MALWARE ANALYSIS

Analisi statica

-Quanti parametri sono passati alla funzione Main()?

I parametri che possiamo notare essere passati alla funzione Main() sono: envp, argv e argc.

Data e hModule possono essere sia variabili che parametri.

-Quante variabili sono dichiarate all’interno della funzione Main()?

Le variabili all’interno della funzione sono: var\_8 e var\_4.

Data e hModule possono essere sia variabili che parametri.

-Quali sezioni sono presenti all’interno del file eseguibile? Descrivete brevemente almeno 2 di quelle identificate.

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

Le sezioni presenti nel file sono 4, ovvero: .text, .rdata, .data, .rsrc

La sezione **.rdata** è una sezione di dati leggibili durante l'esecuzione di un programma.

La designazione rdata sta per read-only data. In un file eseguibile, questa sezione è destinata a contenere dati costanti o inizializzati che non devono essere modificati durante l'esecuzione del programma.

La sezione **.rsrc** in un file eseguibile, incluso un malware, è comunemente utilizzata per memorizzare risorse del programma, come icone, immagini, stringhe localizzate, e altri dati che non sono codice eseguibile ma sono necessari per il corretto funzionamento dell'applicazione.

-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 sono:  **- ADVAPI32.DLL** che è una libreria di Windows che fornisce un'interfaccia per l'Advanced Windows 32 Base API. Questa libreria contiene funzioni che gestiscono vari aspetti avanzati del sistema operativo Windows, inclusi servizi, sicurezza e registro di sistema.

**- Kernel32.dll** è una libreria fondamentale del sistema operativo Windows. Contiene numerose funzioni di basso livello che forniscono servizi essenziali per l'esecuzione di programmi su piattaforma Windows.

1. **Kernel32.dll:**
   * **CreateProcess**: Il malware potrebbe utilizzare questa funzione per avviare nuovi processi, consentendo, ad esempio, l'esecuzione di ulteriori componenti o il caricamento di file dannosi.
   * **CreateFile**, **WriteFile**: Queste funzioni potrebbero indicare che il malware esegue operazioni di creazione e scrittura su file, come la manipolazione o la diffusione di payload.
2. **ADVAPI32.dll:**
   * **RegCreateKey**, **RegSetValue**: Queste funzioni suggeriscono l'interazione con il Registro di sistema di Windows, il che potrebbe indicare che il malware modifica le impostazioni di sistema o le sue proprie configurazioni.

MALWARE ANALYSIS

-Lo scopo della funzione chiamata alla locazione di memoria 00401021:

Nella posizione di memoria designata, troviamo la funzione "RegCreateKeyExA", 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:

I parametri vengono passati tramite PUSH

-Che oggetto rappresenta il parametro alla locazione 00401017:

Il parametro presso l'indirizzo di memoria 00401017, denominato "SubKey", rappresenta un puntatore che si riferisce a una stringa contenente il percorso nel Registro di sistema.

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

**TEST EAX, EAX** è un'istruzione dell'assembly x86/x86-64. In questa istruzione, **TEST** è un'operazione logica che fa un'AND bit a bit tra i registri **EAX** e se stesso (**EAX**). Questa istruzione non memorizza il risultato, ma imposta i flag del registro dei flag dell'architettura x86/x86-64 in base al risultato dell'operazione.

In particolare, l'istruzione **TEST EAX, EAX** è spesso utilizzata per testare se il registro **EAX** è zero. Se **EAX** è zero, allora il flag di zero (**ZF**) sarà impostato a 1; altrimenti, sarà a 0.

Questa istruzione è comunemente utilizzata in condizioni di controllo di flusso, ad esempio nei confronti condizionali, dove si vuole verificare se un registro è zero o meno prima di decidere quale ramo dell'esecuzione del programma seguire.

L'istruzione assembly **JE SHORT Malware\_.00401032** rappresenta un salto condizionale. In particolare:

**JE** sta per "Jump if Equal", che significa "salta se uguale".

**SHORT** indica che l'istruzione di salto è un salto breve, che può coprire solo un intervallo limitato di indirizzi.

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

if (eax == 0) {

// Codice da eseguire se EAX è zero

} else {

// Codice da eseguire se EAX non è zero

}

-Valutate ora la chiamata alla locazione 00401047, qual è il valore del parametro <<ValueName>>?

Il Valore di tale parametro è “GinaDLL”

ANALISI DINAMICA

-Cosa Notate all’interno della cartella dove è situato l’eseguibile del malware?

All’interno della cartella troviamo un ulteriore file: “msgina32.dll”

-Quale chiave di registro viene creata?

Vengono create due chiavi di registro: “GinaDLL” (sottochiave) e “Winlogon” (chiave)

-Quale valore viene associato alla chiave di registro creata?

Il valore associato a questa chiave è:

“C:\Documents andSettings\Administrator\Desktop\Build\_Week\_Unit\_3\msgina32.dll”

-Quale chiamata ha modificato il contenuto della cartella dove è presente l’eseguibile del malware?

La modifica del contenuto della cartella dove è posizionato l’seguibile del malware è stata effettuata da “CreateFile”.

-Unite tutte le Informazioni raccolte fin qui sia dall’analisi statica che adall’analisi dinamica per delineare il funzionamento del malware.

Il sistema sembra essere compromesso da un malware di tipo DROPPER, evidenziato dalle modifiche al Registro di Sistema e l'intento di stabilire una persistente presenza sulla macchina per l'esecuzione automatica al boot. La creazione del file "msgina32.dll" e le modifiche alla sottochiave di registro "GinaDll" indicano che il malware mira ad integrare un componente nel processo di autenticazione di Windows attraverso il file da esso creato.