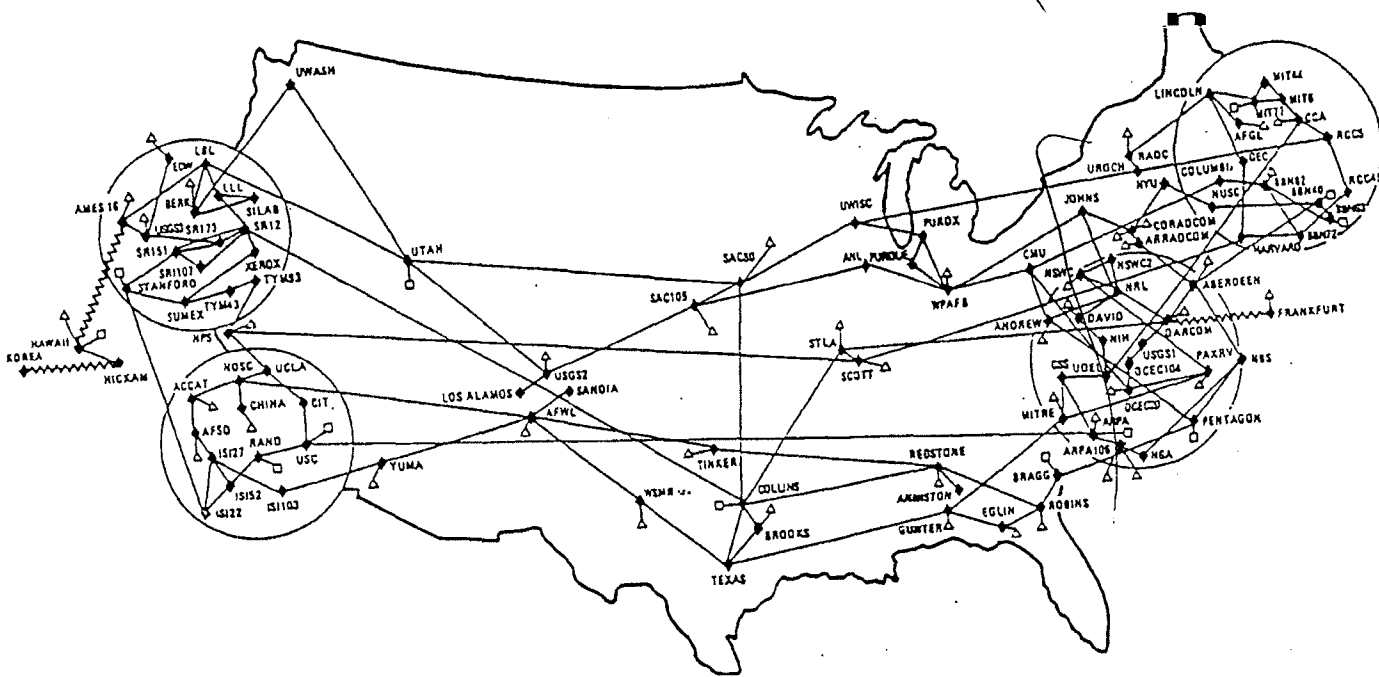


National Computer Security Center
PROCEEDINGS
of the
VIRUS POST-MORTEM MEETING
8 November 1988



ARPANET / MILNET Computer Virus Attack
of
3 November 1988

TABLE OF CONTENTS

MEMORANDUM

RECOMMENDATIONS

AGENDA

INTRODUCTION

THE VIRUS

CHRONOLOGY OF EVENTS

SITE EXPERIENCES

NATIONAL COMPUTER SECURITY CENTER

FORT GEORGE G. MEADE. MARYLAND 20755-6000

Serial: C3-0021-88
14 November 1988

MEMORANDUM FOR DISTRIBUTION

SUBJECT: 8 November Post-Mortem Meeting on the
ARPANET/MILNET Virus Propagation - INFORMATION
MEMORANDUM

The National Computer Security Center (NCSC) hosted a meeting on 8 November 1988 of highly respected researchers from government and university research facilities for the purpose of documenting their unique contribution in categorizing and resolving the recent virus attack. Representatives from Air Force, Army, ASD (C³I), CIA, DARPA, DCA, DOE, FBI, NIST, NCSC, NSA, and their colleagues from academia, recounted their site experiences and shared their respective approaches to thwarting the propagation and purging the virus from their systems. The sharing of information that took place at this meeting was unprecedented and reflected very positively on all participants. The high degree of professionalism and dedication by those involved, particularly in the university research community, was the key to rapidly understanding and ending the propagation of this virus. In the pages that follow, our editors have captured the essence and record of the meeting's presentations and discussions. Some of the material is obviously in "early draft" form; however, we believe that the value of these proceedings will be in its timely dissemination as opposed to its format quality.

This virus attack was the first occurrence of a virus propagating autonomously via a network and affecting host computers throughout the United States. The goal of the post-mortem was to examine this virus incident in depth and develop an assessment of U.S. capability to react and recover from future attacks of this nature. While the DoD ARPANET/MILNET was the focus in this incident, the lessons learned are generic and applicable to all networks or distributed computing systems processing classified or unclassified data.

Serial:

The attendees developed the 11 attached recommendations to reduce the vulnerability of U.S. Government and private networks to virus attack. All unanimously agreed with the recommendations and concluded that the computer security community faces an urgent responsibility to develop the capability to rapidly respond to subsequent attacks. In response to this charge the NCSC in conjunction with the NIST is developing a detailed implementation plan for these recommendations.

Sincerely,

A handwritten signature in cursive script, appearing to read "Lawrence Castro".

LAWRENCE CASTRO
Chief

Research and Development

Encl:
a/s

RECOMMENDATIONS FROM THE 8 NOVEMBER 1988
POST MORTEM OF THE ARPANET/MILNET VIRUS PROPAGATION

1. Establish a centralized coordination center.
This center, supported jointly by NIST and NSA, would also function as a clearinghouse and repository. Computer site managers need a place to report problems and to obtain solutions. This center might evolve into a national level command center supporting the government and private sector networks. The center needs to provide 24 hour service, but not necessarily be manned 24 hours a day (i.e., responding via beeper after hours might be acceptable).
2. Establish an emergency broadcast network.
In the ARPANET/MILNET case, the network was used to disseminate the patches (i.e., antidote) at the same time the virus was still actively propagating. If the net had gone down, there would have been no way to coordinate efforts and disseminate patches. It is recommended that a bank of telephone lines be designated as an emergency broadcast network. The phones would be connected to digital tape recorders and operate in a continuous broadcast mode (or a recorded "binary" announcement mode) to disseminate network status, patches, etc.
3. Establish a response team.
The technical skills required to quickly analyze virus code and develop antidotes or system patches are highly specialized. The skills required are system specific (i.e., UNIX 4.3 in this case), and in many cases exist only at vendor development facilities (e.g., the majority of commercial operating systems are proprietary and source code is not provided to users). The concept of a response team would require advance coordination so that personnel with the requisite skills can be quickly mobilized.
4. Maintain technical relationships with the computer science "old boy network".
The ARPANET/MILNET virus was analyzed and eradicated through the services of this old boy network, not by U.S. Government (USG) personnel. This old boy network is willing to participate in supporting USG initiatives; however, their consensus, support, and trust is required.
5. Centrally orchestrate press relations.
An inordinate amount of time at virtually every site was spent responding to the news media. Multiple press reporting from geographically dispersed sites has the potential for circular reporting of incorrect and misleading data. A single USG focal point at the national level to interact with the press is recommended.

ENCLOSURE

6. Develop etandard procedures for "trusted fixes."
During this recent event, several different fixes or patches to the virus were disseminated to users. There was no method available to determine if the fix was to be trusted (i.e., to authenticate the purported origin of the fix and determine whether the patch itself contained malicious code). A related issue concerns the legal liability of the individual or organization developing and promulgating the fix in the event it causes undesired results. A good samaritan exclusion is desired.

7. Designate a centralized repository for virus infection reports.
The National Computer Security Center (NCSC) has designated a bulletin board on Dockmaster as a central repository for this purpose.

8. Include law enforcement agencies in the planning and implementation phases.
The response and recovery from viral attacks will generate information which may be evidence from the legal perspective. Their input is needed. Participation in response teams should be an option.

9. Training for system operators.
Many system operators lacked the technical ability to understand that a virus had attacked their system. Similarly, those same system operators had difficulty in administering the antidote. It is recommended that standards be established and a training program started. A similar event occurred during the 1986 German hacker penetration of ARPANET/MILNET; i.e., the system operators when informed that their system had been penetrated refused to believe it.

10. Establish etandard backup policies.
The conventional methodology of routinely performing a system backup by saving a "mirror" image on disk, would have been disastrous in the case of this particular virus because the virus would have unwittingly been included on the backup. New standards and criteria for backup should be developed and promulgated by NIST or the NCSC.

11. Develop a common set of virus analysis tools.
The analysis of a virus is initiated by reverse engineering the virus code. The reverse engineering of software is complicated, tedious, and computer specific. A common set of virus analysis tools needs to be developed and available for use by the quick response team.

Caveat: All of the recommendations must be implemented within the constraints of PL 100-235. PL 100-235 assigns responsibilities in computer security to NIST for unclassified systems and the National Security Agency for classified systems. These recommendations clearly fall into both areas.

POST MORTEM OF 3 NOVEMBER ARPANET/MILNET ATTACK

Tuesday, 8 November, 0900

AGENDA

WELCOME	L. Castro
KICKOFF	P. Gallagher
INTRODUCTION	D. Vaurio
SITE EXPERIENCES	
HARVARD	C. Stoll
LAWRENCE LIVERMORE	C. Cole
BERKELEY	P. Lapsley
MIT	D. Alvarez M. Eichen J. Rochlis
LOS ALAMOS NATIONAL LABS	A. Baker
DCA/DDN	G. Mundy
ARMY BALLISTICS RESEARCH LAB	M. Muuss
SRI	D. Edwards
HOW THE ATTACK WORKS	
INTRODUCTION	G. Meyers
CONTRAST WITH OTHER VIRUSES	J. Beckman
RECOMMENDATIONS	R. Brand
DISCUSSION: A GOVERNMENT MALICIOUS CODE INFORMATION NETWORK	
D. Vaurio	P. Fonash
S. Katzke	W. Scherlis
C. Stoll	L. Wheeler

INTRODUCTION

On Wednesday, 2 November 1988, a sophisticated virus attacked host computers throughout the MILNET and the ARPANET computer network communication systems and significantly reduced computer operations at many facilities. Host managers and software experts responded effectively to this challenge. They identified the virus attack routes, analyzed the virus software, developed antidotes, and communicated information about both the attacks and antidotes to other sites. Defensive software was in place and the virus largely purged from the network within 48 hours.

The National Computer Security Center (NCSC) hosted a meeting on Tuesday, 8 November 1988, to review and document the virus attack and its subsequent solution. Over 75 researchers and administrators from government, industry, and university computer facilities recounted their experiences and shared their approaches to stopping the propagation of the virus and purging the virus from their computer systems. This document is a summary of their reports. We would appreciate comments concerning errors or omissions; please contact Dr. C. Terrence Ireland at the NCSC on 301-859-4485.

THE VIRUS

Once introduced into a host computer the virus can automatically propagate itself to other hosts using several different mechanisms. The virus can use a documented feature in the sendmail program that was intended for use during program development. Sendmail is UNIX user interface to the network mail system. A debugging feature in sendmail allows a user to send a program to a host which then goes directly into execution bypassing the standard login procedure.

The virus can use a program error in the finserd program. Finaerd allows a UNIX user to query a remote host about its current activity or the profile of a specific user. The error occurs when specific (and improper) data is passed into the program. When finserd quits, a rogue program contained in the passed data goes into execution.

The virus can masquerade as a legitimate user by discovering a user's password that was not carefully constructed, logging on as that user and starting the entire infection process over. The virus uses host tables maintained by the system and by its legitimate users to select other hosts and gateways to attack. It takes advantage of high levels of trust between remote hosts frequently accessed by users who can connect to trusting hosts without manually having to go through the login procedure.

CHRONOLOGY OF EVENTS

The following chronology is compiled from presentations at the 8 November 1988 Post Mortem review. As in any historical analysis, it is difficult to determine the exact sequence of events.

The format gives the Eastern Standard Time (EST) of the event in the left-hand column, followed by the reported time of the event in parentheses if the report came from a different time zone, then a short description of the event followed by a parenthesized list of the people reporting it. The following list of abbreviations is used extensively.

BRL Army Ballistic Research Laboratory
DCA Defense Communications Agency
DOE Department of Energy
LANL Los Alamos National Laboratory
LLL Lawrence Livermore Laboratory
NASA National Aeronautic and Space Administration
UCB University of California, Berkeley
UCD University of California, Davis UCSD University of California, San Diego

Wednesday, 2 November 1988

1700		Cornell detects virus (Stoll, Myers)
1830		University of Pittsburgh infects RAND (Myers)
2100	(1800 PST)	Stanford and RAND detect virus (Stoll)
2100	(1800 PST)	BRL hears of virus (Muuss)
2200	(1900 PST)	UCB detects virus (Muuss)
2300		Virus spreads from MIT AI Labs (Stoll)
2328	(2028 PST)	Peter Yee sends first notice that UCB, UCSD, LLL, Stanford and NASA Ames have been attacked by a virus (Rochlis)
2345		Virus enters VGR.BRL.MIL at BRL (Muuss)

Thursday, 3 November 1988

0000	(2100 PST)	UCB shuts off <u>sendmail</u> , <u>finserd</u> , etc. (Muuss)
0100		More than 15 ARPANET hosts infected (Stoll)
0105	(2205 PST)	Virus attacks LLL (Cole)
0200		Harvard detects virus (Stoll)
0300		Virus spreads from VGR.BRL.MIL (Muuss)
0300		Virus spreads into most subnets (Stoll)
0310		MIT detects virus (Rochlis)
0330	(0030 PST)	LLL begins virus analysis (Cole)

0334 Virus threat posting from Harvard to TCP-IP with sendmail, finserd, and rexecd warnings; requires 26 hours to reach MIT

0400 Network overloading slows spread of virus; Approximately 1000 hosts infected (Stoll)

0400 (0100 PST) UCB fixes sendmail problem (Lapsley)

0400 (0100 PST) LLL believes problem serious enough to consider disconnecting from network (Cole)

0400 MIT Athena Project detects virus (Schiller)

0448 (0148 PST) LLL disconnects from network (Cole)

0500 Stoll alerts MILNET and ARPANET operations centers (Stoll)

0515 MILNET monitoring center notified of virus by University of Pittsburgh (Mundy)

0530 (0230 PST) LLL notifies DOE Headquarters (Cole)

0600 (0300 PST) UCB posts sendmail antidote on TCP-IP, USENET bulletin boards (Lapsley)

0600 (0300 PST) UCB contacts UCD (Cole)

0630 (0330 PST) LLL installs sendmail antidote on VAX host but it does not prevent reinfection (Cole)

0645 Stoll calls NCSC (Stoll)

0800 Smithsonian Astrophysical Center detects virus (Stoll)

0800 UCB identifies finserd problem (Lapsley)

0806 UCB sendmail fix forwarded to nntp-managers@ucbvax.berkeley.edu (Rochlis)

0900 (0700 MST) DOE Headquarters notifies Los Alamos (Baker)

1000 DOE Headquarters advises its 7 ARPANET hosts to leave the net (Vaurio)

1000 (0700 PST) LLL holds first press conference (Cole)

1000 BRL disconnects from MILNET, DISNET, NSI (Muuss)

1007 MIT receives UCB sendmail fix to MIT Project Athena (Rochlis)

1015 MIT Math department detects virus and shuts down gateway to their Suns (Rochlis)

1028 (0728 PST) NCSC requests copy of virus from LLL (Cole)

1100 MIT begins work on virus (Rochlis)

1130 DCA inhibits mail bridges between ARPANET and MILNET (Mundy)

1130 (0830 PST) LLL tells Lab Directors to remove their hosts from the network (Cole)

1200 BRLNET completes internal checking for virus, concludes virus no longer present (Muuss)

1500 (1300 MST) LANL first receives antidotes (Baker)

1500 (1200 PST) LLL installs antidote and restarts internal networks (Cole)

1500 Antidote published (Stoll)

1800 (1600 MST) LANL receives antidotes (Baker)

1800 MIT observes virus using the finserd attack
(Rochlis)
1852 Risks digest seen at MIT. Includes Stoll
message describing spread and other messages
describing sendmail propagation mechanism
(Rochlis)
2000 (1700 PST) UCB begins decompilation of finserd component
(Lapsley)
2100 MIT decodes most of virus strings; sees the
net address ernie.berkeley.edu to whom the
virus was supposed to send messages
(Rochlis)
2100 First press interviews at MIT (Rochlis)
2300 BRL connects protected host to MILNET in
effort to capture virus (Muuss)

Friday, 4 November 1988

0000 (2100 PST) UCB posts finaerd antidote on TCP-IP, USENET
bulletin boards (Lapsley)
0500 MIT finishes decompilation (Rochlis)
0900 (0600 PST) UCB finishes virus decompilation (Lapsley)
1100 Mailbridges returned to service (Mundy)
1200 (0900 PST) LLL back on network (Cole)
1800 Virus pretty much eliminated (Stoll)

Saturday, 5 November 1988

0030 BRL captures virus in protected host (it's
still out there) (Muuss)

Monday, 7 November 1988

0600 Analysis completed by BRL on 2 virus modules
(Muuss)
1200 BRL "Vulnerability Sweep" programs operating
(Muuss)
1600 Antidotes installed at BRL (Muuss)

Tuesday, 8 November 1988

0900 Post Mortem Review at NCSC

SITE EXPERIENCES

Researchers directly involved with analyzing and stopping the virus attack shared their experiences during a Post Mortem Review at the National Computer Security Center. The following is a summary of their accounts presented at the 8 November 1988 Review.

