Analisi Malware

INDICE

TRACCIA ESERCIZIO

Con riferimento al file Malware_U3_W2_L5 presente all'interno della cartella «Esercizio_Pratico_U3_W2_L5» sul desktop della macchina virtuale dedicata per l'analisi dei malware, rispondere ai seguenti quesiti:

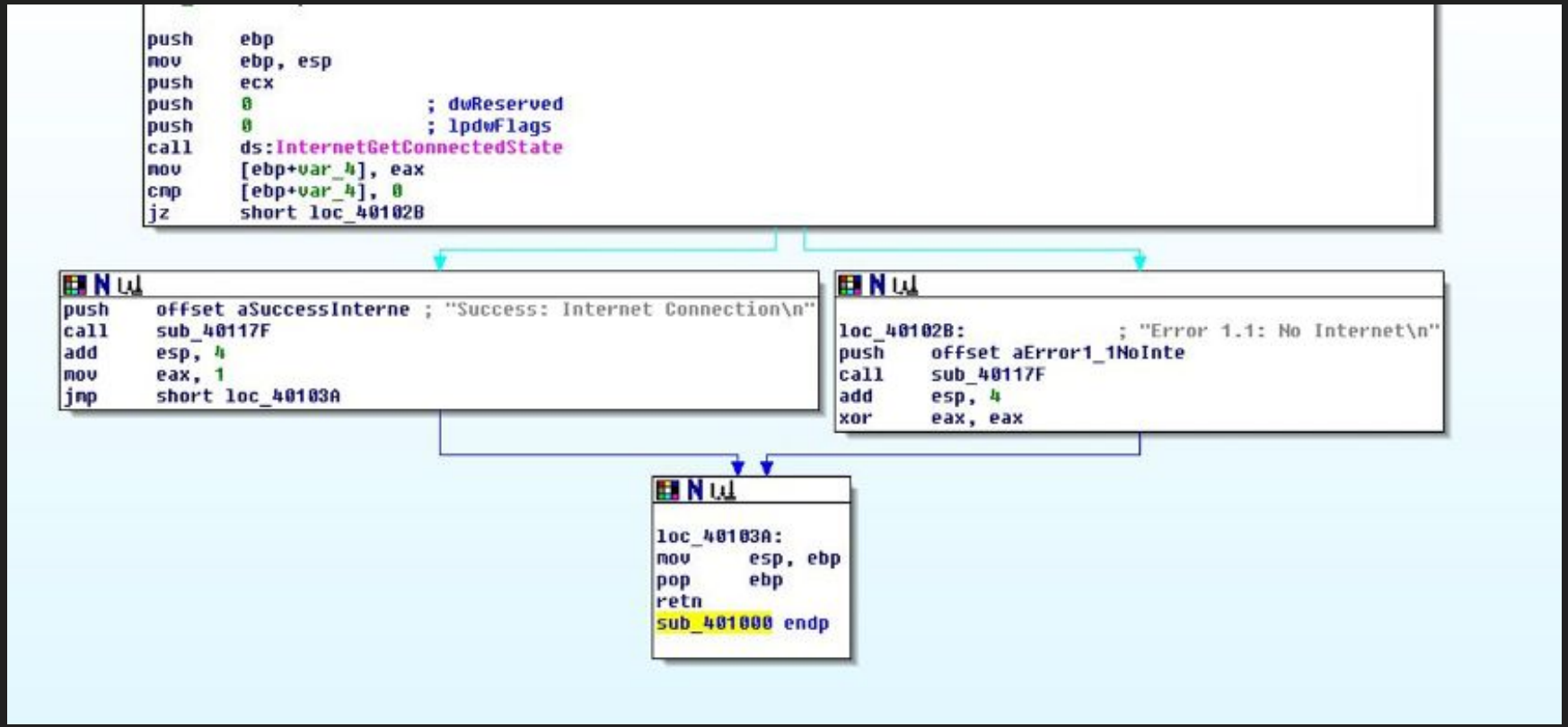
Quali librerie vengono importate dal file eseguibile?

