Analisi Statica

In riferimento al file eseguibile Malware_Build_week_U3 possiamo dire che:

Ci sono 3 parametri della funzione

```
argc= dword ptr  8
argv= dword ptr  0Ch
envp= dword ptr  10h
```

(offset positivo)

Ci sono 4 variabili

```
; int __cdecl main(int arg
_main proc near

hModule= dword ptr -11Ch
Data= byte ptr -118h
var_8= dword ptr -8
var_4= dword ptr -4
```

(offset negativo)

Le sezioni presenti nel file eseguibile sono 4 :

Program Segmentation

Name	Start	End	R	W	X	D	L	Align	Base	Type	Class	AD
.text	00401000	00407000	R	.	X	.	L	para	0001	public	CODE	32
.idata	00407000	004070DC	R	.	.	.	L	para	0002	public	DATA	32
.rdata	004070DC	00408000	R	.	.	.	L	para	0002	public	DATA	32
.data	00408000	0040BEA8	R	W	.	.	L	para	0003	public	DATA	32

La sezione .**text** contiene il codice eseguibile del programma; questa sezione è dedicata alla memorizzazione delle istruzioni del linguaggio macchina che costituiscono il programma; nel nostro caso il programma sta cercando di modificare le chiavi di registro, infatti va a controllare le autorizzazioni di accesso all'oggetto.

La sezione .**rdata** contiene le informazioni sulle librerie (ADVAPI32, KERNEL32) e sulle funzioni inporate ed esportate dall'eseguibile.

La sezione .**data** contiene i dati e le variabili globali del programma. Le Librerie importate sono 2:

1. **ADVAPI32**.dll fornisce funzionalità relative alla sicurezza, alla gestione dei privilegi, ai servizi, e al Registro di sistema, infatti implemaenta la funzione **RegSetValueExA** che è parte delle API di Windows e viene utilizzata per impostare il valore di una chiave nel Registro di sistema, e la funzione RegCreateKeyExA, questa è un'altra funzione delle API di Windows utilizzata per creare o aprire una chiave nel Registro di sistema.
   Quindi il malware può utilizzare questa libreria per manipolare o nascondere le proprie voci di registro e ottenere persistenza nel sistema.
2. **KERNEL32**.dll questa libreria fornisce invece funzioni necessarie per la gestione del sistema operativo Windows e delle applicazioni che lo utilizzano.
   - Fornisce funzioni per la gestione della memoria, inclusa l'allocazione e la deallocazione di blocchi di memoria con funzioni come **VirtualAlloc** e **VirtualFree.**
   - Fornisce funzioni per la creazione, la terminazione e la gestione dei processi e dei thread, ad esempio con **TerminateProcess**, **ExitProcess**, **GetCurrentProcess, CloseHandle.**
   - Offre funzionalità per la gestione dei file e delle directory, inclusi metodi per la creazione, l'apertura, la lettura, la scrittura e la chiusura dei file con funzioni come **ReadFile, CreateFileA, WriteFile, SetEndOfFile,**
   - Include funzioni per la manipolazione delle stringhe e per le conversioni tra formati di stringa con funzioni tipo **GetEnviromentString**, **GetStringTypeW, LCMapString.**
   - è coinvolta nella gestione delle DLL, tra cui funzioni come **LoadLibraryA**, **GetProcAddress.**

La chiamata di funzione alla locazione di memoria 00401021 `call ds:RegCreateKeyExA` è una funzione delle API di Windows utilizzata per creare o aprire una chiave del Registro di sistema. La "A" alla fine del nome indica che si tratta di una versione ASCII della funzione, il che significa che accetta argomenti e restituisce risultati in formato ASCII (caratteri a 8 bit).

I parametri vengono forniti alla funzione tramite **push** che quindi passa i parametri attraveso lo stack per specificare il comporamento desiderato.

Quindi **push eax** mette il valore del registro nello stack e poi una serie di parametri vengono messi nello stack

Successivamente un altro parametro che rappresenta HKEY_LOCAL_MACHINE viene aggiunto.

L'istruzione **push offset SubKey** spinge l'indirizzo di memoria dell'inizio della stringa nello stack. **SubKey** rappresenta un puntatore alla stringa che specifica il percorso della chiave del Registro di sistema che la funzione **RegCreateKeyExA** deve creare o aprire. Nel nostro caso la stringa è "Software\Microsoft\Windows NT\CurrentVersion"; questo è il percorso della chiave del Registro di sistema sotto la radice **HKEY_LOCAL_MACHINE**.

test eax, eax: Questa istruzione esegue un'operazione logica di AND tra i registri **eax** ed **eax**, generalmente usato per verificare se un valore è zero.

jz short loc_401032: Questa istruzione effettua un salto condizionale corto (**jz**), che salterà a **loc_401032** solo se il risultato del test (**eax**) è zero. Quindi il programma si sposterà a **loc_401032** se la chiamata **RegCreateKeyExA** restituisce zero, indicando che l'operazione ha avuto successo.

