40 Years of Internet Security and Perimeters

Bill Cheswick
AT&T Research
ches@research.att.com

Internet Security in a Nutshell

- The third character on the Internet crashed the server (1969)
- The same problems have been repeated repeatedly ever since
- Basically, the Internet is working just fine, warts and all

The Early Internet: the end-to-end principle

- Everyone can talk to everyone else
- The middle of the network is, and must be, dumb
- Any two computers can define and use a new protocol, without further permission
- This was the rule until 1987

1987: Packet filtering

- *Mogul, Rashid, Accetta.* SOSP Nov. 1987
- Found in routers
- Easy to implement
- Efficient, mostly
- Can implement a variety of security policies
- *Mogul: screend*

1987: Application level gateways

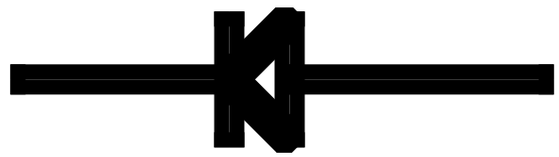
- Dave Presotto at Bell Labs rewrote mailer because he didn't trust *sendmail*

- This is the firewall I inherited.
- DEC Gatekeeper and DEC SEAL
  - Ranum, Avolio, Reid, Vixie

"Design of a Secure Internet Gateway"

- 1990 Summer Usenix paper
- Belt-and-suspenders gateway design
- Described Presotto's work, and my additions
- Coined the term *proxy*.

Original firewall



My (Safer!) Firewall

Referee's suggestion

A simile for the ages?

- "All of [the gateway's] protection has, by design, left the internal AT&T machines untested---a sort of crunchy shell around a soft, chewy center."

Behind firewalls

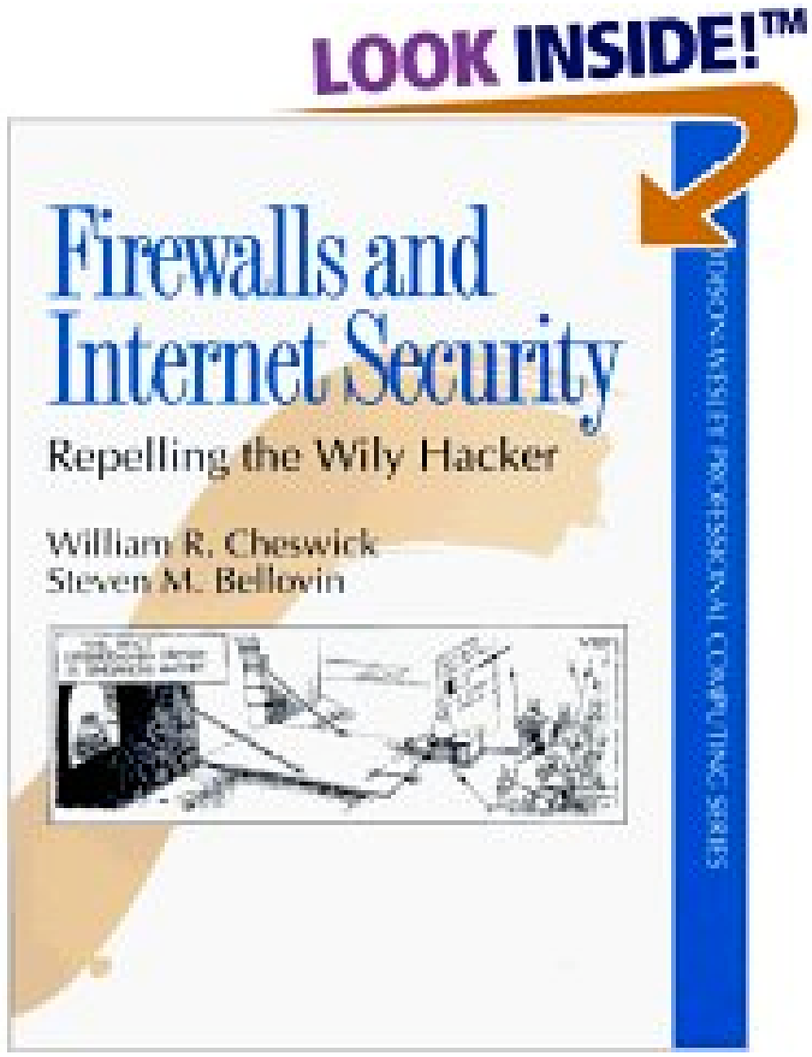
- Standard servers are too dangerous to expose to outside access
- TCP/IP packets are too dangerous
  - No IP connectivity to outside