HARVARD-SMITHSONIAN CENTER FOR ASTROPHYSICS

Personnel were alerted to the situation during the early morning hours on Thursday, 3 November 1988 when the virus was first seen at Harvard. Researchers who responded to the call soon realized that there had been continual network reinfection suggesting that the virus was being spread by the sendmail utility in the UNIX BSD 4.3 and related operating systems.

Five hours later that day the virus reinfected this site. Personnel spent the rest of the day trying to eradicate the virus using the antidote that had been sent our over the network, and dealing with press media inquiries.

Harvard researchers were frustrated in combatting the virus by the lack of coordination with other sites experiencing the same problem; the lack of communication with sites that had been disconnected from the network; the slow network response caused by the saturation of the network by virus packets passing between hosts; and the variety of tactics used by the virus to spread among the hosts.

Harvard researchers provided much-needed assistance to the community by suggesting methods for host cleanup and urging users to change their passwords.

LAWRENCE LIVERMORE LABORATORIES (LLL) OF THE DEPARTMENT OF ENERGY

The LLL security force called the appropriate Laboratory officials just before midnight on Wednesday, 2 November 1988, to report a serious problem with the Laboratory's computer systems. After arriving on the scene the LLL officials assembled a six-person **VIRUS** team as soon as possible and set up a response center to deal with the situation. The six-person team began exploring LLL computer facilities, all the while maintaining close contact with their University of California, Berkeley (UCB) counterparts.

When officials were convinced that the problem was serious enough to sever network connections to prevent internal spreading of the virus, the people responsible for the various interface connections were instructed to disconnect them. At that point UCB researchers informed LLL by phone that they were working on a

fix for the sendmail problem. A fix was later installed on a VAX which was then reconnected to the network to determine if the fix would prevent reinfection -- it did not. LLL officials then notified DOE headquarters and the University of California, Davis.

A memo was distributed to LLL employees as they arrived for work at the laboratory's three entrance gates. The memo advised everyone to turn on their machines. As the workday began, press inquiries multiplied and the LLL community received an update on the virus situation. LLL laboratory directors were told to disconnect from the network: fixes were described at a meeting with 300 people. By noon Thursday the fixes had been installed on all of the LLL computers and they were brought back on line. Later that day a final press conference was held. Not long after the press conference, LLL's DOE headquarters was again called and again headquarters reported that it had not been hit by the virus.

LLL reported that a test fix had been created and was running. LLL expected to know whether the fix worked by late in the day on 8 November 1988. Because the virus probes a password file, all LLL users are in the process of changing their passwords on all systems.

UNIVERSITY OF CALIFORNIA, BERKELEY

Researchers first noticed that their machines had been attacked shortly after dusk (PST) on Wednesday, 2 November 1988. Within a few hours they had determined that the systems involved included, among others, sendmail and telnet. They were able to determine what the virus was doing through a network message from NASA Ames and phone contacts with LLL. UCB researchers were able to work out an initial fix to disable the debug option in the sendmail system. They later sent out a second fix.

Very early Thursday morning, UCB researchers had observed a second virus attack using the finard system and by early evening began decompiling that virus component. The decompiling process lasted into the early morning hours on Friday. Three UCB terminals were still decompiling as of Monday.

The UCB spokesman was quick to acknowledge that he and his colleagues had received expert assistance in the decompiling effort from members of the Berkeley UNIX workshop attendees who, luckily, happened to be in town.

LOS ALAMOS NATIONAL LABORATORY (LANL) OF THE DEPARTMENT OF ENERGY

The DOE Center for Computer Security received the first word on the virus on Thursday, 3 November 1988. When they learned of

the virus, LANL researchers gathered information from DOE headquarters and LLL, then devoted their efforts to analyzing the virus. By the time LANL had learned of the virus attack, others in the computer security community already had been working on virus fixes.

The LANL effort was hampered by a lack of timely information. Most of the information they received was inaccurate and they seldom received followup information. LANL researchers received conflicting information on the fixes; they did not receive a copy of the first patch until Thursday evening. Since LANL does not have a UNIX expert on site, it was difficult to figure out which fixes would work and which would not, whether the fix was reliable, and who had originated the patch. LANL had difficulty dealing with information being passed from on nontechnical person to another and the technical people had problems interpreting this information effectively.

DEFENSE COMMUNICATIONS AGENCY (DCA)

The MILNET monitoring center, housed at DCA, was notified of the virus attack early Thursday morning. Just before noon on Thursday, the ports on both sides of the mail bridges were looped back to prevent any traffic flow between the ARPANET and the MILNET. DCA received phone calls from the Army Ballistic Research Laboratory (BRL) about once every 3 hours. The MILNET was looped back at 1130 a.m. on Thursday and opened early on Friday morning at BRL's request. The rest of the machines were turned back on later on Friday.

The Network Operations Center was not able to identify this virus attack: monitoring the system usage did not yield the necessary information. It is not unusual for a host (or several hosts) to go down on the MILNET or ARPANET. If DCA receives a call about an ARPANET problem, they take it seriously. In this instance they received no calls until early Thursday morning and saw no indication of a virus. The MILNET and ARPANET monitoring centers do receive constant information on network status, but the propagation of the virus appeared to be routine host activity.

DCA is in the process of evaluating the impact of the virus attack and has instructed personnel to set up a mailbox to collect information. The INTERNET address of the infected machines should be useful. DCA researchers are particularly interested in the impact of the virus on the MILNET.

Operations personnel on the MILNET and the ARPANET are concerned about the lack of administrative reporting.

ARMY BALLISTICS RESEARCH LABORATORY (BRL)

BRL researchers first learned of the virus from the attack on RAND on Wednesday. Early on Thursday BRL received phone calls notifying them that the virus had infected other sites, and later that day they began a coordinated effort with various sites. BRL researchers said that their contribution was fairly modest. The virus attacked only one or two BRL hosts. BRL personnel responsible for installing computer systems must adhere to a U.S. Army regulation which states that each host must defend its own host-to-network interface. Every host is set up to defend itself. The mechanisms to block improper entry attempts and to log all entry attempts are built into every host. Since most weapons systems for the year 2000 are being designed at BRL, researchers are forced to take a very conservative approach to computer security.

BRL was able to develop a protected or "test cell" host which they placed back on the network in an effort to capture the virus for analysis. The protected host was placed on the network very late on Thursday evening, but did not capture the virus until early Saturday morning. By noon on Monday they had created vulnerability sweeping modules to check their machines for infestations of the virus. They will reconnect all of their machines to the network once they believe their machines to be clean and protected (most likely, around noon on Tuesday, 8 November 1988).

The effort expended at BRL was estimated to be 500 work-hours. Six four-line telephones were in active use throughout the entire effort. BRL was especially concerned about the virus attack to recover user passwords. They suggested that Berkeley do a code review of this problem.

SRI INTERNATIONAL (SRI)

SRI became aware of the virus late Wednesday night via information received from other infected sites. The SRI Computer Science Laboratory gateway was down for about 2 hours on Thursday morning with several other gateways down until Friday morning. The Computer Science Laboratory remained largely unaffected due to the lack of host table entries. However, the virus had been detected because of unusual command usage and excessive audit entries. Personnel were able to examine finserd and to determine how they had been infected. The virus problem consumed an estimated 3 workhours to shut down the gateway, correct the mailers, clean up the system and return to service.

Since the virus attacked only a small Sun network, SRI researchers feel lucky. Personnel are in the process of downloading to the Suns and hope to use the Sun audit data to

detect the virus path. If the virus had entered the main server, SRI feel that could have done considerable damage.

SRI researchers are working on a real time intrusion-detection expert system called IDES sponsored by a DoD computer security program. The IDES team feels that an IDES-enhanced prototype would have detected the sendmail attack as it would have noted the compiler and command usage by finard, the excessive audit records, and the input-output and CPU usage. Sendmail connects to standard network ports only. The virus was using nonstandard ports to download its binary images. A system such as IDES could have detected the usage of nonstandard ports.

The communication and coordination problem existed at SRI as it did at other sites. System managers needed more instruction. Suggested actions included establishing a better notification and coordination system and general procedures to follow for the INTERNET hosts.

ATTENDEES

Title	FirstName	LastName	Organization	Address	Phone
Mr.	Don	Alvarez	MIT-Center for Space Research 37-618	77 Massachusetts Ave. Cambridge, MA 02139 Boomer@space.mit.edu	617-253-7457
CPT	Bill	Arbaugh	HQDA, OCSA	Attn: CSDS-AI Washington, D.C. 20310 Arbaugh@pentagon-ai.army.mil	202-694-6912
Ms.	Beth	Babyak	FBI-HQ	10th & Pennsylvania Ave., NW Rm 8391, TL245 Washington, D.C. 20535	
Mr.	Dave	Bailey	DOE	Production Operations Division P.O. Box 5400 Albuquerque, NM 87115 DB@a.lanl.gov	505-846-4600
Ms.	Alice	Baker	DOE	P. O Box 1663 MS E541 Los Alamos, NM 87545 Alb@lanl.gov	505-665-2577
Mr.	Joseph	Beckman	NCSC Attn: C31	9800 Savage Road Ft. Meade, MD 20755-6000 Beckman@dockmaster.arpa	301-859-4489
SA	Paul	Boedges	HQAFOSI/IVSC	Bolling AFB Washington, D.C. 20332-6001	202-767-5847
Dr	Russell L.	Brand	Lawrence Livermore National Labs	1862 Euclid Ave., Suite 136 Berkeley, CA 94709 Brand@llnl-crq.llnl.gov	415-548-136L
Dr	Leon	Breault	DOE	Washington, D.C. 20545	301-353-4255
Mr.	Brute	Calkins	NCSC Attn: C31	9800 Savage Road Ft. Meade, MD 20755-6000 BCalkins@dockmaster.arpa	301-859-4488

Title	FirstName	LastName	Organization	Address	Phone
Mr.	Larry	Castro	NCSC Attn: C3	9800 Savage Road Ft. Meade, MD 20755-6000	301-859-4485
SA	Jim	Christy	HQ AFOSI/IVSC	Bolling AFB Washington, D.C. 20332	202-767-5847
Mrs.	Judi	Citrenbaum	NCSC Attn: C34	9800 Savage Road Ft. Meade, MD 20755-6000	301-859-4486
Mr.	Chuck	Cole	Lawrence Livermore National Labs	P.O. Box 808 Livermore, CA 94550 Cole@llnl-crg.llnl.gov	
Mr.	William	Collins	NCSC Attn: C34	9800 Savage Road Ft. Meade, MD 20755-6000	301-859-4486
Mr.	Jared	Dreicer	DOE	P.O. Box 1663 MSE541 Los Alamos, NM 87545 Jzsd@lanl.gov	505-667-0005
Mr.	Dave	Eastep	Attn: T33	9800 Savage Road Ft. Meade, MD 20755-6000	301-688-5456
Mr.	David	Edwards	DCA	Code B602 McLean Washington, D.C. 20305-2000 DLE@cs.sri.com	703-285-5206
Dr.	Mark	Eichin	MIT Project Athena	4 Ames Street, Nichols 201 Cambridge, MA 02139 Eichin@athena.mit.edu	617-253-7788
Mr.	Paul	Esposito	Attn: T44	9800 Savage Road Ft. Meade, MD 20755-6000	
Mr.	Steven D.	Fleshman	Attn: X21	9800 Savage Road Ft. Meade, MD 20755-6000 Fleshman.xeva@dockmaster. arpa	301-688-5726

Title	FirstName	LastName	Organization	Address	Phone
Mr.	Pete	Fonash	DCA	Code H102 8th & Courthouse Road Arlington, VA Fonash@edn-vax.dca.mil	202-746-3642
Mr.	Paul	Franceus	NCSC Attn: C321	9800 Savage Road Ft. Meade, MD 20755-6000 Franceus@tvcho.arpa	301-859-4491
S.A	J. Michael	Gibbons	FBI-WMFO	300 N. Lee Street, Suite 500 Alexandria, VA 22314	703-683-2680
Mr.	Bill	Gordon	DoD	Washington, D.C. 20505	703-482-5493
Mrs.	Kimberly	Hebda	NCSC Attn: C311	9800 Savage Road Ft. Meade, MD 20755-6000	301-859-4488
Mr.	Gericks	Hendricks	NISAC/S2	9800 Savage Road Ft. Meade, MD 20755-6000	
LT	Alan	Hensley	NCSC Attn: C34	9800 Savage Road Ft. Meade, MD 20755-6000 Hensley@dockmaster.arpa	301-859-4494
Mr.	George	Hoover	Attn: V45	9800 Savage Road Ft. Meade, MD 20755-6000 Hoover@dockmaster.arpa	301-859-4374
Mr.	Douglas	Hunt	National Inst. of Standards & Tech.	Computer Security Division - Bldg 225 Gaithersburg, MD 20899 DHunt@ecf.icst.nbs.gov	301-975-5140
Dr.	David J.	Icove	FBI	FBI Academy Quantico, VA	703-640-1176
Dr.	Terry	Ireland	NCSC Attn: C	9800 Savage Road Ft. Meade, MD 20755-6000 Ireland@dockmaster.arpa	301-859-4371
Mr.	John	Jackson	NCSC Attn: C321	9800 Savage Road Ft. Meade, MD 20755-6000 Jackson@tvcho.arpa	301-859-4491

Title	FirstName	LastName	Organization	Address	Phone
Dr.	Mike	Karels	University of California	CSRG Computer Science Div., EECS Berkeley, CA 94720 Karels@ucbarpa.Berkeley. EDU	415-642-4948
Dr.	Stu	Katzke	National Inst: of Standards & Tech.	Technology Bldg; A2 16 Gaithersburg, MD 20899 Katzke@ecf.icst.nbs.gov	975-2929
Mr.	Stephen J.	Kougoures	Attn: 581	9800 Savage Road Ft. Meade. MD 20755-6000	301-688-6026
Mr	Timothy W.	Kreman	NCSC Attn: C31	9800 Savage Road Ft. Meade. MD 20755-6000	301-859-4488
Dr.	Phil	Lapsley	University of California	Experimental Computing Facility 199B Cory Hall Berkeley, CA 94720 Phil@ucbarpa.Berkeley.EDU	415-642-7447
Mr.	Peter	Loscocco	NCSC Attn: C321	9800 Savage Road Ft. Meade, MD 20755-6000 Loscocco@tycho.arpa	301-859-4491
SSA	R. Stephen	Mardigian	FBI	FBI Academy Quantico, VA	704-640-6131
Capt	John	McCumber	NCSC Attn: C	9800 Savage Road Ft. Meade. MD 20755-6000	
Mr.	Jack	Moskowitz	NCSC Attn: C2	9800 Savage Road Ft. Meade, MD 20755-6000 JJMoskowitz@cbckmaster	301-859-4465
LtCol	George R.	Mundy	DCA	Code B602 Washington, D.C. 20305-2000 Mundv@beast.ddn.mil	703-285-5481

Title	FirstName	LastName	Organization	Address	Phone
Mr.	Mike	Muuss	US Army Ballistic Research Lab	Leader, Advanced Computer Systems Team APG, MD 21005-5066 Mike@brl.mil	301-278-6678
Mr.	Eugene	Myers	NCSC Attn: C311 Secure Architectures	9800 Savage Road Ft. Meade, MD 20755-6000 EDMyers@dockmaster.arpa	301-859-4488
Mr.	Gordon R.	Parry	CIA	Washington, D.C. 20505	703-482-6204
Mr.	George	Prettyman	Asst. General Counsel	9800 Savage Road Ft. Meade, MD 20755-6000	301-688-6017
Ms.	Harriet	Roberts	NCSC Attn: C34	9800 Savage Road Ft. Meade, MD 20755-6000	301-859-4486
Mr.	Jon	Rochlis	MIT E40-311	1 Amherst St. Cambridge, MA 02139 Jon@athena.mit.edu	617-253-4222
Mr.	Shawn	Rovanseck	NCSC Attn: C12	9800 Savage Road Ft. Meade, MD 20755-6000 Rovanseck@dockmaster.arpa	301-859-4458
Mr.	Kenneth	Rowe	NCSC Attn: C333	9800 Savage Road Ft. Meade, MD 20755-6000 Rowe@tycho.arpa	301-859-4491
Dr.	William	Scherlis	DARPA	1400 Wilson Blvd. Arlington, VA 22209 Scherlis@vax.darDa.mil	202-694-5800
LTC	James	Sells	NCSC Attn: C33	9800 Savage Road Ft. Meade, MD 20755-6000 JSells@dockmaster.arpa	301-859-4494
Cpt	Richard	Severson	NCSC Attn: C333	9800 Savage Road Ft. Meade, MD 20755-6000 Severson@dockmaster.arpa	301-859-4491
Mr.	Philip L	Sibert	DOE	MA-24 F-315GTN Washington, D.C. 20545	301-353-3307

