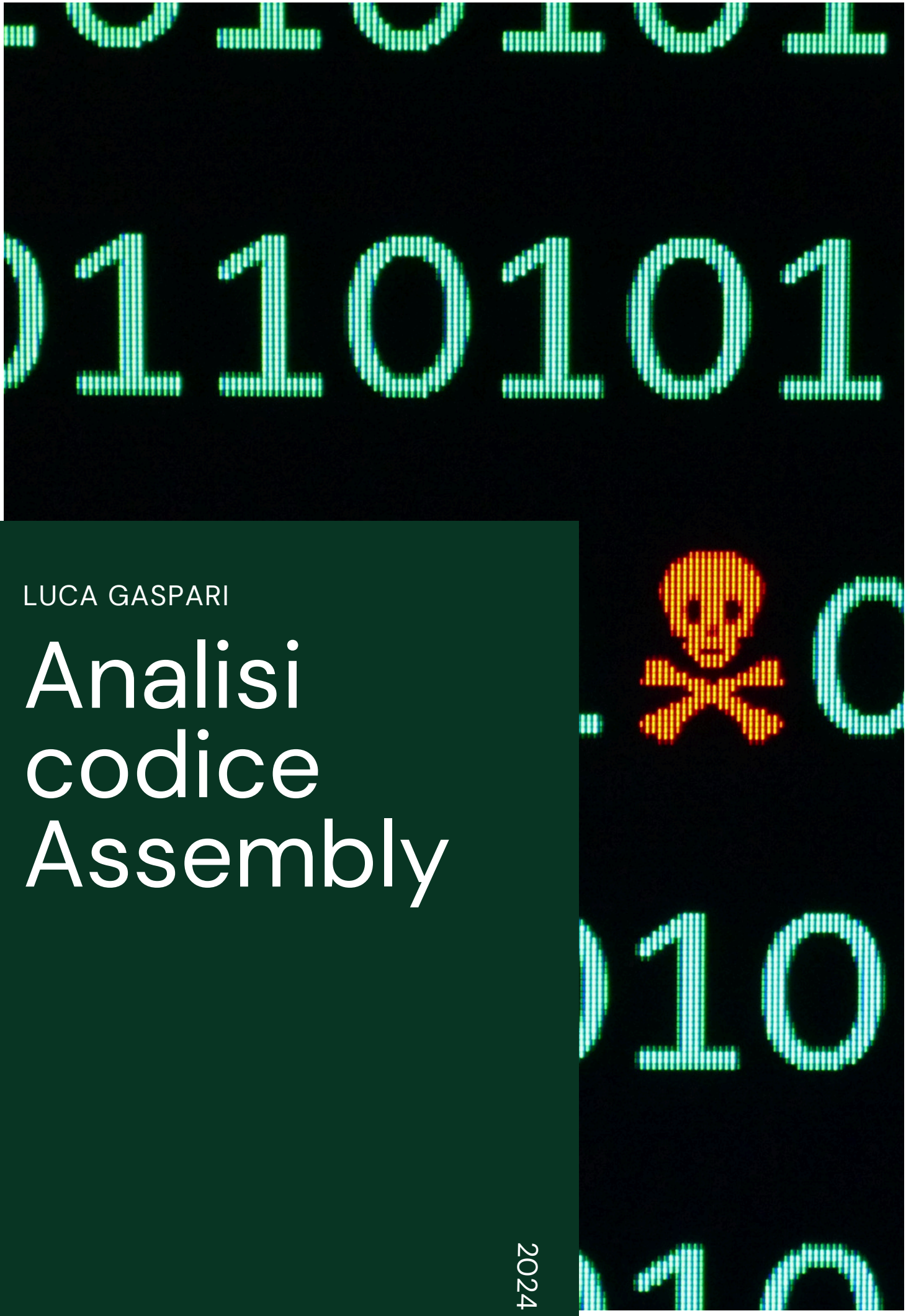
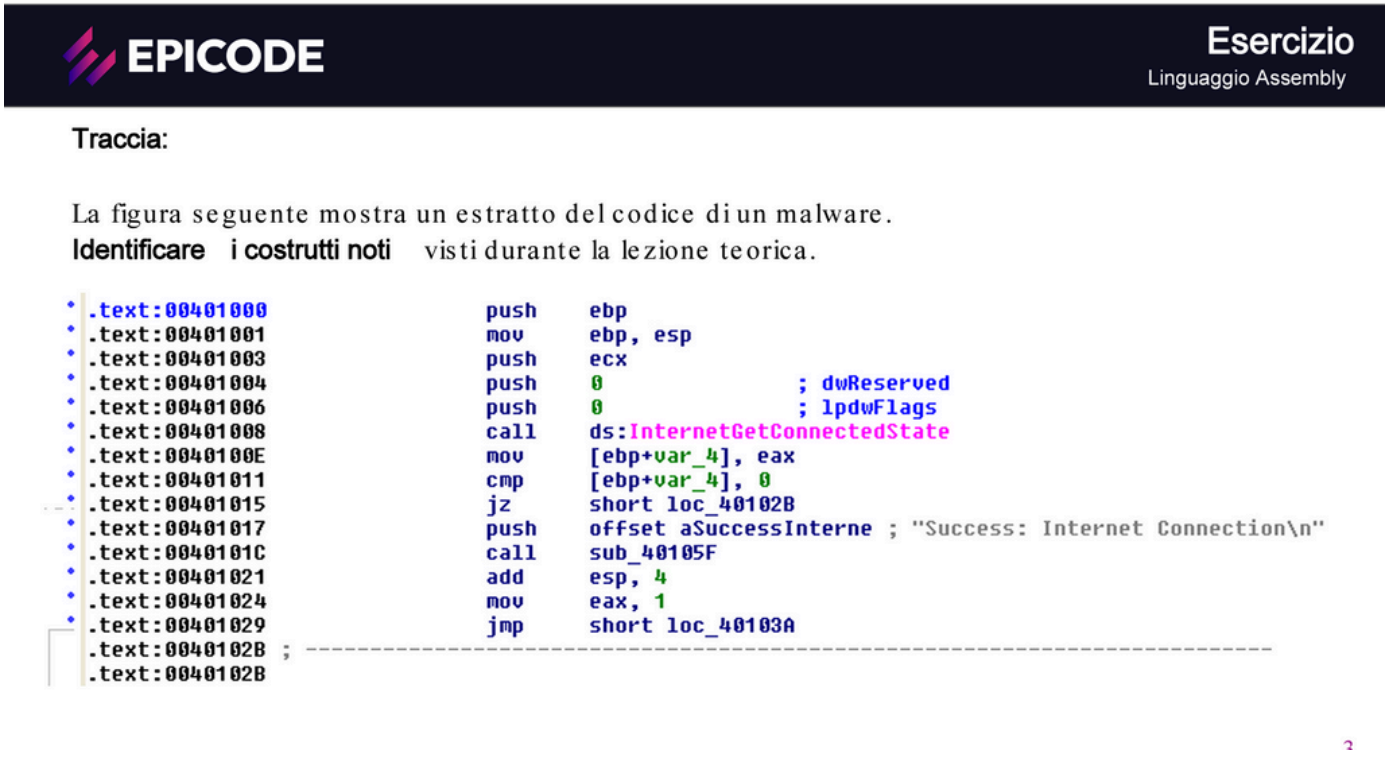


