With Microscope and Tweezers: The Worm from MIT's Perspective

The actions taken by a group of computer scientists at MIT during the worm invasion represents a study of human response to a crisis. The authors also relate the experiences and reactions of other groups throughout the country, especially in terms of how they interacted with the MIT team.

Jon A. Rochlis and Mark W. Eichin

The following chronology depicts the Internet virus as seen from MIT. It is intended as a description of how one major Internet site discovered and reacted to the virus. This includes the actions of our group at MIT which wound up decompiling the virus and discovering its inner details, and the people across the country who were mounting similar efforts.

It is our belief that the people involved acted swiftly and effectively during the crisis and deserve many thanks. Also, there is much to be learned from the way the events unfolded. Some clear lessons for the future emerged, and as usual, many unresolved and difficult issues have also risen to the forefront to be considered by the networking and computer community.[1]

WEDNESDAY: GENESIS

Gene Myers [1] of the National Computer Security Center (NCSC) analyzed the Cornell[2] mailer logs. He found that testing of the `sendmail` attack first occurred on October 19, 1988 and continued through October 28, 1988. On October 29, 1988, there was an increased level of testing; Myers believes the virus author was attempting to send the binaries over the SMTP (Simple Mail Transfer Protocol) connections, an attempt which was bound to fail since the SMTP is only defined for 7-bit ASCII data transfers [7].

The author appeared to go back to the drawing board, returning with the "grappling hook" program on Wednesday, November 2, 1988. The virus was tested or launched at 5:01:59 p.m. The logs show it infecting a second Cornell machine at 5:04 p.m. This may have been the genesis of the virus, but that is disputed by reports in the *New York Times* [4] in which Paul Graham of Harvard states the virus started on a machine at the MIT Artificial Intelligence Lab via remote login from Cornell. Cliff Stoll of Harvard also believes the virus was started from the MIT AI Lab. At the time this article was written, nobody had analyzed the infected Cornell machines to determine where the virus would have gone next if they were indeed the first infected machines.

In any case, Paul Flaherty of Stanford reported to the *tcpgroup@ucsd.edu* mailing list on Friday that Stanford was infected at 9 p.m. and that it got to "most of the campus UNIX® machines (cf. 2,500 boxes)." He also reported the virus originated from *prep.ai.mit.edu.* This is the earliest report of the virus we have seen.

At 9:30 p.m. Wednesday, *wombat.mit.edu,* a private workstation at MIT Project Athena maintained by Mike Shanzer, was infected. It was running a version of `sendmail` with the `debug` command turned on. Shanzer believes the attack came from *prep.ai.mit.edu* since he had an account on *prep* and *wombat* was listed in his `.rhosts`, a file which specifies a list of hosts and users on those hosts who may log into an account over the network without supplying a password. Unfortunately, the appropriate logs were lost, making the source of the infection uncertain. (The logs on *prep* were forwarded via `syslog`, the 4.3 BSD UNIX® logging package, to another host which was down and by the time anybody looked at the `wtmp` log, which records logins, it was truncated, perhaps deliberately, to some point on Thursday. The lack of logging informa-

[1] The events described took place between Wednesday, November 2, 1988 and Friday, November 11, 1988. All times are EST.

[2] Cornell systems personnel had discovered unusual messages in their mailer logs and passed the logs to Berkeley which passed them to the NCSC. Later it was reported that the alleged author of the virus was a Cornell graduate student [3].

® UNIX is a trademark of AT&T.

tion and the routine discarding of what old logs did exist hampered investigations.)

Mike Muuss of Ballistics Research Laboratory reported at the NCSC meeting that RAND was also hit at 9 p.m. or soon thereafter. Steve Miller of the University of Maryland (UMD) reports the campus was first hit at 10:54 p.m.; Phil Lapsley of the University of California, Berkeley, stated that UCB was hit at 11 p.m.

THURSDAY MORNING: "THIS ISN'T APRIL FIRST"

David Edwards, of SRI International, said at the NSCS meeting that SRI was hit at midnight. Chuck Cole and Russell Brand of Lawrence Livermore National Laboratory (LLNL) reported they were assembling their response team by 2 a.m., and John Bruner independently reported spotting the virus on the S1 machines at LLNL about that time.

Pascal Chesnais of the MIT Media Lab was one of the first people at MIT to spot the virus, after 10 p.m. Wednesday, but assumed it was just "a local runaway program." A group at the Media Lab killed the anomalous shell and compiler processes, and all seemed normal. After going for dinner and ice cream, they figured out that it was a virus and it was coming in via mail. Their response was to shut down network services such as mail and to isolate themselves from the campus network. The MIT Telecommunications Network Group's monitoring information shows the Media Lab gateway first went down at 11:40 p.m. Wednesday, but was back up by 3 a.m. At 3:10 a.m. Pascal gave the first notice of the virus at MIT by creating a message of the day on *media-lab.mit.edu* (see Figure 1).

False Alarms or Testing?

Chesnais later reported that logs on *media-lab* show several scattered messages, "ttloop: peer died: No such file or directory," which frequently occurred just before the virus attacked. There were a few every couple of days, several during Wednesday afternoon and many starting at 9:48 p.m.. The logs on *media-lab* start on October 25, 1988 and entries were made by `telnetd` on the following dates before the swarm on Wednesday night:

Oct. 26, 15:01:57;
Oct. 28, 11:26:55;
Oct. 28, 17:36:51;
Oct. 31, 16:24:41;
Nov. 1, 16:08:24;
Nov. 1, 18:02:43;
Nov. 1, 18:58:30;
Nov. 2, 12:23:51;
Nov. 2, 15:21:47.

