COMPUTER VIRUS PREVENTION AND CONTAINMENT ON MAINFRAMES

Ghannam M. Al-Dossary, Manager
Industrial Security Planning and Support Services Department
Saudi Aramco
Box 90, Dhahran 31311 Saudi Arabia

Abstract

A computer virus can be a vicious and insidious form of code. It has the ability to replicate itself, to attach itself to other code, to spread through a computer system or network, and often to initiate a harmful series of instructions when a "trigger" point is reached. Viruses can have a major impact on productivity because of the steadily increasing dependence of industrial, business, and government functions on the availability and integrity of data processing systems. Although mainframe computers have been the target of virus attacks less often than microcomputers up until now, there is no room for complacency when the stakes are so high. The novelty, the technical nature, and the tendency to romanticize this phenomenon, have resulted in a "black-box" syndrome ("I don't know what's going on in there.") and a feeling of overwhelming impotence in the business community.

The risk of viruses can be reduced. One approach is to examine the constituent parts from which a virus is composed, and to design a comprehensive defense which reckons with each of these parts. The protection chain will only be as strong as its weakest link. The author of this paper suggests a classification scheme which is useful in understanding the components of a virus and useful methods for maintaining the integrity of a computer system.

This paper outlines basic prevention, detection, and correction techniques which are available today to reduce the threat of damages caused by viruses. These include software "vaccines" or filters; encryption; access control software (e.g. RACF, ACF2, and Top Secret); "test-to-production" control procedures; back-up and recovery procedures; personnel selection and review controls; and physical access control.

The concepts presented in this paper conform to the "Trusted Computer System Evaluation Criteria" *developed by the United States Computer Security Center and use examples from major published virus incidents to illustrate the price of control weaknesses. The paper concludes that no working computer system is impregnable but that much can be done by industry to make most computer systems less inviting to attacks from viruses.*

A bibliography is included for further study.

I. The Nature of the Virus:

A virus is a parasitic form of computer code which has the particular characteristic of being able to reproduce itself. Not only may it produce exact copies of itself which multiply exponentially through a system [1] but in some cases it may produce an evolved copy of itself which is able to adapt to the conditions which it encounters in a particular environment.[2] Although the vast majority of virus attacks reported to date [3] have been against personal computers, the threat against mainframe computers is continuing and limited publicity should not be an excuse for complacency. The original research on viruses was conducted on mainframe computers [2] and it was demonstrated that viruses are easily created on any commercially available operating system including MVS, VM, VMS, and Unix. The published incidents of mainframe virus attacks show that the creation of a virus may even be accidental rather than intentional [4] when systems are poorly designed.

The virus concept exploits one of the most fundamental qualities of general purpose computing systems: the leveraging effect of making more information available to more people. A lever is a powerful tool in moving a desired object toward a specific goal. However, a very small or subtle shift in the location of the fulcrum (caused, for instance, by the introduction of a virus to a computer program) can cause the desired object to move rapidly away from the goal. The goal of computing is to provide information. A virus can deny the availability of information processing services or can distort the information itself. (Special function viruses can also theoretically increase information availability [2] but that is not included in the subject of this paper.)

The same computer hardware may be used to meet an wide variety of disparate computing needs. Software, like an idea, can often be shared, copied, and used over and over again in thousands of machines with millions of repetitions without dissipating or reducing the power or effectiveness of the original application. A virus may exploit both the known hardware characteristics of a particular system and the shareability of software. It has been theoretically demonstrated that virus contamination cannot be contained without interfering with the shareability of software. [2] Total isolationism is the only complete cure for viruses but a significant risk reduction is possible through a limited compromise in software shareability. [5]

Viruses exploit the Von Neumann computer architecture which is common in almost all business computing systems today. This architecture treats stored software as data which can be dynamically modified to meet varying requirements. This is true of both operating system and applications programs. Viruses take advantage of this ability of executable programs to be changed, to introduce new purposes which maybe at variance with the purposes of the owner of the computer system or application.

