## D6 Progetto

Immagine che contiene testo, schermata, Carattere, documento

Descrizione generata automaticamente

Immagine che contiene testo, schermata, software, schermo

Descrizione generata automaticamente

Immagine che contiene testo, schermata, software, schermo

Descrizione generata automaticamente

Immagine che contiene testo, elettronica, schermata, schermo

Descrizione generata automaticamente

Immagine che contiene testo, elettronica, schermata, schermo

Descrizione generata automaticamenteImmagine che contiene testo, elettronica, schermata, schermo

Descrizione generata automaticamente

Per analizzare il malware Build\_Week\_Unit\_3 presente sul desktop ho utilizzato IDA, uno strumento software utilizzato dagli analisti di sicurezza informatica e dai ricercatori per analizzare il codice eseguibile, come i file binari dei programmi. Questo strumento permette di visualizzare e analizzare il codice assembler di un programma, aiutando a comprendere il funzionamento interno del software.

Per fare l’analisi statica del Malware, ho aperto il malware e ho utilizzato la funziona functions per trovare i parametri e le variabili.

Possiamo procedere a identificare i parametri e le variabili utilizzando una logica basata sui loro valori. In particolare, notiamo che le variabili assumono valori negativi, mentre i parametri hanno valori positivi.

* 4 variabili: hModule, var\_8, var\_4, data
* 3 parametri: argc, argv, envp

Questa logica si basa sulla considerazione che, in un contesto assembly, le variabili assumono valori negativi, mentre i parametri spesso contengono valori positivi. Tale distinzione è utile per identificare e distinguere tra variabili e parametri nella funzione Main.

Con il programma CFF Explorer, uno strumento software utilizzato principalmente nel campo della sicurezza informatica e del reverse engineering, che consente di esplorare e analizzare i file seguibili, fornendo informazioni dettagliate sulla loro struttura interna.

Immagine che contiene testo, elettronica, schermata, schermo

Descrizione generata automaticamente

Nella sezione Section Header si può vedere all’interno del malware: .text/ .rdata / .data / .rsrc

La sezione .text di un file eseguibile è cruciale perché contiene il codice eseguibile del programma. In pratica, questa sezione rappresenta la sequenza di istruzioni che il sistema dovrà seguire per eseguire correttamente il programma. Senza di essa, il software non sarebbe in grado di svolgere nessuna operazione.

La sezione .data è altrettanto importante, poiché contiene dati globali e variabili che possono essere modificati durante l’esecuzione del programma. Questi dati includono variabili globali, informazioni dinamiche e altri dettagli necessari per il corretto funzionamento del programma.

La sezione .rdata è significativa poiché conserva dati di sola lettura utilizzati dal programma durante l’esecuzione. Questi dati possono includere stringhe di testo e altre informazioni che non devono essere alterate durante l’esecuzione del programma. In sostanza, .rdata fornisce una sorta di archivio per informazioni fondamentali e costanti.

Infine, la sezione .rsrc è essenziale per la gestione delle risorse del programma. Al suo interno, si trovano dati come immagini, icone e altri elementi grafici. La sua funzione principale è quella di fornire un’interfaccia utente completa e migliorare l’aspetto visivo del programma, contribuendo a una user experience più ricca.

In sintesi, le diverse sezioni di un file eseguibile svolgono ruoli chiave: .text per il codice eseguibile, .data per i dati modificabili, .rdata per i dati di sola lettura e .rsrc per le risorse grafiche del programma. Queste sezioni collaborano per consentire al software di funzionare correttamente e offrire un’esperienza utente completa.

Immagine che contiene testo, schermata, software, schermo

Descrizione generata automaticamente

Per vedere quali librerie sono importante dal Malware, sempre con CFF explorer clicco su import directory, e trovo due componenti fondamentali: KERNEL32.dll e ADVAPI32.dll. Entrambe queste librerie sono entrambe fondamentali nel contesto del sistema operativo Windows e sono parte integrante delle piattaforma Windows API (Application Programming Interface). Queste librerie forniscono una vasta gamma di funzionalità di sistema utilizzate da molte applicazioni Windows.