It is not clear whether these represent early testing of the virus, or if they were just truly accidental premature closings of the telenet connections. We assume the latter. With hindsight we can say a `telnetd` that logged its peer address, even for such error messages, would have been quite useful in tracing the origin and progress of the virus.

> A virus has been detected on media-lab; we suspect that whole internet is infected by now. The virus is spread via mail of all things. . . . So Mail outside of media-lab will NOT be accepted. Mail addressed to foreign hosts will NOT be delivered. This situation will continue until someone figures out a way of killing the virus and telling everyone how to do it without using email. . . .
>
> —lacsap Nov 3 1988 03:10am

FIGURE 1. Thursday morning's message of the day on *media-lab.mit.edu.*

E-mail Warnings

The first posting mentioning the virus was by Peter Yee of NASA Ames at 2:28 a.m. on Wednesday to the *tcp-ip@sri-nic.arpa* mailing list. Yee stated that UCB, UC San Diego, LLNL, Stanford, and NASA Ames had been attacked, and described the use of sendmail to pull over the virus binaries, including the `x*` files which the virus briefly stored in `/usr/tmp`. The virus was observed sending VAX® and Sun® binaries, having DES tables built in, and making some use of `.rhosts` and `hosts.equiv` files. A phone number at UCB was given and Lapsley and Kurt Pires were listed as being knowledgeable about the virus.

At 3:34 a.m. Andy Sudduth from Harvard made his anonymous posting[3] to *tcp-ip@sri-nic.arpa*.[4] The posting said that a virus might be loose on the Internet and that there were three steps to take to prevent further transmission. These included not running `fingerd` or fixing it not to overwrite the stack when reading its arguments from the net,[5] being sure `sendmail` was compiled without the `debug` command, and not running `rexecd`.

Mike Patton, network manager for the MIT Laboratory for Computer Science (LCS), was the first to point out to us the peculiarities of this posting. It was made from an Annex terminal server[6] at Aiken Laboratory at Harvard, by telneting to the SMTP port of *iris.brown.edu*. This is obvious since the message was from "foo%bar.arpa" and because the last line of the message was "qui\177\177\177," an attempt to get rubout processing out of the Brown SMTP server, a common mistake when faking Internet mail.

It was ironic that this posting did almost no good. Figure 2 shows the path it took to get to Athena. There was a 43-hour delay before the message escaped from *relay.cs.net*[7] and got to *sri-nic.arpa*. Another six

® VAX, and Ultrix are trademarks of Digital Equipment Corp.
® Sun, SunOS and NFS are trademarks of Sun Microsystems, Inc.
[3] In a message to the same mailing list on Saturday, November 5, 1988, he acknowledged being the author of the Thursday morning message and stated he had posted the message anonymously because "at the time I didn't want to answer questions about how I knew."
[4] An "obscure electronic bulletin board," according to the *New York Times* [4]. Nothing could be further from the truth.
[5] This was a level of detail that only the originator of the virus could have known at that time. To our knowledge nobody had yet identified the finger bug, since it only affected certain VAX hosts, and certainly nobody had discovered its mechanism.
[6] Perhaps ironically named *influenza.harvard.edu*.
[7] This is probably because *relay.cs.net* was off the air during most of the crisis.

hours went by before the message was received by *athena.mit.edu*.[8] Other sites have reported similar delays.

Yet More People Notice the Virus

About 4 a.m. Thursday Richard Basch of MIT Project Athena noticed a "text table full" `syslog` message from *paris.mit.edu*, an Athena development machine. Since there was only one message and he was busy doing a project for a digital design lab course, he ignored it.

At 4:51 a.m. Chris Hanson of the MIT AI Laboratory reported spotting anomalous telnet traffic to several gateways coming from machines at LCS. He noted that the attempts were occurring every one or two seconds and had been happening for several hours.

At 5:58 a.m. Thursday morning Keith Bostic of Berkeley made the first bug fix posting. The message went to the *tcp-ip@sri-nic.arpa* mailing list and the newsgroups *comp.bugs.4bsd.ucb-fixes*, *news.announce*, and *news.sysadmin*. It supplied the "compile without the debug command" fix to `sendmail` (or patch the `debug` command to a garbage string), as well as the very wise suggestion to rename the UNIX C compiler and loader (`cc` and `ld`), which was effective since the virus needed to compile and link itself, and which would be effective at protecting against non-`sendmail` attacks, whatever those might have turned out to be. It also told the people that the virus renamed itself to "(sh)" and used temporary files in `/usr/tmp` named XNNN,vax.o, XNNN, sun3.o, and XNNN,l1.c (where NNN were random numbers, possibly process id's), and suggested that one could identify infected machine by looking for these files. That was somewhat difficult to do in practice, however, since the virus quickly got rid of all of these files. A somewhat better solution was proposed later in the day by, among others, John Kohl of Digital Equipment Corp. and Project Athena, who suggested doing a `cat -v/usr/tmp`, thus revealing the raw contents of the directory, including the names of deleted files whose directory slots had not yet been re-used.[9]

The `fingerd` attack was not even known, much less understood, at this point. Lapsley reported at the NCSC meeting that Ed Wang of UCB discovered the `fingerd` mechanism around 8 a.m. and sent mail to Mike Karels, but this mail went unread until after the crisis had passed.

At 8:06 a.m. Gene Spafford of Purdue forwarded Bostic's fixes to the *nntp-managers@ucbvax.berkeley.edu* mailing list. Ted Ts'o of MIT Project Athena forwarded this to an internal Project Athena hackers list (*watchmakers@athena.mit.edu*) at 10:07 a.m. He expressed disbelief ("no, it's not April 1st"), and thought Athena machines were safe. Though no production Athena servers were infected, several private workstations and development machines were, so this proved overly optimistic.