Title	FirstName	LastName	Organization	Address	Phone
SSA	Karen E.	Spangenberg	FBI-HQ	10th & Pennsylvania NW Washington, DC. 20535	202-325-5594
Mr.	K. H.	Speierman	NSA Senior Scientist	9800 Savage Road Ft. Meade. MD 20755-6000	301-688-6434
Dr.	Stephen L.	Squires	DARPA	Information Science and Technology Office Director, Strategic Computing 1400 Wilson Blvd. Arlington, VA 22209 Squires@vax.darpa.mil	202-694-5800
Capt	Michael	St. Johns	DCA	DCA Code 6612 Washington, DC. 20305-2000 StJohns@beast.ddn.mil	703-285-5133
Dr.	Howard	Stainer	NCSC Attn: C32	9800 Savage Road Ft. Meade. MD 20755-6000	301-859-4491
Dr.	Dennis D.	Steinauer	National Inst. of Standards and Tech.	A-2 16 Technology Gaithersburg, MD 20899 Steinauer@ecf.icst.nbs.gov	301-975-3357
Mr.	Jim	Steinmeier	NCSC Attn: C2	9800 Savage Road Ft. Meade, MD 20755-6000	301-859-4467
Mr.	Cliff	Stoll	Harvard-Smithsonian Center for Astrophysics	60 Garden Street M.S. 6 Cambridge, MA 02 138 Cliff@cfa200.harvard.edu	
Mr.	Jeff	Sweet	Attn: X21	9800 Savage Road Ft. Meade, MD 20755-6000	301-688-5724
Maj	Hugh H.	Thomas	NCSC Attn: C25	9800 Savage Road Ft. Meade, MD 20755-6000 Thomas@dockmaster.arpa	301-859-4474
Mr.	Mario	Tinto	NCSC Attn: C1	9800 Savage Road Ft. Meade, MD 20755-6000 Tinto@dockmaster.arpa	301-859-4450

Title	FirstName	LastName	Organization	Address	Phone
CDR	David	Vaurio	NCSC Attn: C3	9800 Savage Road Ft. Meade, MD 20755-6000	301-859-4485
Mr.	Wayne J.	Weingaertner	NCSC Attn: C31 Secure Computer Systems	9800 Savage Road Ft. Meade, MD 20755-6000 WWeingaertner@ dockmaster.arpa	301-859-4488
Mr.	Howard	Weiss	NCSC Attn: C32	9800 Savage Road Ft. Meade, MD 20755-6000 HWeiss@dockmaster.arpa	301-859-4491
Lt Col	Larry E.	Wheeler	OSD (C3I)	Pentagon-Room 3E187 Washington, D.C. 20301	202-695-7181
Mr.	Mark	Woodcock	NCSC Attn: C331	9800 Savage Road Ft. Meade, MD 20755-6000 Woodcock@tycho.arpa	301-859-4494
Mr.	Tom	Zmudzinski	DCA	Code 8602 McLean Washington, D.C. 20305-2000 TomZ@ddn1.arpa	703-285-5206

The Internet Virus of November 3, 1988

Mark W. Eichen, MIT Project Athena

November 8, 1988

Contents

1	Strategies Involved	1
1.1	Attacks	1
1.1.1	Finger bug	1
1.1.2	Sendmail	1
1.1.3	rexec and passwords	1
1.1.4	side effects	2
1.2	defenses	2
1.2.1	covering tracks	2
1.2.2	camouflage	3
1.3	flaws	3
1.3.1	reinfection prevention	3
1.3.2	heuristics	3
2	The program	5
2.1	main	5
2.1.1	initialization	5
2.1.2	Command line argument processing	5
2.2	doit routine	6
2.2.1	initialization	6
2.2.2	mainloop	6
2.3	Cracking routines	6
2.3.1	cracksome	7
2.3.2	crack0	7
2.3.3	crack 1	7
2.3.4	phase2	8
2.3.5	phase3	8
2.4	hroutines	8
2.4.1	hg	8
2.4.2	ha	8
2.4.3	hl	8
2.4.4	hi	8
2.4.5	hul	9
2.5	attack routines	9
2.5.1	hit finger	9
2.5.2	hit rexec	10

2.5.3	hitsmtp	10
2.5.4	makemagic	10
2.6	host modules	10
2.6.1	nametohost	10
2.6.2	address to host	10
2.6.3	add address	10
2.6.4	addname	10
2.6.5	clean up table	11
2.6.6	get. addresses	11
2.7	object routines	11
2.7.1	load object.	11
2.i.2	get object by name	11
2.8	other initialization routines	11
2.8.1	ifinit	11
2.8.2	rtinit	11
A	Credits	12
A.1	The MIT team	12
A.2	The Berkeley Team	12
A.3	Others	12

Abstract

This paper is a thorough analysis of the code of the virus program which attacked the Internet beginning some time November 3, 1988. It discusses the actual code itself, as well as the strategies and ideas involved in the propagation of the virus.

A virus, according to Webster's, is something which causes infectious disease, as well as being capable of growth and multiplication only in living cells. Inasmuch as a computer is analogous to a living entity, this program is a virus; one of its infection methods is very much like an actual virus, in that it actually infects a running program to gain entry into the system.

Also, virii infect. Worms just crawl around.

Chapter 1

Strategies Involved

1.1 Attacks

This virus attacked several things, directly and indirectly. It both picked out some deliberate targets and had interesting side effects.

1.1.1 Finger bug

The virus hit the finger daemon by overflowing a buffer which was allocated on the stack. The overflow was possible because a library function which did not do range checking was used. Since the buffer was on the stack, the overflow allowed a fake stack frame to be created: which caused a small piece of code to be executed when the procedure returned. The write daemon has a similar piece of code, which makes the same mistake, but it exec's `write` directly, and explicitly exits rather than returning, and thus never uses the (damaged] return stack.

1.1.2 Sendmail

The sendmail mechanism is the "debug" function, which enables debugging mode for the duration of the current connection. One thing that this enables is the ability to send a mail message with a piped program as the recipient. This mode is normally allowed in the sendmail configuration file or user .forward file directly: but not for incoming connections. In this case, the recipient was a command which would strip off the mail headers and pass the remainder of the message to a shell. The body was a script which created a C program which would suck over the rest of the modules from the host that sent it, and the commands to compile and execute it.

The fact that debug was enabled by default was reported to Berkeley by several sources during the 4.2 release, however it was not fixed for the 4.3 release (source or binary.) Project Athena was among a number of sites which disabled it, however it is unlikely that many binary-only sites were able to be as diligent.

1.1.3 rexec and passwords

The virus attacked by the Berkeley remote execution protocol, which required the user name and plaintext password to be passed over the net. The program only used pairs of usernames and

passwords which it had already verified to be correct on the local host.

One fundamental security tenet violated here¹ was that passwords should not be at **all** readable by unprivileged entities. Under most forms of UNIX, the passwords are stored in as computationally extensive encryptions. However, this meant that a program needed merely *try* a large number of guesses, “on its own turf” so to speak, without going through any recorded channels. The Kerberos² system, used at Project Athena, keeps passwords only on a secure central machine, which is used as an authentication server. Although once a username was known, the password could be attacked in much the same way, the usernames are also stored centrally, making it more difficult for the virus to find a set of names to attack.

1.1.4 side effects

When it became clear that the virus was propagating via sendmail, the first reaction of many sites³ was to cut **off** mail service. This did not totally stop the progress of the virus, which continued to travel via rexec and finger. It *did* effectively stop communication of information about the virus, slowing down the information about finger and the patches needed to fix the problem. USENET news was an effective side-channel of information spread, although a number of sites disabled that as well.

One program posted after the virus was analyzed was a tool to duplicate the password attack used (including the dictionary that the virus carried with it) to allow system administrators to analyze the passwords in use on their system. The spread of this virus should be effective in raising the awareness of users (and administrators] to the importance of choosing “difficult” passwords.

1.2 defenses

The virus used a number of techniques to hide itself as well, though they had various vulnerabilities.

1.2.1 covering tracks

The program did a number of things to cover its trail. It zeroed out its argument list, once it had finished processing the arguments, so that the process status command would not show how it was invoked.

It also deleted the executing binary, which would leave an inode only referenced by the execution of the program but not appearing in the filesystem. If the machine was rebooted while the virus was actually running, the file system salvager will recover the file after the reboot.

The program also uses resource limit functions to prevent it from using any space in a core dump. Thus, it prevents any bugs in the program from leaving core dumps behind.

¹ref orange book7

²cite paper

³including the Darpa MILNET

1.2.2 camouflage

It was compiled as "sh", the same name used by the Bourne Shell, which is used often in shell scripts and automatic commands. Even a diligent system manager **would probably** not notice a large number of shells running for short periods of time. The **virus** did **fork**, splitting into a parent and child, approximately every three minutes. The parent would then die, leaving the child to continue from the exact same place.

1.3 flaws

The virus also had a number of flaws, varying between the subtle and the clumsy. Keith Bostic of Berkeley, with concurrence of the team at MIT⁴ posted patches for some of the more obvious ones? as a humorous gesture.

1.3.1 reinfection prevention

The code for preventing reinfection of a machine which was actively infected didn't work at all. It was only checked on a one in fifteen random chance, making multiple infections likely. This also lead to the early detection of the virus, since only one in fifteen instances of the virus would actually die; since the virus was careful to clean up temporary files! its presence alone didn't interfere with reinfection.

Also, a multiply infected machine would spread the virus faster! perhaps proportionally to the number of infections it was harboring, since

- The program scrambles the lists of hosts and users it attacks; since the random number generator is seeded with the current time, the separate instances are likely to hit separate targets.
- The program tries to spend a large amount of time sleeping and listening for other infection attempts (which never report themselves) so that the processes would share the resources of the machine fairly well.

Thus, the virus spread much more quickly than the perpetrator expected, and was noticed for that very reason. The MIT Media Lab, for example, cut themselves completely **off** from the network because the computer resources absorbed by the virus were detracting from work in progress, while the lack of network service was a minor problem.

1.3.2 heuristics

One attempt that was made to make the program not waste time on non-UNIX systems was to first try to telnet to the host in the list. If the host refused telnet connections, it was likely to refuse other attacks as well. There were several problems with this attack:

⁴telephone call, around 9AM Friday?

- A number of machines exist which provide sendmail service (for example) that do not provide telnet service, and although vulnerable would be ignored under this **attack**.⁵
- The telnet “probing” code immediately closed the connection upon **finding** that it had opened it. By the time the “inet daemon”, the funnel which handles most **incoming** network services, identified the connection and handed it off to the telnet daemon, the connection was already closed, causing the telnet daemon to indicate an error condition of high enough priority to get logged on most systems. Thus the times of the earliest attacks were noted, if not the machines they came from.

⁵ATHENA.MIT.EDU, for example, was vulnerable to the finger daemon attack, but was untouched because it did not run a telnet daemon.

Chapter 2

The program

2.1 main

The main module made several steps to set itself up.

2.1.1 initialization

First the program takes some steps to hide itself. It changes the “zeroth” argument, which is the process name, to “sh” so that no matter how the program was invoked, it would show up in the process table with the same name as the Bourne Shell, a program which is often running legitimately.

The program also sets the resource limit on core dump size to zero blocks, so that if the program did crash for some reason it would vanish, rather than leaving a core dump behind to help investigators. It also turns off handling of the write errors on pipes, which by default cause the program to **exit**.

The next step is to read the clock, store the current time in a local variable, and using that value to seed the random number generator.

2.1.2 Command line argument processing

The virus program itself takes an optional argument “-p” which must be followed by a decimal number, which seems to be a process id of the parent which spawned it. It uses this number later on to kill that process: probably to “close the door” behind it.

The rest of the command line arguments are “object names”. These are names of files it tries to load in. If it can’t load one of them, it quits. If the “-p” argument was given, it also deletes the file (and later tries to delete the running virus, as well as a file `/tmp/.dumb`).¹

After **all** the arguments have been read, if no objects were loaded the program quits. It then checks for the existence of the object “l1.c” and quits if it is missing.

If the “-p” argument was given, the program closes all of its file descriptors, and then deletes the files

The program then erases the text of all of the arguments.

¹need better explanation of loadobject

It then scans **all** of the network interfaces on the machine, gets the flags and address of each interface. It tries to get the point to point address of the interface; it skips the loopback address. It **also** stores the netmask for that network.

Finally, it **kills off** the process **id** given with the “-p” option. It **also** changes **the** current process group, **so** that it doesn’t die when the parent exits. Once this is cleaned up, it **falls** into the **doit** routine which performs the rest of the work.

2.2 **doit** routine

This routine is the where the program spends most of its time.

2.2.1 **initialization**

Like the main routine. it seeds the random number generator with the clock: and stores the clock value to later measure how long the virus has been running on this system.

It then tries **hg**. If that fails. it tries **h1**. If that fails. it tries **ha**.

It then tries to check if there is already a copy of the virus running on this machine. This code doesn’t work correctly (one of the reasons the virus was using large amounts of computer time.)

It then sends a one byte on a TCP Stream connection to **128.32.137.13**, which is **ernie.berkeley.edu**. There has not been an explanation for this: it only **sends** this packet with a 1 in 15 random chance.

2.2.2 **main loop**

An infinite loop comprises the main active component of the virus. It **calls** the **cracksome** routine² which tries to find some hosts that it can break in to. Then it waits 30 seconds. while **listening** for other virus programs attempting to break in. and tries to break into another batch of machines.

After this round of attacks. it forks. creating two copies of the virus; the original (parent) dies, leaving the fresh copy. The child copy has all of the information the parent had, while not having the accumulated CPU usage of the parent. It also has a new process id, making it hard to find.

The virus then runs the “h routines”, which search for more machines to add to the list of hosts, and then sleeps for 2 minutes (again looking for other virus attempts.) After that, it checks to see if it has been running for more than 13 hours, and if so cleans up some of the entries in the host list.

Finally, before repeating, it checks **pleasequit**. If it is set, **and** it has tried more than 10 words from its own dictionary against existing passwords? it quits. Thus forcing **pleasequit** to be set in the system libraries will do very little to stem the progress of this virus.

2.3 **Cracking routines**

There are a collection of routines which are the “brain” of the virus. There is a **main switch**, which chooses which of four strategies to execute next, and a number of separate strategy routines. It is clearly the central point to add new strategies, were the virus to be further extended.

²This name **was** actually in the symbol table of the distributed binary.

2.3.1 **cracksome**

The **cracksome** routine is the main switch. Again, this routine was named in the global symbol table; though it could have been given a confusing or random **name**, it **was** actually clearly labelled, which lends **some** credence to the idea that the virus was released prematurely.

2.3.2 **crack 0**

The first crack routine read through the `/etc/hosts.equiv` file to find machine names that would be likely targets. While this file indicates what hosts the current machine trusts, it is fairly common to find systems where all machines in a cluster trust each other, and at very least implies that people with accounts on this machine will have accounts on the other machines mentioned in `hosts.equiv`.

It also read the `/.rhosts` file: which lists the set of machines that this machine trusts root access from. Note that it did not take advantage of any knowledge about this trust³ but merely uses the names as a list of additional machines to attack. Often, system managers will deny read access to this file to any user other than root itself, to avoid providing any easy list of secondary targets that could be used to subvert the machine; this practice would also have prevented the virus from discovering those names, although `/.rhosts` is very often a subset of `/etc/hosts.equiv`.

The program then reads the entire local password file `/etc/passwd`. It uses this to find personal `.forward` files for names of other machines it can attack. It also records the username, encrypted password, and “gecos” information string which is also stored in the `/etc/passwd` file. After processing the entire file, it advances the attack type selector: so that the machine proceeds to the next set of attacks.

2.3.3 **crack 1**

The next set of attacks are on passwords on the local machine. It uses several functions to pick passwords which can then be encrypted and matched against the encryptions obtained in phase 0:

- No password at all.
- The username itself.
- The username appended to itself.
- The second of the comma separated “gecos” information fields, which is commonly a nickname.
- The remainder of the full name after the first name in the “gecos” fields, ie. probably the last name, with the first letter converted to lower case.
- This “last name” reversed.

All of these attacks are applied to fifty passwords at a time from those collected in phase 0. If this **pass** finishes all of the passwords, it advances to phase 2.

³such as Bob Baldwin's system **KUANG** would

2.3.4 phase 2

Phase 2 takes the internal word list that the virus distributes with itself, and scrambles it. Then it takes the words one at a time and decodes them (the high bit is set on all of the characters to obscure them) and then tries them against all collected passwords. Thus the check in the main loop against `nextw` only succeeds after 10 of the words have been checked against all of the encryptions in the collected list.

Again, if the word list is exhausted the virus advances to phase 3.

