Security Injections @ Towson

Module: Buffer Overflow - CS2 C++

Student: Ben Wasserman

Date: 09/20/2018

ID: 1541217942

Discussion Questions

Buffer overflows are more troublesome for some programming languages than for others. For example, C and C++ lack the built-in bounds checking facilities that Java provides. Some people have argued that this is a good reason to avoid C and C++ in favor of Java or other "safer" languages. Do you think this is a good idea? Why or why not?

> I don't necessarily think this is a good idea. C and C++ have lots of important qualities, namely speed. Programmers should learn to deal with the problems of bounds that could lead to buffer overflow. I also don't believe that just abstracting the problem away from the developer is the best method of solving it. Developers simply need to be more educated on the topic.

Countless currently running programs were built using C and C++. Buffer overflow vulnerabilities are often found in these programs, often after they have been in use for many years. Why should it be so difficult to find and fix buffer overflow flaws in software?

> It's so difficult to find and fix buffer overflow flaws in software because a lot of developers and industry professionals aren't as well educated on the topic as they should be. Additionally, there are a lot of legacy programs out there that aren't well maintained. This means old errors, like buffer overflow, can easily stick around for a long time without anyone noticing.

Buffer overflows can be troublesome if they are used by hackers to run their own code. What sort of things might an attacker try to do if he or she were able to run arbitrary code on a computer?

> If an attacker was able to run arbitrary code on a computer from buffer overflow by injecting the malicious code in the overflow bytes, there are a lot of things he or she may be able to do. Of course, the primarily goal may be to get shell access or to be able to run a command prompt. The attacker could inject malware to start to infiltrate a network, try to gain access to sensitive information, or try to destroy the vulnerable host.