Quali sono le sezioni di cui si compone il file eseguibile del malware?

Con riferimento alla figura in slide 4, risponde ai seguenti quesiti:

Identificare i costrutti noti (creazione dello stack, eventuali cicli, costrutti)
Ipotizzare il comportamento della funzionalità implementata

TRACCIA ESERCIZIO

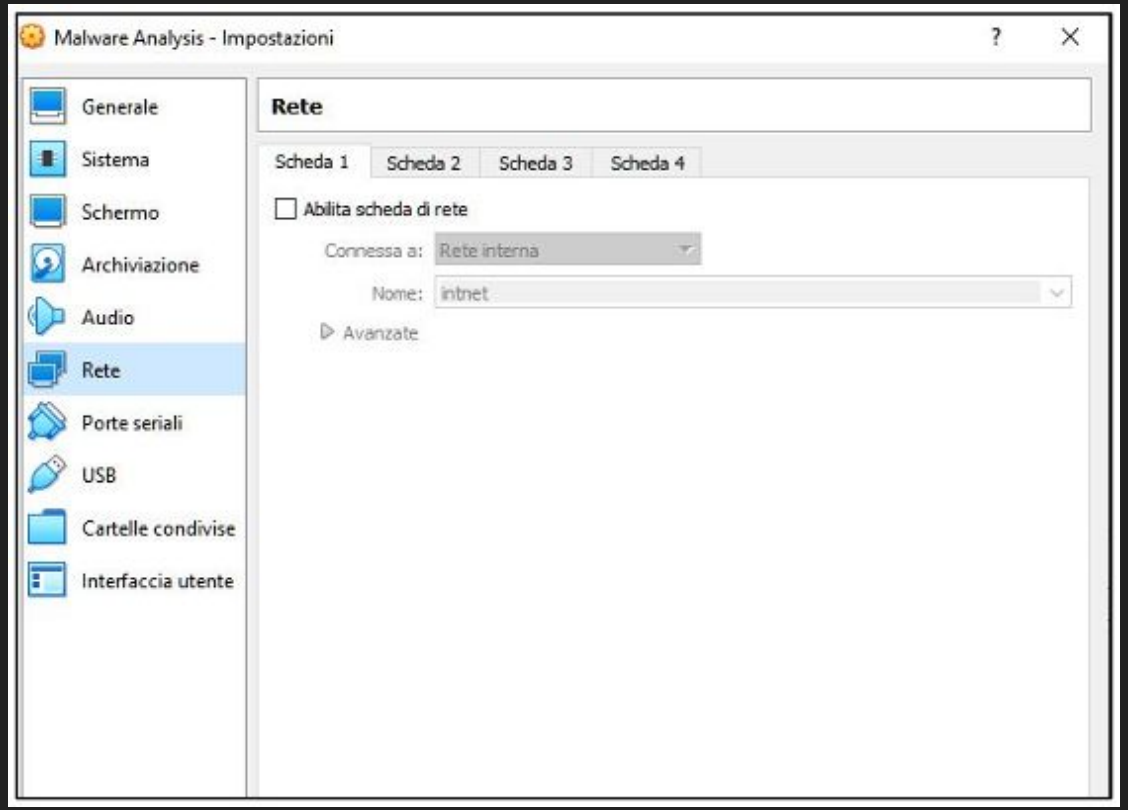


INTRODUZIONE

Prima di svolgere le richieste dell’esercizio è **NECESSARIO** lavorare in sicurezza, per questo l’esercizio andrà svolto su una macchina virtuale installata sul nostro PC ma totalmente isolata da esso, in modo che i malware testati non si propaghino e causino danni.