2.3.5 phase 3

Phase 3 looks at the local `/usr/dict/words` file, a twenty four thousand word dictionary distributed with 4.3BSD and other unix systems. The words are stored in this file one word per line. One word at a time is tried against all encrypted passwords. If the word begins with an upper case letter, the letter is converted to lower case and the word is tried again.

When the dictionary runs out, the phase counter is again advanced to 4 (thus no more password cracking is attempted.)

2.4 h routines

The “h routines” are a collection of routines with short names, including `hg`, `ha`, `hi`, and `hl`, which search for other hosts to attack.

2.4.1 hg

The “hg” routine calls `rt_init` to scan the routing table, which creates a list of gateways. It then tries a generic attack routine⁴ to attack via `rsh`, `finger`, and `smtp`.

2.4.2 ha

The “ha” routine also tries to go through the list of machines and connect to port 25, the SMTP port, to determine if a mailer was running on the machine.

2.4.3 hl

The “hl” routine just looks for certain machines based on their netmasks, and tries to attack them.

2.4.4 hi

The “hi” routine goes through the table of hosts and tries to actually attack a host via “`rsh`”, “`finger`”, “`smtp`”.

⁴s1638 internally

2.4.5 h_ul

The “h_ul” routine is called by the phase one and phase three crack subroutines. Once a user name and password is guessed, this routine is called with a hostname read from either the user’s .forward or .rhosts files. It then runs an rsh to that machine, and has it execute a Bourne Shell, thus allowing it to use standard methods to attack that machine.

2.5 attack routines

There were a collection of attack routines, which all provided a Bourne Shell running on the remote machine if they succeeded.

2.5.1 hit finger

The “hit finger” routine tries to make a connection to the finger port of the remote machine. Then it creates a “magic packet” which consists of

- A 400 byte “runway“ of VAX “nop“ instructions, which can be executed harmlessly.
- A small piece of code which executes a Bourne Shell.
- A stack frame, with a return address which would hopefully point into the code.

Note that the piece of code is VAX code and the stack frame is a VAX frame. in the wrong order for the Sun. Thus, although the Sun finger daemon has the same bug as the VAX one, this piece of code cannot exploit it.

The attack on the finger⁵ can be considered a “viral“ attack. since although the worm doesn’t modify the host machine at all. the finger attack does modify the running finger daemon process. Then, speaking in viral terms, the “injected DNA” component of the virus contained the following VAX instructions:

```
pushl $68732f push '/sh<NUL>'
pushl $6e69622f push '/bin'
movl sp,r10 save address of start of string
pushl $0 push 0 (arg 3 to execve)
pushl $0 push 0 (arg 2 to execve)
pushl r10 push string addr (arg 1 to execve)
pushl $3 push argument count
movl sp,ap set argument pointer
chmk $3b do “execve” kernel call.
```

The execve system call causes the current process to be replaced with an invocation of the named program: /bin/sh is one of the UNIX command interpreters. In this case, the shell wound up running with its input coming from! and its output going to, the network connection. The virus then sent over the same bootstrap program that it used for its sendmail-based attack.

⁵William E. Sommerfeld, of MIT Project Athena. was the first to discover this mode of attack, and provided the description that follows.

2.5.2 hit rexec

The “hit rexec” routine uses the “exec/tcp” service, the remote execution system which is similar to rsh, but is designed for use by programs. It connects, sends the username, the password, and /bin/sh as the command to execute. It checks to see if it succeeded to connect and get access using one of the password/account pairs guessed earlier.

2.5.3 hit smtp

The “hit smtp” routine uses the “smtp/tcp” service to take advantage of the sendmail bug. It attempts to use the “debug” option to make sendmail run a command (the recipient of the message), which compiles a program (which is included as the body of the message.)

2.5.4 makemagic

This routine tries to make a telnet connection to the 6 addresses for the current victim, and then breaks it immediately. If it succeeds, it creates a listening stream socket on a random port number which the infected machine will eventually connect back to. Since it breaks the connection immediately, it often produces error reports from the telnet daemon, which get recorded, and provide some of the earliest reports of attack attempts.

2.6 host modules

There are a set of routines designed to collect names and addresses of target hosts in a master list.

2.6.1 name to host

This routine searches the host list for a given named host, returns the list entry describing it, and optionally adds it to the list if it isn't there already.

2.6.2 address to host

This routine searches the host list for a given host address, returns the list entry describing it, and optionally adds it to the list if it isn't there already.

2.6.3 add address

This routine adds an address to an entry in the host list. Each entry contains up to twelve names, up to six addresses, and a flag field.

2.6.4 add name

This routine adds a name to an entry in the host list, if it doesn't already exist.

2.6.5 clean up table

This routine cycles through the host list, and cleans out hosts which only have flag bits 1 and 2 set (and clears those bits.)

2.6.6 get addresses

This routine takes an element of the host table and tries to find an address for the name it has, or get a name for the addresses it has, and include the aliases it can find in the list as well.

2.7 object routines

These routines are what the system uses to pull all of its pieces into memory when it starts (after the host has been infected) and then to retrieve them to transmit to any host it infects.

2.7.1 load object

This routine opens the file, stats it, allocates enough space to load it in, reads it in as one block. decodes the block of memory (with XOR). If the object name contains a comma, it moves past it and starts the name there.

2.7.2 get object by name

This routine returns a pointer to the requested object. This is used to find the pieces to download when infecting another host.

2.8 other initialization routines

2.8.1 if init

This routine scans the array of network interfaces. It gets the **flags** for each interface, and makes sure the interface is UP and RUNNING (specific fields of the flag structure.) If the entry is a point to point type interface, the remote address is saved and added to the host table. It then tries to enter the router into the list of hosts to attack.

2.8.2 rt init

This routine runs "netstat -r -n" as a subprocess. This shows the routing tables, with the addresses listed numerically. It gives up after finding 500 gateways. It skips the default route, as well as the loopback entry. It checks for redundant entries, and checks to see if there this address is already an interface address. If not, it adds it to the list of gateways.

After the gateway list is collected, it scrambles it and enters the addresses in the host table.

Appendix A

Credits

I'd like to mention a few people who worked on the virus hunt:

A.1 The MIT team

Mark W. Eichen (Athena and SIPB) and Stanley R. Zanarotti (LCS and SIPB) lead the team disassembling the virus code. The team included William E. Sommerfeld (Athena, SIPB, and Apollo), Ted Y. Ts'o (Athena and SIPB), Jon Rochlis (MIT Telecom and SIPB), Hal Birkeland (MIT Media Lab), and John T. Kohl (Xthena, DEC, and SIPB).

Jeffery I. Schiller (Director of the MIT Network, Athena, SIPB) did a lot of work in trapping the virus, setting up an isolated test suite, and dealing with the media. Ron Hoffman (MIT Telecom) was one of the first to notice an MIT machine attacked by finger.

Tim Shepard (LCS) provided information as to the propagation of the virus, as well as large amounts of "netwatch" data and other technical help.

A.2 The Berkeley Team

We don't know how they were organized at Berkeley, however we conversed extensively and exchanged code with Keith Bostic throughout the morning of November 4, 1988.

A.3 Others

Numerous others across the country deserve thanks: many of them worked directly or indirectly on the virus, and helped coordinate the spread of information.

Chronology of Virus from the MIT Perspective

Jon Rochills *jon@bitsy.mit.edu*

The first posting mentioning the virus was by Peter Yee (Nasa Ames) at 8:28pm est on Wednesday to the `tcp-ip` mailing list. Peter stated that **UCB, UCSD, LLNL**, Stanford, and NASA Ames had been attacked, and described the use of `sendmail` to pull over the virus, including the `x*` files found in `/usr/tmp`. The virus was observed to send `vax` and `sun` binaries, have **DES** tables built in, and made some use of `.rhosts` and `hosts.equiv` files. A Berkeley extension was given and Phil Lapsley and Kurt Pires were listed as being knowledgeable about the virus.

At 3:10am the first notice of the virus at **MIT** was posted at AMT by Pascal Chesnais (*lacsap@media-lab.mit.edu*). The motd on *media-lab* read:

--- lacsap Nov 3 1988 03:10am
DO NOT CALL THE GARDEN. IF YOU WANT TO PROTECT YOUR MACHINE TURN OFF SENDMAIL
OR JUST TURN YOUR MACHINE OFF, OR UNPLUG IT FROM THE NETWORK!!!! DO NOT CALL
THE GARDEN" !!!

Pascal had spotted the virus earlier but assumed it was just "a local run away program". The group at AMT figured out after midnight that it was a virus and it was coming in via mail. The response was to such down infected machines. The network groups monitoring information shows the *media lab* gateway first went down at 11:40pm Wednesday, but was back up by 3:00am. Pascal requested that the Network group isolate the building during the Thursday 11:30pm and it remained so isolated until Friday at 2:30pm.

Pascal now reports that logs on *media-lab* show several scattered `ttloop: peer died: No such file or directory` messages. There were a few every couple of days, several during the Wednesday afternoon and many starting at 9:48pm. These are caused by opening a telnet connection and immediately closing it: specifically `inetd` spawns a `telnetd`, but when `telnetd` goes to read from the network, it finds the connection has disappeared. The virus did this in order to determine whether or not to try to infect a target machine.¹ The logs on *media-lab* start on October 25th and the following log entries made before the swarm on Wednesday night.

```
Oct 26 15:01:57 media-lab telnetd[23180] : ttloop: peer died: No such file or
Oct 28 11:26:55 media-lab telnetd[23331] : ttloop: peer died: No such file or
Oct 28 17:36:51 media-lab telnetd[12614] : ttloop: peer died: No such file or
Oct 31 16:24:41 media-lab telnetd[18518] : ttloop: peer died: No such file or
Nov 1 16:08:24 media-lab telnetd[16125] : ttloop: peer died: No such file or
Nov 1 18:02:43 media-lab telnetd[21889] : ttloop: peer died: No such file or
Nov 1 18:58:30 media-lab telnetd[24644] : ttloop: peer died: No such file or
Nov 2 12:23:51 media-lab telnetd[4721] : ttloop: peer died: No such file or d
Nov 2 15:21:47 media-lab telnetd[13628] : ttloop: peer died: No such file or
```

¹The assumption that machines not running a `telnetd` are not vulnerable to attack is quite interesting. It allowed systems like the MIT Project Athena mailhub, *athena.mit.edu*, (on which we preferred to use only `kerberos` authentication), to escape unscathed.

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

At 3:34am est on Thursday, Andy Sudduth from Harvard made his anonymous posting to tcp-ip. The posting said that a virus might be loose on the Internet and that there were three steps to take to prevent further transmission. This included not running fingerd or fixing it not to overwrite the stack when reading its arguments from the net², be sure sendmail was compiled without debug, and not to run rexecd.

The posting was made from an Annex terminal server at from Aiken (sp?) Center (?) at Harvard, by teineting the SMTP port of *iris.brown.edu*. This is obvious since the message was from "foo%bar.apar" and because the last line of the message was "qu\177\177\177", an attempt to get about processing out of the brown SMTP server, a common mistake when faking Internet mail.

It was ironic that this posting did almost no good. The path it took to get to athena was:

```
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 P
Received: from ca.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
```

There was a 20 hour delay before the message escaped from relay.cs.net and got to *sri-nic.arpa*. Another 6 hours went by before the message was received by *athena.mit.edu*. Other sites have reported similar delays.

At 5:58am Thursday morning Keith Bostic (*bostic@okeefe.berkeley.edu*) made the virus bug fix posting. The message went to the *tcp-ip*, *comp.bugs.4bsd.ucb-fixes*, *news.announce*, and *news.sysadmin*. It supplied the compile without debug fix to sendmail (or patch the debug command to a garbage string), as well as the very wise suggestion to rename *cc* and *ld*, which was effective since the virus needed to compile and link itself.

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

²this was a level of detail that only the originator of the virus could have known at that point. To our knowledge nobody had yet identified the finger bug, since it only affected certain vaxen, and certainly nobody had discovered its mechanism.

During Thursday morning Ray Hirschfeld (*ray@math.mit.edu*) spotted the virus on the MIT math department suns and shut down the math gateway at 10:15am. It remained down until 3:15pm.

Gene Spafford posted a message at 2:50pm Thursday to a large number of people and mailing lists include *nntp-managers* which is how we saw it quickly at MIT. It warned 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 Network Group), called Milo Medin (*medln@nsipo.nasa.gov*) and found out much of the above. Many of us had not yet seen the messages. He pointed out that the virus just loved to attack gateways (found via the routing tables) and remarked that it must have not been effective at MIT were we run our own C Gateway code, not Unix. Milo also informed us that DCA had shut down the mailbridges. He pointed us to the group at Berkeley and Peter Yee specifically.

At about 6pm on Thursday, Ron Hoffmann (*hoffmann@bitsy.mit.edu*) 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 immune³. The virus was running under the userid of nobody, and it appeared that it had to be attacking through the finger service, the only network service running under that userid. At that point, we called the group working at Berkeley; they confirmed our suspicions that 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 `exec` it did involved a hard-coded 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"; clearly some sort of overrun buffer wound up corrupting something.

Bill Sommerfeld (*wesommer@athena.mit.edu*) 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 benign way.⁴

A *risks* digest came out at 6:52pm. It included a message from Cliff Stoll of Harvard (*Stoll@dockmaster.arpa*) which described the spread of the virus on *milnet* and suggested that *milnet*

³It was running a mailer with debugging turned off

⁴the test virus sent the string "Bozo!" back out the network connection.

sights might want to remove themselves from the network. **Stoll also** made the wonderful statement, "This is bad news." Other messages were from Spafford, Peter Neumann (*neumann@csl.sri.com*), and Matt Bishop (*mbishop@bear.dartmouth.edu*). They described the sendmail propagation mechanism.

In the SIPB office Stan Zanarotti (*srz@lcs.mit.edu*) and Ted Ts'o had managed to get a core dump from the virus running on a machine in the MIT Lab for Computer Science (LCS) as well as the *vax* binary. Stan and Tim Sheppard (*shep@ptt.cs.mit.edu*) had been dealing with the virus from 11am Thursday over in Tech Square. Their first reaction was to shut down the network by powering off **DELNI's**. By 1pm Tim had verified that no files had been modified on *a//spice./sc.mif.ed* and had installed recompiled sendmail. (Tim also reloaded a root partition from tape, just to ensure that he was running trusted software).

Ted and Stan started attacking the virus. Pretty soon they had figured out the xor encoding of the strings and were manually decoding strings. By 9:00pm Ted 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 setting the meta bit of each character).

At the same time they discovered the ip address of *ernie.berkeley.edu*, 128.32.137.13, and proceeded to take apart the **send-message** routine to figure out what it was sending to ernie, how often, and if a handshake was involved. Stan told Jon Rochlis <*jon@bitsy.mit.edu*> 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.

About this time a camera crew from **WNEV** Channel 7 (the Boston CBS affiliate) showed up at the office of James D. Bruce (*jdb@delphi.mit.edu*), VP for Information Systems. He called Jeff Schiller and headed over to E40. Jeff and Jim were interviewed. The 80,000 number of hosts was stated along with an estimate of 10% infection of the 2000 hosts at MIT. The infection rate was a pure guess. The virus was the lead story on the 11pm news, and we were quite surprised that the real world would pay that much attention. Pieces of the footage shot then were shown on the CBS morning news (but by that point were were too busy to watch).

Sheppard shows up in E40, then punts to Tech Square to check his netwatch data for ernie packets. (The machine with the data had been unplugged from the network.)

Serious decompiling began at midnight. Stan and Ted came to E40.

John Kohl had the virus running by 5am and observed many things. They were confirmed by the decompiling which was almost done.

List times of berkeley conversations and ftp exchanges of source code.

Press conference in E40 at noon. 7 camera crews, tons of print media. Total zoo until 3pm.

Bostic asks for our affiliations and if we like the idea of posting bug fixes to the virus (we did!).

The Today show comes to the SIPB office Saturday to find out about "hackers".