A stored (and potentially modified) program may remain dormant as a file for a significant period of time before it is executed and any changes are detected. This delay allows the infection to spread throughout the computer system and to any other system receiving output from the first system. The original technical analysis of viruses conceived of a very sophisticated strain of virus which would be active across operating systems. (Telecommunications have made significant advancements since that time.) The conclusion: "If a computer virus of this type spread through the computers of the world, it would likely stop the majority of computer use for a significant period of time, and wreak havoc on modern government, financial, business, and academic institutions." [2]

II. The Anatomy of the Virus:

The code in a virus may be divided into two parts. The first part is the reproductive or survival mechanism which copies the code and attaches it to target programs. The second part is the purpose or hidden intention of the virus. This second part has two subsections. The first is the trigger which activates or initiates the purpose. The trigger may be based on a date/time; the discovery of a particular program or account number; a count of the number of interrupts or calls; the extent of infection (e.g. the number of times the virus has copied itself); or any other discreet condition or combination of conditions which the author may conceive. The other subsection is the code which produces the result. This may take the form of a warning ("Got'cha!"); a cute picture on the screen; modified data; destruction of data (e.g. the erasure of a disk or tape); the activation of some other program; or any other action which code may normally accomplish.

Based on these building blocks, known viruses may be classified according to the following categories which are useful in detection/prevention strategies.

GCAM - General Contagion Agent Mechanism
SCAM - Specific Contagion Agent Mechanism

GTAR - General Target Action or Result
STAR - Specific Target Action or Result

The GCAM virus will attach itself to any program or executable file which promises to help perpetuate the existence of the virus. (Some "vaccines" search only certain types of files or the front of a program in ferreting out a virus [6] but the virus may be located at the end or anywhere in the middle.) Early experimenters were able to create functional GCAM mainframe viruses with under 100 bytes of code. Poorly designed GCAM viruses often draw attention to themselves by altering the program's overall length [7]; reinfecting the same program multiple times [8]; or altering the "date-of-last-modification" associated with the program. However, all of these clues can be easily camouflaged by more effective design. The date can often be restored to its former value after the infection has taken place. A subtle "signature" can identify infected programs to prevent reinfection. The length of the program can be maintained by condensing inefficient code, overwriting non-critical code, or by calling subroutines stored in other areas.

The SCAM virus seeks out specific categories (or one category) of an application or system program for its habitat. Therefore the spread of infection is much slower, but it is also much more difficult to detect. In certain types of programs, a well disguised virus may even be impossible to detect by any other means than a byte-by-byte comparison to a copy of the program which is known to be uncontaminated. This approach is certainly difficult if the program is dynamic and there are multiple active versions available. SCAM viruses often exploit known "bugs" or defects in particular programs. These defects can provide a point of entry, a camouflage mechanism, or access to powerful utilities. The author of a SCAM virus must, of course, be intimately familiar with the target program.

The GTAR virus has been the most common type reported by the news media. This is probably because it is easiest to code and the most dramatic or sensational in its result. When the "trigger" is initiated, a whole disk is reformatted or a tape overwritten. The GTAR virus seeks attention but is not too particular about its audience. It does not generally discriminate between target disk drives, target terminals, or target files. Any target which it can reach is good enough. Because the type of commands which it is likely to use can be generalized, the GTAR virus is more easily recognized and neutralized than the STAR virus.

The STAR virus is specific in its target and purpose. Therefore, it requires more sophistication and application-specific knowledge in designing its trigger. A STAR virus may attempt to initiate a financial instrument when an account number is located, or change coordinates in a military application when a particular weapons system is discovered. Even after the virus has executed its purpose, its presence may not be disclosed. Data may be changed and the virus may even self-destruct, leaving no traces. This type of virus does not result so much in denial of service (as with the GTAR virus) but in perversion of the purpose of the services.

These categories may result in the following combinations: GCAM-GTAR, GCAM-STAR, SCAM-GTAR, and SCAM-STAR. Each has direct application and impact on the detection and prevention tools which are used to counteract their effects.

