Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Fall Term 2018

SYSTEMS PROGRAMMING AND COMPUTER ARCHITECTURE
Assignment 7: Understanding Buffer Overflow Bugs - Errata

1 Problem

You have successfully written an exploit for the second stage of `ctarget`, the target prints `NICE JOB`. But when transmitting the result to the server the target receives a segfault and the result is not transmitted.

To fix this behavior, you must ensure the stack remains 16-byte aligned when you call touch2. To send the result to the server we rely on libc and the one that is shipped with Ubuntu 18.04 (and other modern Linux distributions) assume a 16-byte aligned stack.

2 Solution

Make sure the stack is 16-byte aligned when calling the touch functions. You can achieve this by explicitly modifying the stack pointer or ensuring that you call stack operations (such as `pop`, `push`, `ret`) an even number of times.

3 Detailed explanation

The problem affects *phase2* and *phase3* of the ctarget and *phase2* of `rtarget`. In `ctarget`, you have to write a bit of executable code onto the stack. In phase2 for instance, this little bit of code is needed to pass a (student-specific) argument to the function, hence the injected code probably does something not stack related, followed by `ret`. If you do this correctly, the binary prints `NICE JOB` and it goes on to submit the result to the server. This is all done on the modified stack. The first libc function that is called is `gethostbyname` which, after a couple of other calls to libc functions, calls `memset` with a stack variable as destination. As we know, memset is highly optimized and uses XMM (one of Intel's vector extensions) instructions. The XMM move instructions need (at least) a 16 byte aligned destination. New versions of gcc contain compiler optimizations (last year it was not in Ubuntus glibc yet...), that figures out that the stack is always 16 bytes aligned, hence it can skip the alignment check in the inlined memset and executes the XMM instruction with an unaligned destination, which segfaults.