* KERNELL32.dll:

Questa libreria è di vitale importanza per il kernel del sistema operativo Windows, offrendo funzionalità a basso livello che comprendono gestione della memoria, le operazioni sui file, la supervisione dei processi e delle thread, e molto altro ancora. All’interno di KERNELL32.dll, si trovano funzioni fondamentali come la gestione dei file (apertura, lettura, scrittura), la gestione della memoria (allocazione, deallocazione) e operazioni di base del sistema. Esaminando le informazioni interne con CFF Explorer, è possibile approfondire la comprensione di queste funzioni specifiche e di come vengono sfruttate dai programmi.

* ADVAPI32.dll:  
  Questa libreria, indicata dalla sigla “ADV” per “Advanced”, offre funzionalità avanzate relative alla sicurezza, alla gestione degli account e ai servizi di sistema. Tra le sue molteplici responsabilità, ADVAPI32.dll si occupa della gestione delle autorizzazioni, degli account utente e dei servizi Windows, oltre a comprendere funzioni di crittografia e gestione degli eventi di sicurezza. Fondamentale per l’iterazione con il Registro di sistema, per l’impostazione delle autorizzazioni di sicurezza e per l’autenticazione degli utenti, ADVAPI32.dll contribuisce in modo cruciale alla sicurezza e al funzionamento stabile del sistema operativo.

In sintesi, queste librerie rappresentano pilastri fondamentali di Windows, fornendo funzionalità avanzate e svolgendo ruoli chiave nella gestione di operazioni di sistema, dalla sicurezza alla gestione delle risorse.

Immagine che contiene testo, elettronica, schermata, schermo

Descrizione generata automaticamente



Immagine che contiene testo, elettronica, schermata, schermo

Descrizione generata automaticamente



L’analisi delle librerie suggerisce che il malware potrebbe essere un dropper, ovvero un tipo di malware progettato per scaricare e installare ulteriori componenti dannosi sul sistema. Questa ipotesi si basa su alcune funzioni sospette all’interno delle librerie, come CreateFile e WriteFile.  
La funzione CreateFile è nota per la sua capacità di creare o aprire file, mentre WriteFile consente di scrivere o modificare il contenuto di file già aperti. Questo suggerisce la possibilità che il malware stia cercando di salvare sé stesso o altri componenti dannosi su disco.

Immagine che contiene testo, elettronica, schermata, schermo

Descrizione generata automaticamente



Ulteriori indizi di un comportamento di dropper emergono analizzando attentamente le funzioni FindResourceA, LoadResource e LockResource nel contesto del programma analizzato. Queste funzioni sono spesso indicative di un processo in cui il malware cerca di manipolare risorse presenti nei programmi Windows, come immagini o altri file eseguibili.

1. FindResourceA:  
   La funzione FindResourceA viene impiegata per individuare risorse all’interno degli eseguibili. Ad esempio, può essere utilizzata per individuare un file con estensione .dll tra le risorse del programma. Questa fase è cruciale per il dropper in quanto cerca risorse specifiche per eseguire operazioni successivamente.
2. LoadResource:  
   LoadResource è coinvolto nel caricamento di una risorsa di memoria. Questa funzione prende un handle a una risorsa, restituito da FindResourceA, e restituisce un handle di blocco di memoria contenente la risorsa. Questo passaggio è spesso cruciale quando il dropper intende caricare nella memoria del sistema una risorsa specifica identificata in precedenza.
3. LockResource:  
   Per accedere ai dati della risorsa caricata in memoria, è necessario utilizzare la funzione LockResource. Questo passaggio consente al dropper di ottenere un puntatore al blocco di memoria che contiene la risorsa, consentendo ulteriori manipolazioni o utilizzo di tali dati per il suo scopo maligno.
4. SizeofResource:  
   Tra le altre funzioni coinvolte nella manipolazione delle risorse, vi è anche SizeofResource, che restituisce la dimensione in byte di una risorsa specifica. Questa informazione può essere fondamentale per il dropper, poiché fornisce una stima delle dimensioni della risorsa, consentendo una gestione adeguata della memoria durante il processo di caricamento.

