Viruses and Worms--What Can You Do?

© Stanley A. Kurzban 1989
International Business Machines Corporation
Systems Research Educational Center
500 Columbus Avenue (798/1)
Thornwood, New York 10594

ABSTRACT

"Computer viruses" have received a lot of attention lately. In fact, the best-known "viruses" have not been viruses at all, but "worms," programs that spread through networks instead of modifying programs. Both viruses and worms reproduce themselves and defensive measures have focused on stopping or slowing their spread. But that is only one tack you can take. Preventing or limiting the effect of the harm that their components can cause is another. Still other measures are specific to known viruses. Ultimately, though, there is no defense better than a comprehensive security strategy that embraces user education, crisis-response teams, and technologically sound security measures including, but not limited to, those that relate specifically to the threats posed by viruses and worms.

Viruses and Worms--What Can You Do?

"Computer viruses" have attracted a great deal of attention from the media. Now, for the first time, corporate executives, as well as teenagers and retired laborers, are hearing and reading about computer security the way they hear and read about inflation, politics, and war. For those of us who believe that the subject is important, this is an opportunity to convince others that we are right. The spotlight is on us. But the opportunity, like all others, comes with a responsibility as well. We must be accurate, we must be prudent, and we must be effective.

Accuracy demands that we choose our words carefully. For that reason, our first sections deal with precise definitions of the terms we use and actual examples of the types of attacks that opponents may wage against computers.

Prudence demands that we consider carefully what protective measures are available to us, their costs and benefits, and our recommendations in the context of the environment as we see it.

Effectiveness can only derive from clear summation of the situation today, what we can reasonably foresee, and the issues that apply to both.

Definitions and Examples

While the media have been devoting a lot of their attention to "computer viruses," they have been giving different and, most often, no definitions for the term. In fact, authoritative definitions leave the most widely covered harm-doing "computer viruses" out altogether. The program (11, 33, 35) that disabled the Internet and ARPANET in early November of 1988 and the EXEC (22) that degraded service on several networks before Christmas of 1987 are worms, not computer viruses. Contrary to the Associated Press story of September 20, 1988, the code that Texan Donald Gene Burleson was convicted of using to harm a former employer's data in 1987 is a logic bomb, not a virus.

Definitions

A Trojan horse (1) is harmful code concealed within an attractive program.

A logic bomb is code that does widespread and sudden damage to data.

A time bomb (29) is a logic bomb that some event triggers.

A worm (34) is a program that plants copies of itself in remote, electronically connected nodes.

A virus (6) is code that plants a version of itself in any program it can modify.

A worm or virus may contain something else, for example, a logic bomb. Since a logic bomb is presumably something that a person would not wish to execute, it is likely to be concealed, that is, a Trojan horse.

Note that nothing in the definition of "virus" or "worm" necessarily connotes harm. That fact contributes to the difficulty of defense against the harm that either may do. Either may be beneficial. In fact, the concept of a worm was introduced (34) in the context of a useful application. (The same may not be said about a virus, although (6) implies so. Its reference to (13) is incorrect as to the latter's date and place of publication; its definition of the term, "virus"; and its main thrust.) In trying to defend yourself against the harm that either may do, you must not deprive yourself of even greater benefit that a program with similar, but benevolent, behavior might provide. That means that you must find, rather than eradicate, such programs and then judge subjectively, on a case-by-case basis, whether to destroy what you have found.

Note that the difference between a virus and a worm is that the former requires a program in which to reside, while the latter does not. The distinction parallels the one in the biological analog, where a virus invades a cell and forces the cell to make copies of the virus in other cells, while a worm need not have a host to invade and can exist by itself without a larger, encompassing organism. (Some (10) have suggested the term "bacterium" to replace "worm"; the biological analogy seems better, but the term "worm" was already well-established before the term "virus" gained currency, so it is not likely to be superseded.) The cell in which a virus resides is called its "host." In this sense, a virus requires a host that is a program; a worm does not. Because, however, a worm travels among what we in data processing call "hosts," the potential for confusion is great. Both viruses and worms copy themselves and are potential carriers of code that can do great harm.

Examples

Examples help to explain the implications of raw definitions. We chose the examples below to illustrate the points that are most important to the subject of defenses against harm-doing self-replicating programs.

Our appendices contain more detail on several viruses and the Internet worm, used below as an example.

