Computer Security
CS 426
Lecture 17

Market Failure of Secure Software

Announcements

- Project deadline extended to Friday Oct 15
- Join class mailing list
  - CS426_Fall2010@cs.purdue.edu
- Guest lecture on Digital Forensics on Monday Oct 4
- Guest lecture by Prof. Chris Clifton on Friday Oct 8
- Mid-term exam the week after October break week

How does a computer get infected with malware or being intruded?

- Executes malicious code via user actions (email attachment, download and execute trojan horses)
- Buggy programs accept malicious input
  - daemon programs that receive network traffic
  - client programs (e.g., web browser, mail client) that receive input data from network
  - Programs Read malicious files with buggy file reader program
- Configuration errors (e.g., weak passwords, guest accounts, DEBUG options, etc)
- Physical access to computer

Why Software Has (or appear to have) So Many Bugs?

- Software is complicated, and created by human
- Software is no more buggy, is just more targeted?
- Unique nature of software
  - Near-zero marginal cost
- Market failure for secure software
  - Market failure: a scenario in which individuals' pursuit of self-interest leads to bad results for society as a whole
  - Users cannot just vote for security with their money.
    - lack of measurement for security
  - Vendor has no incentives to produce higher quality software.

Guy Kawasaki: **“The Art of Innovation”**

- Don't worry, be crappy.
  - An innovator doesn't worry about shipping an innovative product with elements of crappiness if it's truly innovative.

- Churn, baby, churn.
  - I'm saying it's okay to ship crap--I'm not saying that it's okay to stay crappy. A company must improve version 1.0 and create version 1.1, 1.2, ... 2.0.

Why Vendors Lack Incentive to Produce More Secure Software

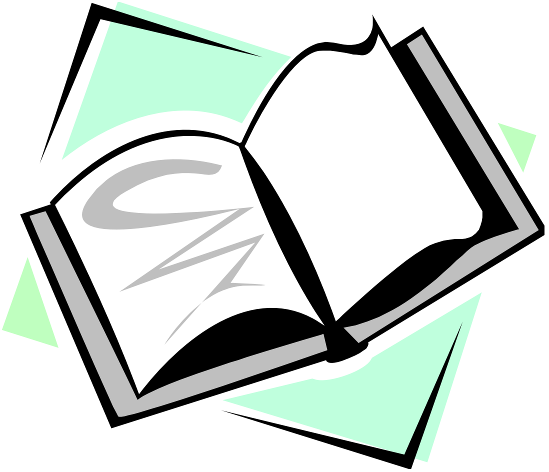
- Cash flows when product starts shipping.
- Market dominance is key to success
  - being first often means becoming de facto standard
- No liability.
- Bugs can be patched with little cost. No expensive recall.
- Thorough testing is inefficient. Let the users test it and fix only the bugs that affect users

The Perversity of Patching

- Releasing a patch costs little
- Buggy software can force users to upgrade
  - Achieving market dominance means competing with previous versions
  - Stop releasing patches for old versions can force users to upgrade
- Patching provide an opportunity of offering new licensing terms

Readings for This Lecture

- Wikipedia



Coming Attractions …

- Dealing with Malwares