# Internet Security in my Crystal Ball

*Steven M. Bellovin*

smb@research.att.com

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

+1 973-360-8656

AT&T Labs Research

Florham Park, NJ 07932

# What's the Problem?

- Bugs.
- Availability.
- Sniffing and spoofing.

$\Rightarrow$ None of these will go away!

# Bugs

- Most “Internet” security problems are caused by buggy code (plus administrators not installing patches for known security bugs). *This is not just the high-order bit of Internet security, it’s the high-order byte.*
- Buggy code won’t go away.
- The only solution is to keep bad guys away from the bugs.
- Today, we use firewalls for that. Is there an architecturally clean, scalable alternative?

N.B. This is a neither a requirement for, nor a condemnation of, a topological solution. It is a statement of a ongoing need.

# Availability

- We can't fix DDoS problems on the end systems — the network's resources are being abused, so the fix has to be in the net.
- We lack proven defenses (though there are several schemes on the table).
- Should a new network design improve our ability to control such abuse? How?
- What, if anything, should such a solution have in common with a congestion or flash crowd solution?

# Sniffing and Spoofing

- There have been a variety of attacks that are preventable by cryptography.
- But cryptography is used very rarely.
- Will this change in the future?

# Encryption

- Symmetric-key encryption in software is cheap, and getting cheaper. (On a 450 Mhz Pentium II, AES has been measured at about 243 Mbps.)
- Some NIC cards have IPsec on-board.
- RC4 is twice as fast, and has a very small footprint.
- **Assertion:** Asymptotically, encryption is free.

# What About the Low End?

- RC4 is very fast, even on wimpy processors.
- Besides, wimpy processors don't need to talk at such data rates.
- Caveat: encryption generally needs to be accompanied by authentication, *especially* with stream ciphers like RC4.
- Authentication is more expensive than RC4 encryption, though not as expensive as AES encrytion.

# What About the High End?

- We can encrypt *blocks* very fast; we can't encrypt *messages* much faster than OC-192.
- Well, we can, but we can't authenticate faster than that.
- Some work going on for new combined encryption/authentication modes of operation.
- Tentative conclusion: the high end won't be a problem, either.

# But. . .

- Public key encryption is considerably more expensive, and is likely to remain so.
- Will it ever be cheap enough for low-end devices?
  Unknown — there are some new schemes, but they're new, and their strength is quite unclear.
- Regardless, public key cryptography will always be more expensive than symmetric crypto, especially at the very low end.

# Key Management

- Cryptography generally requires key management.
- It's almost always mandatory for our nice, cheap, stream ciphers.
- Key management is expensive:
  - In round trips — keys must be "fresh" and authentic.
  - In hardware — you may need a timer.
  - In infrastructure — KDC and/or CA.
  - In CPU, if you use public key.

# Authorization

- An encrypted channel to a bad guy isn't very useful.
- You need to know to *whom* you're talking, and whether or not they're authorized.

$\Rightarrow$ May require technical infrastructure; *will* require management infrastructure.

# What Does This Mean?

- We may need to offload some functions from some boxes:
  - Proxy key management?
  - Proxy authorization?
  - Proxy encryption?
- Authorization management is very difficult, and is getting worse. (Can the manufacturer of your washing machine query its MIB? What about the owner of a leased car?)
- “Scale is the only interesting problem.”