Mark Reinhold, a MIT LCS graduate student, reacted to the virus around 8 a.m. by powering off some network equipment in LCS. Tim Shepard, also a LCS graduate student, soon joined him. They were hampered by a growing number of people who wanted information about what was happening. Reinhold and Shepard tried to call Yee several times and eventually managed to get through to Lapsley who relayed what was then known about the virus.

At about this time, Basch returned to his workstation (a person can only do so much schoolwork after all) and noticed many duplicates of the "text table full" messages from *paris* and went to investigate. He discovered several suspicious logins from old accounts which should have been purged long ago. The load was intolerably high, and he only managed to get one line out of a `netstat` command before giving up, but that proved quite interesting. It showed an outgoing `rsh` connection from *paris* to *fmgc.mit.edu*, which is a standalone non-UNIX gateway.

Ray Hirschfeld of the MIT Math Department at the MIT AI Lab spotted the virus Thursday morning on the Sun workstations in the math department and shut down the math gateway to the MIT backbone at 10:15 a.m. It remained down until 3:15 p.m.

Around 11 a.m. the MIT Statistics Center called Dan Geer, manager of system development at Project Athena. One of their Sun workstations, *dolphin.mit.edu* had been infected via a Project Athena guest account with a weak password, along with the account of a former staff member. This infection had spread to all hosts in the Statistics Center. They had been trying for some time prior to call Geer to eradicate the virus, but the continual reinfection among their local hosts had proved insurmountably baffling.

Bostic sent a second virus fix message to *comp.4bsd.ucb-fixes* at 11:12 a.m. It suggested using 0xff instead of 0x00 in the binary patch to `sendmail`. The previous patch, while effective against the current virus, would drop into debug mode if an empty command

```
Received: by ATHENA.MIT.EDU (5.45/4.7) id AA29119; Sat,
          5 Nov 88 05:59:13 EST
Received: from RELAY.CS.NET by SRI-NIC.ARPA with
          TCP; Fri, 4 Nov 88 23:23:24 PST
Received: from cs.brown.edu by RELAY.CS.NET id
          AA05627; 3 Nov 88 3:47 EST
Received: from iris.brown.edu (iris.ARPA) by cs.brown.edu
          (1.2/1.00) id AA12595; Thu, 3 Nov 88 03:47:19
          est
Received: from (128.103.1.92) with SMTP via tcp/ip
          by iris.brown.edu on Thu, 3 Nov 88 03:34:46 EST
```

FIGURE 2. Path of Andy Sudduth's warning message from Harvard to MIT.

[8] Phil Lapsley and Mike Karels of Berkeley reported at the NCSC meeting that the only way to get mail to *tcp-ip@sri.nic.arpa* to flow quickly is to call up Mark Lottor at SRI and ask him to manually push the queue through.

[9] Jerry Saltzer, MIT EECS professor and technical director of Project Athena, included similar detection advice in a message describing the virus to the Athena staff sent at 11:17 a.m. on Friday.

line was sent. He also suggested using the UNIX `strings` command to look in the `sendmail` binary for the string "debug." If it didn't appear at all then that version of `sendmail` was safe.

About 11:30 a.m. Chesnais requested the Network Group isolate the Media Lab building and it remained so isolated until Friday at 2:30 p.m.

Russ Mundy of the Defense Communications Agency reported at the NCSC meeting that the MILNET to ARPANET mailbridges were shut down at 11:30 a.m. and remained down until Friday at 11 a.m.

In response to complaint from non-UNIX users, Reinhold and Stan Zanarotti, another LCS graduate student, turned on the repeaters at LCS which had been previously powered down and physically disconnected UNIX machines from the network around 11:15 a.m. Shepard reloaded a root partition of one machine from tape (to start with known software), and added a feature to `find`, a UNIX file system scanner, to report low-level modification times. Working with Jim Fulton of the X Consortium, Shepard inspected *allspice.lcs.mit.edu.* By 1 p.m. they had verified that the virus had not modified any files on *allspice* and had installed a recompiled `sendmail`.

THURSDAY AFTERNOON: "THIS IS BAD NEWS"

By the time Jon Rochlis of the MIT Telecommunications Network Group arrived for work around noon on Thursday, November 3, 1988, the Network Group had received messages from MIT Lincoln Laboratory saying they had "been brought to their knees" by the virus, from Sergio Heker of the John Von Neumann National Supercomputer Center warning of network problems, and from Kent England of Boston University saying BU had cut their external links. The MIT Network Group loathed the thought of severing MIT's external connections and never did throughout the crisis.

At 1:30 p.m. Geer and Jeff Schiller, manager of the MIT Network and Project Athena Operations Manager, returned to the MIT Statistics Center and were able to get both VAX and Sun binaries from infected machines.

Spafford posted a message at 2:50 p.m. Thursday to a large number of people and mailing lists including *nntp-managers@ucbvax.berkeley.edu*, which is how we saw it quickly at MIT. It warned that the virus used `rsh` and looked in `hosts.equiv` and `.rhosts` for more hosts to attack.

Around this time the MIT group in E40 (Project Athena and the Telecommunications Network Group) called Milo Medin of NASA and found out much of this information. Many of us had not yet seen the messages. He pointed out that the virus just loved to attack gateways, which were found via the routing tables, and remarked that it must have not been effective at MIT where we run our own C Gateway code on our routers, not UNIX. Medin also said that it seemed to randomly attack network services, swamping them with input. Some daemons that run on non-standard ports had logged such abnormal inputs. At the time we thought the virus might be systematically attacking all possible network services exploiting some unknown common flaw. This was not true but it seemed scary at the time. Medin also informed us that DCA had shut down the mailbridges which serve as gateways between the MILNET and the ARPANET. He pointed us to the group at Berkeley and Yee specifically.