Advantages

- Expertise focused at the gateway
- Security is cheaper
- Stopped the Morris worm, and many many other evil probes
- Isolated address space doesn't leak information, maybe easier to manage

Firewalls book (1994)

- The timing was perfect
- The world adopted many of our suggestions



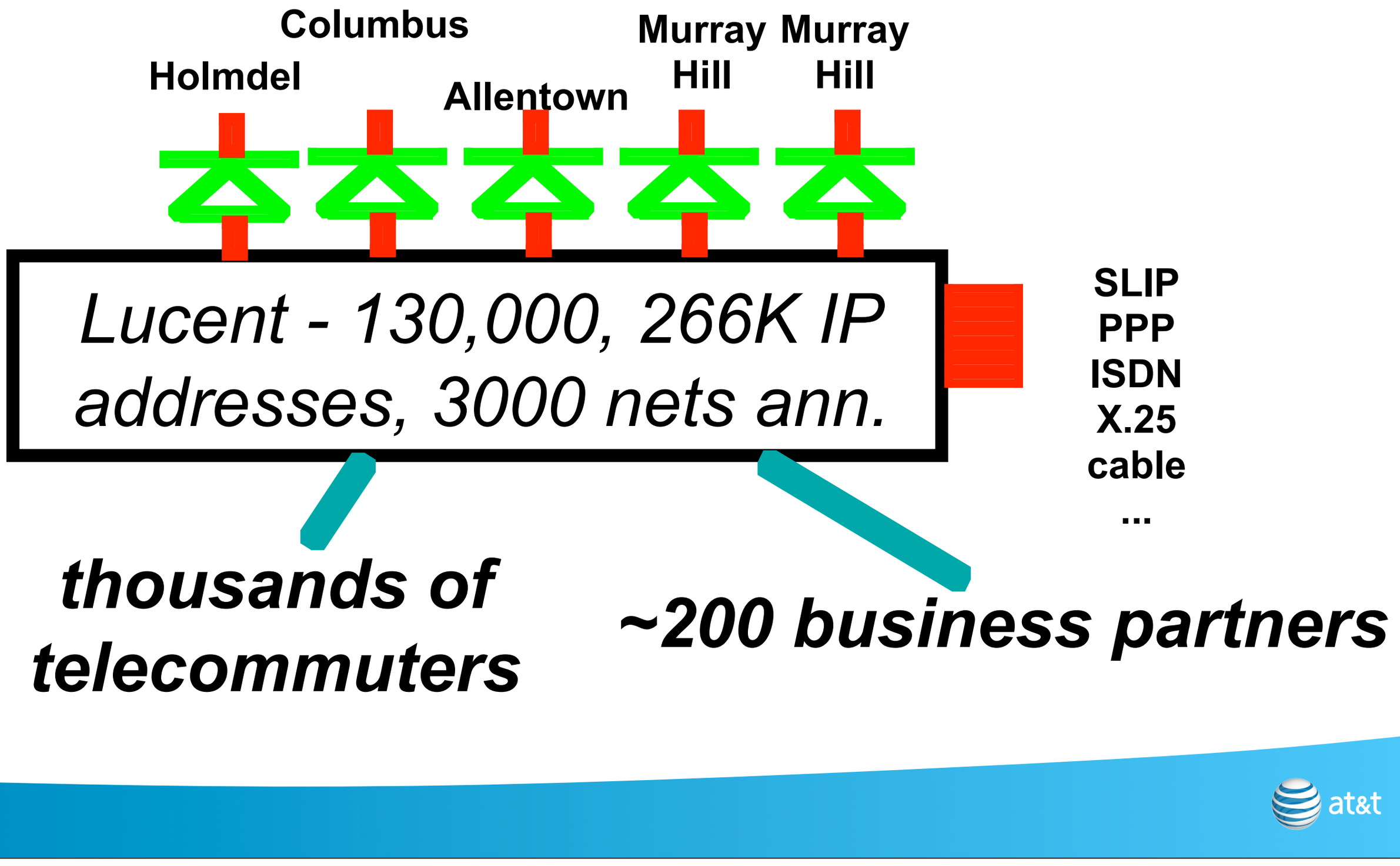
Disadvantages

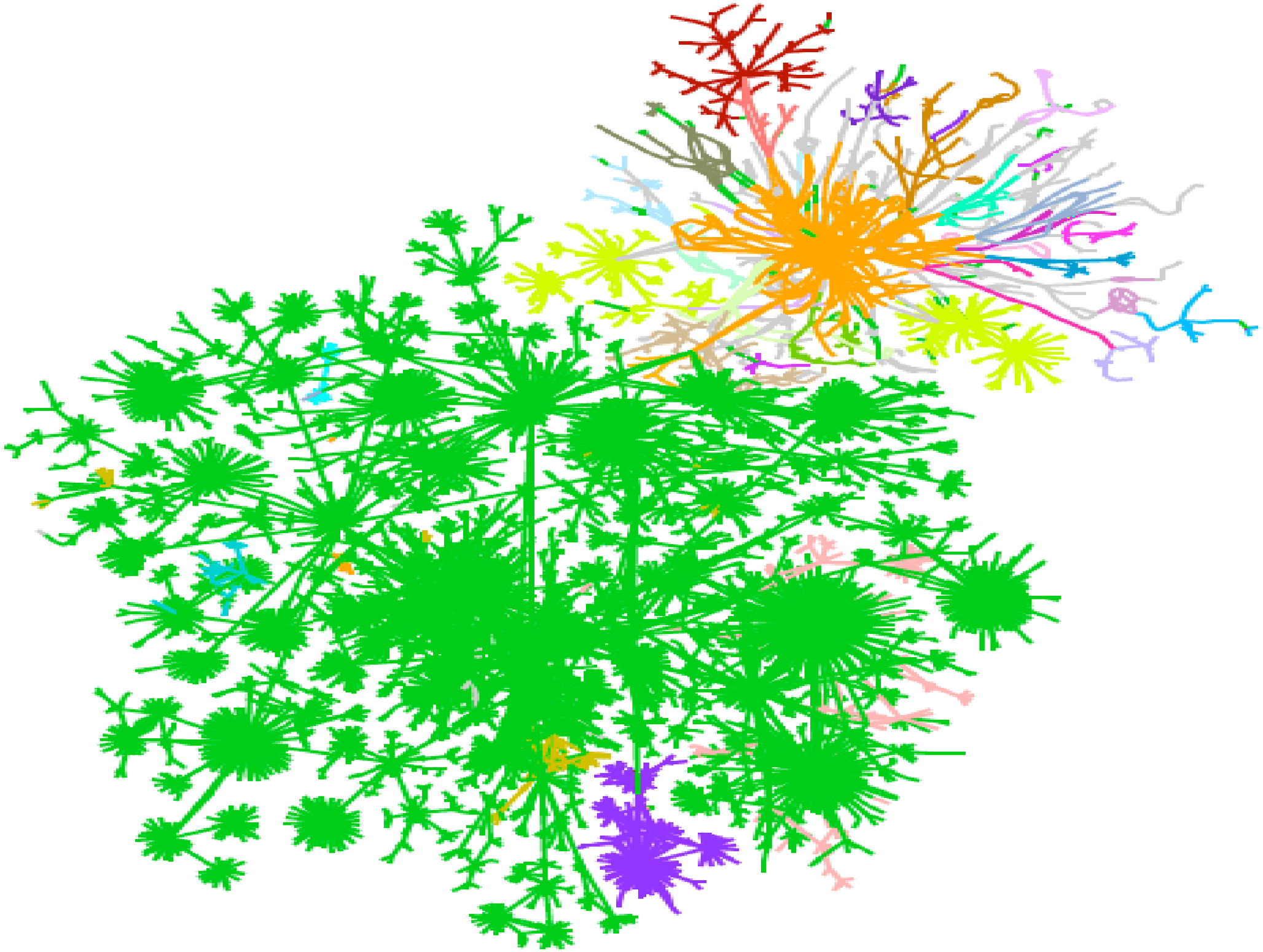
- Lose much of the innovation potential of the end-to-end principle
- Hard to keep up with new desired services
- Mechanism for outgoing TCP connections very helpful
  - reflected in modern NAT security

Chewy Center is a problem

- Host weakness "OK" if firewall is present, but isn't really
- By 1996, AT&T/Lucent had 130,000 hosts "inside" the perimeter

The Internet





Internet Skinny Dipping

Research question

- Can one use the Internet in a rich way, safely, without perimeter defenses?
- If so, what does it take?

