Outline for November 17, 2005

Reading: K. Thompson, “Reflections on Trusting Trust,” *Communications of the ACM* **27** (8) pp. 761–763 (Aug. 1984).

1. Malicious logic
    a. Trojan horses, including replicatiing Trojan horses
    b. Computer viruses
        i. Boot sector infectors
        ii. Executable infectors
        iii. Multipartite viruses
        iv. TSR viruses
        v. Stealth viruses
        vi. Encrypted viruses
        vii. Polymorphic viruses
        viii. Macro viruses
    c. Computer worms
    d. Rabbits, bacteria
    e. Logic bombs
2. Defenses
    a. Cannot write a program to detect computer viruses without error
    b. Can detect all such programs if willing to accept false positives
    c. Can constrain case enough to locate specific malicious logic, using:
        i. Type checking (data vs. instructions)
        ii. Limiting rights (sandboxing)
        iii. Limiting sharing
        iv. Preventing or detecting changes to files
        v. Prevent code from acting beyond specification (proof carrying code)
        vi. Check statistical characteristics of programs (more authors than known, constructs in object files not corresponding to anything in the source)

Puzzle of the Day

Dr. Solomon, one of the earliest anti-virus software developers, wrote the following:

> .... Fanfare of trumpets ...
>
> .... Roll of the drums ...
>
> .... Very loud noise from 76 trombones ....
>
> **THE PERFECT ANTIVIRUS**
>
> Definition. I shall now give you, free of charge, an antivirus that if used correctly, detects all past, present and future viruses, never gives a false alarm, and has a zero cost. Sceptical? Then watch carefully ...
>
> ```
> P1.BAT
> Echo %1 is infected by a virus!!!
> ```
>
> You'll agree, I think, that P1.BAT will detect all past present and future viruses. That alone meets the "mathematically impossible" task! But, I hear you thinking, aren't there rather a lot of false alarms? Well, you didn't say you wanted a low false alarm rate....
>
> OK, OK. I'm used to projects where the user specification changes in the midle. Never mind. I can deal with the false alarms ...
>
> ```
> P2.BAT
> Echo %1 is NOT infected by a virus!!!
> ```
>
> You'll agree, I think, that P2 will never, ever, tell you that you have a virus when you don't. Of course, it has a pretty poor detection rate. I admit that. But I can fix it. See here ...
>
> ```
> PERFECT.BAT
> Echo Is %1 a virus? (Y/N)
> ```
>
> If the user types Y, you run P1. If the user types N, you run P2. Remember what I promised you? An antivirus that *if used correctly*, detects all past, present and future viruses, never gives a false alarm, and has a zero cost. All very amusing, but what can we learn from this?

What lessons can you draw from this exercise?