Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

**1.**

**Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente**

La funzione chiamata alla locazione di memoria 00401021 è **RegCreateKeyExA**.

La funzione RegCreateKeyExA è una funzione della libreria di Windows API utilizzata per creare una nuova chiave o aprire una chiave esistente nel Registro di sistema di Windows. Questa funzione viene spesso utilizzata per manipolare le chiavi nel Registro di sistema da un'applicazione Windows o da un eventuale malware per ottenere la **persistenza** sul sistema.

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

**2.**

**Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente**

I parametri della funzione vengono passati tramite **push** nello **stack** con successiva chiamata alla funzione con **call**.

**3.**

Alla locazione 00401017 viene pushato nello stack l’oggetto **lpSubKey**; esso rappresenta il nome di una sottochiave aperta o creata dalla funzione. La sottochiave specificata deve essere una sottochiave della chiave identificata dal parametro hKey; questo parametro non può essere NULL. Nel caso del nostro malware la chiave del registro da modificare si trova al percorso **"SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Winlogon"**; essa rappresenta una parte specifica del Registro di sistema di Windows che contiene le impostazioni relative al processo di accesso a Windows, noto come Winlogon. Questa chiave contiene vari valori e impostazioni che influenzano il comportamento e l'aspetto del processo di accesso a Windows. In particolare alcuni dei valori che si trovano all’interno di questa chiave riguardano **GINA** (Graphical Identification and Authentication), ossia una componente del sistema operativo Windows che gestisce il processo di identificazione e autenticazione degli utenti; GINA è responsabile per la presentazione dell'interfaccia grafica di accesso a Windows, dove gli utenti inseriscono le proprie credenziali (nome utente e password) per accedere al sistema.

**4.**



In assembly, l'istruzione test viene utilizzata per eseguire un'operazione logica di AND tra due operandi senza memorizzare il risultato, ma impostando solo i flag della CPU in base al risultato dell'operazione. In particolare, test confronta due valori e imposta i flag della CPU in base al risultato dell'AND bit a bit tra i due valori.

Nel caso di **test eax, eax** l'istruzione esegue un'operazione di AND bit a bit tra il registro eax e se stesso, cioè lo stesso registro eax. Questa operazione è equivalente a verificare se il valore di eax è uguale a zero. Se il risultato è zero, allora tutti i bit del registro eax sono impostati a zero. Altrimenti, se il risultato non è zero, almeno uno dei bit nel registro eax è impostato su uno.

Quindi, test eax, eax è spesso utilizzato per verificare se un registro è zero o no, e imposta di conseguenza i flag della CPU, in particolare il **flag Zero (ZF).** Se eax è zero, il flag ZF viene impostato a 1; altrimenti, se eax non è zero, il flag ZF viene impostato a 0.

Dopo l'istruzione test eax, eax, l'istruzione **jz** è un'istruzione di salto condizionale che salta all'indirizzo specificato solo se il flag Zero (ZF) è impostato a 1. jz sta per "**jump if zero**".

Pertanto:

* test eax, eax verifica se il registro eax è zero e imposta il flag Zero (ZF) di conseguenza.
* jz salta all'indirizzo specificato solo se il flag Zero (ZF) è impostato a 1, cioè se il registro eax è zero.

Nel caso eax non sia uguale a zero, l’esecuzione del codice procede con le istruzioni **mov eax,1** (che assegna al registro eax il valore 1) e con un successivo **salto incondizionato (jmp)** all’indirizzo di memoria specificato.

**5.**

Le precedenti righe di codice analizzate rappresentano in C il **costrutto IF**; di seguito la traduzione del codice assembly in C.

If (x==0) {

Esegui il blocco di istruzioni che inizia dall’indirizzo di memoria 00401032

}

else {

Esegui il blocco di istruzioni che inizia dall’indirizzo di memoria 0040107B

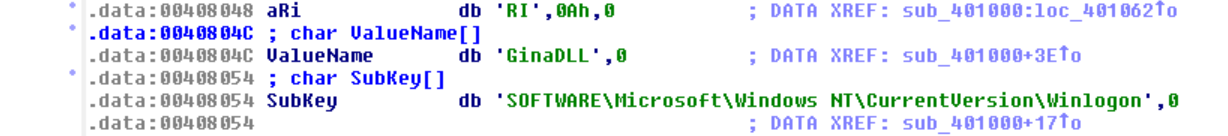
}

**6.**

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Il valore del parametro **ValueName** passato alla funzione **RegSetValueExA** è “GinaDLL”; lo vediamo anche analizzando la porzione di codice in cui viene definito il suo valore tramite l’istruzione **db “GinaDLL”, 0** che indica la definizione di una variabile stringa che contiene i caratteri "GinaDLL".



**7.**

In queste righe di codice analizzate è possibile notare che il malware sta implementando la funzionalità di **persistenza** sul sistema infetto; tramite le funzioni delle API di Windows **RegCreateKeyExA** e **RegSetValueExA** modifica una chiave di registro tramite la quale si sostituisce al servizio legittimo di accesso di Windows chiamato GINA.