Threat Model

- Attacks from without: evil software actively probing our software
- Invited attacks: clicking on the wrong thing
- Eavesdropping in the endpoints or in transit data

Security elements

- Secure servers, highly resistant to crafted attacks
- Secure communication, resistant to man-in-the-middle attacks and eavesdropping
- Clients strong enough to protect their users' secrets and software integrity
- The bozo in the chair

Guiding security principle for servers

- "You've got to get out of the game." - Fred Grampp
- "Best block is not be there." - Mr. Miyagi, Karate Kid 2

Secure Servers

We can do pretty well with servers

- If we try. Ask Amazon, Fedex, etc., etc.
- We have experts designing and running these machines
- Server software can be quite robust
  - sshd, postfix, apache (maybe)
- Systems don't default to safe servers

Win ME

```
Active Connections - Win ME

  Proto  Local Address          Foreign Address        State
  TCP    127.0.0.1:1032         0.0.0.0:0              LISTENING
  TCP    223.223.223.10:139     0.0.0.0:0              LISTENING
  UDP    0.0.0.0:1025           *:*
  UDP    0.0.0.0:1026           *:*
  UDP    0.0.0.0:31337          *:*
  UDP    0.0.0.0:162            *:*
  UDP    223.223.223.10:137     *:*
  UDP    223.223.223.10:138     *:*
```

Win 2K

```
Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1029         0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1036         0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1078         0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1080         0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1086         0.0.0.0:0              LISTENING
  TCP    0.0.0.0:6515         0.0.0.0:0              LISTENING
  TCP    127.0.0.1:139        0.0.0.0:0              LISTENING
  UDP    0.0.0.0:445          *:*
  UDP    0.0.0.0:1038         *:*
  UDP    0.0.0.0:6514         *:*
  UDP    0.0.0.0:6515         *:*
  UDP    127.0.0.1:1108       *:*
  UDP    223.223.223.96:500   *:*
  UDP    223.223.223.96:4500  *:*
```

Win XP pre-SP2

```
Proto  Local Address          Foreign Address        State
 TCP    ches-pc:epmap          ches-pc:0              LISTENING
 TCP    ches-pc:microsoft-ds   ches-pc:0              LISTENING
 TCP    ches-pc:1025           ches-pc:0              LISTENING
 TCP    ches-pc:1036           ches-pc:0              LISTENING
 TCP    ches-pc:3115           ches-pc:0              LISTENING
 TCP    ches-pc:3118           ches-pc:0              LISTENING
 TCP    ches-pc:3470           ches-pc:0              LISTENING
 TCP    ches-pc:3477           ches-pc:0              LISTENING
 TCP    ches-pc:5000           ches-pc:0              LISTENING
 TCP    ches-pc:6515           ches-pc:0              LISTENING
 TCP    ches-pc:netbios-ssn    ches-pc:0              LISTENING
 TCP    ches-pc:3001           ches-pc:0              LISTENING
 TCP    ches-pc:3002           ches-pc:0              LISTENING
 TCP    ches-pc:3003           ches-pc:0              LISTENING
 TCP    ches-pc:5180           ches-pc:0              LISTENING
 UDP    ches-pc:microsoft-ds   *:*
 UDP    ches-pc:isakmp         *:*
 UDP    ches-pc:1027           *:*
 UDP    ches-pc:3008           *:*
 UDP    ches-pc:3473           *:*
 UDP    ches-pc:6514           *:*
 UDP    ches-pc:6515           *:*
 UDP    ches-pc:netbios-ns     *:*
 UDP    ches-pc:netbios-dgm    *:*
 UDP    ches-pc:1900           *:*
 UDP    ches-pc:ntp            *:*
 UDP    ches-pc:1900           *:*
 UDP    ches-pc:3471           *:*
```

FreeBSD

```
Active Internet connections (including servers)
Proto Recv-Q Send-Q  Local Address
tcp4       0      0  *.22
tcp6       0      0  *.22
```

Microsoft wasn't the first

SGI Irix

```
ftp      stream  tcp     nowait  root    /v/gate/ftpd
telnet   stream  tcp     nowait  root    /usr/etc/telnetd
shell    stream  tcp     nowait  root    /usr/etc/rshd
login    stream  tcp     nowait  root    /usr/etc/rlogind
exec     stream  tcp     nowait  root    /usr/etc/rexecd
finger   stream  tcp     nowait  guest   /usr/etc/fingerd
bootp    dgram   udp     wait    root    /usr/etc/bootp
tftp     dgram   udp     wait    guest   /usr/etc/tftpd
ntalk    dgram   udp     wait    root    /usr/etc/talkd
tcpmux   stream  tcp     nowait  root    internal
echo     stream  tcp     nowait  root    internal
discard  stream  tcp     nowait  root    internal
chargen  stream  tcp     nowait  root    internal
daytime  stream  tcp     nowait  root    internal
time     stream  tcp     nowait  root    internal
echo     dgram   udp     wait    root    internal
discard  dgram   udp     wait    root    internal
chargen  dgram   udp     wait    root    internal
daytime  dgram   udp     wait    root    internal
time     dgram   udp     wait    root    internal
sgi-dgl  stream  tcp     nowait  root/rcv dgld
uucp     stream  tcp     nowait  root    /usr/lib/uucp/uucpd
```