Typical virus

A virus copies itself, sometimes imprecisely. (Imprecise copying is the analog of biological evolution.) Because we are concerned here with defense, we shall assume that it also does harm; let us say that it contains a time bomb. We can represent the whole of the virus in pigeon code, then, as follows:

```
VIRUS-IDENTIFIER
SEARCH FOR A MODIFIABLE PROGRAM
IF ONE IS FOUND,
  TEST FOR THE VIRUS-IDENTIFIER
  IF IT IS FOUND IN THE PROGRAM, LOOK FOR ANOTHER
    ELSE
      COPY THIS CODE INTO IT SUCH THAT EXECUTION OF
        THE PROGRAM WILL BEGIN WITH THIS CODE'S "SEARCH"
      LOOK FOR ANOTHER
  ELSE CONTINUE
TEST FOR TRIGGER
IF ON, EXPLODE TIME BOMB
  ELSE CONTINUE
GO TO WHAT WAS THE PROGRAM'S PREINFECTION ENTRY POINT.
```

Note the salient points:

1. To spread, a virus must execute under conditions that permit it to copy its own code and to modify one or more programs other than the one in which it resides.
2. When it spreads, it modifies programs.

Thus, it follows that you can prevent a virus from spreading by preventing it from copying itself or preventing it from modifying any program in which it does not reside. Also, it follows that you can detect its spread by noticing that a program has been modified. Finally, if you can neither prevent nor detect its spread, you can (try to) acquire no virus in the first place or deal after the fact with the harm that is done by the code in the virus.

Viruses can spread very quickly and very widely. Consider, for example, the role that a program almost everybody uses, like an editor, can play. A virus may spread slowly until someone who can modify the editor executes it. Then, the virus infects the editor and then it infects every program that can be modified by anyone who uses the editor. Since every program is likely to be susceptible to modification by some user of the editor, infection soon becomes universal.

CHRISTMAs EXEC

The CHRISTMAs EXEC is a REXX EXEC for IBM's VM/370 family of operating systems that offered the recipient a Christmas greeting, but also sent copies of itself to all those with whom it could detect that the recipient routinely or recently communicated. We can represent it in pigeon code as follows:

```
COMMENT: "RECEIVE ME ONTO YOUR MINIDISK AND EXECUTE ME BY
  KEYING IN 'CHRISTMA' DON'T BOTHER LOOKING AT MY CODE IN
  YOUR READER"
DISPLAY: (a drawing of a Christmas tree with a greeting)
READ user_identifier NETLOG
EXTRACT NAMES OF ALL OTHERS ON NETWORK WITH WHOM THIS USER
  HAS COMMUNICATED SINCE LAST PURGE OF NETLOG
READ user_identifier NAMES
EXTRACT NAMES OF ALL THOSE FOR WHOM THIS USER HAS NICKNAMES
SEND A COPY OF THIS TO EVERY USER WHOSE NAME HAS BEEN EXTRACTED
```

Note that the effect of CHRISTMA is to send many copies of itself to many people very quickly. Because each copy will come to a user from someone with whom that user has regularly or recently communicated, the recipient is quite likely to do as the "COMMENT" suggests without suspecting anything untoward.

Internet worm

The Internet worm did harm because it busied systems that it invaded. (11, 33, 35) In our context, however, what is interesting is not so much the many things that it did in each computer as the ways that it propagated among systems. The worm's first propagational step is devising the addresses of other systems it might invade. It does this in three different ways: (33)

1. finding addresses in a system table.
2. Finding addresses in a program.
3. Randomly generating addresses.

It tries to enter the system at each address in three different ways: (11)

1. Through a feature in a program that receives mail; the feature permits immediate execution, under some well-documented circumstances, of code contained in the incoming mail. (Note that those who were known to use the mail-receiving program had been advised to disable the feature.)
2. Through a fairly widely known bug in a data-transfer program; the bug caused data that overflowed a buffer to be executed as code.
3. Through accounts whose user identifiers and passwords the worm could "guess," using a table it contained, or find in a system that it had previously penetrated.

Note that all propagational means the worm used could have been foreclosed by standard security practices, had they been employed:

1. Disabling a debugging feature that was known to create an integrity exposure.
2. Applying a fix for a well-known bug.
3. Using passwords that are difficult for others to guess.
4. Concealing even encrypted passwords and logging repeated entry failures.

Defenses

Defense against harm can consist of preventing the harm from occurring, limiting the extent of the harm, or recovering from the harm after it has occurred. Defenses of all three types are included below.

Observations that Arise from the Definitions

What distinguishes worms and viruses from other things that may cause harm is that both may spread rapidly, thwarting any attempt to identify their origin and carrying some form of harm-doing code with them. This means that our defensive plan must include the ability to react more rapidly and on a larger scale than we ever have before. What distinguishes worms and viruses from each other (11, 33, 35) is the way they spread. We must think of communication lines and system entry points when we try to prevent or detect the spread of worms. We must think of programs when we think about detecting or preventing the spread of viruses.

