

Forum:[Hack Forums](#)

Thread: [\[mdev\] - RtlUserFiberStart Shellcode execution in Rust](#)

CL1B - February 4, 2026, 3:24 am

Executing shellcode using RtlUserFiberStart in Rust

Yeehaw!

Shellcode execution in Rust using RtlUserFiberStart. Manipulates TEB fiber structures to redirect execution flow through the fiber mechanism.

Getting TEB Pointer

TEB (Thread Environment Block) is accessible via gs:[0x30] on x64. We need this to manipulate fiber data.

Resolving RtlUserFiberStart

Get the address of RtlUserFiberStart from ntdll. This function is called when a fiber starts and reads execution address from fiber context.

Setting HAS_FIBER_DATA Bit in TEB

TEB has a flags field that indicates if the thread has fiber data. We set bit 2 (0b100) at offset 0x20 to make the system think we have valid fiber data.

Allocating RWX Memory for Shellcode

Allocate executable memory and copy shellcode into it.

Creating Fake Fiber Data

Allocate a fake fiber data structure on the heap. RtlUserFiberStart reads the execution address from offset 0x0A8 in the fiber context.

Setting Fiber Data Pointer in TEB

TEB stores the fiber data pointer at gs:[0x20]. point it to our fake fiber data.

Executing via RtlUserFiberStart

Call RtlUserFiberStart. It reads our fake fiber context, gets the RIP value, and jumps to it.

Cleanup

Free allocated memory after execution.

AV Performance [\(Click to View\)](#)

For educational purposes only