MALWARE ANALYSIS

ANALISI STATICA

- Quanti parametri sono passati alla funzione Main()?
I parametri passati alla funzione Main() sono 5 ovvero hModule, Data, argc, argv e envp.
- Quante variabili sono dichiarate all'interno della funzione Main()?
Le variabili dichiarate all'interno della funzione Main() sono 2 ovvero var_8 e var_4.

```
_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
```

- Quali sezioni sono presenti all'interno del file eseguibile? Descrivete brevemente almeno 2 di quelle identificate.
Le sezioni presenti all'interno del file eseguibile sono ".text", ".rdata", ".data" e ".rsrc".

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

La sezione ".text" contiene il codice che viene eseguito dalla CPU e di conseguenza all'interno di essa si trova il cosiddetto "entry point" ovvero il primo indirizzo di memoria che il sistema operativo eseguirà quando il programma viene avviato.
La sezione ".rsrc" contiene le risorse del programma come ad esempio icone e stringhe. Queste risorse possono essere utilizzate per personalizzare l'aspetto e il comportamento dell'applicazione. Questa sezione è generalmente di sola lettura e contiene dati che possono essere richiamati dal programma durante l'esecuzione, ma non è eseguibile come codice.

- 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.
Le librerie importate sono "*KERNEL32.dll*" e "*ADVAPI32.dll*".

Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.dll	51	00007534	00000000	00000000	0000769E	0000700C
ADVAPI32.dll	2	00007528	00000000	00000000	000076D0	00007000

Funzioni importate da "KERNEL32.dll"
FindResource() e LoadResource() permettono di determinare la posizione di una risorsa e successivamente caricarla in memoria per l'esecuzione o da salvare sul disco.
Funzioni importate da "ADVAPI32.dll"
La funzione "RegCreateKeyExA" della libreria "ADVAPI32.dll" viene utilizzata per creare una nuova chiave di registro o aprirne una già esistente.

La funzione “RegSetValueExA” consente l'aggiunta di un nuovo valore al Registro di sistema e la configurazione dei dati associati a tale valore.

- Lo scopo della funzione chiamata alla locazione di memoria 00401021

```
00401021 . FF15 04704000 CALL DWORD PTR DS:[<&ADVAPI32.RegCreate RegCreateKeyExA
```

Alla locazione di memoria abbiamo la funzione “RegCreateKeyExA” che viene utilizzata per creare una nuova chiave o aprirne una già esistente nel Registro di sistema di Windows.

- Come vengono passati i parametri alla funzione alla locazione 00401021

```
00401004 . 6A 00          PUSH 0                                  pDisposition = NULL
00401006 . 8D45 FC        LEA EAX,DWORD PTR SS:[EBP-4]
00401009 . 50             PUSH EAX                                pHandle
0040100A . 6A 00          PUSH 0                                  pSecurity = NULL
0040100C . 68 3F000F00    PUSH 0F003F                             Access = KEY_ALL_ACCESS
00401011 . 6A 00          PUSH 0                                  Options = REG_OPTION_NON_VOLATILE
00401013 . 6A 00          PUSH 0                                  Class = NULL
00401015 . 6A 00          PUSH 0                                  Reserved = 0
00401017 . 68 54804000    PUSH Malware_.00408054                  Subkey = "SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon"
0040101C . 68 02000080    PUSH 80000002                           hKey = HKEY_LOCAL_MACHINE
00401021 . FF15 04704000  CALL DWORD PTR DS:[<&ADVAPI32.RegCreate RegCreateKeyExA
```

I parametri della funzione sono passati sullo stack tramite l’istruzione “PUSH”.

- Che oggetto rappresenta il parametro alla locazione 00401017

```
00401017 . 68 54804000    PUSH Malware_.00408054                  Subkey = "SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon"
```

Il parametro alla locazione 00401017 ovvero “SubKey” è un puntatore che punta ad una stringa (LPCSTR) contenente il percorso nel Registro di sistema e pertanto il nome della sottochiave che vogliamo creare o aprire tramite la chiamata di funzione “RegCreateKeyExA”.