INTRODUZIONE

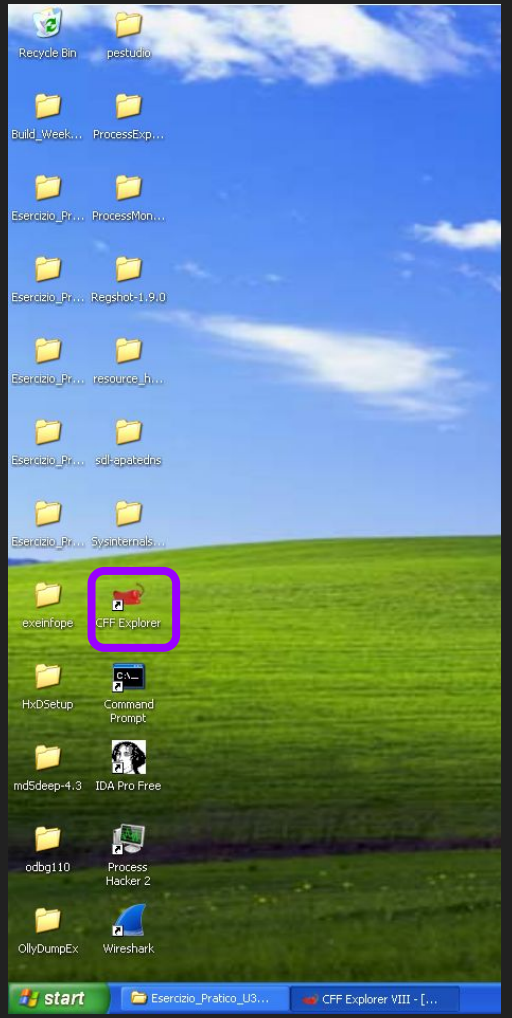
Aprendo VirtualBox ci assicuriamo che non ci siano schede di rete abilitate e che anche tutti gli altri metodi di ingresso (esempio USB) siano disabilitati



INTRODUZIONE

Ora che tutto è stato isolato possiamo avviare il software che ci consentirà di analizzare staticamente il malware ovvero CFF Explorer, un software che permette di controllare le funzioni importate ed esportate da un malware.

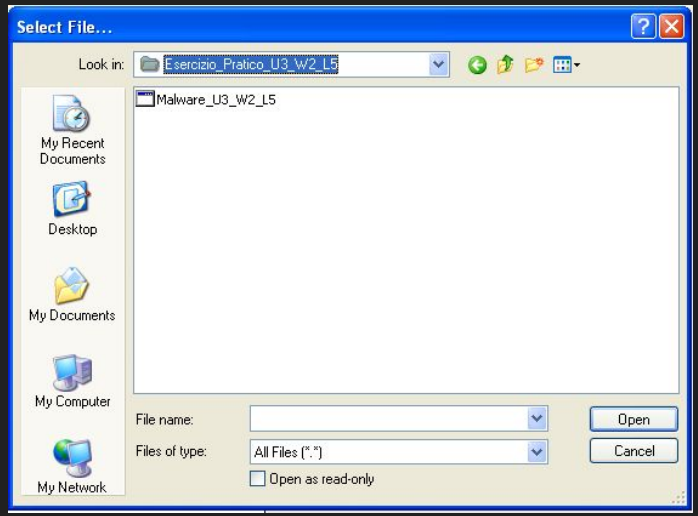
Per avviare il software è sufficiente cliccare due volte sulla sua icona