MIT Cast of Characters

Media Lab

Pascal Chesnais <lacsap@media-lab.media.mit.edu>

VP Information Services

James D. Bruce <jdb@delphi.mit.edu>

Network Group/Athena/SIPB

Jeff Schiller <jis@bitsy.mit.edu>

Athena/SIPB

Mark Eichin <eichin@athena.mit.edu>

LCS/SIPB

Stan Zananottl <srz@lcs.mit.edu>

Athena/SIPB

Ted Ts'o .ztysto@athena.mit.edu

Apollo/Athena/SIPB

William Sommerfeld <wesommer@athena.mit.edu>

DEC/Athena/SIPB

John Kohl <jtkohl@athena.mit.edu>

Athena/SIPB

Ken Raeburn <raeburn@athena.mit.edu>

Network Group/SIPB

Jon Rochlis <jon@bitsy.mit.edu>

Media Lab

Hal Birkeland <hkbirke@athena.mit.edu>

Network Group

Ron Hoffmann <hoffmann@bitsy.mit.edu>

Athena/SIPB

Richard Basch <probe@athena.mit.edu>

LCS

Tim Sheppard <shep@ptt.lcs.mit.edu>

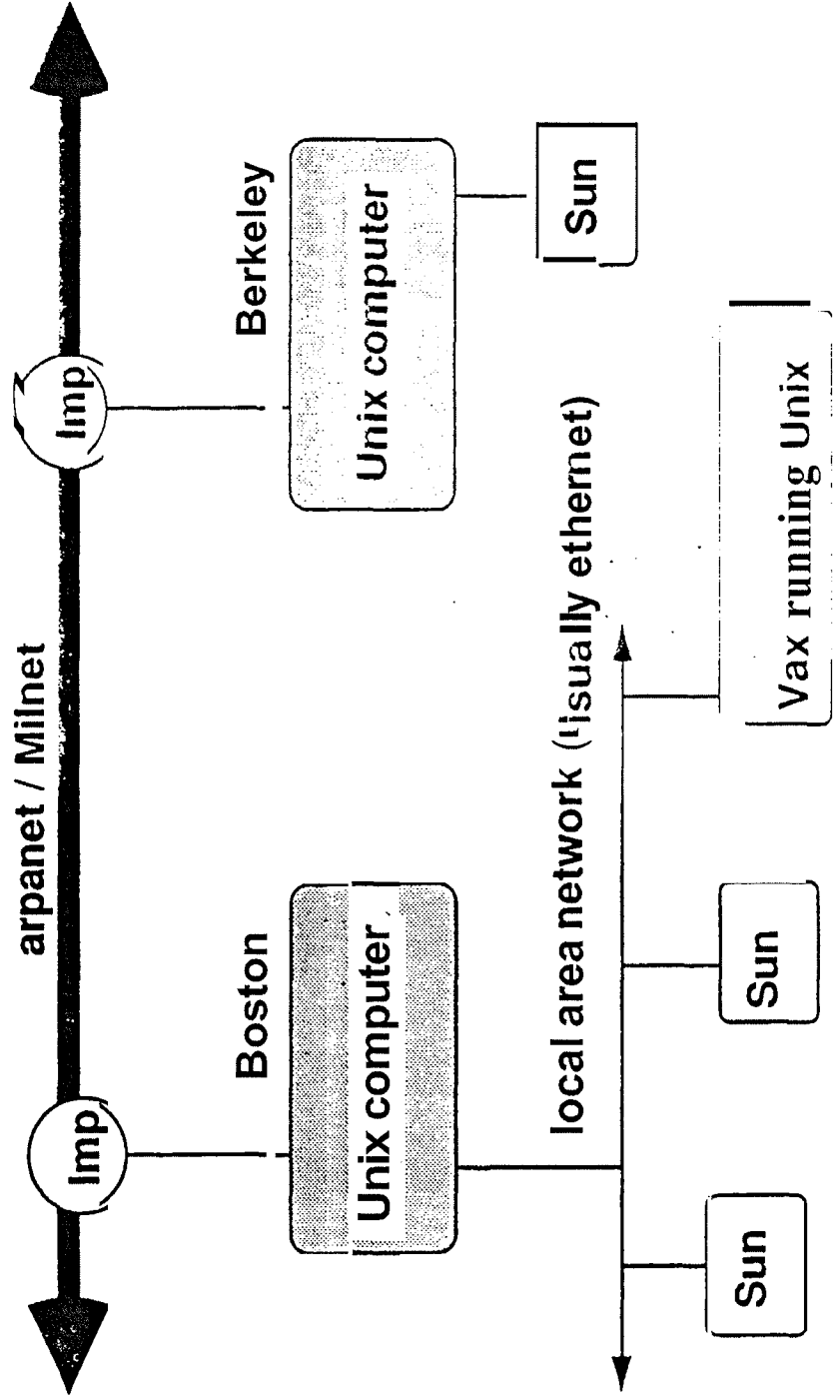
SITE EXPERIENCE

HARVARD

CLIFF STOLL

Arpanet as a Backbone

4



What holes did the Virus exploit?

- **Sendmail**
Utility to copy network packets into mail files
Sometimes used to move packets into processes
(news feeds)
- **Finger Daemon**
Utility to find out where someone is
- + **The virus was specifically designed for Unix 4.3BSD**
it could not spread to non-unix computers, like a VMS system, or an IBM PC.
- + **Sun workstations, Vax 780's and Vax 8800's were hit.**

H'OWTO SOLVE A VIRUS

SOLVING A VIRUS

WAYS TO PREVENT SOLVING

1 REVERSE ENGINEERING:
DISASSEMBLE THE MACHINE CODE
RECONSTRUCT THE ORIGINAL PROGRAM
TRY TO UNDERSTAND IT

1 HIDE THE CODE BY ENCRYPTION
MAKE SELF-MODIFYING CODE
ADD MISLEADING SEGMENTS
INSERT NON-OPERATING CODE

2 TREAT IT AS A BLACK BOX:
MONITOR ALL ITS INPUTS & OUTPUTS
FIND WHAT IT RESPONDS TO
... ITS TRANSFER FUNCTION

2 BUILD MANY DIFFERENT MODULES
MAKE IT TIME DEPENDENT
HAVE IT SENSE LOTS OF PARAMETERS
USE SEVERAL ATTACK MECHANISMS

3 TRACK IT BACK TO THE AUTHOR

3 START THE VIRUS FROM A DISTANT SITE
DON'T PUT YOUR NAME ON THE VIRUS

SENDMAIL BUG

- SENDMAIL: MOVES NETWORK PACKETS INTO MAIL FILES
TRANSFERS NETWORK TRAFFIC INTO MAIL FILES
CAN MOVE TRAFFIC INTO CERTAIN PROCESSES (FOR NETNEWS FEEDS)

- WHEN COMPILES WITH DEBUG, AND DEBUG IS SET
LETS YOU SEND TRAFFIC INTO ANY PROCESS
THROUGH A UNIX PIPE, WITHOUT CHECKING

FROM: </DEV/NULL >
RECEIVE TO: < /SED >
MAIL BODY DATA TO SEND TO THE PROCESS

- SUN & BERKELEY UNIX DISTRIBUTED W/DEBUG ENABLED

SO THIS BUG WAS IN 20,000 + COMPUTERS

PASSWORD GUESSING

- **VIRUS ATTEMPT TO GUESS PASSWORDS
BY READING THE LISTS OF USERS
NAMES AND PERMUTATIONS OF THEIR NAMES**

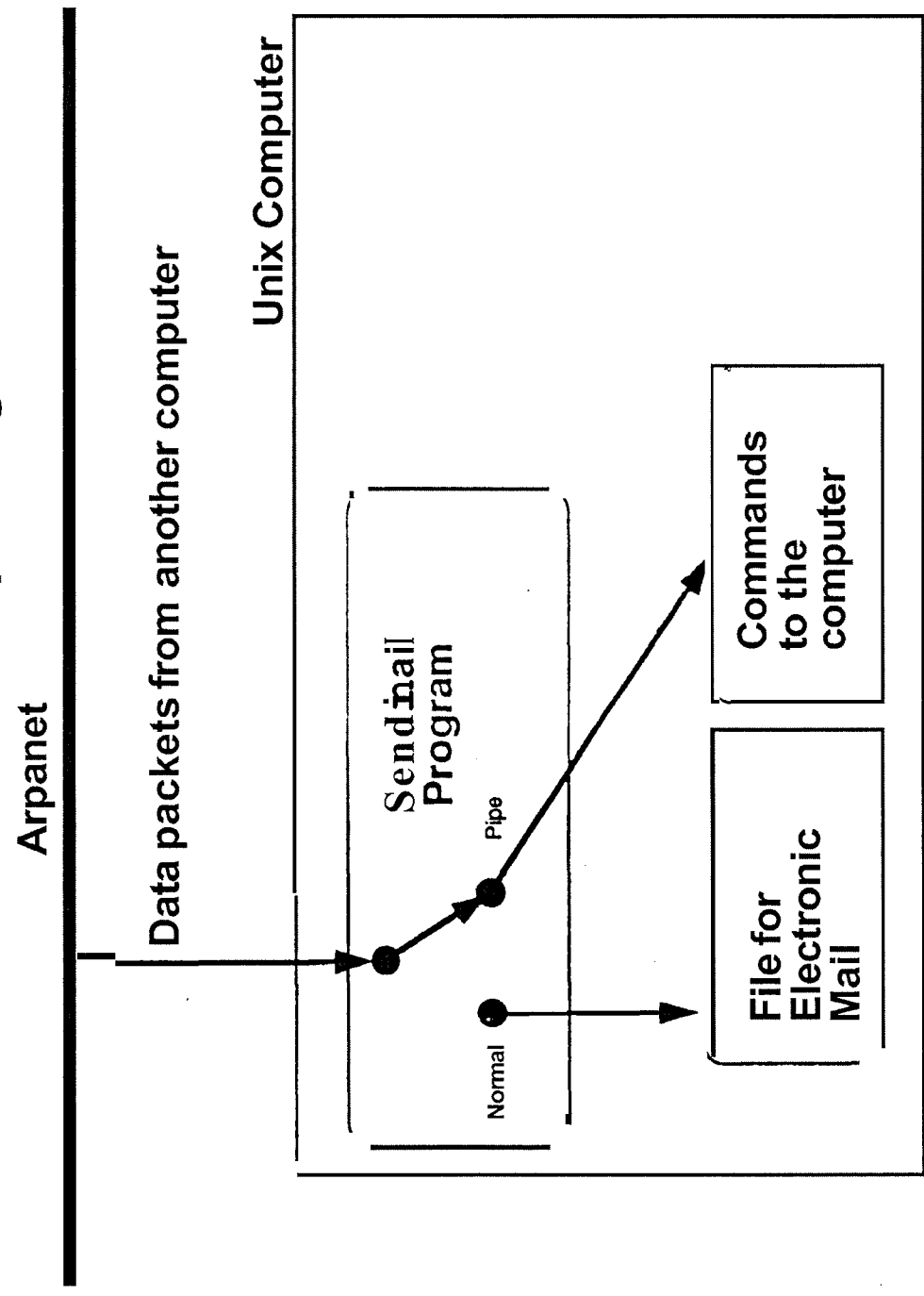
Quick Reaction Across the Nation

- **UC Berkeley -
Experimental Computing Center for Disease Control**
- **Stanford**
- **NASA/AMES**
- **Ballistics Research Lab**
- **MIT**
- **Lawrence Berkeley Labs**
- **Lawrence Livermore Labs**
- **Univ. Rochester**
- **Harvard-Smithsonian Center for Astrophysics**

Stamping it Out

- Initial cures: disconnect from networks
reboot standalone
erase the x files
disable sendmail
boot nearby computers
- Problem: virus reinfected from nearby computers (.rhosts especially)
virus used other holes (fingerd, password crackinj)
very frustrating
- Hard to communicate with other sites:
many disconnected from network
all the virus packets saturated some nets
nobody was coordinating
- Hard to understand: encryption tough to disassembly

Exploiting a hole in Sendmail 17



Normally, data goes through the mailer into mail files.
Data can be sent as commands to special programs.
When Debug is enabled, data can be sent as commands any program

HOW MANY COMPUTERS INFECTED?

- THESE ARE GUESSES. I KNOW OF NO CENSUS
- HOW MANY COMPUTERS ARE ON THE ARPANET?
ABOUT 100 CLASS A NODES
CLASS A NODES ARE EXPLICITLY TARGETED
- HOW MANY NODES ON THE SUBNETS?
ABOUT A HUNDRED PER CLASS A NODE?
- WHAT PERCENTAGE WERE INFECTED?
10%? 50%? AT HARVARD/SMITHSONIAN, ABOUT 80%
(NONE OF OUR DISKLESS NODES, BUT THEN THEY WERE USELESS WHEN THE
FILESERVER WAS DEAD)
AT LAWRENCE BERKELEY LABS, ABOUT 50% WERE INFECTED
- SO ABOUT 1000 TO 10,000 COMPUTERS WERE HIT.

Virus or Worm?

Virus: Self replicating program that infects other programs

Worm: Program that snakes through computers, copying itself from one system to another.

**Purists would call this a worm, not a virus.
Makes'no difference to me.**

Previous Viruses & Hacks

- '84 - '88 On personal computers: replication by infecting programs.
Medium of transport: floppy discs & phone lines to bulletin boards
 - '86 - '87 Intruders manually break into computers to embarrass companies, wreck programs, or steal information.
Medium of transport: dial-up phone lines, networks
 - '87 IBM Christmas tree virus: Replication by distributing a **command** file to many people. Each person executes the file & it mails itself to many others.
Medium of transport: SNA networks, Bitnet
 - '88 Arpanet virus: self replicates by entering Unix systems & breaking security to obtain a root shell. Medium of transport: TCP/IP networks (Arpanet/Milnet, local area networks)
- ☞ This is the first virus to spread automatically across the networks.
The first virus to exploit multiple security **holes**

REAL EFFECTS

- **HOW MUCH DAMAGE WAS DONE?
10,000 PEOPLE LOST 2 DAYS OF WORK; AT \$100/PERSON-DAY = \$2,000,000**
- **INDIRECT COSTS - OPERATIONS DISRUPTED, SCHEDULES DELAYED**
- **CONSCIOUSNESS RAISING ABOUT COMPUTER SECURITY**
- **DID THIS GUY DO US A FAVOR BY SHOWING OUR VULNERABILITIES?
WAS IT NECESSARY?
A MONTH AGO, COVER OF TIME MAGAZINE WAS ABOUT VIRUSES!**

What to learn

- Networking makes the problem much worse.
- Our society depends heavily on interlinked computers military, university, commercial systems are intertwined
- There's no central coordinating center or clearing house for emergencies.
- Nobody's in charge of our networks
- Security holes are subtle; introduced from strange sources and exploited by competent, aware people.

SITE EXPERIENCE

BERKELEY

P. LAPSLEY

U.C. BERKELEY TIMELINE (PST)

WED NOV 2	1900	BERKELEY HIT
	2028	PETER YEE'S MESSAGE
	2100	SENDMAIL, FINGERD, ETC. SHUT OFF LOCALLY
THU NOV 3	0100	SENDMAIL BUG FIXED
	0300	SENDMAIL BUG POSTED TO 'I-CP-IP, USENET
	0500	FINGERD BUG - ED WANG
	1700	FINGERD BUG DECOMPILATION STARTS
FRI NOV 4	2100	FINGERD POSTED
	0600	DECOMPILATION "FINISHED"
MON NOV 7	1900	(DITTO)

HOST CLEANUP

1. FIX SENDMAIL - DISABLE "DEBUG" COMMAND
2. FIX/DISABLE FINGERD (REPLACE GETS WITH FGETS)
3. DISALLOW RSH FROM INFECTED HOSTS -
-DISABLE RSHD
-RENAME USERS' RHOSTS FILES AND VERIFY EQUIVALENT HOSTS ARE CLEAN
4. REMOVE WORMS FROM MAIL QUEUE
5. KILL RUNNING WORMS (OR REBOOT)
(SH)
Om-91*)
RSH HOSTNAME EXEC/BIN/SH
6. USERS WHOSE ACCOUNTS WERE BROKEN - CHANGE PASSWORDS

SITE EXPERIENCE

**ARMY BALLISTIC
RESEARCH LAB**

M. MUUSS

Post-Mortem of 3-Nov ARPANET Incident

The Ballistic
Research Laboratory
Anti-Viral
Program

Michael J. Muuss

The Advanced Computer **Systems** Team
U S Army Ballistic Research Laboratory

“VIRUS” From Websters 9th

- From Latin: slimy liquid, poison, stench.
- Causative agent of an infectious disease.
- Complex molecules capable of growth and multiplication only in living cells.

GLOBAL OUTLINE

- BRL
- History of Events
- The People Involved
- The BRL Approach
- Attack & Propagation Modes
- Network Sweep Tools
- Fixes
- BRL Status

What is BRL?

- U. S. Army Ballistic Research Laboratory
- One of America's foremost research and development labs.
- 700 Scientists & Engineers pursuing in-house research programs
- 5 Scientific Divisions
- 3 Support Divisions
- Networked Computers are all pervasive: throughout research **and** administrative staffs
- > 200 systems
- UNIX Cray X-MP/48 and Cray-2

History, Part 1

- o 1800 PST Wed: virus seen at Rand Corp.
- 2345 EST Wed: virus enters VGR.BRL.MIL.
- 0300 Thu: VGR was seen attacking other machines.
- 1000 Thu: BRL disconnected from MILNET, DISNET. NSI; VGR totally isolated.
- 1200 Thu: BRLXET checking complete: no virus on inside.
- 1600 Thu: Coordinating w/other researchers. DCA orders MILNET hosts shutdown. blows MIL/ARPA gws.
- 2200 Thu: Virus was Lead story on CNN
- **2300 Thu: VGR "Test Cell" prepared, connected to MILNET.**

History, Part 2

- 0645 Fri: MIL/ARPA gateways restored
- 0030 Sat: Virus trapped in "Test Cell", UCB src revd.
- 0630 Sat: BRL-wide power outage (sigh)
- 0600 Mon: 2 Additional attack modules rev-eng.
- 1200 Mon: BRL "Vulnerability Sweep" programs operating
- 1600 Mon: Patched servers installed
- ~1200 Tue: reattach BRL to network

Who BRL Worked With Through the Night

- Tim Smith, US Naval Academy
- Cliff Stoll, Harvard
- Keith Bostic, Berkeley
- Rick Adams, Seismo
- a Jenny, CONUS MILNET Monitoring
- e Bob Fields, CONUS MILNET Monitoring
- CPT Bill Arbaugh, Pentagon
- Peter Yee, NASA/Berkeley

The BRL Approach

- Use instrumented “Test Cell”
- Analyze attack modes
- Coordinate community efforts via telephone
- Assist with reverse engineering
- Relay info on attack modes (incl flukes):
 - 2nd priv inetd (3 sites)
 - Ingres lock daemon
 - System accounting

The Attack Modes

External

- a Sendmail SMTP Server
- Finger Daemon

Internal

- e Password attack [word list]
- /.rhosts
- /etc/hosts.equiv
- .forward

After Penetration

- “Gorch Attack” — sends ll.c sources, compiles and run.
- “L1, Loading” — gets Sun and VAX obj from network.
- “L1, Shell” — Links 2nd stage: “P”
- “P Attack” — Crack & Propagate

Network Sweep Tool

- + Finger Daemon buffer over-run
- FTP bugs
- TFTP bugs
- passwd/rsh
- + SMTP/Sendmail [Wiz, Debug]

Fixes

- Improved fingerd, with logging
- FTPD fixes
- TFTPd fixes

- e Code installed on VAXen, Suns: Goulds
- In progress on Crays, Alliant, Convex: SGI
- BRL has source code licenses.