LUCA GASPARI

Analisi codice Assembly

2024

Traccia esercizio S10L4



La traccia dell'esercizio richiede l'analisi di un estratto di codice assembly di un malware, identificando i costrutti noti, ipotizzando la funzionalità implementata e (opzionale) spiegando ogni singola riga di codice.

Identificazione dei costrutti noti

Chiamata di funzione:
La funzione **InternetGetConnectedState** viene chiamata per verificare la connessione Internet, il risultato della verifica viene memorizzato nel registro **eax.**

La funzione **sub_40105F** viene chiamata con un argomento, che è l'offset della stringa "Success: Internet Connection\n".
Questa è una sottoroutine definita nel codice (non mostrata nel dettaglio), presumibilmente, gestisce o registra il messaggio di successo della connessione Internet.

If condizionale: Il costrutto cmp [ebp+var_4], 0 seguito da jz short loc_40102B implementa una logica condizionale simile a un'istruzione if in linguaggi di alto livello, se la condizione è vera (cioè, se var_4 è uguale a 0), il codice esegue un salto all'etichetta loc_40102B.

Salvataggio e ripristino dello stack: I registri vengono salvati e ripristinati usando le istruzioni **push** e **mov**.

Ipotesi della funzionalità e dell'esecuzione

Analizzando questo estratto di codice di un malware sembra che abbia lo scopo di verificare se il sistema ha accesso a Internet utilizzando la funzione InternetGetConnectedState, in caso di connessione, esegue una procedura che probabilmente registra un messaggio di successo e imposta il registro eax a 1 per indicare un risultato positivo.

Spiegazione dettagliata del codice

push ebp:
• Salva il valore corrente del base pointer (ebp) sullo stack, questo è fatto per preservare il valore di ebp prima di modificarlo.

mov ebp, esp
• Imposta il base pointer (ebp) al valore corrente dello stack pointer (esp), creando un nuovo frame dello stack. Questo è l'inizio della creazione di un nuovo frame di stack per la funzione.

push ecx
• Salva il valore del registro ecx sullo stack, questo è fatto per preservare il valore di ecx che verrà utilizzato successivamente.

push 0
• Salva il valore 0 sullo stack, questo rappresenta il parametro dwReserved passato alla funzione InternetGetConnectedState.

push 0
• Salva il valore 0 sullo stack, questo rappresenta il parametro lpdwFlags passato alla funzione InternetGetConnectedState.

call ds:InternetGetConnectedState
• Chiama la funzione InternetGetConnectedState, che verifica lo stato della connessione Internet, i risultati della funzione vengono restituiti nel registro eax.

mov [ebp+var_4], eax
• Salva il valore di ritorno della funzione (contenuto in eax) in una variabile locale (var_4), questo memorizza lo stato della connessione Internet.

cmp [ebp+var_4], 0
• Confronta il valore della variabile locale var_4 con 0, questo controllo determina se il sistema è connesso a Internet.

jz short loc_40102B
• Se il valore della variabile locale var_4 è 0 (non connesso a Internet), salta all'indirizzo loc_40102B, che gestisce il caso di mancata connessione.

push offset aSuccessInterne
• Salva l'offset della stringa "Success: Internet Connection\n" sullo stack. Questa stringa viene passata come argomento alla sottoroutine sub_40105F.

call sub_40105F
• Chiama una sottoroutine a sub_40105F, questa sottoroutine presumibilmente registra o gestisce il messaggio di successo della connessione Internet.

add esp, 4
• Aggiusta lo stack pointer (esp) per pulire l'argomento passato alla sottoroutine. Questo è fatto per mantenere l'integrità dello stack dopo la chiamata alla funzione.

mov eax, 1
• Imposta il registro eax a 1, questo indica un successo nell'operazione.

jmp short loc_40103A
• Salta all'indirizzo loc_40103A, che probabilmente è l'epilogo della funzione, questo permette di uscire dalla funzione in modo ordinato.