

CS 6V81-002: Quiz 18 Solutions

April 7, 2008

1. A “return-to-libc” attack involves... (circle all that apply)

- (a) injecting malicious code into the standard C libraries of a target system
- (b) exploiting a buffer overflow vulnerability to overwrite a return address on the stack**
- (c) overwriting a return instruction in the code segment with a jump to libc
- (d) crafting program input in such a way that the victim program stores attacker-supplied executable code into one of its stack buffers

The main distinguishing characteristic of jump-to-libc attacks is that they do not involve any injection of attacker-supplied code. Pre-existing libc code is used by the attacker in a malicious way.

2. PaX ASLR enforces which security policies? (circle all that apply)

- (a) memory safety
- (b) control-flow safety
- (c) type safety
- (d) buffer overflow prevention

None of these policies are enforced! PaX does not even prevent buffer overflows; it just tries to make the results of such an attack non-useful to the attacker (with debatable success).

3. Switching from a 32-bit to a 64-bit architecture would help PaX protect against the attack described in the paper because... (circle one)

- (a) segment content could be randomized at a finer granularity
- (b) binaries could be re-obfuscated more frequently
- (c) the attacker’s probing procedure would take prohibitively long on average**
- (d) none of the above

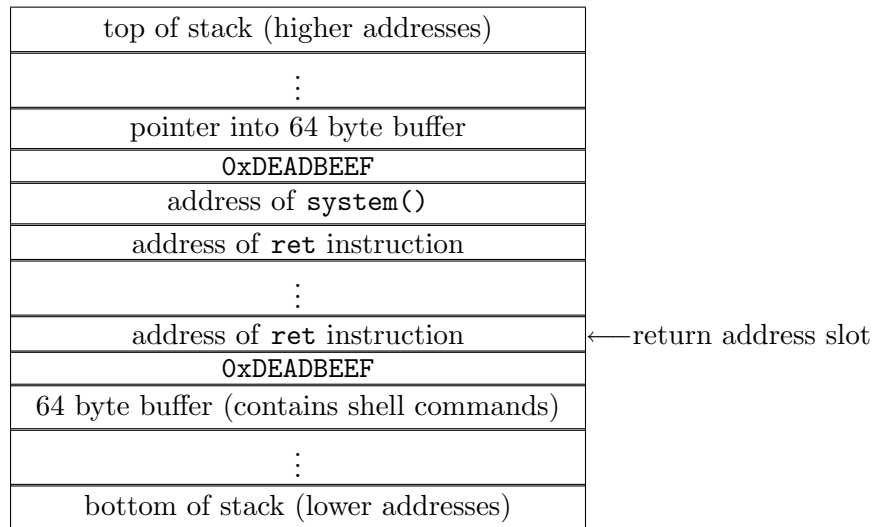
A 64-bit architecture introduces a much larger key space, making the probing phase of the attack take prohibitively long on average.

4. The attack described in the paper took about how long on average to compromise an Apache server? (circle one)

- (a) 12 hours
- (b) 1 hour
- (c) 5 minutes**
- (d) 20 seconds

5. (continued on back)

5. The following diagram depicts the stack contents after overflow (Figure 4 from the paper):



Consider the following simplified attack: Instead of writing all those “address of `ret` instruction” values into the stack, just write the address of `system()` directly into the return address slot. Why wouldn’t this simpler attack work just as well?

The simplified attack would only work if the attacker knew the location of the 64-byte buffer containing the shell commands and could therefore write it into an appropriate place on the stack. Rather than discover this location through a separate probe attack, the authors use a series of `ret` pointers to move the stack pointer up to where the buffer pointer already exists on the stack.