Where we Stand

As of the end of 1988, viruses and worms had generated far more publicity than actual damage. Especially in the case of the Internet worm, however, there is no question that the publicity itself was damage of a sort. Murray (24) is among those who have observed that much greater damage would result if the threat of self-replicating harm-doing programs were to inhibit exploitation of data processing. If use of PCs were to be greatly curtailed for fear of viral infection or networks shut down for fear of worms, great resources would be lost to the entire data processing community. In that regard, ″where we stand″ may be at the edge of a precipice, needing only the push of a few more widely publicized incidents to plunge us into the abyss.

The good news is that such a reaction would hardly be justified by what we have witnessed to date. There is some evidence that people victimized by viruses have recovered most of what they lost, (9) albeit with considerable discomfort, because they had created back-up copies of their most valuable data. They did so not because they had foreknowledge of viral attacks, but because back-up copies are prudent protective devices for a wide variety of reasons. People victimized by worms were able to restore their networks to sound working order relatively quickly because they had procedures in place for dealing with disruptions of their networks. Again, no foreknowledge was involved; the procedures were justified by different considerations that had similar implications.

Attacks we should anticipate

Realistic planning of our defense requires an understanding of what threats we are likely to face. Certainly, we know that those who wish to steal from us need not (yet) go to the trouble of loosing a virus or worm on us. (12) There are simply too many easier ways. (14) Viruses and worms are far better agents of malice than of greed. Their rapid dispersion can do a great deal of damage very quickly.

The worms we′ve seen so far (11, 22, 33, 35) seem to have done most of their harm because of the carelessness, rather than malice, of their creators. Simple bugs turned an experiment or a prank into a cause celebre.

Accordingly, we know to fear people who are malicious and/or careless. What little we do know of perpetrators to date suggests that they have been bright young people. The correlation of ″bright and young″ with ″malicious and careless″ is sociologists′ concern, not ours.

We know to fear greater sophistication in attacks. CHRISTMA was easily stifled because it never evolved, never even changed its name. We cannot expect to be so lucky again, especially if the factor of malice, absent from CHRISTMA, should be added. Viruses are becoming more complicated as time goes by. We must expect that trend to continue. It bespeaks another instance of the ″evolution″ metaphor: survival of the nastiest, if you will.

Viruses: Because a virus modifies a program, only a programmer can create one (without a do-it-yourself virus-construction kit (of which, unfortunately, one has been reported). Because malicious people who write programs for large systems can find easier ways to accomplish their ends than creating viruses, viral attacks of the future are likely to take place exactly where all the previous ones have: on microprocessors. (12) Malefactors can afford to buy them and do all the necessary testing in the privacy of their own homes or dorm rooms. The problem of creating a virus is still sufficiently novel and difficult to appeal to the mischievous. They will improve on past viruses, but they will continue to attack the most popular systems and the most sensitive areas: hard disks and system data areas.

Worms: Creating a worm is sufficiently difficult that, the Internet experience notwithstanding, we can expect perpetrators to favor easier-to-use languages, like the REXX of the CHRISTMA EXEC. While some, like the Internet worm, may principally affect the systems on the network they inhabit, most will probably do their damage by sheer proliferation on the networks' interconnections, as CHRISTMA did. Worms that evolve will be harder to recognize and rout out of our systems. New dispersion techniques will undoubtedly replace the ones we have already seen, so to combat them, we shall have to focus on the general problem of dispersion rather than the particular of some avenue.

Defenses we are employing

The limited harm done by the worms and viruses we have encountered gives evidence that we must be doing some things right. Mostly, we are employing some of the defenses that auditors and consultants have been recommending for years to limit general security exposures. (23) We have been controlling access to critical resources and monitoring for unexpected or excessive use of some of our resources. We have been creating back-up copies of sensitive data and restoring data from those copies when the originals suffered damage. But the magnitude of the threat posed by self-reproducing harm-doing programs implies that we shall have to do much more.

Viruses: Most users of large computers today control "WRITE" access to programs. That control limits the speed, if not the breadth, of a virus's spread. The same cannot be said for smaller systems, where, not coincidentally perhaps, all viruses have appeared. Vendors of software test their products, yet some viruses have been disseminated by vendors whose good reputations are unquestioned. Many users of computers are discriminating in their choice of sources for software, yet many more are not. With little hard evidence, some have alleged that one virus was disseminated exclusively through the medium of illegally distributed software. (16) If the allegation is true, it speaks ill of the community's care in regard to software acquisition.

