Problem 1 Ken Thompson’s paper, “Reflections on Trusting Trust,” describes a technique for installing an undetectable login backdoor by adding a second backdoor to the compiler. The backdoored compiler inserts the appropriate backdoors when compiling the login program and the compiler itself. Once the binary of the compiler, used for bootstrapping future systems, implements the backdoor, any trace of tampering can be removed from the source. In this problem, we will explore a technique for detecting such an attack.

Assume we have two C-language compilers: GCC and Clang. We suspect that nefarious hackers have inserted the Thompson backdoor into the GCC binary on our Linux system, but we believe that these hackers are haven’t managed to corrupt the faculty and students at UIUC responsible for Clang. The entire Linux system doesn’t yet build with Clang, since many programs were written to expect GCC’s extensions to the C language. But we can get Clang to build GCC. (The usual procedure is to compile a new version of GCC using GCC itself; this is known as “self-hosting.”)

Describe how we can reliably detect the presence of a GCC backdoor using the fact that we can compile GCC with Clang.

In this problem, you have access to a Linux runtime environment in which the GCC compiler is possibly compromised, but you may assume that other critical components (such as the kernel, shell, and standard utilities) are clean.

Problem 2 Last week I stopped by my favorite independent coffee shop, Starbucks, for a delicious Caramel Macchiato. While there, I connected my computer to the “Totally Free Starbucks WiFi” network, and proceeded to catch up on the latest posts on reddit.com.

Strangely, though, the site my browser brought up wasn’t Reddit. In fact, when I checked the DNS resolution (by typing “host reddit.com” on the command line) I found that my computer believed that Reddit was being served not from its usual IP address but from the IP address associated with attacker.com.

I happen to know that my computer’s operating system uses cryptography to generate random, difficult-to-guess DNS query-ID values and randomizes the source port on DNS queries. What’s more, I had hard-coded Google’s public DNS recursive resolver (“8.8.8.8”) in my system DNS settings.

How, then, was this attack possible?
Problem 3 Below is a picture of a UCSD police car parked near the Hopkins Parking Structure on campus. There is no one in the car.

Why was this police car parked here? Explain. Be sure to specify the assumptions your answer makes about police behavior and driver behavior.