SGI Irix (cont.)

```
mountd/1     stream  rpc/tcp wait/lc    root    rpc.mountd
mountd/1     dgram   rpc/udp wait/lc    root    rpc.mountd
sgi_mountd/1 stream  rpc/tcp wait/lc    root    rpc.mountd
sgi_mountd/1 dgram   rpc/udp wait/lc    root    rpc.mountd
rstatd/1-3   dgram   rpc/udp wait       root    rpc.rstatd
walld/1      dgram   rpc/udp wait       root    rpc.rwalld
rusersd/1    dgram   rpc/udp wait       root    rpc.rusersd
rquotad/1    dgram   rpc/udp wait       root    rpc.rquotad
sprayd/1     dgram   rpc/udp wait       root    rpc.sprayd
bootparam/1  dgram   rpc/udp wait       root    rpc.bootparamd
sgi_videod/1 stream  rpc/tcp wait       root    ?videod
sgi_fam/1    stream  rpc/tcp wait       root    ?fam
sgi_snoopd/1 stream  rpc/tcp wait       root    ?rpc.snoopd
sgi_pcsd/1   dgram   rpc/udp wait       root    ?cvpcsd
sgi_pod/1    stream  rpc/tcp wait       root    ?podd
tcpmux/sgi_scanner stream tcp nowait    root    ?scan/net/scannerd
tcpmux/sgi_printer stream tcp nowait    root    ?print/printerd
9fs          stream  tcp     nowait     root    /v/bin/u9fs u9fs
webproxy     stream  tcp     nowait     root    /usr/local/etc/webserv
```

And they are still making mistakes

- *Finding User/Kernel Pointer Bugs with Type Inference*. Rob Johnson, David Wagner, Usenix Security 2004
  - Unchecked user-space pointers in systems calls on Linux
- New bugs appearing in secure OSes

Secure Communications

- The crypto export wars of the 90s are over
- In June 2003, NSA said that a properly implemented and vetted version of AES is suitable for Type 1 cryptography
- SSL is holding up well
- So is ssh

Secure Clients: Windows

- Has had server problems (see above) and poor or no software containment
- Microsoft's security press is real, and Vista is going to be an improvement
  - This is going to take time: an Augean stable

Vista: good signs

- It took longer than they expected to get it out
  - Not a mythical man month problem, they had to dig deeper
- A lot of applications need modifications to run (that first trip to the dentist is painful)

http://www.matasano.com/log/611/gunar-petersons-os-security-features-chart/

		Windows Vista	Windows XP SP2	RHEL 4	OpenBSD 3.x	Mac OS X
images	Section Reordering			■	■	
	EXE Randomization	■		□		
	DLL Randomization	■		■	■	
stack	Frame Protection	■	□	□	□	
	Exception Protection	■	□			
	Local Variable Protection	■	□	□	□	
	Randomization	■		■	■	
	Non-Executable	■		■	■	
heap	Metadata Protection	■	■			
	Randomization	■		■	■	
	Non-Executable	■	□	■	■	

■ full support □ partial support

Vista: bad signs

- blacklisting, not whitelisting, of attachments
- DRM requirements force software breakage (see Peter Guttman's work)
- I haven't heard of useful sandboxing yet

Secure clients: *nix

- Runs firefox, thunderbird, and other giant client programs, without containment

Macintosh clients

- Have been below the radar, making it an uneconomical target
- I expect Apple to double or quadruple their current market share. Still tiny.
- Basic OS is probably a better platform
- Open source software versions lagging

Bozo in the Chair

- These attacks will continue indefinitely
- Attackers' ingenuity is endless
- Unreasonable to expect users to understand security implications of most computer decisions
  - Experts can easily lack enough data

Resistance to Secure Clients

- Many clients haven't demanded secure host
- Naive users have high tolerance for infection
  - lost weekends for techies

How has skinny dipping worked for me?

- FreeBSD and Linux hosts
- Very few, very hardened network services
- Single-user hosts
- Dangerous services placed in sandboxes
  - Much too hard to do

How has skinny dipping worked for me?

- Quite well, but I give up services
- No undetected break-ins
- Not all my hosts and services are skinny dipped

Limitations to host-level security

- Cannot stop DDoS attacks
  - so we are still going to need walled gardens
- Giving up a layer is an important security decision, once the inside is toughened

Future technologies

- Looking for virtualization of client software, in all operating systems
- Virtualization will help servers, nicely
- Beyond the DMZ: a quasi-walled garden?

40 Years of Internet Security and Perimeters

Bill Cheswick
AT&T Research
ches@research.att.com