III. Virus Prevention:

The only known method of completely securing a computer system against viruses is total isolation. That would remove the functionality of most business computers. However, good management and security practices can greatly reduce the risk and provide an effective defense against most viruses. The broad concepts presented here do not go into technical details but references are provided for further study. These concepts are more comparable to driving a car than to expounding on the mysteries of the internal combustion engine. These practices include:

- software "vaccines"
- encryption
- access control software
- "test-to-production" procedures
- back-up and recovery plans
- personnel selection and review controls

and

- physical access controls.

Vaccines:

The purpose of a virus filter is to recognize a particular type (or types) of virus to prevent it (or them) from entering into a system, and to prevent reproduction. Vaccines attempt to identify viruses which are already present in the system; to eliminate them, if possible; and to prevent reproduction. The means of accomplishing these ends are many and varied. A Swedish company called Secure Transmission AB has developed a set of programs called "T-Cell" which they say "several major Swedish manufacturer's have incorporated into their mainframe systems."[9] One of their obvious precautions was to try to protect the T-Cell programs themselves from becoming infected by strictly limiting access to those programs. Every data file and program in the system is protected by a "seal" which is difficult to reproduce. If the seal is altered, the system assumes an unauthorized intrusion. Standard "safe" copies of operating system software and utilities are maintained for comparison to operational versions and for recovery, if necessary. Many different strategies are employed by other vendors. None of these vaccines has been used long enough to measure their operational effectiveness. For maximum protection they should be incorporated with access control software such as ACF2.

The most useful MVS operating system virus detection tool known to the author is CA-Examine. This set of programs has compiled an extensive knowledge base of the internal structure of MVS in order to analyze key system libraries and memory resident modules. It is capable of detecting superzaps, corezaps, modreps, and other system modifications. Using expert system techniques, these programs conduct an analysis of operating system parameters to identify abnormalities and unauthorized changes. It provides interactive followup which allows the user to decide which problems deserve the most attention. CA-Examine is not an automatic panacea and is not of much use to the MVS-novice but can be very effective in detecting viruses in MVS in the hands of a knowledgeable user.

By their nature, filters and vaccines concentrate on GCAM and GTAR viruses. They are only useful against SCAM or STAR viruses after a specific virus is known and analyzed. This is similar to closing the barn door after the horses have escaped.

IBM, Corporation installed a "filter" to prevent viruses after an inadvertent virus brought down their international VNET network and left 100,000 users in 80 countries without telecommunications in December of 1987. The problem began when a law student in Germany (not a professional programmer) sent a friendly holiday message to an associate. The greeting also included a command to copy the recipient's mail distribution list and send the same message to everyone on the list. Each new recipient's mail distribution list was copied and the message multiplied. Within hours VNET collapsed under the weight of too many messages. [10]

Encryption:

Encryption appears to be a virus' worst enemy. However, the care and feeding of the encryption package can be expensive. This technique, for the present, is usually reserved for highly critical systems. Pozzo and Gray [11] have contributed the most extensive discussion of the tradeoffs of encryption protection against viruses to date. They argue that the most effective plan is to encrypt all executables within a system. (An "executable" is any software which can be understood and initiated by the operating system or other software.) If the author of the virus cannot get to the executable before it is encrypted or cannot break the encryption algorithm, the virus can never be executed. Even if the virus does get to a particular executable before it is encrypted, the virus will not be able to spread to another executable, and a virus which does not reproduce is not a virus (which does not mean that is is not irritating).

A less expensive approach is to append an encrypted signature block to the plain-text executable ("plain-text" to the programmer who understands the language.) The signature block would be the result of applying a strong one-way encryption function (e.g. a cryptographic checksum) to the entire executable plus a password from the authorizer and a time stamp. The entire signature block would then be encrypted. Before execution, a public key would be used to decipher the block and the checksum would be recomputed. This process is trivial compared to encrypting the entire executable but still expensive enough to require careful analysis before the investment is made. A third and still weaker approach is to use encrypted signature blocks for only critical or highly used programs. Pozzo and Gray also examine the tradeoffs between public key and private key designs.

