Assembly Code Analysis Report

Objectives:
1. Identify known constructs (e.g., while, for, if, switch, etc.).
2. Hypothesize the code's functionality – high-level execution.
3. Bonus: Study and explain each individual line of code.

Prompt:
The figure below shows a snippet of malware code. The task is to identify known constructs seen during the theoretical lesson, hypothesize the functionality implemented in the code, and explain each individual line of code.

Traccia:

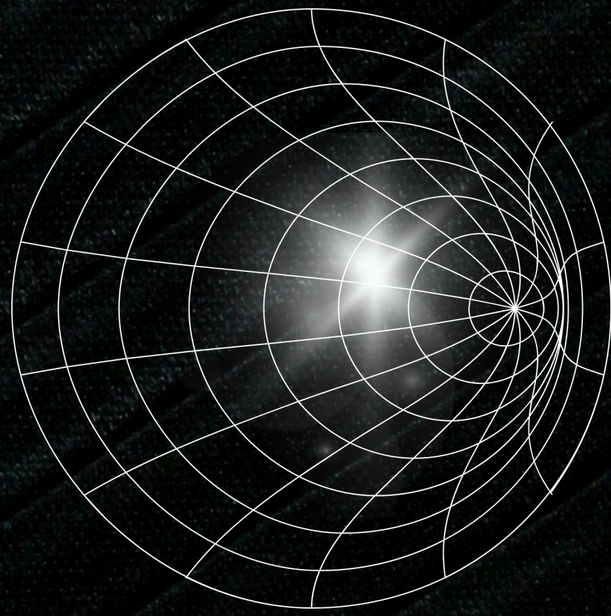
La figura seguente mostra un estratto del codice di un malware.
Identificare i costrutti noti visti durante la lezione teorica.

```
.text:00401000                push    ebp
.text:00401001                mov     ebp, esp
.text:00401003                push    ecx
.text:00401004                push    0               ; dwReserved
.text:00401006                push    0               ; lpdwFlags
.text:00401008                call    ds:InternetGetConnectedState
.text:0040100E                mov     [ebp+var_4], eax
.text:00401011                cmp     [ebp+var_4], 0
.text:00401015                jz      short loc_40102B
.text:00401017                push    offset aSuccessInterne ; "Success: Internet Connection\n"
.text:0040101C                call    sub_40105F
.text:00401021                add     esp, 4
.text:00401024                mov     eax, 1
.text:00401029                jmp     short loc_40103A
.text:0040102B ; ---------------------------------------------------------------------------
.text:0040102B
```

Michele Covi

SIOL4

QUESTION I



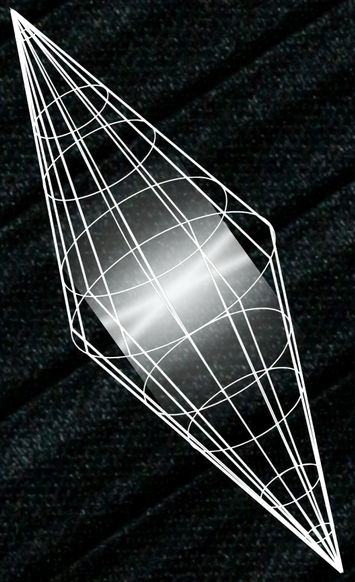
I. Identification of Known Constructs:

The code contains an if construct used to check whether an Internet connection is active. Specifically:

- The cmp instruction compares the value of a variable with zero.

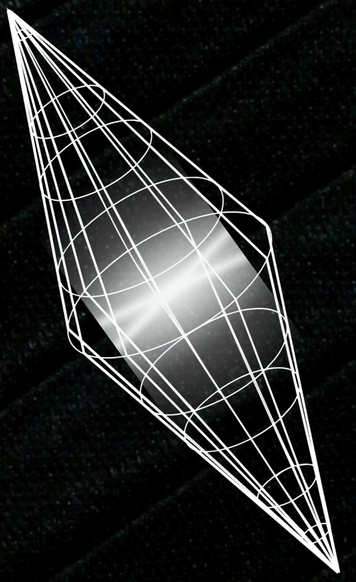
- The jz (jump if zero) instruction jumps to a specific location if the result of the comparison is zero.

QUESTION 2



2. Hypothesizing the Functionality of the Code:

The assembly code implements a check for Internet connectivity using the InternetGetConnectedState function.
This function verifies whether the computer has access to the Internet. If the connection is active, the program prints a success message. If the connection is not active, the program flow jumps to the next location without printing the message.



QUESTION 3 – PTI

1-

Stack Frame Setup:

.text:00401000 push ebp
.text:00401001 mov ebp, esp

- push ebp: Saves the current base pointer value onto the stack.
- mov ebp, esp: Sets EBP to the current ESP value, creating a new stack frame.

2-

Register Saving and Parameter Preparation:

.text:00401003 push ecx
.text:00401004 push 0 ; dwReserved
.text:00401006 push 0 ; lpdwFlags

- push ecx: Saves the ECX register onto the stack.
- push 0: Passes the value 0 as the dwReserved parameter to the InternetGetConnectedState function.
- push 0: Passes the value 0 as the lpdwFlags parameter to the same function.

3-

Function Call:

.text:00401008 call ds:InternetGetConnectedState

- call ds:InternetGetConnectedState: Calls the InternetGetConnectedState function, which checks the status of the Internet connection.
- The result of the function is stored in the EAX register.

4-

Saving the Result:

.text:0040100E mov [ebp+var_4], eax

- mov [ebp+var_4], eax: Saves the function's result (stored in EAX) into a local variable var_4.

5-

Comparison and Conditional Jump:

.text:00401011 cmp [ebp+var_4], 0
.text:00401015 jz short loc_40102B

- cmp [ebp+var_4], 0: Compares the value of var_4 with 0.
- jz short loc_40102B: If the value is zero (no Internet connection), jumps to location loc_40102B.

6-

Success Message:

.text:00401017 push offset aSuccessInternet ; "Success: Internet Connection\n"
.text:0040101C call sub_40105F
.text:00401021 add esp, 4

- push offset aSuccessInternet: Pushes the address of the success message onto the stack.
- call sub_40105F: Calls a subroutine to print the message.
- add esp, 4: Cleans up the stack by removing the message address.

QUESTION 3 – PART 2 + CONCLUSION

2-

Setting the Return Value:

.text:00401024 mov eax, 1

- mov eax, 1: Sets EAX to 1, indicating a successful outcome.

3-

Jump to End of Code:

.text:00401029 jmp short loc_401030
.text:0040102B loc_40102B

- jmp short loc_401030: Jumps to location loc_401030, skipping any following code.
- loc_40102B: Label that marks the jump target when no Internet connection is detected.

Conclusion

This assembly code performs a check to verify whether the system has access to the Internet by using the InternetGetConnectedState function.

If a connection is detected, it prints a success message and sets a positive return value. If no connection is detected, the program jumps directly to the end without printing the message.

This detailed analysis of known constructs and individual assembly instructions provides a clear understanding of the implemented functionality and the overall program execution flow.