Security: Present and Future

Steven M. Bellovin
smb@research.att.com
http://www.research.att.com/~smb

Today's Environment

- Some UNIX, especially on servers.
- Lots of Windows 98; some Windows NT.
- Decent (but not great) Internet connectivity, especially for Web and email.
- A few internal firewalls.
- A fair number of extranet connections.

But…

Today's product systems designs are quite complex.

- Complex topologies.
- Some outsourced functions.
- Liberal use of "point" firewalls and encryption.
- Many exposed machines.
- Overlapping networks, functions, etc.

Today's Internet Infrastructure

- Varying collection of ISPs -- some big, some small; some sharp, some clueless.
  - "Big" does not necessarily imply "smart"…
- Unauthenticated routing protocols control Internet reachability.
  - AT&T Worldnet has been taken off the air by routing errors.
- Little or no accountability by ISPs.
- External problems rarely affect internal nets.

Today's Threats

- Buggy code, especially buffer overflows.
- Automated hacking tools
  - Scanners.
  - Toolkits.
- Weak authentication.
  - Systems still employ reusable passwords.
- Viruses and worms.
- Buggy code!

Where are Today's Failures?

- Mostly due to scans of exposed machines.
- Most penetrations due to bugs.
- Viruses and worms are mostly nuisances.
- Attacks aren't sophisticated.

Tomorrow's Environment

- Will still have UNIX servers; more will be Linux.
- More Linux desktops; more Windows NT desktops.
  - Windows 98 will (and should) fade away from the corporate environment.
- Much richer Internet connectivity will be *required*. Business-critical functions will require the Internet (example: ASPs).

Tomorrow's Firewalls

- Corporate firewall becomes much less useful.
  - Too much connectivity through and around it.
- Too many firewall-evading protocols.
  - Mismatch between security groups and productivity needs.
- Firewalls will move towards the edges.
  - Local administrators understand local needs, local connectivity.

Culture versus Security

- "You can't solve social problems with software" (Marcus Ranum).
- Security people will have to pick their battles.
- Managed holes are better than internal warfare.
- User (and administrator) education is a priority.
  - Secrecy about attacks *hurts* security!

Tomorrow's Systems

- Even more complex.
- Much more out-sourced functionality.
- Much more complex topology.
- Today's platforms are not able to segregate data properly.
- How will we secure such systems?

Tomorrow's Internet

- Considerable merger with phone net.
- Difference between Internet, Intranet, and Extranet will disappear.
  - Today's VPNs are just the start.
  - Control of QoS important.
- Must find way to authenticate routing.
- Operations will be harder. Will there be more non-hostile failures?

Tomorrow's Threats

- We'll still have buggy code problems.
- Newer scanners, newer toolkits, etc.
- Tailored worms and viruses:
  - Controlled rate of spread.
  - Environment-aware; targets selected systems.
  - Threat will *not* diminish from demise of Windows 98.
- Serious attackers.
- Buggy code!

Tomorrow's Attackers

- Today's attackers are mostly bored kids.
- Will we see (more?) professionals?
  - This week, someone tried to steal Microsoft's garbage. What about the Recyle Bin?
  - Is information warfare a real threat?
- There is no reliable data on insider attack rates. It's high -- but no one knows how high.

Target Selection

- Today:
  - Most exposed systems are informational, customer-facing, or for Internet operations.
- Tomorrow:
  - Internal corporate operations will use the Internet.
  - National (and international) infrastructure will *rely* on the Internet.

Privacy Issues

- Customer privacy becoming a major issue.
  - Insecure systems cannot protect private data
- How do you authenticate people while still protecting their privacy?
  - Biometrics are a dangerous path (and probably won't do the job).
  - How will users carry their authentication data?

Managing Complexity

- Complexity is the enemy of security.
- We're building more complex systems -- how will we keep them secure?
- *Must* separate complex, user-facing functionality from security-critical sections.
  - Of course, sometimes that makes for complex interfaces…

Integrating Security

- Add-on security is harder on users, and doesn't protect as well.
- Application and systems developers *must* build in security from the start.
  - But they don't understand the threats. Education is crucial.
- Obtrusive security won't be used. Security under the hood works just fine.
  - Why isn't all internal email encrypted?

New Application Styles

- We are moving towards more distributed applications.
- Client/server model will become much less common.
- Intelligence everywhere -- but that can mean more management woes.
  - Who is the systems administrator for my VCR? Does my oven trust my cell phone?

Example: Gnutella

- Pure peer-to-peer protocol.
- Flooding-style topology discovery and queries.
  - Leaks information.
  - Rogue node in the middle can spy.
  - Hard to prevent spoofed addresses; can lead to flooding attacks.
- Firewall evasion commands.

(See http://www.research.att.com/~smb/talks/NapsterGnutella/index.htm)

Conclusions

- Increased interconnectivity will make security harder.
- Increased reliance on the net will make security more important.
- Buggy code won't go away.
- We won't be out of work any time soon...