It Uses Finger

At about 6 p.m. on Thursday, Ron Hoffman, of the MIT Telecommunications Network Group, observed the virus attempting to log into a standalone router using the Berkeley remote login protocol; the remote login attempt originated from a machine previously believed immune since it was running a mailer with the `debug` command turned off. The virus was running under the user name of `nobody`, and it appeared that it had to be attacking through the finger service, the only network service running under that user name. At that point, we called the group working at Berkeley; they confirmed our suspicions that the virus was spreading through `fingerd`.

On the surface, it seemed that `fingerd` was too simple to have a protection bug similar to the one in `sendmail`; it was a very short program, and the only program it invoked (using the UNIX *exec* system call) was named using a constant pathname. A check of the modification dates of both `/etc/fingerd` and `/usr/ucb/finger` showed that both had been untouched, and both were identical to known good copies located on a read-only filesystem.

Berkeley reported that the attack on finger involved "shoving some garbage at it," probably control A's; clearly an overrun buffer wound up corrupting something.

Bill Sommerfeld of Apollo Computer and MIT Project Athena guessed that this bug might involve overwriting the saved program counter in the stack frame; when he looked at the source for `fingerd`, he found that the buffer it was using was located on the stack. In addition, the program used the C library *gets* function which assumes that the buffer it is given is long enough for the line it is about to read. To verify that this was a viable attack, he then went on to write a program which exploited this hole in a benign way. The test virus sent the string "Bozo!" back out the network connection.

Mike Rowan and Mike Spitzer also report having discovered the `fingerd` mechanism at about the same time and forwarded their discovery to Spafford and Bostic, but in the heat of the moment the discovery went unrecognized. Liudvikas Bukys of the University of Rochester posted to the *comp.bugs.4bsd* newsgroup a detailed description of the `fingerd` mechanism at 7:21 p.m. The message also stated that the virus used telnet but perhaps that was only after cracking passwords. In reality it only sometimes used telnet to "qualify" a machine for later attack, and only used `rsh` and `rexec` to take advantage of passwords it had guessed.

A *risks@kl.sri.com* digest [6] came out at 6:52 p.m. It included a message from Stoll describing the spread of the virus on MILNET and suggested that MILNET sites might want to remove themselves from the network. Stoll concluded by saying, "This is bad news." Other messages were from Spafford, Peter Neumann of SRI, and Matt Bishop of Dartmouth. They described the sendmail propagation mechanism.

THURSDAY EVENING: "WITH MICROSCOPE AND TWEEZERS"

In the office of the Student Information Processing Board (SIPB), Zanarotti and Ts'o had managed to get a VAX binary and core dump from the virus while it was running on a machine at LCS.

The duo started attacking the virus. Pretty soon they had figured out the xor encoding of the text strings embedded in the program and were manually decoding them. By 9 p.m. Ts'o had written a program to decode all the strings and we had the list of strings used by the program, except for the built-in dictionary which was encoded in a different fashion (by turning on the high order bit of each character).

At the same time they discovered the IP address of *ernie.berkeley.edu*, 128.32.137.13, in the program; they proceeded to take apart the virus routine *send message* to figure out what it was sending to *ernie*, how often, and if a handshake was involved. Zanarotti told Rochlis in the MIT Network Group of the SIPB group's progress. The people in E40 called Berkeley and reported the finding of *ernie*'s address. Nobody seemed to have any idea why that was there.

At 9:20 p.m., Spafford created the mailing list *phage@purdue.edu*. It included all the people he had been mailing virus information to since the morning; more people were to be added during the next few days. This list proved invaluable, since it seemed to have many of the "right" people on it and seemed to work in near real time despite all the network outages.

At 10:18 p.m. Bostic made his third bug fix posting. It included new source code for `fingerd` which used *fgets* instead of *gets* and did an *exit* instead of *return*. He also included a more general `sendmail` patch which disabled the `debug` command completely.

The Media Descends

About this time a camera crew from WNEV-TV Channel 7 (the Boston CBS affiliate) showed up at the office of James D. Bruce, MIT EECS Professor and Vice President for Information Systems. He called Jeff Schiller and headed over to E40. They were both interviewed and stated that there were 60,000 Internet hosts,[10] along with an estimate of 10 percent infection rate for the 2,000 hosts at MIT. The infection rate was a pure guess, but seemed reasonable at the time. These numbers were to stick in a way we never anticipated. Some of the press reports were careful to explain the derivation of the numbers they quoted, including how one could extrapolate that as many as 6,000 computers were infected. However, many reports were not that good and simply stated things like "at least 6,000 machines had been hit." We were unable to show the TV crew anything "visual" caused by the virus, something which eventually became a common media request and disappointment. Instead, they settled for people looking at workstations talking "computer talk."

The virus was the lead story on the 11 p.m. news and was mentioned on National Public Radio as well. We were quite surprised that the real world would pay so much attention. Sound bites were heard on the 2 a.m. CBS Radio News, and footage shot that evening was shown on the CBS morning news (but by that point we were too busy to watch).

After watching the story on the 11 p.m. news we realized it was time to get serious about figuring out the detailed workings of the virus. We all agreed that decompiling was the route to take, though later we also mounted an effort to infect a specially instrumented machine to see the virus in operation. As Saltzer said in a later message to the Project Athena staff, we undertook a "wizard-level analysis" by going over the virus "with microscope and tweezers."