Worms: Some of us have been sufficiently concerned about connections between networks that we are placing flexible code at those connection points (2) to allow us to react to threatening situations that may arise. Some of our networks are configured much like the spoke-and-hubs arrangement of United States airports, as perceived and served by air carriers, so that the hubs provide points at which we can implement defenses against worms. Both of the defenses to which the preceding sentences alluded played a role in the International Business Machine Corporation's rapid response to the CHRISTMA EXEC when it entered IBM's VNET from an academic network.

Windows of vulnerability

Nonetheless, much vulnerability clearly remains. Back-up is still regarded as an exceptional, not a routine, step. Few if any users of computers can say that they strictly adhere to the principle of least privilege: (4) People should be authorized to do all and only what they must do to do their jobs. In all, the long-known methods we could be using to protect ourselves from the Trojan horses that worms and viruses may carry are unarguably underused.

Viruses: Whatever we do to control the modification of programs impedes the spread of viruses. Auditors have long recommended that the right to modify a program be restricted to those whose jobs are program modification. (23) That is merely an instance of "least privilege." (4) But control over the development of programs is usually a far cry from what texts (23) recommend. Where the programming staff is small and well-defined, adherence to that principle might so retard the spread of a virus that it could be detected and eliminated before it does significant damage. Actual viruses, however, have appeared only on computers considered "personal." On such computers, the need for control of program modification has been unobvious to most people. (20) They reason that a "personal" computer is the province of a single person and, since that person presumably has no desire to harm him- or herself, control facilities are unneeded. The flaw in that reasoning arises because "personal" computers have long ago ceased to be "personal" in that sense, if they ever were. As soon as people use programs written by others on their "personal" computers, other people are involved (because those programs may contain viruses or Trojan horses of any nature).

The same applies to connection of the "personal" computer to other computers. When data, which may be executable, enter "personal" computers, the people who created the data are intruding into that "personal" realm.

We are not totally unconcerned about such intrusions today. An acquired program that merely malfunctions can be as great a threat to us as any virus. Accordingly, we have taken steps to encourage prudence in the acquisition of software for "personal" computers. These steps are among today's defenses against computer viruses. They may be the only defenses of any value that many of us employ today.

Worms: The novel threat represented by worms derives from their rapid proliferation within and across networks. Even before we recognized the nature of this particular threat, we were concerned about overloading of a network. Whatever we did to try to recognize the approach of such a condition and deal with it acted as a defense against the direct harm that worms can do. Monitoring network activity and being able to isolate and deactivate parts of a network are among such defenses.

Because we have recognized the requirement for management of networks, we have organized groups of people to administer them. Because the networks are so useful, however, we have tended to rely on them for communications among administrators. (11, 33, 35) A worm like the CHRISTMA EXEC can disable the network and render the administrators incommunicado. (5, 26) This is a fate we must avoid. (27)

If we assume that people who are not authorized to use our networks are more likely to inflict worms on us than authorized users, then we can conclude that all the measures we take to exclude unauthorized users from our networks are also defenses against worms. Since experience to date offers no evidence to support that assumption, we do not treat it further here.

Available Defenses

In defending ourselves against viruses and worms, we can employ both defensive measures that address specifically the way that those things reproduce themselves and measures that address the harm-doing code that they can carry within them.

Defenses against viruses

You can protect yourself specifically against a viral attack by preventing viruses from entering your system, detecting them and eliminating them after they have entered, and employing measures that are specific to viruses that are known. (21, 38) Prevention of entry involves not only technological measures but also procedural ones. The latter are the first discussed below.

Procedural/educational defenses: Procedural defenses against viruses derive from accumulated experience with viruses to date. While exceptions to the "conventional wisdom" exist and may increase in incidence, it is nonetheless a guide to prudent behavior.

Murray (24) states that most known viruses have entered establishments via software "of dubious pedigree" for personal computers. It is no more than good common sense to purchase a potentially harmful item from a vendor whom you have sufficient reason to trust. (25) While software was known to be "potentially harmful," because of bugs it might have, before there were viruses, the fact is far more obvious now. We therefore consider "specific to viruses" the defense that consists of acquiring and encouraging one's employees and co-workers to acquire software for personal computers only from reputable individuals by conventional and contractual means. The procedure is meaningless without education. You cannot expect those who acquire software for personal computers to acquire it prudently unless you educate them as to what constitutes prudence and why it is necessary.

Education applies as well to early warning signs of infection. A virus must modify programs on diskettes to spread rapidly in personal computers. Many known computer viruses do other unexpected things, for example:

- Cause the display of error messages that users should not expect.
- Degrade the system's performance.
- Write to disk drives at unexpected times.
- Use up storage space on some medium.