Access Control Software:

The most effective general purpose virus prevention technique is access control software such as ACF2, RACF, or Top Secret. (In the author's experience, ACF2 is the most cost-effective of these three.) There are many other reasons for installing access control software, but virus prevention is a fortuitous side effect. Because this software restricts access to critical files (programs), modifications cannot be made unless the virus has already attached itself to a program with high level access. This points out the necessity of constructing useful rule sets across the system. Access control software will not protect against poor rule-writing nor against users with high level privileges such as the security administrator, system auditor, or system programmers. These users must be competent and trusted in order to preserve the integrity of the system.

Test-to-production Controls:

"Test-to-production" software control procedures limit the introduction of new software to the operating environment and restrict access to high level programs where viruses are likely to be effective. These procedures include software testing, verification, quality assurance review, and written authorization before implementation. Usually production programs are only executed from controlled libraries and those libraries are carefully restricted. Providing a separate but controlled development environment keeps viruses out of the production system. These procedures apply to software modifications and enhancements as well as to completely new software. The ability to track and keep a historical record of software introduction is critical to the integrity of the whole system. The purpose and nature of any changes should be recorded along with the authorization. Software purchased from outside vendors should go through the testing procedure just like the internally developed software. Contracts with software vendors should include provision for liability in the case any undetected virus is latent in the vendor's product. No modification or enhancement should ever by made directly to base code. This rule may delay an urgent "fix" but it will save even greater delays from "fixes" which don't work or contain unauthorized side effects. High- privilege system users should check each other's work. Systems programmers' work should be randomly (or constantly) reviewed by the Auditors and Computer Security Personnel.

Back-up and Recovery:

If a virus is discovered and there is no reliable method of deleting it, or if a file has been erased by the virus, often the only solution is to reload a clean copy of the original program or data. For most large data processing centers, monthly, weekly, daily, or in some cases even hourly backups are standard procedure. There are many sound reasons for backing up software and data; protection against viruses is a secondary benefit. However, it is also necessary; without backup, recovery from a virus attack might be impossible.

The backup copies also provides a standard to compare against operational versions in the case that a modification is suspected but not proved. Of course, the system must be virus-free at the time of the recovery or the backup copy will also become infected. The process of insuring that the system is virus-free may cost time and effort, but the alternative is even more costly. Also if it is not known how long the virus has been in the system, a particular backup copy may already be infected. In that case, an earlier generation must be recalled.

Personnel Selection and Review:

Some people create viruses. Other people prevent or reduce the risk of viruses. Untrusted data processing personnel can be a system's worst enemy. Viruses have been possible for over twenty years. The U.S. Military has been in contact with live viruses for over ten years [12], but only recently have viruses become a significant problem to the general public. The "popularity" (as well as the population) of viruses seems to be closely correlated to the negative feelings of certain individuals against computer owners or users. Careful screening of employee candidates can help to prevent many problems before they develop. Many difficulties can be prevented by clearly defined termination procedures, including cancellation of computer privileges before other termination processing begins.

Physical Access Controls:
Physical access controls are an effective virus prevention technique to the extent that "safe" users can be identified and allowed to complete assigned work. Perimeter access controls reduce the number of potential virus perpetrators. Keeping computer terminals in locked or closely supervised rooms has proved to be a cost-effective technique for many companies. Physical access controls applies to input devices (terminals, key boards, and disk/tape drives) and to the data itself (which may be stored on disks, tapes, or other media.) Shredding unneeded output and systems manuals can also reduce exposure. Many virus authors have begun their efforts by pulling a discarded computer manual out of a trash bin. A corporation's trash may be a virus author's treasure.