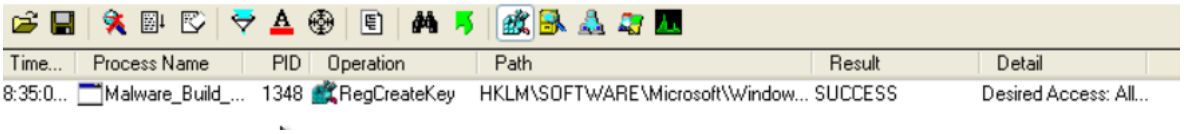
Questo costrutto in C sarebbe

```
if (eax == 0) {
   // Salta a loc_401032
} else {
   // Continua con l'esecuzione del codice successivo
}
```

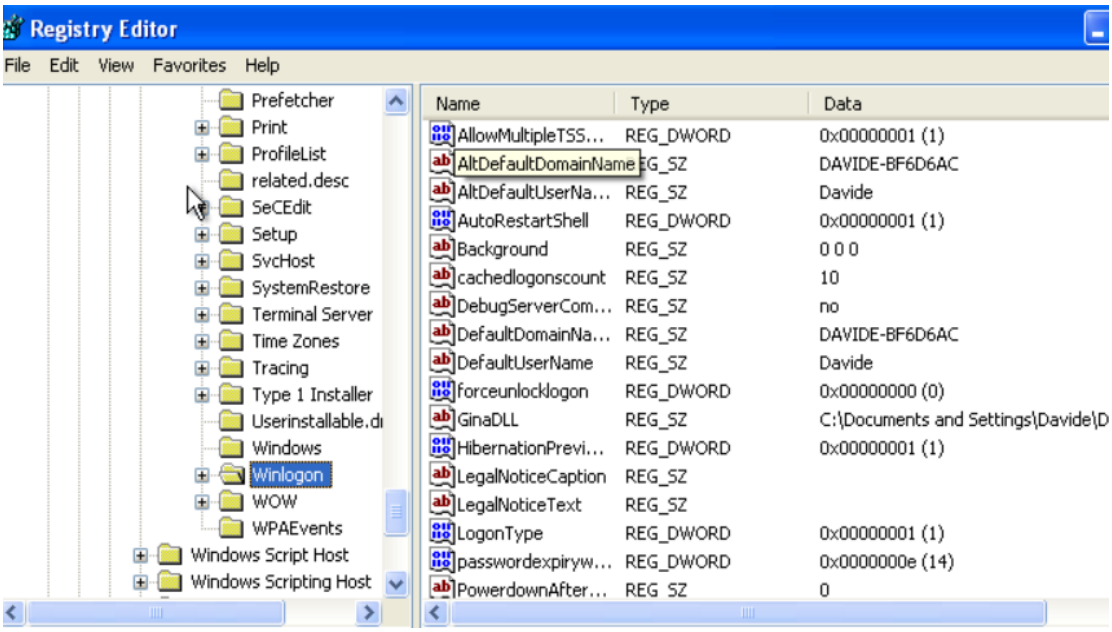
Il valore del parametro **ValueName** è l'indirizzo (offset) di una stringa nella memoria, e il testo di questa stringa è "GinaDLL"; quindi sta dicendo al sistema di impostare un valore di registro con la chiave **hKey** usando la stringa "GinaDLL" come nome del valore.

Analisi Dinamica

Dopo aver settato correttamente i filtri di Procmon lasciando attivi sono quelli del Malware e della creazione di chiave di registro questo è il risultato che otteniamo:

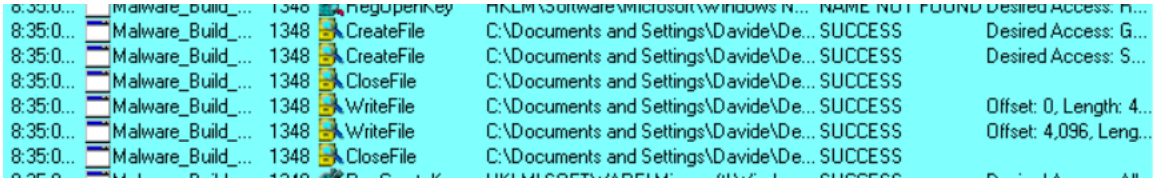


quindi la chiave di registro è stata creata sul path HKLM\SOFTWARE\Microsoft\WindowsNT\CurrentVesrion\Winlogon



Per vedere i valori associati alla chaive di registro creata possiamo usare regedit, a questo punto seguiamo il percorso della chiave di registro e troviamo i valori: 0x00000001 (1), 0x00000001 (1), 0x00000000 (0), 0x00000001 (1), 0x00000001 (1), 0x0000000e (14), 0x00000000 (0), 0x00000000 (0).

Visualizzando invece l'attività del File System possiamo vedere quale chaiamta ha modificato il contenuto della cartella abbiamo infatti l'operazione crea File msgina32.dll sul percorso della cartella e la scrittura del file stesso.



In conclusione unendo i risulati dell'analisi statica e di quella dinamica possiamo dire che il funzionamento del malware è di modificare i permessi di sistema tramite la modifica dei registri di sistema, questo gli permette di poter apportare modifiche al sistema come l' installazione di componenti aggiuntivi, successivamente andrà infatti ad installare una nuova DinamicLoadLibrary (msgina.dll) che potrebbe poi avere vari scopi; come la persistenza, potrebbe essere progettato per iniettare il proprio codice malevolo in processi in esecuzione o in applicazioni specifiche oppure potrebbe creare un file .dll per aggiungere funzionalità dannose al sistema, come keylogging, intercettazione delle comunicazioni di rete.