In sintesi, le funzioni analizzate suggeriscono che il malware potrebbe agire come un dropper, focalizzato sull’individuazione, il caricamento e la manipolazione di risorse all’interno dei programmi Windows. Un’analisi dettagliata di queste operazioni è essenziale per ottenere una comprensione approfondita degli obbiettivi e delle tattiche specifiche adottate dal dropper nel contento in cui opera.  
Il malware sembra sfruttare tali funzioni per identificare risorse, come file.dll, presenti negli eseguibili. Questo comportamento è tipico di un dropper, il quale cerca di localizzare risorse specifiche all’interno del sistema. Una volta individuata la risorsa, il dropper procede al caricamento in memoria attraverso l’uso di LoadResource, ottenendo un handle e creando un blocco di memoria contenente la risorsa stessa.  
Questa manipolazione delle risorse può indicare una strategia del dropper finalizzata a depositare o eseguire ulteriori componenti dannosi sul sistema. L’approfondimento dell’analisi potrebbe rilevare la presenza di una seconda fase di attacco o il download di payload aggiuntivi, aspetti che vanno esaminati attentamente per comprende appieno la minaccia e adottare misure adeguate di mitigazione.

Immagine che contiene testo, schermata, schermo, numero

Descrizione generata automaticamente

Le funzioni di ‘RegSetValueExA’ e ‘RegCreateKeyExA’ sono due delle principali funzioni della libreria ADVAPI32.dll e sono utilizzate per manipolare il Registro di sistema di Windows.

1. RegSetValueExA:

Questa viene utilizzata per impostare il valore di una chiave di registro esistente o crearne uno nuovo. La “Ex” nel nome indica che si tratta di una versione estesa, in grado di gestire dati di tipo variabile (come stringhe, numeri interi, binari, ecc..).   
Viene utilizzata quando il malware o un’applicazione legittima desidera assegnare un valore specifico a una chiave di registro esistente o crearne una nuova. Ad esempio, potrebbe essere utilizzato per impostare configurazioni, percorsi dei file, o informazioni di stato.

1. RegCreateKeyExA:

Questa funzione è utilizzata per creare o aprire una chiave di registro. La “Ex” nel nome indica, anche in questo caso, una versione estesa in grado di gestire operazioni più avanzate. Viene utilizzata quando il malware o un’applicazione vuole creare una nuova chiave di registro o aprire una chiave esistente per eseguire operazioni successive. Ad esempio, può essere utilizzo per stabilire un percorso nel Registro di sistema in cui il malware memorizzerà o cercherà informazioni.

Entrambe queste funzioni sono fondamentali per le operazioni di manipolazione del Registro di sistema e sono spesso coinvolte in attività maligne o dannose, come nel caso di un malware che cerca di ottenere persistenza, nascondere le proprie tracce o influenzare la configurazione del sistema. La loro presenza all’interno di ‘ADAVPI32.dll’ sottolinea l’importanza di questa libreria nella gestione della sicurezza e delle configurazioni di sistema avanzate.

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente

Locazione di Memoria 00401201 – funzione RegSetValueExA:  
In questa locazione di memoria, si trova la chiamata alla funzione ‘RegSetValueExA’. Questa funzione ha la responsabilità di creare o aprire una chiave di registro. I parametri necessari vengono passati tramite l’istruzione ‘PUSH’, che mette i valori nello stack prima di chiamare la funzione.

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente

Funzione 0401017 – Offset Subkey – RegCreateKeyExA:  
L’offset SubKey specifica o identifica la chiave di registro e, in questo caso sembra correlato alla funzione ‘RegCreateKeyExA’. Questa funzione è comunemente utilizzata per creare una nuova chiave di registro o aprire una chiave esistente.

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente

Istruzioni Test eax, eax e jz short loc\_401032:  
‘test’ eax, eax’ esegue AND logico del registro EAX con sé stesso, influenzando il flag Zero (ZF) nei registri EFFLAGS. Se EAX è zero, ZF diventa 1.