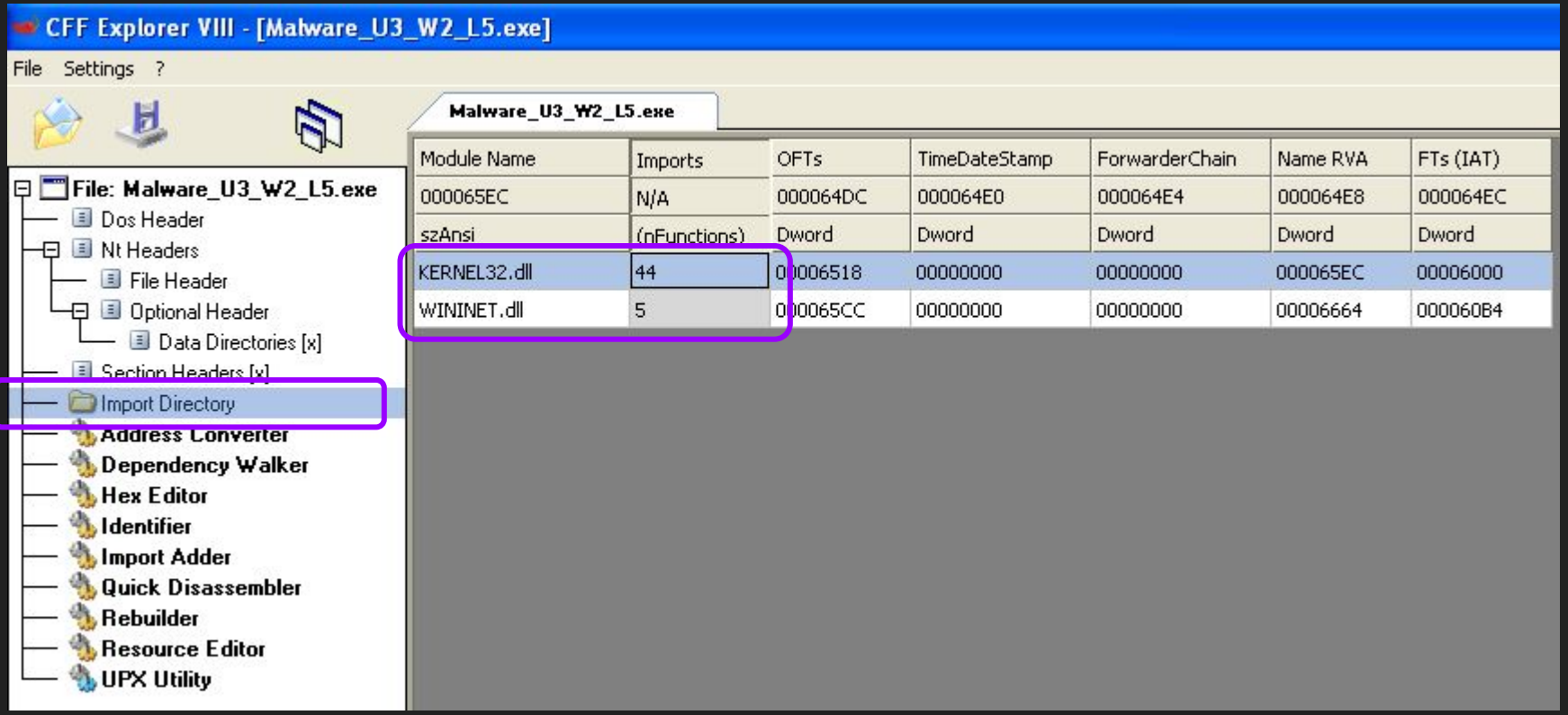
INTRODUZIONE

In alto a Sx cliccando sull’icona della cartella gialla ci comparirà la finestra di selezione del file.