FRIDAY: "WHERE'S SIGOURNEY WEAVER?"

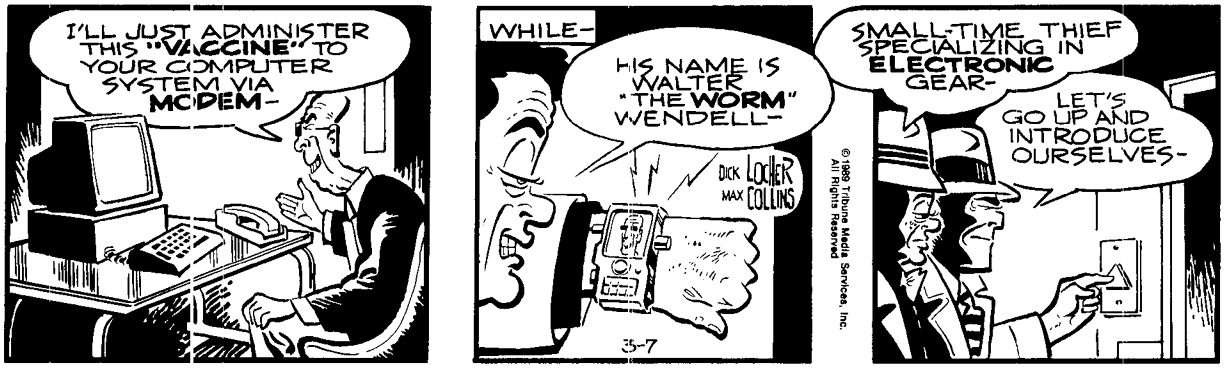
Tim Shepard joined the group in E40, just before midnight on Thursday. We thought we saw packets going to *ernie* and replies coming back, though this later proved to be an illusion. Shepard had hundreds of megabytes of packet headers gathered Thursday morning from a subnet at LCS which was known to have had infected machines on it. Unfortunately, the data was sitting on a machine at LCS, which was still off the network, so Shepard decided to go back and look through this data. Within an hour or two, Shepard called back to say that he found no unusual traffic to *ernie* at all. This was our first good confirmation that the *ernie* packets were a red-herring or at least they did not actually wind up being sent.

Serious decompiling began after midnight. Zanarotti and Ts'o soon left the SIPB office and joined the group working in E40, bringing with them the decoding of the strings and much of the decompiled main module for the virus. Mark Eichin, who had recently spent a lot of time disassembling-assembling some ROMs and thus had recent experience at reverse engineering binaries, took the lead in dividing the project up and assigning parts to people. He had also woke up in late afternoon and was most prepared for the night ahead.

At 1:55 a.m. Eichin discovered the first of the bugs in the virus. A *bzero* call in *if init* was botched. At 2:04 a.m. Zanarotti had a version of the main module that compiled. We called Bostic at Berkeley at 2:20 a.m. and arranged to do FTP exchanges of source code on an MIT machine (both Berkeley and MIT had never cut their outside network connections). Unfortunately, Bostic was unable to get the hackers at

[10] This was based on Mark Lottor's presentation to the October 1988 meeting of the Internet Engineering Task Force.

DICK TRACY®



Reprinted with permission: Tribune Media Services.

Berkeley to take a break and batch up their work, so no exchange happened at that time.

At 2:45 a.m. Eichin started working on *checkother*[11] since the Berkeley folks were puzzled by it. Rochlis was working on the later *cracksome* routines. By 3:06 a.m. Ts'o had figured out that *ha* built a table of target hosts which had telnet listeners running. By 3:17 a.m. Ts'o and Hal Birkeland from the Media Lab had determined that the *crypt* routine was the same as one found in the C library. Nobody had yet offered a reason why it was included in the virus, rather than being picked up at link time.[12] Eichin had finished *checkother* and Ts'o had finished *permute* at 3:28 a.m. We worked on other routines throughout the morning.

Observations from Running the Virus

The first method of understanding the virus was the decompilation effort. A second method was to watch the virus as it ran, in an attempt to characterize what it was doing—this is akin to looking at the symptoms of a biological virus, rather than analyzing the DNA of the virus. We wanted to do several things to prepare for observing the virus.

- Monitoring: We wanted to set up a machine with special logging, mostly including packet monitors.
- Pointers: We wanted to "prime" the machine with pointers to other machines so we could watch how the virus would attack its targets. By placing names of the target machines in many different places on the "host" computer we could also see how the virus created its list of targets.
- Isolation: We considered isolating the machines involved from the network totally (for paranoia's sake) or by a link-layer bridge to cut down on the amount of extraneous traffic monitored. True isolation proved more than we were willing to deal with at the time, since all of our UNIX workstations assume access to many network services such as nameservers and file servers. We did not want to take the time to build a functional standalone system, though that would have been feasible if we had judged the risk of infecting other machines too great.

Mike Muuss reported that the BRL group focused on monitoring the virus in action. They prepared a special logging kernel, but even in coordination with Berkeley were unable to re-infect the machine in question until Saturday.

By 1 a.m. Friday we had set up the monitoring equipment (an IBM PC running a packet monitor) and two workstations (one acting as the target, the other running a packet monitoring program and saving the packet traces to disk), all separated from the network by a link-layer bridge and had dubbed the whole setup the "virus net." We, too, were unsuccessful in our attempt to get our target machine infected until we had enough of the virus decompiled to understand what arguments it wanted. By 3:40 a.m. John Kohl had the virus running on our "virus net" and we learned a lot by watching what it did. The virus was soon observed trying telnet, SMTP, and finger connections to all gateways listed in the routing table. Later it was seen trying `rsh` and `rexec` into one of the gateways.