Books, News

- “Adolescence of P1”
- “Sole on Sapphire”
- Press Coverage was remarkable good. My congratulations to the Public Relations folks.
- My fear: these headlines:

“Computer Virus Spreads to Humans: 96
Left Dead...”

BRL Status

- No information lost
 - Minor disruption of work schedules due to network disconnection
 - BRL Computers now secure against this threat
 - Anti-Viral Team used ~500 man-hours
 - Incidental people used ~1000 man-hours
- Copy of virus still captive in test cell

Who is This MUUSS Fellow, Anyway?

Michael Muuss

Leader, Adv. Computer Systems Team

Ballistic Research Laboratory

APG, MD 21005-5066, U.S.A.

(301)-278-6678

AV 283-6678

ArpaNet: <Mike @ BRL.MIL >

The BRL ~~Virus~~ Busters”

- Mike Muuss
- Phil Dykstra
- Doug Gwyn
- Terry Slattery
- Bob Reschly
- Sue Muuss
- Lee Butler [NASA STScI]

SITE EXPERIENCE

MIT

D. ALVAREZ

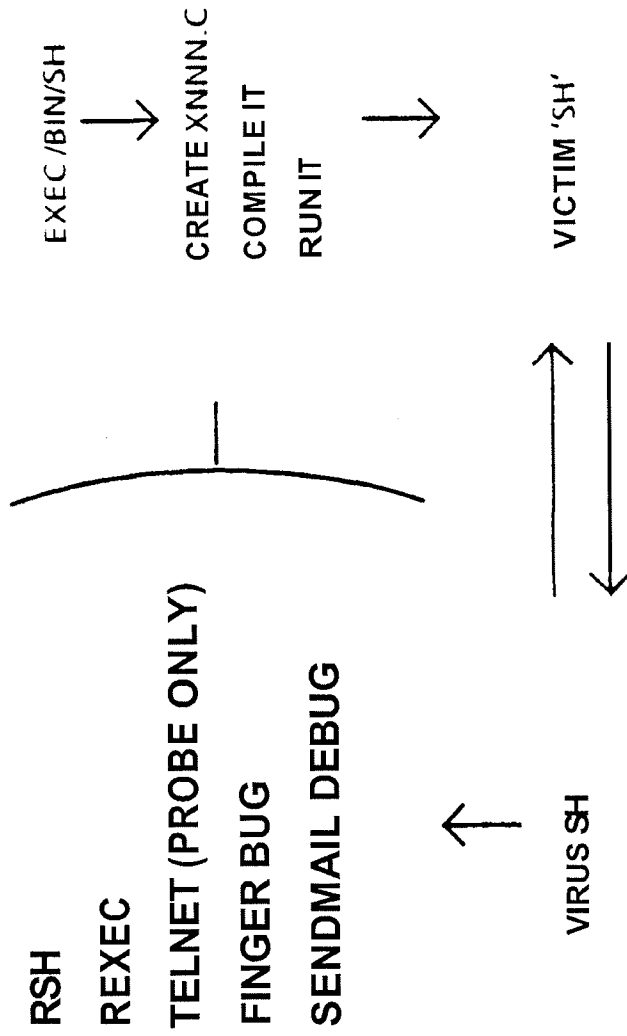
M. EICHIN

J. ROCHLIS

TACTICAL/MANAGEMENT ISSUES

- o SMALL GROUPS 3 TO 5 +
- o PHYSICAL PROXIMITY
- o FUNCTIONAL BREAKDOWN
 - COORDINATING
 - PROTECTING
 - RESEARCHING
- INTERGROUP CONTACTS
 - "OLD BOY NETWORK"
 - TELEPHONES - CAREFUL OF IDSN
- 16 HOURS TO COMMANDS POST
- 3 HOURS TO SECURE (THANKS PETER YEE)
- FEAR/MORALE/COCKPIT ERRORS
- REPORTING KEY TO INTEGRATING - SECURITY COMMUNITY INTO SYSTEM MANAGER COMMUNITY
- EMERGENCY BROADCAST NETWORK - 1200 BAND DIGITAL TAPE RECORDER

MIT SITE EXPERIENCES



MARK EICHIN

MIT SITE EXPERIENCES

JOHN ROCHLIS, MIT NETWORK GROUP

MARK EICHIN, MIT PROJECT ATHENA

- **STUDENT INFORMATION PROCESSING BOARD**
- **PROJECT ATHENA 'WATCHMAKERS'**
- **MIT LAB FOR COMPUTER SCIENCE**
- **MIT MEDIA LAB**

"THE INTERNET VIRUS OF NOVEMBER 3, 1988"

MIT SITE EXPERIENCES

/ETC/HOSTS.EQUIV
/,RHOSTS



HOSTS LIST



ROUTING TABLES
INTERFACE LISTS

USER NAME & PASSWORD



- PERMUTATION OF USER NAMES
- BUILT-IN DICTIONARY
- /USR/DICT/WORDS



FILE/. FORWARD
.FILE/.RHOSTS

Observations on the蔓延 of Virus at MIT

1. Work was performed primarily by small, isolated groups.
Three to five members seems typical.
2. Groups seem to form first by physical proximity, then connect to other groups through "old boy network".
3. Groups seem to break along functional lines:
Coordinating and communicating information,
Protecting and disinfecting machines,
Researching and disassembling virus.
4. Most sites were able to isolate and secure their machines in about three hours after receipt of Peter Yee's message.
5. Very little effort made to contact persons not in "old boy" network
ie. little effort to contact government, etc. until quite late.
6. Initial inter-group communications primarily over telephones
Some later communications possible by computer mail.
7. Groups worked largely in a vacuum, isolated from others simply
because they did not try to contact outsiders.
8. Little amount of unnecessary duplication of effort.
9. Took almost 16 hours before any kind of central command post
was set up at MIT. Post came about largely when two very
competent groups began working on disassembly of virus and
needed to pool resources.
10. Most sites seemed to have expected (and experienced) relapses due
to incomplete inoculation, but were not concerned by this.
11. Group members seemed to be hit by fear only when the virus
reinfected supposedly "safe" machines long after the threat
was believed over (as with the finger daemon attacks). The
illusion of security was shattered.

om ions for the the Future.

1. **Safe use of telephones** is essential. Information on the virus could not have been transmitted between workers without them. **Mixed voice/data** systems make cleanup much more difficult and **dangerous**
2. **Greater mixing** between system managers and government security professional is necessary if a **nationally coordinated** response is to be possible in the future. Most system managers don't know any security professionals, and hence can not include them in their "**old boy network**"
3. A two-pronged, **time-delayed** attack would be extremely demoralizing, particularly if the second attack was timed to hit just when groups were **disbanding** and felt a sense of confidence and security from their work.
4. A computer equivalent of the **Emergency Broadcasting Network** could be extremely important. **Peter Yee's** message was probably the single most decisive factor in a timely response to this virus. Suppose **UUNET** had gone down. The emergency system could take the form of a large bank of phone lines terminating in a **digital tape recorder** containing short recordings at 1200 or 3000 baud. (This solution would be much cheaper than an equivalent bank of modems and less susceptible to hacking). Users would be able to upload system patches and code from this **clearing house** in a timely manner

SITE EXPERIENCE

SRI

D. EDWARDS

IDES:

REAL-TIME
INTRUSION
DETECTION
EXPERT
SYSTEM

SRI-CSL-88-12

SUPPORTED BY USN SPAWAR

SRI SITE EXPERIENCES

- 1. COMPILER USAGE BY DAEMON**
- 2. UNUSUAL COMMAND USAGE BY DAEMON**
- 3. EXCESSIVE AUDIT RECORDS, 10 USAGE, & CPU USAGE**
- 4. NET ACTIVITY BY TYPE**

TIMELINE

WED

9:00PM TERP INFECTS CSLB

10:45 PM OTHER OUTSIDE INFECTIONS

THURS

8:00 AM

9:00AM CSL GW DOWN

11:00 AM CSL BACK

| SRINET HOSTS DOWN

FRI

9:00AM

SITE EXPERIENCE

LOS ALAMOS

ALICE BAKER

LOS ALAMOS NATIONAL LAB

- **DOE CENTER FOR COMPUTER SECURITY**
- **INTEGRATED COMPUTER NETWORK (ICN)**
- LINKED TO ARPANET & MILNET (**UNCLASSIFIED**)

DOE CENTER FOR COMPUTER SECURITY

- FIRST NOTIFICATION FROM HQ
- 0700 MST NOVEMBER 3
- MAJOR EFFORT

NOTIFICATION EFFORT

DOE SITES

DISCONNECT

GATHERING INFORMATION

HQ

LIVERMORE

FOLLOW-UP INFO TO DOE SITES

WORK WITH EXPERTS ON VIRUS

CONFLICTING INFO ON FIXES/ELIMINATION

PREVENTION PATCHES

INTEGRATED COMPUTER NETWORK

- **WHEN NOTIFIED - HAD ALREADY SEEN VIRUS
(NOT IN CLASSIFIED)**
 - **ISOLATED VAX IN UNCLASSIFIED NETWORK**
 - **CAPTURED PART OF VIRUS**
 - **FIX TO SENDMAIL**
 - **ELIMINATION/CLEARING EFFORT**
- LINKED TO ARPANET & MILNET (UNCLASSIFIED)**

PROBLEMS SEEN

DIFFERENT PATCHES -

WHICH ARE OK?

WHO DO YOU BELIEVE?

IS THERE A VIRUS IN THE PATCH?

MESSAGES/INFORMATION

PASSED BY NON-TECHNICAL PEOPLE TO NON-TECHNICAL

PEOPLE

SLOW DISSEMINATION OF INFORMATION

CONFLICTING INFORMATION

SITE EXPERIENCE

**FIXES AND
PATCHES**

P. LAPSLEY

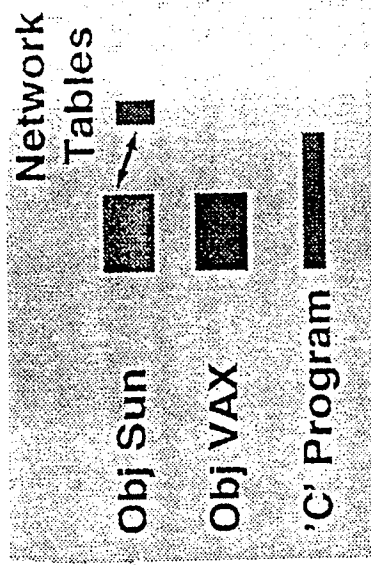
FIXES & PATCHES

- **NIGHT PLAN**
- **BUG FIX PERMANENCE**
- **BACK UPS**
- **NET EXCLUSIONS**
- **MONITOR**
- **CLASSIFIED**
- **MEDIA CONTROL**
- **DIGITAL SIGNATURE**
- **MAIL BRIDGES?**
- **ADMINISTRATOR COOPERATION**
- **OPEN QUESTIONS**

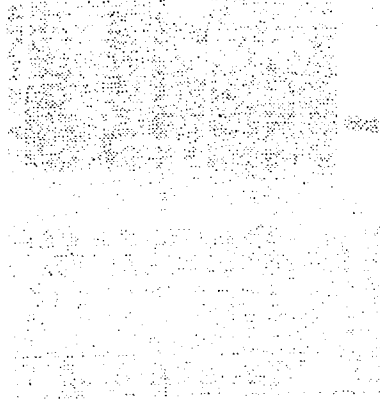
Y

OTHERS

SOURCE



DESTINATION



1 FIND A VICTIM

EXAMINE NETWORK TABLES

SOURCE

Obj Sun

Obj VAX

'C' Program



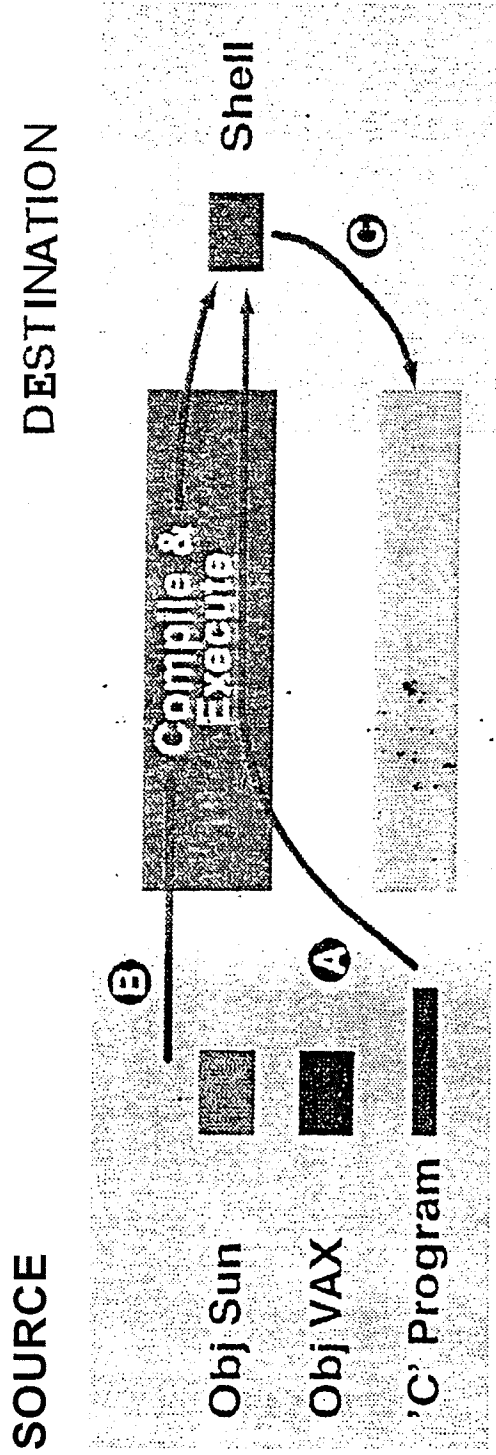
finger
rlogin/rsh

Shell

DESTINATION

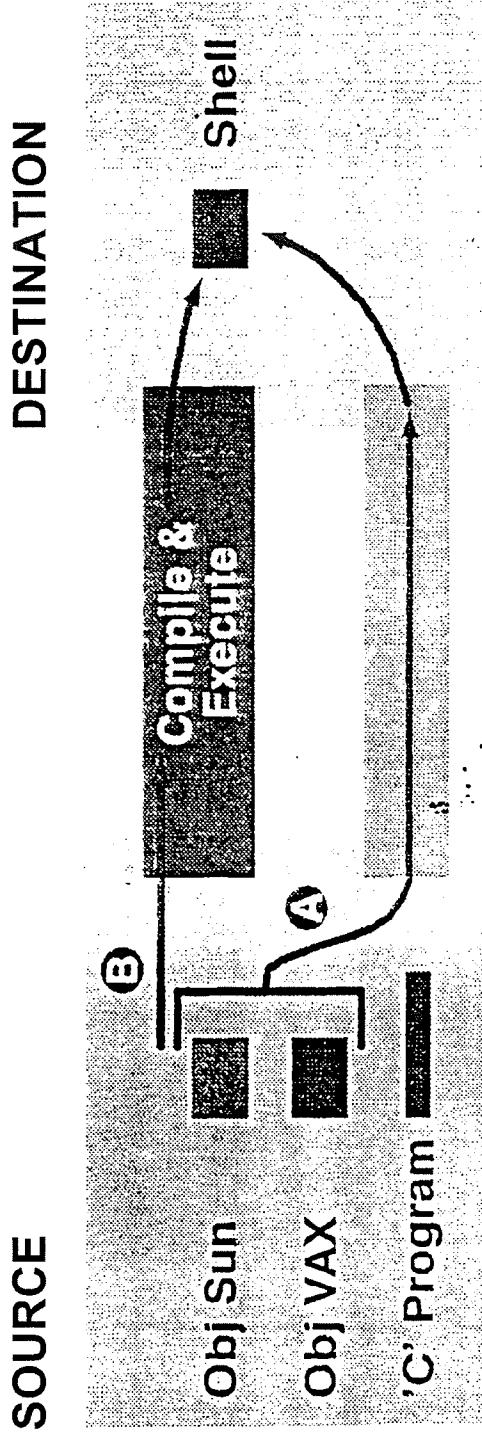
2 GET VICTIM TO OPEN DOOR

PASSWORD ATTACK FOR rLogin/rsh



③ THROW GRAPPLING HOOK IN

- A. TRANSFER HELPER PROGRAM TO DESTINATION
- B. COMPILE & EXECUTE HELPER ON DESTINATION
- C. ESTABLISH "IMAGE" CONNECTION WITH SOURCE



4 PULL ACROSS "REAL" VIRUS CODE

- A. TRANSFER TWO OBJECT MODULES
- B. COMPILE & EXECUTE



DEFENSE ADVANCED RESEARCH PROJECTS AGENCY
1400 WILSON BOULEVARD
ARLINGTON, VA 22209-2308

8 November 1988

MEMORANDUM FOR THE DIRECTOR

SUBJECT: Account of the 2 November 1988 internet virus (updated)

The swiftness of onset and scale of infection of the recent Internet virus reinforce the need to make more aggressive steps in developing appropriate technology and policy for computer security.

