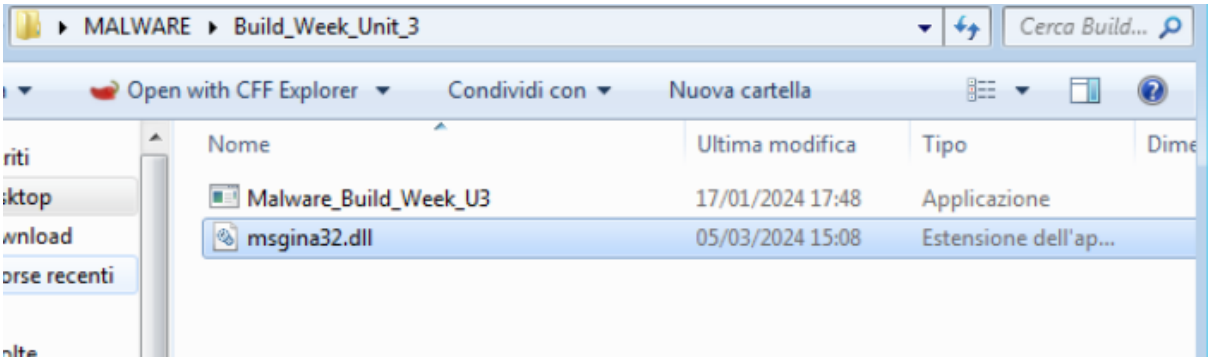
```
.text:0040103C                push    0               ; Reserved
.text:0040103E                push    offset ValueName ; "GinaDLL"
.text:00401043                mov     eax, [ebp+hObject]
.text:00401046                push    eax             ; hKey
.text:00401047                call    ds:RegSetValueExA
.text:0040104D                test    eax, eax
.text:0040104F                jz      short loc_401062
```

Malware Analysis

Analisi dinamica

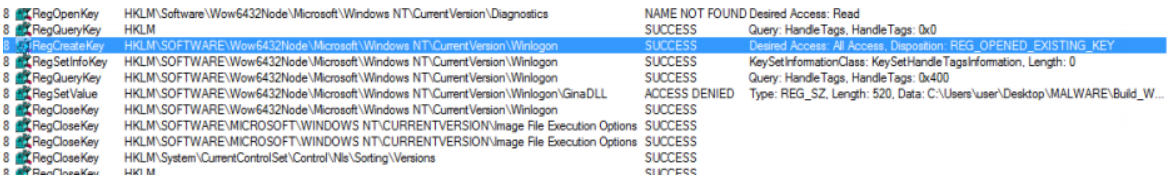
Preparate l'ambiente ed i tool per l'esecuzione del Malware (suggerimento: avviate principalmente Process Monitor ed assicurate di eliminare ogni filtro cliccando sul tasto «reset» quando richiesto in fase di avvio). Eseguite il Malware, facendo doppio click sull'icona dell'eseguibile





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.

REGISTRO WINDOWS:



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.
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: Overwritelf, 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.