Understanding TEB and PEB in Malware Development and Reverse Engineering

By Reda Ouzudane

Introduction

In the field of malware development and reverse engineering, understanding low-level Windows internals is essential. Two crucial structures used by malware and reverse engineers alike are the **TEB (Thread Environment Block)** and the **PEB (Process Environment Block)**.

PEB — Process Environment Block

The **PEB** is a data structure in the Windows operating system that holds information about the current process. It is located in user-mode memory and can be accessed without calling any Windows APIs, which is why it is often used in stealthy malware.

Key Contents of the PEB:

- Image base address
- Loaded modules (DLLs)
- Process parameters (like command line and environment variables)
- Heap information

Location:

On 32-bit systems, it can be accessed via `fs:[0x30]`, and on 64-bit systems via `gs:[0x60]`.

Use in Malware and RE:

- Retrieve loaded modules without using API calls
- Process hollowing and manual mapping
- Anti-debugging tricks (e.g., checking the `BeingDebugged` flag)

TEB — Thread Environment Block

The **TEB** is a structure associated with each thread in Windows. It contains information specific to the thread and also includes a pointer to the PEB.

Key Contents of the TEB:

- Thread ID
- Stack base and limit
- Pointer to the PEB
- Last error value
- Structured Exception Handling (SEH) chain

Use in Malware and RE:

- Traversing to the PEB
- Implementing anti-debugging techniques
- Stack manipulation and custom thread management

Summary Table

Term	**Stands for**	**Purpose**	**Use in Malware / RE**
PEB	Process Environment Block	Process-level info	DLL listing, anti-debugging, stealth
TEB	Thread Environment Block	Thread-level info	PEB access, error checking, SEH

Conclusion

Understanding and leveraging the PEB and TEB structures allows malware developers and reverse engineers to perform advanced techniques, often bypassing traditional API monitoring and making analysis more difficult for defenders.

Written by Reda Ouzudane