Ora selezioniamo il malware da analizzare



LIBRERIE IMPORTATE



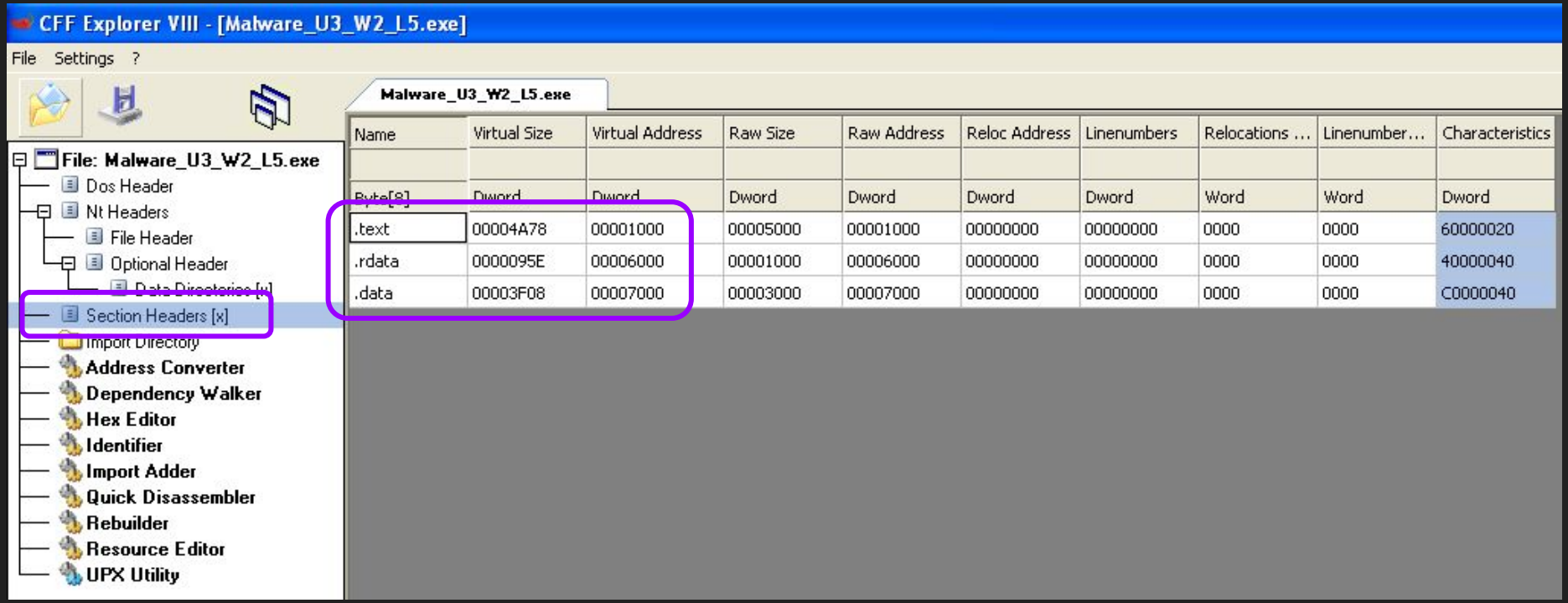
LIBRERIE IMPORTATE

Come mostrato nella figura precedente cliccando a Sx su Import Directory verranno mostrate le librerie importate dal malware. In questo caso sono:

Kernel32.dll: libreria piuttosto comune che contiene le funzioni principali per interagire con il sistema operativo, ad esempio: manipolazione dei file, la gestione della memoria.

Wininet.dll: libreria che contiene le funzioni per l'implementazione di alcuni protocolli di rete come HTTP, FTP, NTP.

SEZIONI MALWARE



SEZIONI MALWARE

Sempre nel menu di Sx troviamo la voce Section Headers, qui verranno mostrati invece le sezioni da cui è composto il software malevolo. In questo caso sono:

.text: la sezione «text» contiene le istruzioni (le righe di codice) che la CPU eseguirà una volta che il software sarà avviato. Generalmente questa è l'unica sezione di un file eseguibile che viene eseguita dalla CPU, in quanto tutte le altre sezioni contengono dati o informazioni a supporto.

.rdata: la sezione «rdata» include generalmente le informazioni circa le librerie e le funzioni importate ed esportate dall'eseguibile, informazione che come abbiamo visto possiamo ricavare con CFF Explorer.

.data: la sezione «data» contiene tipicamente i dati / le variabili globali del programma eseguibile, che devono essere disponibili da qualsiasi parte del programma.

