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.

[illegible]

William L. Scherlis
Information Science and Technology Office

[illegible]

Stephen L. Squires
Information Science and Technology Office

1. THE VIRUS.

1.1. ONSET. The virus appeared on computers interconnnected **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 esisting 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 protoco! 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 grearer.

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 wih few privileges) on the infected host and other local machines by guessing passwords. It does this by trying as passwords (I) 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 techniqes 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 occured 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 occured 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 sires esrabiished 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. Becuase 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 functionalty 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 authon 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 supprt 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.