Intro

- The number of digits that we receive from log10() is the number of digits we want to remove.
- If our number is 12521, the result of log10() will be 4.

Student Choice Monday

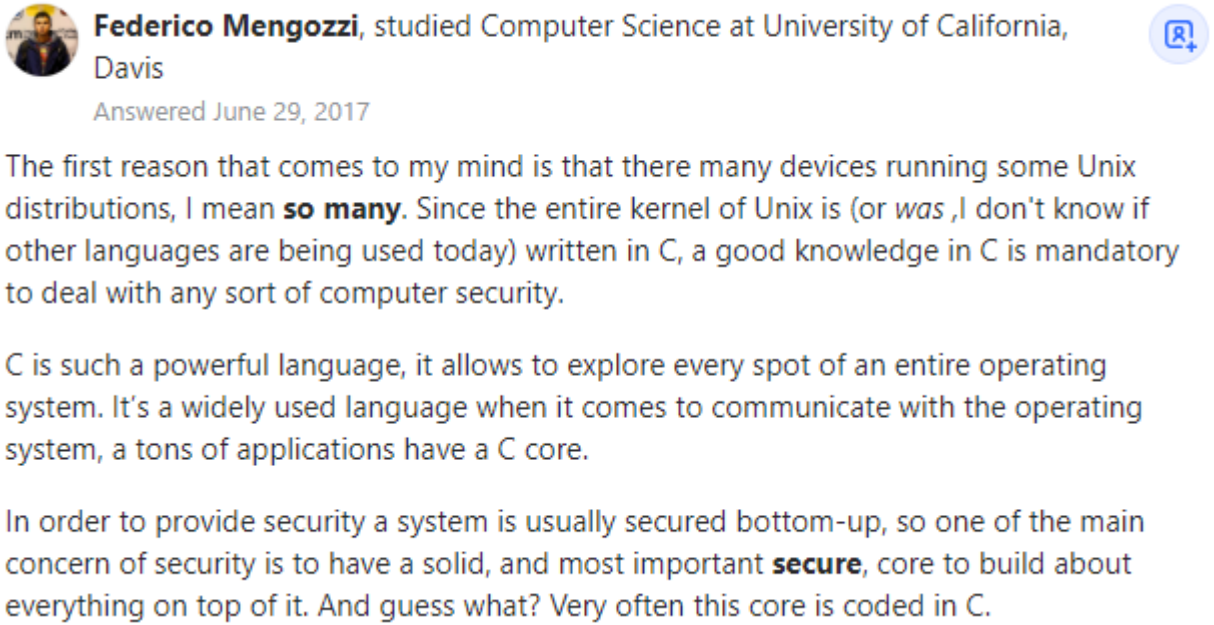
if we get to it.

Generic Makefile

- /public/examples/generic_makefile/makefile
- Show with my program 2

Cybersecurity in C

- https://cybersecurityguide.org/resources/coding-for-cybersecurity/
- Why is C popular for cybersecurity?



Federico Mengozzi, studied Computer Science at University of California, Davis

Answered June 29, 2017

The first reason that comes to my mind is that there many devices running some Unix distributions, I mean **so many**. Since the entire kernel of Unix is (or *was* ,I don't know if other languages are being used today) written in C, a good knowledge in C is mandatory to deal with any sort of computer security.

C is such a powerful language, it allows to explore every spot of an entire operating system. It's a widely used language when it comes to communicate with the operating system, a tons of applications have a C core.

In order to provide security a system is usually secured bottom-up, so one of the main concern of security is to have a solid, and most important **secure**, core to build about everything on top of it. And guess what? Very often this core is coded in C.

Security Issues with C

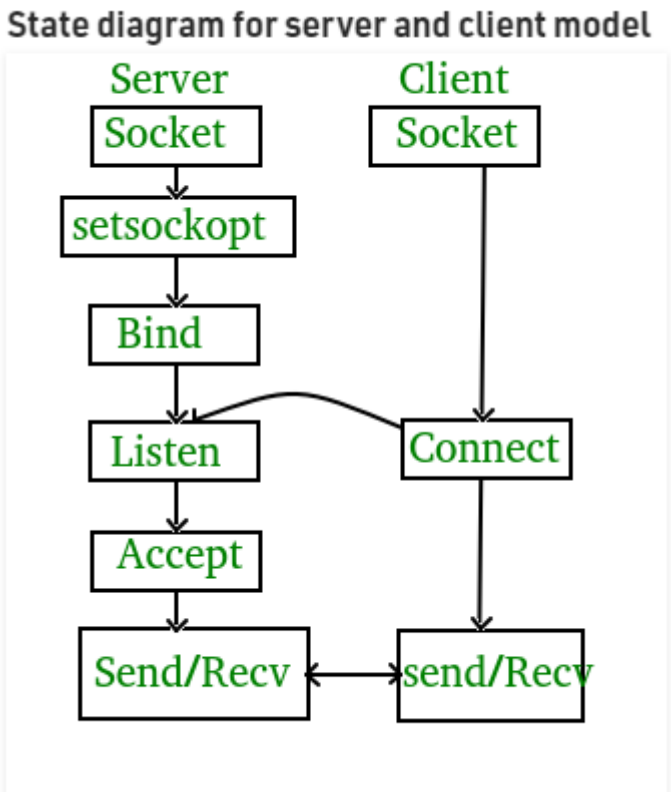
- Buffer overflow
  - Wikipedia: By sending in data designed to cause a buffer overflow, it is possible to write into areas known to hold executable code and replace it with malicious code, or to selectively overwrite data pertaining to the program's state, therefore causing behavior that was not intended by the original programmer. Buffers are widespread in operating system (OS) code, so it is possible to make attacks that perform privilege escalation and gain unlimited access to the computer's resources. The famed Morris worm in 1988 used this as one of its attack techniques.

Security Issues with C

- Formatting string attacks
  - Using printf and not checking user inputs
  - Using %s and %x to print data from locations in memory
  - You can also use %n to write data to arbitrary locations
- Integer overflow - http://projects.webappsec.org/w/page/13246946/Integer%20Overflows#:~:text=An%20integer%20overflow%20during%20a,when%20the%20data%20is%20copied.
  - **Security Impact of Integer Operations**
  - Attackers can use these conditions to influence the value of variables in ways that the programmer did not intend. The security impact depends on the actions taken based on those variables. Examples include, but are certainly not limited, to the following:
    - An integer overflow during a buffer length calculation can result in allocating a buffer that is too small to hold the data to be copied into it. A buffer overflow can result when the data is copied.
    - When calculating a purchase order total, an integer overflow could allow the total to shift from a positive value to a negative one. This would, in effect, give money to the customer in addition to their purchases, when the transaction is completed.
    - Withdrawing 1 dollar from an account with a balance of 0 could cause an integer underflow and yield a new balance of 4,294,967,295.
    - A very large positive number in a bank transfer could be cast as a signed integer by a back-end system. In such case, the interpreted value could become a negative number and reverse the flow of money - from a victim's account into the attacker's.

Communication between 2 programs

- sockets or pipes



At least one program has to run in the background
To run a program in the background: &

write.c/read.c

Pipes vs Sockets

- Use pipes:
  - when you want to read / write data as a file within a specific server. If you're using C, you read() and write() to a pipe.
  - when you want to connect the output of one process to the input of another process... see popen()
- Use sockets to send data between different IPv4 / IPv6 endpoints. Very often, this happens between different hosts, but sockets could be used within the same host