IV. CONCLUSION

These protection efforts (vaccines/filters, encryption, access control software, test-to-production controls, backup and recovery plans, personnel selection and review, and physical access controls) are not independent measures but part of an interlocking protection chain. Some are more important than others in some situations, but none of them should be ignored or eliminated from consideration without evaluating the risks. By way of analogy, underwater mines might be a necessary safeguard against a potential naval attack, but that does not eliminate the need for radar against an unanticipated air attack. Virus perpetrators are looking for the easiest entry point to a system. The protection chain is only as strong as its weakest link.

The virus problem was identified in 1983: "Virus attacks are easy to develop in a very short time, can be designed to leave few if any traces in most current systems, are effective against the average modern security policies for multi-level usage, and require only a minimal effort to implement."[2] The answer to the problem is better than "average" (in historical terms) security controls. Basic management and security policies can greatly reduce the exposure and provide cost-effective protection for corporations and government agencies who take the effort to implement them.

REFERENCES

[1] ___, *'Computer Virus Incidents'*, **Edpacs** (February, 1988),p.9.

[2] Cohen, F., *'Computer Viruses'*, **Proceedings of the 7th DOD/NBS Computer Security Conference** (1984), pp.240-263.

[3] Ali, R., *'Electronic Viruses and Vaccines, a History'*, **The Arab News** (August 2, 1988), p.5.

[4] Goodwins, R., *'Understanding the Minds Behind Viruses'*, *Computer Fraud and Security Bulletin' (July, 1988), pp.2-4.*

[5] Ponting, B., *'Some Common Sense About Viruses'*, *Data Communications* (April, 1988), pp. 60-62.

[6] Menkus, B., *'No Vaccine Wards Off Effects of Virus Attacks'*, **Computerworld** (July 11, 1988), p.S8.

[7] Keefe, P. *'Checkpoints Against Viruses'*, **Computerworld** (September 13, 1988), p.54.

[8] Elmer-DeWitt, P. et al., *'Invasion of the Data Snatchers!'*, **Time** (September 26, 1988), pp.40-45.

[9] Ernsberger, R. et al., *'Vaccines for Computers'*, **Newsweek** (May 16, 1988), p.3.

[10] Marback, W. et al., *'Infecting Computers'*, **Newsweek** (February 1, 1988), p.50.

[11] Pozzo, M. and Gray, T., *'Computer Virus Containment in Untrusted Computing Environments'*, **IFIP/SEC** Fourth International Conference on Computer Security (December 1986), pp.321-331.

[12] Meeks, B., *'Will Viruses Go Away?'*, **Microbytes** (February 26, 1988).

BACKGROUND REFERENCES

PERIODICALS:

AIFIPS/ACM:

AIFIPS, 1979.

ACM, "*Use of Virus Functions to Provide a Virtual APL Interpreter Under User Control*", 1974.

ACM, "*The Worms' Programs — Early Experience with a Distributed Computation*", 1982.

Garey M. R. and D. S. Johnson. "*Computers and Intractability*". Freman, 1970.

Gold, B. D., R. R. Linde. R. J. Peeler, M. Schaefer, J. F. Scheid, and P. D. Ward; "*A Security Retrofit of VM/370*"; in National Computer Conference, pages 335-344. **AIFIPS**, 1978.

Harrison, M. A., W. L. Ruzzo, and J. D. Uilman. "*Protection in Operating System*". In Proceedings. **ACM**, 1976.

Hoffman, I. J., "*Impacts of information system vulnerabilities on society*"; in National Computer Conference, pages 461-467. **AIFIPS**, 1982.

Klein, M. H., ed.; "*DOD Trusted Computer System Evaluation Criteria*"; 1983, CSC-STD-001-83.

Linde, R. R., "*Operating System Penetration*". In National Computer Conference, pages 361-368. **AIFIPS**, 1979.

McCauley and P. J. Drongowski. Ksos. "*The Design of a Secure Operating System*". In National Computer Conference, pages 345-353. **AIFIPS**, 1979.

U.S. Dept. of Justice, Bureau of Justice Statistics. **Computer Crime Computer Security Techniques.** U.S. Government Printing Office, Washington, DC, 1982.

