

Reverse Engineering Course

AntiDebug Techniques

Barak Gonen



Intro

- PE format
- DLL files
- Loader

AntiDebug Techniques

- We shall discuss only FEW AntiDebug methods:
 - IsBeingDebugged
 - NtGlobalFlag
 - CheckRemoteDebuggerPresent
 - Check for breakpoints
 - Check code integrity
 - Monitor thread
 - TLS Callback
 - Packed file

PEB

- Process Environment Block – windows data structure
 - We shall use 32 bit PEB values
- X32dbg:
 - peb() - view PEB address
- Windbg:
 - dt ntdll!_peb – view PEB structure
 - !peb – view specific process values
 - r \$peb – view PEB address
 - eb \$peb - edit PEB values

IsBeingDebugged

- IsBeingDebugged –at \$PEB + 2
- Can be called by IsDebuggerPresent
- How to bypass?
 - Edit the PEB
 - Simply jump over IsDebuggerPresent
 - Patch code (change conditions etc)

NtGlobalFlag

- \$PEB + 0x68
- A debugger will usually set the flags:
 - FLG_HEAP_ENABLE_TAIL_CHECK (0x10)
 - FLG_HEAP_ENABLE_FREE_CHECK (0x20)
 - FLG_HEAP_VALIDATE_PARAMETERS (0x40)

CheckRemoteDebuggerPresent

- Checks if a process has a non zero debug port
- How to bypass?
 - Same idea as IsDebuggerPresent

Monitor Thread

- Demo



Hands On

- <https://data.cyber.org.il/reversing/Inception.exe>
- Remove defense using patching

TLS Callback

- Thread Local Storage
- TLS callbacks - Mechanism provided by the loader
- Initialize / clean when
 - Thread starts / ends
 - Process starts/ ends
- Demo
- <https://data.cyber.org.il/reversing/TLS.exe>

Check Breakpoints

- Debuggers replace a byte with “CC” (int 3)
- Search for “CC” opcode in the text section

Check Code Integrity

- Pre-Calculate the sum (or other hash function) of all bytes
- Test during runtime that value is same

ClassWork

- <https://data.cyber.org.il/reversing/BispooKeygen1.exe>

Homework

- <https://data.cyber.org.il/reversing/BispooKeygen2.exe>

Packed File

- <https://data.cyber.org.il/reversing/packed.exe>
 - Credit: Polarium
- Demo



Exercise 2

- <https://data.cyber.org.il/reversing/ShabakSecond.exe>
- ONLY solve antidebugging