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

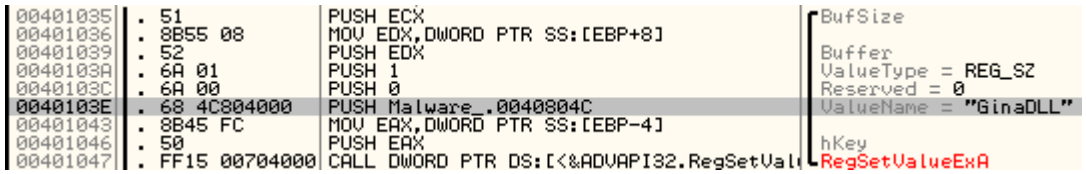
```
00401027 . 85C0           TEST EAX,EAX
00401029 .v74 07          JE SHORT Malware_.00401032
```

TEST EAX, EAX -> questa istruzione effettua un'operazione di AND logico bit a bit tra il registro EAX e sé stesso lasciandoli entrambi inalterati, al contrario di quanto farebbe invece l'effettiva AND. L’istruzione modifica il flag ZF (Zero Flag) del registro EFLAGS, che viene settato ad 1 se e solo se il risultato dell’AND è 0.
JE SHORT Malware_.00401032 -> questa è un'istruzione di salto condizionale. Se il flag ZF è uguale a 1 il programma salterà alla locazione 00401032 in caso contrario l'esecuzione continuerà normalmente con l'istruzione successiva.

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

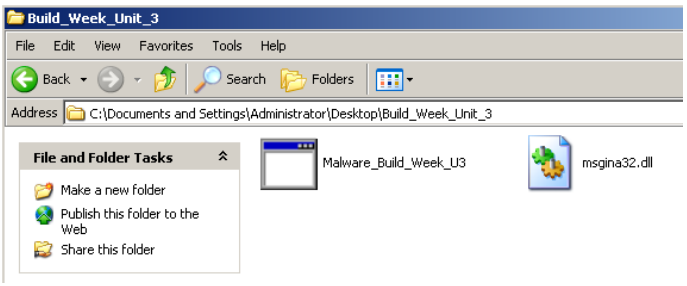
```
if (eax == eax) {
        ecx = ebp+12;
}
else {
        eax = 1;
}
```

- Valutate ora la chiamata alla locazione 00401047, qual è il valore del parametro «ValueName»?
Il valore del parametro “ValueName” è “GinaDLL”.

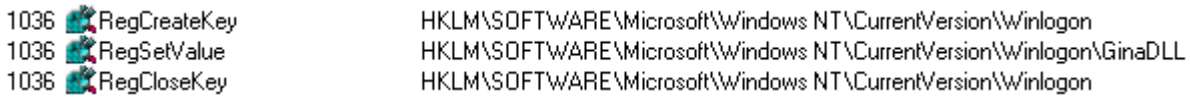


ANALISI DINAMICA

- Cosa notate all’interno della cartella dove è situato l’eseguibile del malware?
All’interno della cartella è comparso il file denominato “*msgina32.dll*”.



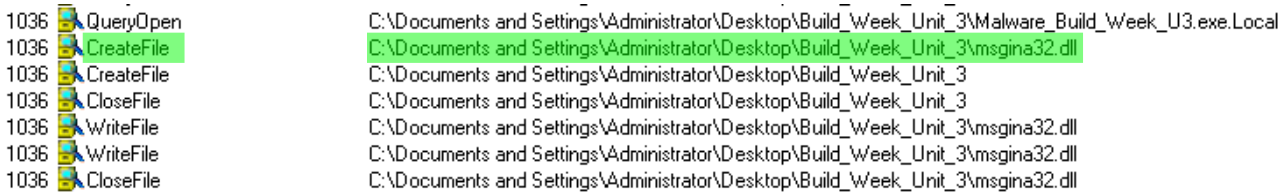
- Quale chiave di registro viene creata?
Viene creata la sottochiave “GinaDLL” alla chiave “Winlogon”.



- Quale valore viene associato alla chiave di registro creata?
Alla sottochiave “GinaDLL” viene associato il seguente valore: ”C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\msgina32.dll”

Type: REG_SZ, Length: 520, Data: C:\Documents and Settings\Administrator\Desktop\Build_Week_Unit_3\msgina32.dll

- Quale chiamata di sistema ha modificato il contenuto della cartella dove è presente l’eseguibile del malware?
La chiamata che ha modificato il contenuto della cartella dove è presente l’eseguibile del malware è stata “CreateFile”.



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

Le varie informazioni fin qui analizzate conducono all'ipotesi che il sistema è stato compromesso da un malware di tipo *dropper.* Alcuni indicatori inoltre, come le modifiche al Registro di sistema, suggeriscono che il malware stia cercando di stabilire una presenza persistente sulla macchina garantendo così l'esecuzione automatica all'avvio del sistema operativo.
La creazione del file "msgina32.dll", avvenuta una volta avviato il file eseguibile, e le modifiche alla sottochiave di registro "GinaDLL"[1], presente all'interno della chiave "Winlogon"[2], indicano che il malware ha come obiettivo l'integrazione di un componente aggiuntivo nel processo di autenticazione di Windows rappresentato appunto dal file creato dal malware.

[1] https://learn.microsoft.com/it-it/windows/win32/secauthn/gina
[2] https://learn.microsoft.com/it-it/windows/win32/secauthn/winlogon