COSTRUTTI NOTI

Il malware è stato scritto in linguaggio Assembly, linguaggio di programmazione a basso livello che fornisce un'interfaccia simbolica per il linguaggio macchina di un computer o di un'altra architettura.

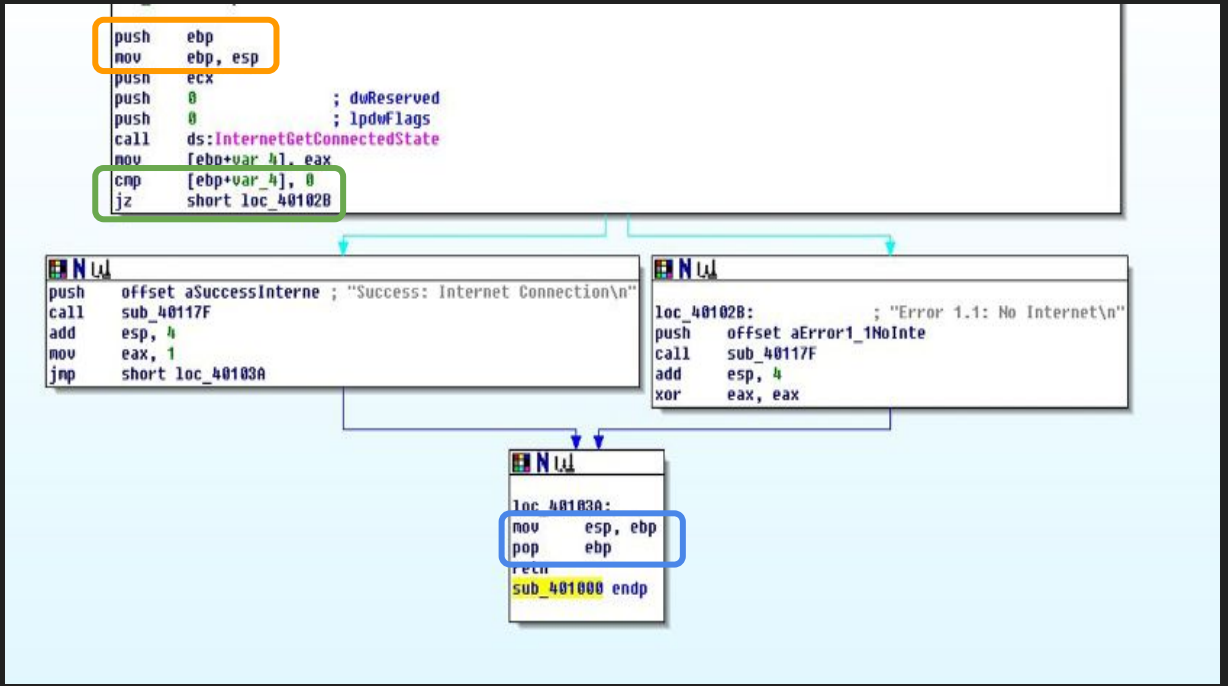
Viene considerato di basso livello perché le sue istruzioni sono strettamente correlate alle istruzioni della CPU e alla struttura del set di istruzioni della macchina.

COSTRUTTI NOTI

Creazione dello stack

Ciclo IF

Rimozione dello stack



FUNZIONALITA' IMPLEMENTATA

Nel codice della slide precedente si nota una scritta in rosa, ovvero InternetGetConnectedState questa non è altro che una funzione che ha lo scopo di controllare se sulla macchina è presente una connessione ad internet.

Tramite il ciclo IF sottostante se il risultato della funzione è 0 allora non è presente nessuna connessione internet e la funzione stamperà il messaggio "Error 1.1: No Internet";

Altrimenti se il valore di ritorno della funzione è diverso da 0 vi è connessione e la funzione stampa "Success: Internet Connection".

GRAZIE