At 4:22 a.m., upon hearing of the virus going after yet another host in a "new" manner, Rochlis remarked "This really feels like the movie *Aliens*. So where's Sigourney Weaver?" Seeing the virus reach out to infect other machines seemed quite scary and beyond our control.

At 5:45 a.m. we called the folks at Berkeley and finally exchanged code. A number of people at Berkeley had punted to get some sleep, and we had a bit of difficulty convincing the person who answered Bostic's phone that we were not the bad guy trying to fool them. We gave him a number at MIT that showed up in

[11] The routines mentioned here are not intended to be an exhaustive list of the routines we worked on

[12] It turned out that we were wrong and the version of *crypt* was *not* the same as library version [8]. Not everything one does at 3 a.m. turns out to be right.

the NIC's whois database, but he never bothered to call back.

At this point a bunch of us went out and brought back some breakfast.

The Media Really Arrives

We had been very fortunate that the press did not distract us, and that we were thus able to put most of our time into our decompilation and analysis efforts. Bruce and the News Office did a first rate job of dealing with most of the press onslaught. By early morning Friday there was so much media interest that MIT News Office scheduled a press conference for noon in the Project Athena Visitor Center in E40.

Just before the press conference, we briefed Bruce on our findings and what we thought was important: the virus did not destroy or even try to destroy any data; it did not appear to be an "accident;" many people (especially the people we had talked to at Berkeley) had helped to solve this.

We were amazed at the size of the press conference—there were approximately 10 TV camera crews and 25 reporters. Schiller spent a good amount of time talking to reporters before the conference proper began, and many got shots of him pointing at the letters "(sh)" on the output of a `ps` command. Bruce and Schiller answered questions as the decompiling crew watched from a vantage point in the back of the room. At one point a reporter asked Bruce how many people had enough knowledge to write such a virus and, in particular, if Schiller could have written such a program. The answer was, of course, many people could have written it and yes, Schiller was one of them. The obvious question was then asked: "Where were you on Wednesday night, Jeff?" This was received with a great deal of laughter. But when a reporter stated that sources at the Pentagon had said that the instigator of the virus had come forward and was a BU or MIT graduate student, we all gasped and hoped it had not really been one of our students.

After the conference the press filmed many of us working (or pretending to work) in front of computers, as well as short interviews.

The media was uniformly disappointed that the virus did nothing even remotely visual. Several reporters also seemed pained that we were not moments away from World War III, or that there were not large numbers of companies and banks hooked up to "MIT's network" who were going to be really upset when Monday rolled around. But the vast majority of the press seemed to be asking honest questions in an attempt to grapple with the unfamiliar concepts of computers and networks. At the NCSC meeting Muuss said, "My greatest fear was that of seeing a *National Enquirer* headline: 'Computer Virus Escapes to Humans, 96 Killed.'" We were lucky that didn't happen.

Perhaps the funniest thing done by the press was the picture of the virus code printed in Saturday's edition of the *Boston Herald* [2]. Jon Kamens of MIT Project Athena had made a window dump of the assembly code for the start of the virus (along with corresponding decompiled C code), even including the window dump command itself. The truly amusing thing was that the *Herald* had gotten an artist to add tractor feed holes to the printout in an attempt to make it look like something that a computer might have generated. We are sure they would have preferred a dot matrix printer to the laser printer we used.

Bostic called in the middle of the press zoo, so we cut the conversation short. He called us back around 3 p.m. and asked for our affiliations for his next posting.[13] Keith also asked if we liked the idea of posting bug fixes to the virus itself, and we instantly agreed with glee. Bostic made his fourth bug fix posting at 5:05 p.m., this time with fixes to the virus. Again he recommended renaming `ld`, the UNIX linker.

Things began to wind down after that, though the press was still calling and we managed to put off the NBC *Today* show until Saturday afternoon. Most of us got a good amount of sleep for the first time in several days.

SATURDAY: SOURCE CODE POLICY

Saturday afternoon, November 5, 1988, the *Today* show came to the SIPB Office, which they referred to as the "computer support club" (*sic*), to find a group of hackers. They interviewed Eichin and Rochlis and used Eichin's description of what hackers really try to do on Monday morning's show.

After the *Today* show crew left, many of us caught up on our mail. It was then that we first saw Andy Sudduth's Thursday morning posting to *tcp-ip@sri-nic.arpa* and Mike Patton stopped by and pointed out how strange it was.

We soon found ourselves in the middle of a heated discussion on *phage@purdue.edu* regarding distribution of the decompiled virus source code. Since we had received several private requests for our work, we sat back and talked about what to do, and quickly reached a consensus. We agreed with most of the other groups around the country who had come to the decision not to release the source code they had reverse engineered. We felt strongly that the details of the inner workings of the virus should *not* be kept hidden, but that actual source code was a different matter. We (and others) intended to write about the algorithms used by the virus so that people would learn what the Internet community was up against. This meant that somebody could use those algorithms to write a new virus; but the knowledge required to do so is much greater than what is necessary to recompile the source code with a new, destructive line or two in it. The energy barrier for this is simply too low. The people on our team (not the MIT administration) decided to keep our source private until things calmed down; then we would consider to whom to distribute the program. A public posting of the MIT

[13] He almost got them right, except that he turned the Laboratory for Computer Science into the Laboratory for Computer Services.

code was not going to happen.

Saltzer, among others, has argued forcefully that the code itself should be publicly released at some point in the future. After sites have had enough time to fix the holes with vendor supplied bug fixes, we might do so.

