Heavy Wizardry

Shellcodes, Backdoors, Droppers, and Worms

by David Martínez Oliveira, aka Pico

Errata updated to print 1

Page	Error	Correction	Print corrected

125–126	The parameters are stored at `RBP - 0x74` and `RBP - 0x78`, which are actually 100 + 8 + 8 = 116 (`0x74` in hex) and 100 + 8 + 8 + 4 = 120 (`0x78` in hex). The `DWORD` specifier ensures that 32 bits (4 bytes) are copied for each instruction.
	In all, then, the program is actually using **124** bytes in the stack. As we've discussed, though, the compiler will try to keep the stack aligned to multiples of 16 bytes for Intel 64-bit platforms, so that value is rounded up to the closest multiple, which in this case is 128. This means there are **4** unused bytes between the buffer and the first parameter. Figure 4-6 shows what the stack frame for `func3` looks like.
	
	`0x00 ->`
	`0x08 -> Stack protector (8 bytes)`
	`...`
	`0x6c -> Buffer (100 bytes)`
	`...`
	`0x70 -> Padding (closest 16-byte multiple)`
	`0x74 -> First parameter (4 bytes)`
	`0x78 -> Second parameter (4 bytes)`
	`0x80 -> Closest 16-byte multiple`
	
	Figure 4-6: The `func3` stack frame
		The parameters are stored at `RBP - 0x74` and `RBP - 0x78`, which are actually 100 + 8 + 8 = 116 (`0x74` in hex) and 100 + 8 + 8 + 4 = 120 (`0x78` in hex). **This means the buffer is extended with 4 padding bytes.** The `DWORD` specifier ensures that 32 bits (4 bytes) are copied for each instruction.
		In all, then, the program is actually using **128** bytes in the stack. As we've discussed, though, the compiler will try to keep the stack aligned to multiples of 16 bytes for Intel 64-bit platforms, so that value is rounded up to the closest multiple, which in this case is 128. This means there are **8** unused bytes between the buffer and the first parameter. Figure 4-6 shows what the stack frame for `func3` looks like.
		
		`0x00 ->`
		`0x08 -> Stack protector (8 bytes)`
		`padding`
		`0x0d -> Buffer[99] <----+`
		`... | 100 bytes`
		`0x70 -> Buffer[0] <----+`
		`0x74 -> First parameter (4 bytes)`
		`0x78 -> Second parameter (4 bytes)`
		`0x80 -> Closest 16-byte multiple`
		
		Figure 4-6: The `func3` stack frame
			Pending