‘jz short loc\_401032’ è un salto condizionale che si verifica solo se ZF è 1, portando l’esecuzione a loc\_401032. In altre parole, questa istruzione salta a loc\_401032 solo se il risultato del test precedente è che EAX è zero

Traduzione in C:

int main() {

int a; // EAX

int b; // ECX

int c; // [EBP + cbData]

if (a == 0) {

b = c;

} else {

a = 1;

}

return 0;

}

Il codice C tradotto riflette l’andamento delle istruzioni assembly. Se il valore iniziale di EAX è zero, allora ECX assume il valore di ‘c’, altrimenti, ‘a’ viene impostato a 1.

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

Riferimento a RegSetValueExA – Valore di ValueName:  
L’indirizzo di memoria 00401047 sembra fare riferimento alla funzione ‘RegSetValueExA’. Salendo nel codice, si nota che il valore di ‘ValueName’ è “GinaDLL”. Questo parametro specifica il valore o il nome della chiave di registro che la funzione ‘RegSetValueEax’ sta cercando di impostare o creare.

Questi dettagli forniscono un’analisi più approfondita del codice assembly, inclusi i riferimenti alle funzioni di registro e alle operazioni condizionali.

Immagine che contiene testo, schermata, algebra, ricevuta

Descrizione generata automaticamente

Per condurre un’analisi dinamica, userò Process Monitor, uno strumento per Windows che fornisce un dettagliato resoconto in tempo reale delle attività di sistema, inclusi i processi in esecuzione.  
Dopo l’avvio di Process Monitor, comparirà una schermata iniziale che consente di impostare i primi filtri, i quali possono essere successivamente modificati.

Immagine che contiene testo, elettronica, schermata, software

Descrizione generata automaticamente

Al momento dell’avvio iniziale di Process Monitor, il computer risultava in uno stato normale ora, avviando il malware ne potrò osservare il comportamento. Durante questa fase Process Monitor catturerà e registrerà le attività del sistema, consentendomi di esaminare le interazioni del malware con il sistema operativo.

Immagine che contiene testo, schermata, software, Icona del computer

Descrizione generata automaticamenteImmagine che contiene testo, schermata, Viso umano, software

Descrizione generata automaticamente

Una volta aperto il malware mi comparirà una nuova icona, un nuovo file di nome msgina32.dll

Immagine che contiene testo, elettronica, schermata, software

Descrizione generata automaticamente

Ho impostato i filtri per definire il tipo di ricerca, e ho trovato delle attività come RegCreateKey e RegSetValue, che indicano rispettivamente la creazione e la modifica delle chiavi di registro. L’analisi di queste attività ha rivelato operazioni insolite, con particolare attenzione alle chiavi di registro HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon. Questa chiave di registro, situata in HKLM (HKEY\_LOCAL\_MACHINE), è stata soggetta a modifiche da parte del malware.

Immagine che contiene testo, schermata, software, numero

Descrizione generata automaticamente

Esaminando più da vicino le specifiche, ho notato che il malware ha creato una nuova chiave di registro all’interno del percorso. Allo stesso modo, l’analisi ha evidenziato che il malware ha impostato un nuovo valore per questa chiave di registro, utilizzando il nome del file precedentemente creato, ovvero msgina32.dll.

Passando all'analisi delle attività del file system, ho notato che il malware ha effettuato operazioni di creazione e modifica di file, sempre con il nome msgina32.dll. Questo conferma che il malware ha non solo manipolato le chiavi di registro ma anche creato o modificato file specifici, contribuendo a una comprensione più approfondita delle azioni dannose intraprese. In base all’analisi dinamica con Process Monitor, è possibile affermare con una certa sicurezza che il malware ha un obbiettivo preciso: influenzare le configurazioni del sistema che infetta. Lo fa manipolando la chiave di registro attraverso un altro malware denominato msgina32.dll. Questo file malizioso è responsabile di apportare modifiche significative alla chiave di sistema GINADLL.

È importante notare che gina.dll, è un file utilizzato in sistemi operativi Windows precedenti a Windows Vista, che gestiva il processo di autenticazione degli utenti durante i login. Modificando questo file, il malware rischia di compromettere la sicurezza del sistema, aprendo la possibilità di accessi non autorizzati o indesiderati. In sostanza, l’obbiettivo sembra quello di sfruttare la manipolazione della chiave di sistema e del file gina.dll per compromettere l’autenticazione degli utenti, un’azione che può portare a gravi rischi per la sicurezza del sistema.