Tuesday: The NCSC Meeting

On Tuesday, November 8, 1988, Eichin and Rochlis attended the Baltimore post-mortem meeting hosted by the NCSC. We heard about the meeting indirectly at 2 a.m. and flew to Baltimore at 7 a.m. Figuring there was no time to waste with silly things like sleep, we worked on drafts of this document. The meeting will be described in more detail by the NCSC, but we will present a very brief summary here.

Attending the meeting were members of the National Institute of Science and Technology (NIST), formerly the National Bureau of Standards, the Defense Communications Agency (DCA), the Defense Advanced Research Projects Agency (DARPA), the Department of Energy (DOE), the Ballistics Research Laboratory (BRL), the Lawrence Livermore National Laboratory (LLNL), the Central Intelligence Agency (CIA), the University of California at Berkeley (UCB), the Massachusetts Institute of Technology (MIT), SRI International, the Federal Bureau of Investigation (FBI), and of course, the National Computer Security Center (NCSC). This is not a complete list. The lack of any vendor participation was notable.

Three-quarters of the day was spent discussing what had happened from the different perspectives of those attending. This included chronologies, actions taken, and an analysis of the detailed workings of the virus. Meanwhile our *very* rough draft was duplicated and handed out.

The remaining time was spent discussing what we learned from the attack and what should be done to prepare for future attacks. This was much harder and it is not clear that feasible solutions emerged, though there was much agreement on several motherhood and apple-pie suggestions. By this we mean the recommendations sound good and by themselves are not objectionable, but we doubt they will be effective.

Wednesday-Friday: The Purdue Incident

On Wednesday evening, November 9, 1988, Rich Kulawiec of Purdue posted to *phage@purdue.edu* that he was making available the unas disassembler that he (and others at Purdue) used to disassemble the virus. He also made available the output of running the virus through this program. Rumor spread and soon the NCSC called several people at Purdue, including Spafford, in an attempt to get this copy of the virus removed. Eventually, the President of Purdue was called and the file was deleted. The *New York Times* ran a heavily slanted story about the incident on Friday, November 11, 1988 [5].

Several mistakes were made here. First, the NCSC was concerned about the wrong thing. The disassembled virus was not important and was trivial for any infected site to generate. It simply was not anywhere near as important as the decompiled virus, which could have very easily been compiled and run. When the MIT group was indirectly informed about this and discovered exactly what was publicly available, we wondered what was the big deal. Secondly, the NCSC acted in a strong-handed manner that upset the people at Purdue who got pushed around.

Other similar incidents occurred around the same time. Jean Diaz of the MIT SIPB forwarded a partially decompiled copy of the virus[14] to *phage@purdue.edu* at some time on Friday, November 4, 1988, but it spent several days in mail queue on *hplabs.hp.com* before surfacing. Thus it had been posted before any of the discussion of the source code release had occurred. It was also very incomplete and thus posed little danger since the effort required to turn it into a working virus was akin to the effort required to write the virus from scratch.

These two incidents, however, caused the press to think that a second outbreak of the virus had once again brought the network to its knees. Robert French, of the MIT SIPB and Project Athena, took one such call on Thursday, November 10, and informed the reporter that no such outbreak had occurred. Apparently, rumors of source code availability (the Purdue incident and Diaz's posting) led to the erroneous conclusion that enough information of some sort had been let out and damage had been done. Rumor control was once again shown to be important.

LESSONS AND OPEN ISSUES

The virus incident taught many important issues. It also brought up many more difficult issues which need to be addressed in the future.

The Community's Reactions

The chronology of events is interesting. The manner in which the Internet community reacted to the virus attack points out areas of concern or at least issues for future study.

- *Connectivity was important.* Sites which disconnected from the network at the first sign of trouble hurt themselves and the community. Not only could they not report their experiences and findings, but they couldn't get timely bug fixes. Furthermore, other sites using them as mail relays were crippled, thus delaying delivery of important mail, such as Sudduth's Thursday morning posting, until after the crisis had passed. Sites like MIT and Berkeley were able to collaborate in a meaningful manner because they never took themselves off the network.
- *The "old boy" network worked.* People called and sent electronic mail to the people they knew and trusted and much good communication happened. This cannot be formalized but it did function quite well in the face of the crisis.

[14] This was the work of Don Becker of Harris Corporation.

DICK TRACY®

Reprinted with permission: Tribune Media Services.

- *Late night authentication is an interesting problem.* How did you know that it really is MIT on the phone? How did you know that Bostic's patch to `sendmail` is really a fix and isn't introducing a new problem? Did Bostic really send the fix or was it his evil twin, Skippy?
- *Whom do you call?* If you need to talk to the manager of Ohio State University network at 3 a.m., whom do you call? How many people can find that information, and is the information up to date?
- *Speaker phones and conference calling proved very useful.*
- *How groups formed and who led them is a fascinating topic for future study.* Don Alvarez of the MIT Center for Space Research presented his observations on this at the NCSC meeting.
- *Misinformation and illusions ran rampant.* Muuss categorized several of these at the NCSC meeting. Our spotting of a handshake with *ernie* is but one example.
- *Tools were not as important as one would have expected.* Most of the decompiling work was done manually with no more tools than a disassembler (`adb`) and an architecture manual. Based on its experience with PC viruses, the NCSC feels that more sophisticated tools must be developed. While this may be true for future attacks, it was not the case for this attack.
- *Source availability was important.* All of the sites which responded quickly and made progress in truly understanding the virus had UNIX source code.
- *The academic sites performed best.* Government and commercial sites lagged behind places like Berkeley and MIT in figuring out what was going on and creating solutions.
- *Managing the press was critical.* We were not distracted by the press and were able to be quite productive. The MIT News Office did a fine job keeping the press informed and out of the way. Batching the numerous requests into one press conference helped tremendously. The Berkeley group, among others, reported that it was difficult to get work done with the press constantly hounding them.