Users warned to look for such things and report them to administrators, such as those who work at "HELP" desks, may permit early discovery of viruses.

The principle of least privilege, enunciated above, always acts to reduce risk. The fewer people who are allowed to modify programs, for example, the slower the spread of viruses is likely to be. Another time-honored generally accepted standard of good practice, "separation of duties," (4) holds that if a sequence of operations can put an organization at risk, then they should be performed by individuals with potentially conflicting motives (so that collusion, coercion, or duping is necessary to successful fraud). Applying both principles scrupulously can lead to a situation wherein no one individual or very few are both susceptible to receipt of a virus and able to propagate one. Thus, access control, in implementation of the two principles, is a defense against viruses.

Preventive software: Preventive software is code that may keep a virus from ever reproducing itself. Controls on program modification are obviously included, but they are not the only weapons of this type at our disposal. Another is a tool that establishes a special environment for the execution of a program that is not known to be free of viral contamination. (36) Advance of the clock should be simulated to coax any time bomb to show itself. Such an environment should be designed to make it possible to detect any programmed attempt to learn of the existence of a modifiable program. Any unexplained attempt suggests the existence of a virus in the program being tested and alerts the tester to the need for further examination of its logic. If examination fails to satisfy, the provider of the program might be asked to provide a satisfactory explanation or documentation, for example, source code, that would satisfy concerns about viral infection.

You could try to retard propagation instead of preventing it altogether. The principle of least privilege comes into play here. Some (6) have gone further, however, suggesting that people be forbidden to share programs. There is probably no environment in which that measure would be practical.

Detective software: Since a virus, by definition, modifies programs, you can use software to check for irregular modifications of programs, that is, modifications that did not occur in the way you have established for program development. The most obvious way to do this is to establish a protected (for example, offloaded) copy of each program every time it is modified in accordance with your established procedures and then check the production copy against the protected copy from time to time (preferably at irregular intervals so that an opponent cannot anticipate your check and restore the correct code just in advance of it). As with so much we say here, the control just described has long been recommended, even in the absence of a viral threat; it guards against all fraudulent modifications of programs.

Mere comparison has the drawbacks that infection may occur in source before the first time you stow a protected copy and each program is vulnerable between comparisons. The former implies that other protective measures must be used in combination with copy-compare. The latter implies that you may need to employ a more elaborate detective defense.

A detective defense can operate at each invocation of a program. (17, 31) Such a defense, of course, may cost far more than it is worth, and its use should be undertaken only with strong justification. Such a dynamic protective defense involves self-checking. This can be as simple as a checksum, like those that have long been used to check the integrity of data on magnetic tapes (for example, the storage at data's end, of the result of successive exclusive ORs of each four-byte, or other size, block of data in the program). Far more elaborate schemes have also been proposed (17, 31), employing encryption of only the checksum or of the entire program. Some machine architectures afford a far simpler alternative. Those of the System/38 (30) and AS/400 (18), for example, absolutely prevent modification, as opposed to replacement, of a compiled program without system privilege. A virus in a machine with such an architecture would have to function at the level of source code, because the object code is invulnerable.

It is impossible to write code that will detect every possible virus. (6) Therefore, code designed to detect viruses in general is vulnerable to an opponent who can create a virus that it cannot detect. Anyone who wrote such code would want to distribute it widely for profit, but could not prevent a potential perpetrator from being among the customers. For that reason, it seems unlikely that anyone will ever try to write a program that will detect the presence of any virus at all in any program. Programs designed to detect viruses by looking at their code will very probably be limited to those that are specific to particular, known viruses, so such programs are subsumed under our next topic, below.

You could try to analyze the code of programs that you acquire from sources of dubious trustworthiness, but that obviously entails forgoing the use of some programs, supplied without source code, that might be very useful to you. Some (6) have gone so far as to suggest that you demand and analyze the source code of every program you use. There is probably no environment in which that measure would be practical.

Virus-specific software Many viruses that have been discovered are probably not yet completely eradicated. Therefore, prudence may dictate the use of software that searches for and eliminates them on your systems. Lists and descriptions of such software are published from time to time in various places. (15, 32) Use of such techniques is especially vital when restoring programs from back-up copies. You would not want to restore the very virus that you just eliminated!