The attached memorandum provides details concerning the technical characteristics of the **virus**, and it **makes** preliminary recommendations concerning associated policy issues and areas for accelerated research.

A handwritten signature in cursive script, appearing to read "Bill Scherlis", is written above a horizontal line.

William L. Scherlis

Information Science and Technology Office

A handwritten signature in cursive script, appearing to read "Stephen L. Squires", is written above a horizontal line.

Stephen L. Squires

Information Science and Technology Office

1. THE VIRUS.

1.1. ONSET. The virus appeared on computers interconnected by the **ARPANET**, MILNET, and associated regional and local networks. These are unclassified networks linking tens of thousands of users and supporting a wide range of research and military applications.

The onset of the virus was extremely rapid. There was an initial report from Cornell at 1700 EST Wednesday 2 November 1988, but the first reports outside Cornell occurred four hours later at approximately 2100 EST, when the virus appeared at more than a dozen major sites. Sites affected include UC Berkeley, University of Maryland, Cornell, Carnegie Mellon, NASA Ames, MIT, University of Southern California, UCLA, Livermore Laboratories, BRL, and many others. By midnight, the virus had spread through more than a thousand computers (workstations, minis, and mainframes) on both the ARPANET and MILNET and on connected local networks. The virus first appeared on DARPA/ISTO computers just before midnight.

1.2. SYMPTOMS AND BEHAVIOR. The principal *symptoms of* the virus, as perceived by computer users, are degradation of system response and loss of available space in the file system. These are benign symptoms in the sense that (1) the virus does not delete or alter existing files, and (2) it does not compromise files by transmitting them to remote sites or by altering protections.

The principal activity of the virus is to replicate itself and spread to other machines. The virus runs as a background process on its host, so its presence is not immediately obvious to a user. In many cases, large numbers of multiple independent instances of the virus appear on single machines, with resultant degradation of performance.

1.3. METHODS OF ATTACK. The virus attempts to propagate itself using four methods of attack. Two of the four methods (SENDMAIL and FINGERD) relied on implementation errors (now fixed and distributed to most major sites) in network protocol server programs. A third method (PASSWORD) is a "brute force" method. The last method (RSH) exploits security assumptions in local networks that are violated as a result of successful attack on the external local net security perimeter using one of the other three propagation methods.

It must be emphasized that the implementation errors that permit spread of this particular kind of virus are NOT in the network protocols themselves, or in the host operating system designs or their implementations, or in the computer or network hardware. They are in specific implementations of programs running on hosts that provide specific network services.

SENDMAIL ATTACK. In most cases, the virus propagates itself to a remote machine by exploiting an error in a server program called SENDMAIL that handles the sending and receiving of computer mail. The program implements a network mail protocol called SMTP. There is nothing wrong with the protocol in this case. The error was that the program that uses this protocol adds a new feature. Ordinarily, the program

receives a block of text along with header information indicating which user is the recipient of the message. The block of text is inserted at the end of the user's mail file and a record is added to a log file. Erroneous messages are logged and returned to the sender and possibly a postmaster mailbox as well.

The developer of the **SENDMAIL** program had included a special feature, however, to facilitate his debugging. Mail messages whose headers contain a special **DEBUG** flag are interpreted not as text but as programs to be executed. It must be emphasized that this feature is not part of the protocol, but was included by the developer for his own convenience. It transpired that when the program went into formal distribution the feature was not disabled.

The virus propagates itself by exploiting this feature to create a running process on a remote machine with whatever access and privileges were available to the **SENDMAIL** process. In most cases, because of file protections and operating system safeguards, these privileges are sufficient to do moderate damage at most. In some cases (usually involving poor systems configuration), the potential for damage is much greater.

But, as indicated above, the virus does not remove files even when it is possible for it to do so. In this sense, it is benign.

PASSWORD ATTACK. The virus tries to establish itself as a legitimate user (rather than remaining a system process with few privileges) on the **infected host** and other local machines by guessing passwords. It does this by trying **as passwords (1)** the words in the standard online spelling dictionary, **(2)** various transformations on the users name, and **(3)** words in a special list of possible passwords included in the virus itself. Ordinary login attempts cannot use this technique because time delays are generally inserted on all password failures. In this case, the virus uses its own implementation of the DES algorithm to generate the encoded password representation used in Unix password files. (This could imply that the virus is subject to export control in the same way that Unix with DES is currently subject to export control.)

There were cases in which this guessing of passwords by the virus was successful, and the virus often appeared running as if it were a legitimate user. The attacking program contained no indication of any intent to exploit special access it might acquire as a result of this attack.

RSN ATTACK. Once established on a local network, the virus could propagate itself by exploiting a feature, called **RSH**, that enables local machines to authenticate users for each other. This feature is convenient when a local network is itself well protected, and when users on that network must interact frequently. If the feature is enabled in a local network, and if the virus had succeeded as masquerading as a legitimate user, then it could spread quickly in a local net since the machines in the net would assume that the virus had already been authenticated.

FINGERD ATTACK. A fourth method of entry was to exploit an error in a different protocol server program, for locating users on remote machines. This program error is exploited by the virus to establish a running process on the remote machine.

1.4. ESTABLISHING THE INFECTION. After a successful **attack**, the first stage of infection is the infiltration of a small "bootstrap" program onto the remote machine. The bootstrap program then retrieves from the previous point of infection a much larger main program. Both the bootstrap program and the main program were designed to evade detection by masquerading as system or user processes and by removing the programs from the disk once they are running in memory.

The bootstrap program is transmitted in source form, and it compiles and loads itself on the remote machine. Its main function is to retrieve the main program. As the virus propagates, the bootstrap program is adapted (by the main program that propagates it), so that it refers only to the most immediate infected source.

The main program, which contains the actual code for assaulting remote machines (using the four methods detailed above), is transmitted in object form. Actually, two versions of the program are transmitted for two different instruction set architectures. A portion of the data of the main program is encrypted (using a simple XOR code). When the program starts, it decrypts most of its data area that is the main memory of the newly infected host. The disk version remains in encrypted form, and is eventually deleted as the virus covers its tracks.

Once the main program is running, the machine is in an infectious state. In many cases, multiple instances of the program were running simultaneously, each attempting to infect other machines on the network. Randomization techniques are used to ensure that the multiple instances did not overly interfere with each other. The virus would also occasionally spawn a clone of itself and then terminate, with the effect that no large accumulations of CPU time would be evident on casual browsing of process status information.

1.5. DETECTION AND DIAGNOSIS. The presence of a large scale virus infection is readily detectable by casual users due to its effect on machine performance. Small scale infections are not as easily noticed (and, indeed, it is easy to imagine that the virus could have been tuned to be less readily detectable by decreasing the extent of denied service). Expert users generally could spot the spurious running processes and remove them as they appeared. This provides fast symptomatic relief, but not immunization.

When detection first occurred Wednesday night, many sites disconnected themselves from the network and powered down critical machines. Both Livermore Labs and NASA Ames disconnected themselves from the network. Bridges between MILNET and ARPANET were closed, but only after the infection had already spread to MILNET. Many sites left one or two machines running in order to enable communication with other sites and to permit study of the virus activity. In the DARPA/ISTO local network, for example, infection occurred around midnight Wednesday. (The other DARPA offices were unaffected because they are not on the network.) The network connections were disabled during the night, and machines were powered down Thursday morning.

As the virus spread; systems programmers at the various network sites established close communication and were able to share observations and results on an hourly

basis. By continually killing spurious processes as they appeared on computers, most of the systems programmers were able to stay online and share results using network mail and bulletin boards. The virus did, however, have the effect of slowing communications on the network as it spread Wednesday night and Thursday morning. Because of the close working relationship DARPA has with the research community affected, it was able to facilitate communication among groups, track the situation, and keep appropriate people advised. Many of the procedures followed at **DARPA** were based on a prior experience with the 13 May 1988 virus hoax.

Monitoring of the virus process activities revealed the various methods of attack that were used, which led to the development of immunization techniques and implementation of preventive measures.

1.6. IMMUNIZATION AND PREVENTION. For each of the four methods of attack, immunization and/or prevention measures were developed. Many major sites had already eradicated the virus and were immunized by Thursday evening or early Friday morning. **DARPA** machines were running and connected to the network within 18 hours of appearance of the virus at **DARPA**.

SENDMAIL -- IMMUNIZATION. This method of attack was permitted due to an error in a widely distributed mail protocol server program. Within hours of discovery of the virus, fixes were in general distribution. The first posting was made at 0600 EST Thursday, with corrections that followed. The fixes were sufficiently simple that they could be carried out by instructions given over the telephone. These fixes generally prevented infection of a site, if it was not infected already.

PASSWORD -- PREVENTION. This method of attack works only in cases where users fail to follow conventional password guidance, which is not to use dictionary words or their own names. Affected users and potentially affected users were instructed to change their passwords.

RSH -- PREVENTION. This method of attack works only because of a failure of the external security perimeter of a local network. In most cases, the level of trust among machines in local networks was temporarily reduced (i.e., by disabling RSH) pending full eradication and immunization.

FINGERD -- IMMUNIZATION. A day after discovery of the virus, fixes for FINGERD were in general circulation. The error was a common programming error. Input to FINGERD that was too long resulted in certain unrelated internal data areas being overwritten by portions of the input. The virus exploited this by using overlong input values that overwrote the unrelated data areas with data that resulted in the virus being able to start a new process. The fix to FINGERD is to insert a check for incorrect input.

1.7. ASSESSMENT AND RECOVERY. Other than denial of service and lost time, no specific unrecoverable damage was caused by the virus. As indicated above, no files are known to be lost, and no information is known to be compromised.

Once eradication and immunization were underway, the systems programmers at Berkeley and MIT embarked on a project to analyze the 60000 bytes of encrypted object code and data for the main program. A special program was written to decrypt the data for the main program. The dictionary of common passwords stored in the virus was extracted and distributed to many sites.

The major challenge of the analysis project was reverse-engineering the object code into source programs. A preliminary version was completed on Saturday 5 November. MIT has released a preliminary document describing the actions of the object code.

The derived source code itself is not being released, however, since many systems are not yet fully immunized, and the code exposes specific vulnerabilities. The program is sophisticated and was written by someone with considerable systems expertise.

The smaller "bootstrap" program used upon initial penetration is propagated in source form. Copies of the messages were obtained when mail to remote sites not running the bad SENDMAIL program returned the message back to the Postmaster mailbox of the originating (previously infected) site. These intercepted mail messages contain the source text.

2. PRELIMINARY OBSERVATIONS.

2.1. **RISKS.** The ARPANET is a dual-use network. It serves as a laboratory for performing experiments in large scale networking while providing services for the research community. Because of the leverage it provides, this dual-use approach is common in the computing research community, and applies to other large scale technologies such as operating systems, parallel computers, user interaction systems, experimental expert systems shells, and the like.

Historically, the research community has been willing to sustain the additional risk in order to obtain functionality beyond the state of the art

Policy requires that no classified data be accessible on the ARPANET, MILNET, and interconnected networks, except through NSA certified private line interfaces. Messages encrypted using approved devices are unclassified. The Internet community consists of 300 or more sites, some of which have hundreds (and in some cases thousands) of computers attached to local networks. A common set of protocols, called TCP/IP, enables communication in the net despite the wide range of computers and operating systems employed.

A key issue is the extent to which improved security safeguards are required by the Internet community.

2.2. **COSTS.** Current systems that have high security requirements generally achieve this through (1) physical isolation of the network or computing installation (an exception is the use of NSA approved private line interfaces), (2) provision of access only to cleared personnel, and (3) use of design and engineering principles including

redundancy, tagging, and precise specifications. Satisfying these requirements generally means making sacrifices in functionality, performance, and cost. Interoperability and open interfaces are also often sacrificed, making it difficult to incrementally improve the capabilities of the systems after deployment.

In research systems, on the other hand, security is often sacrificed in order to maximize functionality, performance, and flexibility. In general, however, there are tradeoffs among these characteristics, with security currently exacting a very high cost.

2.3. VULNERABILITIES. The virus exploited errors in the implementation of two protocol server programs. Installation of correct versions of the programs, as was done as part of the response to this virus! resulted in immunization.

The virus exploited implementation errors. The vulnerabilities exploited by the virus are NOT in the network protocol design: the operating system design, or the underlying hardware design.

(This is in distinction with the PC community, in which viruses are able to propagate and cause damage as a result of specific shortcomings of design of the PC operating systems. In the PC community, virus detection and eradication are often quite difficult, and immunization is often impossible.)

It should be noted that if the author of this virus had chosen to be destructive, wanton destruction of user files would ~~nonetheless~~ have been prevented by a properly implemented and configured operating system. Errors in implementation can result in vulnerabilities, of course.

For this reason, formal security guidelines such as those articulated in the Orange Book emphasize good implementation practice. B1 secure implementations of Unix now exist, and implementations at higher levels of security are being developed for Unix/POSIX (B3 level) and for Mach (A level and beyond). Confidence in security in these cases is achieved through a social process involving attention to design principles and inspection of code. Higher confidence can be obtained using the formal methods approaches that now being developed.

It is probably fair to conjecture, however, that even if the operating system kernel was trusted at B3 or A level, a virus would still be able to propagate itself by exploiting server errors (in cases where servers are outside the kernel). Of course, this hypothetical virus would not be able to damage or compromise protected data.

3. TECHNOLOGY AND POLICY ISSUES

3.1. GOALS. In the near term, effective procedures must be developed that can provide suitable response to viruses that can spread to thousands of computers across the country in a matter of hours. as this one did. In the longer term, policies and technology solutions must be developed to reduce vulnerability of both classified and unclassified networks and systems while not sacrificing functionality and performance.

3.2. RESPONSE PROCEDURES. We recommend the formation of a National Computer Infection Action Team (**NCIAT**) to work in the Defense and national research communities.

FUNCTION. The NCIAT would have three functions: (1) It would provide a mechanism for coordinating response in acute situations. As the recent virus episode demonstrates, extremely rapid mobilization and coordination with the community is essential. (2) It would provide a coordination point for rumors of viruses. In the recent virus episode there was no advance warning; the virus simply appeared. Several months earlier, however, there was a case in which there were rumors of a virus about to strike, with tremendous resulting defensive activity in the community. The virus was a hoax. (3) It would provide a focal point for discussion of prevention, coordination, and awareness in the community, perhaps through publications.

ORGANIZATION. The NCIAT would operate at three organizational levels. (1) The top level would consist of an "Executive Group" at the level of flag officers who would empower the group and have sufficient authority and access to permit fast response when required. (2) The middle level "Action Group" would provide working level support in the government. (3) The operating level "Associates Group" would include elite systems programmers from industry, government, and the research community. This group is the heart of the NCIAT. These positions **would be** assigned in such a way that appointment as an Associate is a **mark of significant** recognition and accomplishment as a senior systems programmer. Rotating terms of appointment would enable a new set of Associates to be designated each year after a formal selection process. This would ensure effective community representation. Retired Associates remain a source of expertise, though they are not expected to provide the same rapid response as Associates. Associates would become a primary means of access for the community to NCIAT both for routine and emergency operations.

Membership in NCIAT Executive and Action groups would include Services and Agencies in DoD, NSA, NCSC, NIST, NSF, the FBI, and other appropriate organizations. Close coordination contacts would be developed with industry and with major research laboratories, including the National Labs. A database of key experts and industry and government contacts would be maintained. NCIAT would have a small core staff to support routine operations, data collection and dissemination, and, in acute situations, communications with NCIAT group members and others. The NCIAT would focus its initial efforts in the Internet community.

NCIAT would have a well-known network mailbox, an 800 number, and a computer facility to provide database service and to enable emergency data and authentication communications. The computer facility would consist of a primary system that is connected to the Internet and a secondary system that is not connected to the network: but only to the first system, and through a protected interface. The primary system would serve as a database platform and would support routine operations. The second system, through provision of dial-up or other special access support, would provide

NCIAT members and others in the community with a known communications point to be used in an emergency, even if the Internet should become damaged or unavailable.