BIX (Byte Information Exchange), Microbytes

Loeb, L. "*Another MAC Virus; This One Shows Up At Apple Too*"; April 7, 1988.

Meeks, B.: "*Will Viruses Go Away? Or Wreak Havoc On Software Industry?*", **Microbytes;** February 26, 1988.

BUSINESS WEEK

Hafner, K. with Lewis, G.; Kelly, K.; Shao, M.; Hawkins, C.; and Angiolillo, P.; "*Is Your Computer Secure?*"; **Business Week**; August 1, 1988; pp. 50-56.

Port, O.; "*How Uncle Sam's Cloak-And-Data Boys Are Fighting Back*"; **Business Week;** August 1, 1988; pg. 58.

Schares, G.; **Business Week;** August 1, 1988; pg. 55.

CSA NOTES:

Peterson, D. ed.; "*Worldwide Virus Invasion*" **CSA Notes;** October, 1988; pg. 1.

Peterson, D. ed.; "*Viral Attacks*" **CSA Notes;** February, 1988; pg. 2.

COMPUTERS & SECURITY:

Cohen, F.; "*On the Implications of Computer Viruses and Methods of Defense*"; **Computers & Security;** Vol. 7:2; (April 1988) pp. 167-184.

"*Models of Practical Defenses Against Computer Viruses*"; **Computers & Security;** Vol. 8:2 (April 1989); pp. 149-160.

Fak, Viivek; "*Are We Vulnerable to a Virus Attack? A Report from Sweden*; **Computers & Security;** Vol. 7:2 (April 1988); pp. 151-155.

Gleissner, W; "*A Mathematical Theory for the Spread of Computer Viruses*"; **Computers & Security**; Vol. 8:1 (Feb. 1989); pp. 35-42.

Herschberg, I. S.; "*Make the Tigers Hunt for You*"; **Computers & Security;** Vol. 7:2 (April 1988); pp. 197-204.

Highland, H. S., "*Computer Virus — Post Mortem*"; **Computers & Security;** Vol. 7:2 (April 1988); pp. 117-128.

"*Anatomy of a Virus Attack*"; **Computers & Security;** Vol. 7:2; pp. 145-150.

"*An Overview of 18 Virus Protection Products*; **Computers & Security;** Vol. 7:2; (April 1988); pp. 159-164.

"*How to Combat a Computer Virus*"; **Computers & Security;** Vol. 7:2; (April 1988); pp. 157-158.

"*The Italian Virus, et. al.*"; **Computers & Security;** Vol. 8:2 (April 1989); pp. 91-100.

"*Virus Victims*"; **Computers & Security;** Vol. 8:2 (April 1989); pp. 101-103.

"*A Macro Virus*"; **Computers & Security;** Vol. 8:3 (May 1989); pp. 178-182.

Murray, W. H.; "*The Application of Epidemiology to Computer Viruses*"; **Computers & Security;** Vol. 7:2 (April 1988); pp. 145-150.

Radai, Y.; "*The Israeli PC Virus*"; **Computers & Security;** Vol. 8:2 (April 1989); pp. 111-114.

Straub, D. W.; "*Organizational Structuring of Computer Security*"; **Computers & Security;** Vol. 7:2 (April 1988); pp. 115-120.

Webster, A. E.; "*University of Delaware and the Pakistani Virus*"; **Computers & Security;** Vol. 8:2 (April 1989); pp. 103-106.
Wyk, K. R.; "*The Lehigh Virus*"; **Computers & Security;** Vol. 8:2 (April 1989); pp. 107-110.
Zajac, B. P.; "*Legal Options to Computer Viruses*"; **Computers & Security;** Vol. 8:1 (Feb. 1989); pp. 25-28.