Defenses against worms

The difference between a harm-doing worm and useful processing can be determined only by a person who is in a position to define what constitutes "harm." (5, 26) Therefore, defenses against worms involve alerting people to the possibility that a harm-doing worm might be at work; giving alerted individuals the tools they need to determine whether a worm does indeed threaten their system; and means for working with others to eradicate the harmful worm after one has been found. (5, 26, 27)

Usage alarms: Whenever a resource may become scarce, sound management includes observation of the rate at which it is being consumed and of the quantity remaining. This is no less true of bandwidth on a network's communication lines and computing power within a network than it is of anything else. Since just those resources have been depleted, respectively, by the two worms that have caused noticeable damage, the CHRISTMA EXEC and the Internet worm, they are the resources that concern us in this context.

If you perceive no extraordinary threat to your resources, no extraordinary defense may be justified. You may feel certain that resource-starved users will scream soon enough and loudly enough to alert your administrative personnel to any threat that a worm may pose. However, experience has indicated that worms spread so rapidly and so destructively that complacence is unlikely to be a good strategy for very many network administrators. Monitoring tools greatly mitigated CHRISTMA's effect on International Business Machines' VNET and they are likely to help others just as well. Response to the Internet was not so rapid (5, 26) and the consequences rather more dire. (5, 26)

Gateways and filters Once administrators recognize a threat, they must have means for dealing with it. Gateways, chokepoints through which much network traffic must travel, can be a great boon. Administrators can concentrate their efforts on those few systems rather than having to act separately and urgently at all of the network's nodes. What they must do is to insert code that will locate the worm and exterminate it. This was done very successfully in the case of the Internet worm, (26) a model for all such efforts in the future.

Crisis teams: On the other hand, post mortems (5, 26) revealed that efforts against the Internet worm were severely hampered by the fact that the administrators involved had become accustomed to total dependence on the network for communication among themselves and had no emergency procedure in place to deal with a situation like the one that confronted them. Network administrators who fear worms, as all should, should form teams of experts (27) who can act when worms are suspected and know on whom they can draw by extra-network means for assistance in a crisis.

Defenses against weapons that viruses or worms might harbor

Any harm-doing code that has traditionally threatened data processing is more frightening now that we understand that a virus or worm might propagate it. Thus, the defensive measures we need are not so much new as they are more urgently needed.

Access Control: The principle of least privilege, enunciated above, always acts to limit risk. The fewer things people are able to do, the less harm they can cause if they are duped into running a virus.

Back-up: Electronically processed data have always been exposed to numerous hazards. Electronic malfunctions, physical mishaps, and program bugs are all capable of erasing valuable data. For that reason, auditors have long been advising people to make copies of their important data and to store those copies securely. What has always been good advice has simply risen to the level of incontestable wisdom with the advent of viruses and worms.

Conclusions

Viruses and worms do pose threats of new magnitude, but the threats are not so new in type. All the harm-doing programs they can harbor can also exist independently. By themselves, the harm they can do is limited to expenditure of resources for the storage of programs, in the case of viruses, and for the carrying of data in networks, in the case of worms. The scarcity of these resources is not a new concern either.

Present Situation

Viruses and worms are receiving a great deal of attention, not only from the media interested in mysterious topics that seem to threaten machines' and technologists' alleged battle for supremacy over mere nontechnological mortals, but also from various communities concerned about computer security. It is precisely that concern and the lessons of history that lead this author, among others, (28) to conclude that the defense will prevail in the battle against self-replicating harm-doing programs. Certainly, active defense is now demanded by prudence, in the face of the rapidity with which these things can propagate, (7) but spending more for defense than one stands to recoup in terms of reduced loss would be as inadvisable in this case as in any other. (8)

Private sector

People who use data processing equipment are taking prudent steps to reduce their exposure to risk.

Awareness: Attendance at the events described below testifies to management's concern with the problem. We can presume that this concern is being passed on to their employees, the people who are the first line of defense.

Software acquisition: While this author lacks data, we have to believe that people are being more cautious in acquiring software. The phenomenon has been reported so often in connection with analogous sexually transmitted diseases (STDs), that we must believe that it applies as well to software-transmitted discomforts (STDs).

Government

The copious publicity attending incidents of self-replicating harm-doing programs and the current attention to division of responsibility between the United States (US) Department of Defense

(DoD) National Computer Security Center (NCSC) and the US Department of Commerce (DoC) National Institute for Standards and Technology (NIST, formerly the National Bureau of Standards, NBS) insure that both the NCSC and NIST will devote considerable attention to the subject.

Viruses: Governmental activity related to viruses has included not only efforts by the NCSC and NIST, discussed below, but also a resolution of the United States Congress. (14) It refers to knowing insertion of loss-causing code rather than to viruses, however. We can predict that the law-drafting process, drawing on a study recently prepared by the Congressional Research Service, (14) will sharpen its focus eventually. That study raises the possibility that research on computer viruses will be treated as research with recombinant DNA is, "within a regulatory framework," because of inherent possibilities for catastrophic mischance or misuse.