Community support for NCIAT is essential, since discussion of local viruses and vulnerabilities can require a high level of trust and respect for privacy. It is anticipated that much coordination with the user and systems support community would occur at the Associates level.

3.3. TECHNOLOGY CHALLENGES. We recommend that security assessments should be done for existing nonclassified systems in order to determine (1) what are appropriate natural levels of security that can be achieved with reasonable impact (e.g., cryptographic checksums for configuration management, validation viruses, server and gateways audits, audit trails, authentication service), and (2) what mid-term technology steps can be made that will provide significant improvements.

For the longer term, we recommend acceleration of investment in technology for the development of trusted and secure systems. The challenges are (1) to increase the absolute level of security attainable and (2) to reduce drastically the functionality and performance premium for security and trust. The first challenge must be met if we are to build systems that provide the very high levels of security assurance and trust that are required in highly sensitive applications and in life-critical systems.

A basic technology in this area is formal methods, which also has applications to parallel programming and program optimization. The European defense community is already moving towards use of formal methods for systems acquisitions in which safety and security are critical. A verified microprocessor chip design has already been produced by RSRE.

Major areas for development with more immediate payoff include (1) operating systems security; particularly for parallel operating systems, (2) secure network technology, (3) trusted servers, including authentication service and network file service, and (4) trusted hardware designs, such as for embedded 32 bit RISC processors.

3.4. POLICY AND BALANCE. We recommend that closer working relationships be developed among the various organizations involved in computer security and trust. At a minimum this includes NSA (as a user), NCSC (as a policy and certification organization), NIST (as a policy and certification organization), DARPA (as a technology developer), DCA (as a network operator), and Service agencies.

In the recent episode, an informal open process in the community led to fast eradication and immunization. It is obvious that any formalized response mechanism must be at least as efficient as the current process. This requires clear channels of communication, trust and cooperation among the parties involved, effective two-way information flow, and, most importantly, the empowerment of the best technical people available in the community to work together to detect, diagnose, and resolve acute problems when they occur.



ACQUISITION

THE UNDER SECRETARY OF DEFENSE
WASHINGTON, DC 20301

OFFICE OF THE
SECRETARY OF DEFENSE

88 DEC -5 PM 2:37

DEC 5 1988

DEPT. OF DEFENSE

5 DEC 1988

W/T

W/S/SE

MEMORANDUM FOR SECRETARY OF DEFENSE

SUBJECT: Summary Report of the OASD(C3I) Executive After Action Assessment Team on the Computer Virus of November 1988
-- ACTION MEMORANDUM

An Executive After Action Assessment Team met on November 14 to assess the Internet computer virus attack which was first detected on November 2. The review team was composed of senior representatives from OASD(C3I), DCA, DARPA, NSA and JCS/J6 (Tab H). The team reviewed the events and actions taken after the detection of the virus on ARPANET and MILNET on November 2; reviewed the DARPA report on the technical characteristics of the virus (Tab G); reviewed the report by the National Computer Security Center of the Proceedings of the Virus Post-Mortem Meeting held November 8 (Tab F); and concluded with recommendations for improving the Department's responsiveness to future attacks.

The team generally recognized that due to the extraordinary efforts of a few talented people and the specific nature of this virus, the Department of Defense did not experience a major catastrophe. However, preventive actions should now be taken to reduce the DoD's exposure to future, potentially more destructive viruses. The team concluded that improvements at the national level and within the Department of Defense and other Federal Agencies are advisable and could be grouped in two general categories--response organization and improved awareness.

In order to provide a rapid response capability, there should be a "central coordination center" established as quickly as possible with the following characteristics:

- national level center
- manned 24 hours/day, 7 days/week (could be an extension to an existing center like the NCC under the NCS)
- emergency alerting procedures including key personnel recall (to include key network operations centers and investigative (DoJ/FBI) poc's)
- access to executive level decision makers if necessary

4

48555

- establish contact to technical experts both in industry and academia
- focal point when major problems (viruses as well as other computer security related vulnerabilities) are identified
- receive problem reports
- coordinate solutions
- able to authenticate source of corrections
- emergency communications capability
- available as the single interface point to press
- archival repository

This central coordination center should be designed under the joint auspices of the National Computer Security Center (NCSC) under NSA and the National Institute of Standards and Technology (NIST, formerly NBS) under Commerce, with technical assistance from DARPA. Its primary focus would be in the unclassified domain, but extensions to the procedures should be developed to deal with classified network/computer events. The Joint Staff is aware of the potential impact on DoD's classified networks and is working that issue in parallel. Current prototype coordination center efforts being initiated by the Software Engineering Institute for DARPA provide conceptual demonstrations and should be the design model for the center.

There is also a need for increased security awareness relating, for example, to passwords and file backups. Lessons learned from this particular virus attack should be documented jointly by the NCSC and NIST and then widely published. Additionally, the Office of Personnel Management (OPM) should be provided with a copy of this report for use in their future training endeavors.

In addition to the general recommendations above, there were events that occurred during the virus that warrant further specific DoD actions. Due to the limited information available and the rapid changes that have occurred in all local and wide area data networks in the past several years, a current vulnerability assessment of all major DoD networked systems should be completed. This action may well uncover additional actions which should be taken to reduce the risk of or the effect of future virus attacks. Consideration should be given to assembling a minimum set of virus analysis tools. The memorandum to NSA (Tab B) includes both of these requirements. The need for intensified research and development in this particular computer security area is also stressed to both NSA and DARPA in the memorandum. Further, responsibilities for the security, management and operation of the Defense Data Network and ARPANET should be more clearly defined, coordinated and documented. The memorandum to DCA and DARPA (also Tab B) includes this requirement.

Recommendations: Because of the joint effort required from Commerce and NSA, recommend that you sign the letter (Tab A) to Mr. Verity, Secretary of Commerce, requesting their collaboration in the development of the response organization and improved awareness.

Recommend you sign the memorandum at Tab B asking NSA, DCA and DARPA to support the findings of the After Action Assessment Team.

Recommend you sign the letter at Tab C to the Department of Justice asking for their support in the development of the central coordinating center.

Recommend you sign the letter at Tab D to OPM forwarding the findings and recommendations to them for their consideration in future computer security training.

Once these actions are complete the press release prepared by OASD (Public Affairs) (Tab E) is recommended for immediate release.

Coordination:

DCA/BGen Bracher *via phone/18 Nov88
 DARPA/Dr. Fields *via phone/18 Nov88
 NSA/Mr. Gallagher — *via phone/18 Nov88
 Joint Staff/J6, Dr. Bialick *via phone/18 Nov88

*signature copies will be added as soon as available

Coord: ASD(FM&P)

Grant S. Green, Jr

Prepared by: DFountaine/IS/57181

Coord: ASP (PA)

Fred S. Hoffman
 Principal Deputy Assistant Secretary

DEC 15 1988



ACQUISITION

THE UNDER SECRETARY OF DEFENSE
WASHINGTON, DC 20301

OFFICE OF THE
SECRETARY OF DEFENSE

88 DEC -5 PM 2: 37

DEC 1 1988

DEFENSE

5 DEC 1988

MEMORANDUM FOR SECRETARY OF DEFENSE

SUBJECT: Summary Report of the OASD(C3I) Executive After Action
Assessment Team on the Computer Virus of November 1988
-- ACTION MEMORANDUM

An Executive After Action Assessment Team met on November 14 to assess the Internet computer virus attack which was first detected on November 2. The review team was composed of senior representatives from OASD(C3I), DCA, DARPA, NSA and JCS/J6 (Tab H). The team reviewed the events and actions taken after the detection of the virus on ARPANET and MILNET on November 2; reviewed the DARPA report on the technical characteristics of the virus (Tab G); reviewed the report by the National Computer Security Center of the Proceedings of the Virus Post-Mortem Meeting held November 8 (Tab F); and concluded with recommendations for improving the Department's responsiveness to future attacks.

The team generally recognized that due to the extraordinary efforts of a few talented people and the specific nature of this virus, the Department of Defense did not experience a major catastrophe. However, preventive actions should now be taken to reduce the DoD's exposure to future, potentially more destructive viruses. The team concluded that improvements at the national level and within the Department of Defense and other Federal Agencies are advisable and could be grouped in two general categories--response organization and improved awareness.

In order to provide a rapid response capability, there should be a "central coordination center" established as quickly as possible with the following characteristics:

- national level center
- manned 24 hours/day, 7 days/week (could be an extension to an existing center like the NCC under the NCS)
- emergency alerting procedures including key personnel recall (to include key network operations centers and investigative (DoJ/FBI) poc's)
- access to executive level decision makers if necessary

~~45555~~

45555

- establish contact to technical experts both in industry and academia
- focal point when major problems (viruses as well as other computer security related vulnerabilities) are identified
- receive problem reports
- coordinate solutions
- able to authenticate source of corrections
- emergency communications capability
- available as the single interface point to press
- archival repository

This central coordination center should be designed under the joint auspices of the National Computer Security Center (NCSC) under NSA and the National Institute of Standards and Technology (NIST, formerly NBS) under Commerce, with technical assistance from DARPA. Its primary focus would be in the unclassified domain, but extensions to the procedures should be developed to deal with classified network/computer events. The Joint Staff is aware of the potential impact on DoD's classified networks and is working that issue in parallel. Current prototype coordination center efforts being initiated by the Software Engineering Institute for DARPA provide conceptual demonstrations and should be the design model for the center.

There is also a need for increased security awareness relating, for example, to passwords and file backups. Lessons learned from this particular virus attack should be documented jointly by the NCSC and NIST and then widely published. Additionally, the Office of Personnel Management (OPM) should be provided with a copy of this report for use in their future training endeavors.

In addition to the general recommendations above, there were events that occurred during the virus that warrant further specific DoD actions. Due to the limited information available and the rapid changes that have occurred in all local and wide area data networks in the past several years, a current vulnerability assessment of all major DoD networked systems should be conducted. This action may well uncover additional actions which should be taken to reduce the risk of or the effect of future virus attacks. Consideration should be given to assembling a minimum set of virus analysis tools. The memorandum to NSA (Tab B) includes both of these requirements. The need for intensified research and development in this particular computer security area is also stressed to both NSA and DARPA in the memorandum. Further, responsibilities for the security, management and operation of the Defense Data Network and ARPANET should be more clearly defined, coordinated and documented. The memorandum to DCA and DARPA (also Tab B) includes this requirement.

Recommendations: Because of the joint effort required from Commerce and NSA, recommend that you sign the letter (Tab A) to Mr. Verity, Secretary of Commerce, requesting their collaboration in the development of the response organization and improved awareness.

Recommend you sign the memorandum at Tab B asking NSA, DCA and DARPA to support the findings of the After Action Assessment Team.

Recommend you sign the letter at Tab C to the Department of Justice asking for their support in the development of the central coordinating center.

Recommend you sign the letter at Tab D to OPM forwarding the findings and recommendations to them for their consideration in future computer security training.

Once these actions are complete the press release prepared by OASD (Public Affairs) (Tab E) is recommended for immediate release.

Coordination:

DCA/BGen Bracher *via phone/18 Nov88
 DARPA/Dr. Fields *via phone/18 Nov88
 NSA/Mr. Gallagher *via phone/18 Nov88
 Joint Staff/J6, Dr. Bialick *via phone/18 Nov88

*signature copies will be added as soon as available

Coord: ASD(FM&P)

Grant S. Green, Jr

Prepared by: DFountaine/IS/57181

Coord: ASP (FA)

Fred S. Hoffman
 Principal Deputy Assistant Secretary

DEC 15 1988

MEMORANDUM

OFFICE OF THE DEPUTY SECRETARY

... should
Office of the Deputy Secretary of Defense

December 12, 1988

MEMORANDUM FOR: Assistant Secretary of
Defense (PA)

Please see comment on attached from Mr. Taft
which reads:

"Dan Howard should look over the press
release here. We should probably have a
briefer available on it. WHT, IV"

respectfully,*

James R. Brout. III

Military Assistant

c: USD(A)

Attachment

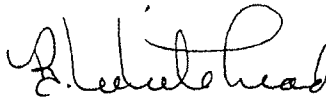


OFFICE OF THE SECRETARY OF DEFENSE

MAD 12 Dec 88
MEMO FOR ASD(PA) Rev.

1. See note next under for required action.
2. Please return complete package to CTD, 3A948.

VIR



Beverly C. Whitehead
Staff Assistant
Corres & Directives

cc: USDO(A)

W.

OFFICE OF THE SECRETARY OF DEFENSE

The Senior Military Assistant

SECDEF

Return to [Signature]

Principal concern:

We need to incorporate some qualifications to this!

e.g.

-No computers associated with our Nuclear War Plans or Release procedures, or any of our sensitive computers within OSD were or could have been affected.

SEC DEF HAS SEEN

DEC 20 1988

[Handwritten initials]

V/R [Signature]

12/20/88

C+D

Note 'Klm Owens' "So

2

Nancy
✓

14 November 1988

The Center would appreciate knowing of any errors in the enclosed Proceedings of the Virus Post-Mortem Meeting. Please provide corrections to:

National Computer Security Center
Attn: C34
9800 Savage Road
FT. George G. Meade, MD 20755-6000

If comments are received before 10 December 1988, we will publish a set of corrections to be mailed by 17 January 1989.

DRAFT PRESS RELEASE

IMMEDIATE RELEASE

No.
(202) 695-0192 (info.)
(202) 697-3189 (Copies)
(202) 697-5737
(Public/Industry)

IMPROVEMENTS IN COMPUTER SECURITY PROCEDURES

Secretary of Defense Frank C. Carlucci has authorized several measures to improve computer security procedures within the Department of Defense (DoD). These steps resulted from an internal assessment of the Internet computer virus attack that was first detected on November 2, 1988. The preventive measures are designed to reduce DoD's exposure to future, potentially more destructive viruses and to provide fast, effective response should unauthorized intrusions happen again in government computer networks.

Essentially these initiatives call for greater awareness of the dangers of virus attacks and the establishment of a central response organization. Implementation will require cooperation from other Government Agencies; to that end, the Department of Commerce, through its National Institute of Standards and Technology (NIST), the Department of Justice, and the Office of Personnel Management have been asked to join in combatting the problem of computer viruses.

To increase awareness, the National Computer Security Center (NCSC), under the National Security Agency (NSA), and Commerce's NIST will develop a report on the lessons learned from the early November attack. Among the lessons already identified are requirements for frequent backup procedures to prevent loss of data and the need to discourage the use of common passwords, such as proper names or words found in the dictionary. This report will be available to users and training officials throughout the government.

The DoD is proposing the establishment of a 'central, nationally-based coordination center to handle emergency situations involving computers and networks. This center would be in operation 24 hours a day, have contact with technical experts both in industry and academia, and be the focal point--for operating and investigative personnel--when major problems are identified. The center would receive problem reports, coordinate solutions, be able to authenticate sources of corrections, and provide information to the public on the attack.

Secretary Carlucci has also directed NSA to undertake a current vulnerability assessment of all major DoD networked computers. In addition, DoD will be reviewing the need for intensified research and development against virus attacks. DARPA is implementing a coordination center at the Software Engineering Institute to provide direct support to the Internet community which consists primarily of research institutions. This center will be developed in close coordination with NCSC and NIST, and will provide a prototype for the operational systems of broader scope that they will be developing.

TAB E

RECOMMENDATIONS FROM THE 8 NOVEMBER 1988
POST MORTEM OF THE ARPANET/MILNET VIRUS PROPAGATION

1. Establish a centralized coordination center.
This center, supported jointly by NIST and NSA, would also function as a clearinghouse and repository. Computer site managers need a place to report problems and to obtain solutions. This center might evolve into a national level command center supporting the government and private sector networks. The center needs to provide 24 hour service, but not necessarily be manned 24 hours a day (i.e., responding via beeper after hours might be acceptable).
2. Establish an emergency broadcast network.
In the ARPANET/MILNET case, the network was used to disseminate the patches (i.e., antidote) at the same time the virus was still actively propagating. If the net had gone down, there would have been no way to coordinate efforts and disseminate patches. It is recommended that a bank of telephone lines be designated as an emergency broadcast network. The phones would be connected to digital tape recorders and operate in a continuous broadcast mode (or a recorded "binary" announcement mode) to disseminate network status, patches, etc.
3. Establish a response team.
The technical skills required to quickly analyze virus code and develop antidotes or system patches are highly specialized. The skills required are system specific (i.e., UNIX 4.3 in this case), and in many cases exist only at vendor development facilities (e.g., the majority of commercial operating systems are proprietary and source code is not provided to users). The concept of a response team would require advance coordination so that personnel with the requisite skills can be quickly mobilized.
4. Maintain technical relationships with the computer science "old boy network".
The ARPANET/MILNET virus was analyzed and eradicated through the services of this old boy network, not by U.S. Government (USG) personnel. This old boy network is willing to participate in supporting USG initiatives; however, their consensus, support, and trust is required.
5. Centrally orchestrate press relations.
An inordinate amount of time at virtually every site was spent responding to the news media. Multiple press reporting from geographically dispersed sites has the potential for circular reporting of incorrect and misleading data. A single USG focal point at the national level to interact with the press is recommended.

ENCLOSURE