General Points for the Future

More general issues have popped to the surface because of the virus. These include the following:

- *Least privilege.* This basic security principle is frequently ignored and this can result in disaster.
- *"We have met the enemy and he is us."* The alleged author of the virus has made contributions to the computer security field and was by any definition an insider; the attack did not come from an outside source who obtained sensitive information, and restricting information such as source code would not have helped prevent this incident.
- *Diversity is good.* Though the virus picked on the most widespread operating system used on the Internet and on the two most popular machine types, most of the machines on the network were never in danger. A wider variety of implementations is probably good, not bad. There is a direct analogy with biological genetic diversity to be made.
- *"The cure shouldn't be worse than the disease."* Chuck Cole made this point and Stoll also argued that it may be more expensive to prevent such attacks than it is to clean up after them. Backups are good. It may be cheaper to restore from backups than to try to figure out what damage an attacker has done [1].
- *Defenses* must *be at the host level, not the network level.* Muuss and Stoll have made this point quite eloquently [1]. The network performed its function perfectly and should not be faulted; the tragic flaws were in several application programs. Attempts to fix the network are misguided. Schiller likes to use an analogy with the highway system: anybody can drive up to your house and probably break into your home, but that does not mean we should close down the roads or put armed guards on the exit ramps.
- *Logging information is important.* The `inetd` and `telnetd` interaction logging the source of virus attacks turned out to be a lucky break, but even so many sites did not have enough logging information available to identify the source or times of infection.

This greatly hindered the responses, since people frequently had to install new programs which logged more information. On the other hand, logging information tends to accumulate quickly and is rarely referenced. Thus it is frequently automatically purged. If we log helpful information, but find it is quickly purged, we have not improved the situation much at all. Muuss points out that frequently one can retrieve information from backups [1], but this is not always true.

- *Denial of service attacks are easy.* The Internet is amazingly vulnerable to such attacks. These attacks are quite difficult to prevent, but we could be much better prepared to identify their sources than we are today. For example, currently it is not hard to imagine writing a program or set of programs which crash two-thirds of the existing Sun Workstations or other machines implementing Sun's Network Filesystem (NFS). This is serious since such machines are the most common computers connected to the Internet. Also, the total lack of authentication and authorization for network level routing makes it possible for an ordinary user to disrupt communications for a large portion of the Internet. Both tasks could be easily done in a manner which makes tracking down the initiator extremely difficult, if not impossible.
- *A central security fix repository may be a good idea.* Vendors *must* participate. End users, who likely only want to get their work done, must be educated about the importance of installing security fixes.
- *Knee-jerk reactions should be avoided.* Openness and free flow of information is the whole point of networking, and funding agencies should not be encouraged to do anything damaging to this without very careful consideration. Network connectivity proved its worth as an aid to collaboration by playing an invaluable role in the defense and analysis efforts during the crisis despite the sites which isolated themselves.

The preceding article is part of a detailed report by the authors entitled "With Microscope and Tweezers: An Analysis of the Internet Virus of November 1988." A version of the paper was presented at the 1989 IEEE Symposium on Research in Security and Privacy.

REFERENCES

1. Castro, L., et al. Post mortem of 3 November ARPANET/MILNET attack. National Computer Security Center, Ft. Meade, Md., November 1988.
2. Computer whiz puts virus in computers. *Boston Herald* (Nov. 5, 1988), 1.
3. Markoff, J. "Author of computer 'virus' is son of U.S. electronic security expert." *New York Times* (Nov. 5, 1988), A1.
4. Markoff, J. Computer snarl: A "back door" ajar. *New York Times* (Nov. 7, 1988), B10.
5. Markoff, J. U.S. is moving to restrict access to facts about computer virus. *New York Times* (Nov. 11, 1988), A28.
6. Neumann, P.G. ed. Forum of risks to the public in computers and related systems. 7, 69. ACM Committee on Computers and Public Policy, November 3, 1988.
7. Postel, J.B. Simple mail transfer protocol. Request For Comments NIC/RFC 821. Network Working Group, USC ISI, August 1982.
8. Spafford, E.H. The internet worm program: An analysis. *ACM SIGCOM 19* (Jan. 1989).

CR Categories and Subject Descriptors: C.2.0 [**Computer-Communication Networks**]: Security and Protection; C.2.3 [**Computer-Communication Networks**]: Network Operations; D.4.6 [**Operating Systems**]: Security and Protection—*BSD UNIX*; K.6.m [**Management of Computer and Information Systems**]: Miscellaneous—*security*; K.4.2 [**Computers and Society**]: Social Issues—*abuse and crime involving computers*

General Terms: *Security*

Additional Key Words and Phrases: "Back door" consequences, security penetration reactions

ABOUT THE AUTHORS:

JON A. ROCHLIS is an Assistant network manager for M.I.T. He is responsible for software aspects of managing a growing network of several thousand hosts and interconnecting with regional and national networks. His interests range from routing protocols to network management to security. Author's Present Address: MIT, 77 Massachusetts Ave., E40-311, Cambridge, MA 02139.

MARK W. EICHIN was an systems programmer for M.I.T.'s Project Athena designing and writing network and graphics software. Author's Present Address: Mark Eichin, 8816 Hunting Lane, #104, Laurel, MD 20708.

DICK TRACY®

Reprinted with permission: Tribune Media Services.