S11/L4

Traccia: la figura mostra un estratto del codice di un malware. Identificate: 1. Il tipo di Malware in base alle chiamate di funzione utilizzate. 2. Evidenziate le chiamate di funzione principali aggiungendo una descrizione per ognuna di esse. 3. Il metodo utilizzato dal Malware per ottenere la persistenza sul sistema operativo 4. BONUS: Effettuare anche un'analisi basso livello delle singole istruzioni.

```
.text: 00401010    push eax
.text: 00401014    push ebx
.text: 00401018    push ecx
.text: 0040101C    push WH_Mouse              ; hook to Mouse
.text: 0040101F    call SetWindowsHook()
.text: 00401040    XOR ECX,ECX
.text: 00401044    mov ecx, [EDI]             EDI = «path to
                                              startup_folder_system»
.text: 00401048    mov edx, [ESI]             ESI = path_to_Malware
.text: 0040104C    push ecx                   ; destination folder
.text: 0040104F    push edx                   ; file to be copied
.text: 00401054    call CopyFile();
```

Tipo di malware
Basandosi sulle chiamate di funzione e sull'uso di un hook (SetWindowsHook), potrebbe trattarsi di un keylogger o comunque di un malware progettato per monitorare le attività dell'utente. I keylogger sono un tipo di malware che registra i tasti premuti dall'utente per rubare informazioni sensibili come password e dati di accesso.

Chiamate di funzione principali
push eax, push ebx, push ecx: queste istruzioni inseriscono i valori dei registri eax, ebx, ecx nello stack. Questo è tipicamente fatto per preservare lo stato dei registri prima di una chiamata di funzione.
*push WH_Mous*e: questo impila il valore associato con l'hook del mouse. WH_Mouse è un tipo di hook che monitora gli eventi del mouse.
call SetWindowsHook(): questa funzione installa un hook di procedura per monitorare il sistema. In un keylogger, potrebbe essere usato per intercettare e registrare i tasti premuti o le azioni del mouse.
XOR ECX,ECX: azzera il registro ECX utilizzando l'operazione XOR.
mov ecx, [EDI]: carica il valore puntato da EDI nel registro ECX. Qui EDI potrebbe essere l'indirizzo del percorso di avvio.
mov edx, [ESI]: carica il valore puntato da ESI nel registro EDX. ESI potrebbe puntare al percorso del malware.
push ecx, push edx: queste istruzioni impilano i valori di ECX e EDX (i percorsi) nello stack prima di chiamare CopyFile().

call CopyFile(): una funzione che copia un file da un percorso sorgente a un percorso di destinazione. Questo è spesso usato da malware per copiarsi in una nuova posizione.

Metodo di persistenza
Il malware sembra utilizzare la funzione SetWindowsHook() per ottenere la persistenza. Installando un hook, il malware può rimanere attivo e monitorare le azioni dell'utente dopo ogni riavvio del sistema. Inoltre, il codice indica l'uso del percorso startup_folder_system (anche se il percorso specifico non è visibile), suggerendo che il malware si copia in una cartella di avvio del sistema per assicurarsi di essere eseguito ad ogni avvio del sistema operativo.

Analisi a basso livello
push eax/bx/cx: preservazione dello stato dei registri per le chiamate di funzione.
call SetWindowsHook(): installazione di un hook per intercettare gli eventi del sistema.
XOR ECX,ECX: azzeramento del registro per la preparazione a un'operazione.
mov ecx/edx, [EDI/ESI]: caricamento dei percorsi da puntatori in registri.
call CopyFile(): copia del file malware in una nuova posizione per la persistenza.
Questa analisi è basata sull'ipotesi che EDI e ESI siano stati precedentemente impostati per puntare ai percorsi di interesse. L'operazione di copia del file suggerisce un tentativo di auto-replicazione in un percorso che garantirà l'esecuzione del malware all'avvio del sistema.