COMPUTERWORLD:

Bloombecker, B.; "*Careful Now — It's Catching*"; **Computerworld;** May 9, 1988; pg. 21.
Hall, C.; "*Virus Scare Infects Clone Sales*"; **Computerworld;**
Menkus, B.; "*No Vaccine to Ward Off Effects Of Virus Attacks*"; **Computerworld;** July 11, 1988.
Ryan, A. J.; "*Firm Finds Antidote for PC Virus*"; **Computerworld;** March 28, 1988; pp. 35, 43.
Ryan, A. J.; "*Viruses Infect Corporate MIS*"; **Computerworld;** May 30, 1988; pg. 29.
Scisco, P.; "*No Such Thing As A Small Mishap*"; **Computerworld;** July 11, 1988; pp. S1-S7.

DAILY BREEZE:

———; "*Virus' Multiples in Computers*"; **Daily Breeze;** January 25, 1988; Business Section; pg. 1.
Lev, M.; "*Computer Viruses*"; **Daily Breeze;** March 23, 1988; pg. C-1.

INFORMATION WEEK:

Iida, J. B. "*Computer Crime: MIS Confronts A Cancer*"; **Information Week;** February 22, 1988; pp. 24-25.
Schindler, P. E.; "*Computer Viruses Spread*"; **Information Week;** February 22, 1988; pp. 16-17.

INFO WORLD:

Arnett, N.; "*Mac Virus Surfaces in Washington*"; **Info World;** April 11, 1988; pp. 1, 3.
Buerger, D. J.; "*Detecting And Combating Computer Viral Infections*"; **Info World;** March 21, 1988; pg. 14.
Johnston, S.; "*Computer Virus Spreads To Commercial Software*"; **Info World;** March 21, 1988; pg. 85.
Mace, S.: "*New Virus Damaging MacData Group Says*"; **Info World;** April 11, 1988; pg. 29.
Warner, E. and Mace, S.; "*Vaccines' Offered To Cure Viruses*"; **Info World;** March 21, 1988.

INSIGHT:

Shear, J.; "*Nightmare Of Digital Deviltry May Not Be Worth Lost Sleep*"; **Insight;** April 18, 1988; pp. 40-41.

LOS ANGELES TIMES:

Finke, N.; "*Does Your Computer Have A Virus?*"; **Los Angeles Times;** January 31, 1988; Part VI; pp. 1, 7-8.

RISKS-FORUM DIGEST (Forum on risks to the public in computers and related systems, ACM Committee on Computers and Public Policy. Peter G. Neumann, moderator (SRI):
Volume 6: Issue 22; February 8, 1988.
Spector, David H. M.; "*Macintosh Virus Hits CompuServe*".
Volume 6: Issue 25; February 25, 1988.
Baker, Bruce N.; "*Yet Another Virus — Brain Virus*".
McLellan, Vin; "*Virus Code and Infected Definitions*".
Radai, Y; "*Another PC Virus*", From Info-IBM PC Digest.
Volume 6: Issue 49; March 27, 1988.
Brunnstein, Klaus; "*Nightmare Virus Construction Set*"; "CCC Hackers Revenge Threat".
Gross, Steve; "*Computer Virus' Creating Entreprenaurial Opportunity*", From Minneapolis Start Tribune — Business Section.
McLellan, Vin; "*Virtous Virus Language*".
Minow, Martin; "*Arari ST Virus*".
Volume 6: Issue 55, April 5, 1988.

TIME:

Elmer-Dewitt, P. with Brown, S. and McCarroll, T.; "*Invasion Of The Data Snatchers*"; **Time;** September 26, 1988; pp. 40-45.
Elmer-DeWitt; P. with Munro, R.; "*You Must Be Punished*"; **Time;** September 26, 1988; pg. 44.

REFERENCE MANUALS/BOOKS:

Anderson, J. P.; **Computer Security Technology Planning Study,** USAF Electronic Systems Division, Bedford, MA., Oct. 1972, ESD-TR-73-51.
Bailey, N.T.; **The Mathematical Theory of Epidemics,** Hafner Publishing Co., NY, 1957.
Baker, D. B.; **Department of Defense Trusted Computer System Evaluation Criteria** (First Draft) private communication; The Aerospace Corporation, 1983.
Bell, D. E., and LaPadula, LJ.; **Secure Computer System: Unified Exposition and Multics Interpretation,** MITRE Technical Report, MTR-2997, July 1975.

