The State of Software Security

`smb@research.att.com`

`http://www.research.att.com/~smb`

973-360-8656

AT&T Labs Research

Florham Park, NJ 07932

Today’s Status

- Constant reports of new security holes.
- Frequent exploitation of these holes.
- Worms, viruses, “distributed denial of service” (DDoS) attacks are common.

Bugs!

- Most security holes are caused by buggy code.
- A National Research Council study* found that 85% of CERT advisories decribed problems not fixable by crypto (primarily buggy software or misconfigured systems).
- Patches for these security bugs are often not applied.

**Trust in Cyberspace*, 1999.

Major Types of Bugs

- 31 CERT (Computer Emergency Response Team) Advisories this year.
  Microsoft, Sun, Oracle, AOL, and assorted open source products implicated.
- 21 refer to “buffer overflows”.
  ⇒Major warning in 1980; exploited very publicly in 1988.
- Two “integer overflows”.
  ⇒New as a security problem, but preventable with mid-1960’s technology.
- Two “format string” errors.
  Relatively new, but impossible with modern programming languages.
- All are geek-speak for preventable errors.

Effects

- DDoS attacks can take almost any site off the air.
- Major worm outbreaks can clog corporate mail servers.
- There have been a few worms with malicious payloads that delete files, steal passwords, leak documents, etc.

Distributed Denial of Service Attacks

- Attacker uses known, unpatched hole to take over many “zombie” machines.
- On command, the zombies all bombard some other site.
- The victim’s Internet link is clogged, even if the victim is not running any insecure software.
- Defending against this is very difficult.

What is the Cause?

- Vendors often ship too soon.

  ⇒Must ship on Internet time; “first to market” often wins.
- Overly-complex designs.

  ⇒Complex code is very often buggy code.
- Inadequate underlying operating system?

It’s Possible to do Better

- Phone switch failures due to sofware problems are at about the same rate as hardware problems.
- The hardware is ultra-reliable to start with!
- Total down time is measured in minutes per year.
- **But. . .** that requires a specialized, very expensive development process.

The Future?

- Will users pay for better software?
- Will innocent parties pay to be protected from other users' buggy code?
- Is there an economic incentive for vendors to do a better job?