National Computer Security Center (NCSC) The NCSC's virus-related activity is reflected in (19, 38). Both are short papers of some practical use. They are undoubtedly indicative of more intense activity yet to surface.

National Institute of Standards and Technology (NIST): NIST runs a Computer and Telecommunications Security Council (CTSC). This author has drafted a position statement on viruses and worms that is due for consideration by the CTSC in 1989 and parallels this paper.

NIST is in the process of preparing guidelines for federal computer security administrators to help them deal with viruses.

Worms: NCSC and NIST have collaborated with /usr/group (5, 26) to address the threat of worms in Internet and ARPANET. The result has been the formation by the Advanced Research Projects Agency (ARPA) of a Computer Emergency Response Team (CERT) at the Software Engineering Institute (SEI), Carnegie Mellon University. NCSC and the NCSC will coordinate CERT's activities, which will be along the lines suggested hereinabove and will serve as a stimulator and focal point for research as well.

NIST is in the process of preparing guidelines for federal computer security administrators to help them deal with worms.

Academia

While the academic community has been working on viruses and worms, little of the work is yet evident outside of the matter of the Internet worm, which a member of that community allegedly perpetrated and several (11, 33, 35) have studied. In connection therewith, there has been considerable discussion of the ethics of loosing a self-replicating program on an unsuspecting community. A consensus in the negative quickly formed, but consequent action is not yet apparent beyond the IEEE Committee on Public Policy's Subcommittee on Computer Ethics, which is considering a draft position paper on the subject.

Data processing community

Aside from /usr/group's activity, (5, 26) little concrete has been seen from the data processing community at large. Deloitte Haskins & Sells hosted a Computer Virus Workshop in New York City October 10-11, 1988, and various publishers and consultants have made a great deal of information available. International Business Machines Corporation (IBM), has provided a great deal of information on the subject, including this paper and (37), to its customers and to the public at large.

Software for creating computer viruses for one manufacturer's personal microcomputer is available.

Education: Many for-profit courses on Computer Security have added material on viruses and worms. It is of varying quality and usefulness and should be approached with some caution.

Ralf Burger has written *"Computer Viruses"--A High Tech Disease*, whose English language publisher is Abacus of Grand Rapids, Michigan. The book describes how to create computer viruses. Philip Fites, Peter Johnson, and Martin Kratz wrote *The Computer Virus Crisis*, published in 1989 by VanNostrand Reinhold of New York City, New York. The book discusses viruses, worms, Trojan horses, and other phenomena in less detail. Ralph Roberts' *Computer Viruses*, published in 1988 by Compute! Books of Radnor, Pennsylvania, dwells on stories of computer viruses and recommendations for defensive measures. The Internet worm occurred too recently for mention in any of the books.

Preventing distribution of viruses: The author has seen no publicly available material on this topic of obvious importance and urgency.

Network control features that could inhibit worms: (2) is the best, if not the only, paper on this subject currently available to the general public.

Recommendations

While there are several ways one can try to protect oneself from the harm that viruses and worms can do, the most efficacious measures would seem to be those that are more broadly applicable and time-tested:

1. Limit privilege.
2. Use only software you have good reason to trust.
3. Control program modification.
4. Monitor resource usage.
5. Educate users to report unexpected events.
6. Protect and back up sensitive data.
7. Form teams of individuals to deal with extraordinary problems.

In addition, there are a couple of things you might discuss with those from whom you buy software for personal computers:

1. Optionally, have any system for a personal computer check with the human user, via an immediate message that no software can circumvent, whether an attempted program modification is in accord with the user's wishes. (20)
2. Take all prudent steps to ensure that no virus is shipped with the vendor's code.

Summary

Viruses and worms are likely to be problems of greater significance in the next few years, but they are unlikely to overwhelm us and we are likely to get them under control eventually. Existing security measures, because they are applicable to the threats posed by viruses and worms, have thus far averted catastrophe, but it is clear that their importance is growing and the penalties we may pay for failing to use them are growing as well, because of viruses and worms. Moreover, they will not suffice to meet the threat of increasingly significant attacks. Specialized defenses, now in their infancy, will have to be developed and marshaled.

Acknowledgment

Much of the information herein came from conversations the author had with Steve White and Dave Chess of IBM's T. J. Watson Research Laboratory in Yorktown Heights, New York. Their (37) is a most knowledgeable and thoughtful view of this same topic.

Appendices

The following appendices contain descriptions of some representative viruses and a worm and a list of representative antiviral software.