Biba, K. J.; **Integrity Considerations for Secure Computer Systems,** MITRE Technical Report, MTR-3153, June 1975.

Boebert, W. E., and Ferguson, C. T.; **A Partial Solution to the Discretionary Trojan Horse Problem,** Honeywell Secure Technology Center, Minneapolis, MN.

Bourne, S. E., **The UNIX Systems.** International Computer Science Series. Addison-Wesley Publishing Company, 1983.

Denning, D. E.; **Cryptography and Data Security,** Addison-Wesley Publishing Co., Reading, Ma. 1982.

Diffie, W., and Hellman, M. E.; **Privacy and Authentication: An Introduction to Cryptography,** Proceedings of the IEEE, Vol. 67, No. 3, March 1979.

DOD Computer Security Center; **Department of Defense Trusted Computer System Evaluation Criteria: DOD,** CSC-STD-001-83, 1983.

Fenton, J. S.; **Information Protection Systems,** PhD thesis, U. of Cambridge, 1973, cited in Denning.

Kahn, D.; **The Codebreakers,** MacMillan, New York, 1972.

Klein, H. D.; **Department of Defense Trusted Computer System Evaluation Criteria;** Department of Defense, Fort Meade, Md. 20755, 1983.

Kline, C. S., Popek, G. J., Thiel, G., and Walker, B. J.; **Digital Signatures: Principles and Implementations;** Journal of Tele-Communication networks, Vol. 2, No. 1, 1983, pp. 61-81.

Lampson, B. W.; "*A Note on the Confinement Problem*"; **Communications of the ACM 16** (10): 613-615.

Landberg, Ted; "*Computer Viruses and Trojan Horses, A Guide to Protecting Your Computer*"; **National Bureau of Standards Microcomputer Electronic Information Exchange;** March 8, 1988.

Landreth, B.; **Out of the Inner Circle: A Hacker's Guide to Computer Security;** Microsoft Press, Bellevue, WA., 1985.

Meyer, C. H. and S. M. Matyas; **Crytography: A New Dimension in Computer Data Security;** John Wiley & sons, 1976.

Nimmer, Raymond T.; "*Computer Crime*"; **The Law of Computer Technology;** Warren, Gorham & Lamont, Boston: 1985, section 9-1 -9-29.

Perry, William E.; **A Standard for Auditing Computer Applications;** Selected Audit Areas; Auerbach Publishers, Inc., N. J.: 1986, pp. 184-240.

Popek, G. J., M. Kampe, C. S. Kline, A Stoughton, M. Urban, and E. J. Walton, "*UCLA Secure Unix*"; in National Computer Conference.

Pozzo M. M. and T. E. Gray; "*Computer Virus Containment in Untrusted Computing Environments*", IFIP/SEC Fourth International Conference and Exhibit on Computer Security, December 1986.

Pozzo, M. M. and T. E. Gray; "*Managing Exposure to Potentially Malicious Program*"; Proceedings of the 9th National Computer Security Conference, Sept. 1986.

Roberts, R. S.; "*Computer Virus*"; **COMPUTE!** Books; Radnor, Pennsylvania, 1988.

Shannon, C. E.: "*Communication Theory of Secrecy Systems*", Bell System Technical Journal, 28, 1949, pp. 656-715.

Shoch, J. F., and J. A. Jupp; "*The 'Worm' Programs — Early Experience with a Distributed Computation; Communications of ACM 25, 3 (March 1982), 172-180.*

Walker, B. J., G. J. Popek R. English, C. S. Kline, C. S., and G. Thiel; "*The Locus Distributed Operating System*", Proceedings of the Ninth ACM Symposium on Operating System Principles, October 1983.