Harmful Self-Replicating Programs

This appendix serves to give you a feeling for the types of harmful self-replicating programs that people have encountered. Consult the references for more detail on each.

Internet Worm

The Internet worm (5, 11, 26, 31, 35) first appeared at Cornell University just after 5 P. M., Wednesday, November 2, 1988. Except for a brief resurgence, the worm, which infected the Advanced Research Projects Agency ARPANET and Internet, was cotrolled within 48 hours. As noted above, it propagated in three ways: Via a trap door, via a bug, and via remote execution under an account whose password it had cracked. (A "trap door" is a security bypass deliberately installed in system code to permit the installer to do things that a security administrator might act to prevent.)

The worm obtained passwords in three ways, as noted above. Two were routine, but the third involved simulating the system's action in performing one-way (that is, irreversible) encryption upon an entered password. [1] All of the worm's code relates to its propagation or defense; it contains no bomb. Its "explosion" occurred because it spread so rapidly, reinfecting systems repeatedly, and performing so much processing in its attempt to propagate further, that it overwhelmed the processing capacity of each system it entered successfully. Much of the excess, that is, the reinfection and the protracted processing, appears to have resulted from bugs rather than from design.

BRAIN Virus

The BRAIN virus (3, 16, 21) afflicts PC-DOS. It derives its name from the fact that it indicates a copyright of "BRAIN" in the label of every diskette it infects. Some call it the "Pakistani virus" because the names of two Pakistani brothers were found in its code. The code's behavior depends on what already exists on each diskette it infects, but most often it does no great damage, moving some data without destroying any or making any unavailable, even temporarily. If a diskette's File Allocation Table has certain characteristics, then the virus will destroy some data. When an infected disk is used to boot a computer, the virus enters the computer and it will thereafter infect other diskettes with boot records that are used without write-protection on the same computer.

Israeli Virus

The Israeli virus (3, 22) was first discovered in Israel and contains a time bomb set to explode on Friday, May 13, 1988, the fortieth anniversary of the last day of the existence of the nation of Palestine, as ruled by Great Britain under the Balfour mandate, (23) and on every Friday the Thirteenth thereafter. It will infect any vulnerable file whose name's second qualifier is "COM" or "EXE" that runs on an infected PC-DOS system, increasing the file's size by about 1800 bytes. The first time any infected program runs on a system, the virus infects its DOS's "execute program" facility. Thereafter, it will infect any suitable program run on the system. Because code intended

[1] Robert T. Morris, now the Chief Scientist of the NCSC and also the father of the man, Robert T. Morris, Jr., alleged to have perpetrated the Internet worm, and Ken Thompson, whose ACM Turing Award lecture that was published in the *Communications of the ACM* (*CACM*) in 1984 was about trap doors and Trojan horses, discussed the attack and how to thwart it in "Password Security: A Case History," Computing Science Technical Report #71, dated April 3, 1978. The same work appeared in *CACM* on Pages 594-7 of the November 1979 issue, wherein the authors wrote, "On the issue of password security, UNIX is probably better than most systems. The use of encrypted passwords appears reasonably secure in the absence of serious attention of experts in the field. (paragraph break) It is also worth some effort to conceal even the encrypted passwords." One might conjecture that the preceding passage might have held more significance for a son of one of its authors than it held for security administrators of systems penetrated by the Internet worm.

to prevent reinfection has a bug, some multiply infected files become very large, slowing systems and revealing the virus's presence. The time bomb erases files that the user tries to execute.

Lehigh Virus

The Lehigh virus first appeared at Lehigh University. It infects COMMAND.COM on PC-DOS. After its fourth (or, for an evolved strain, tenth) infection, it destroys all vulnerable data.

Antiviral Software

The list below comes from (15, 32)[00] and is organized as (21) suggests. It is incomplete and provided for informational purposes only, with no representation as to the quality, usefulness, or safety of any program listed. See (15, 38) for an alternative suggestion for organization.

Preventive

1. Bombsquad, Swarthmore Software Systems, Swarthmore, Pennsylvania
2. Check-4-Bomb, Swarthmore Software Systems, Swarthmore, Pennsylvania
3. Disk Defender, Director Technologies, Incorporated, Evanston, Illinois
4. Disk Watcher, RG Software Systems, Willow Grove, Pennsylvania
5. Dr. Panda Utilities, Panda Systems, Wilmington, Delaware
6. Dprotect, Gee Wiz Software Company, East Brunswick, New Jersey
7. Flu Shot 3, Ross Greenburg, New York, New York
8. Novirus, Digital Dispatch, Incorporated, Minneapolis, Minnesota
9. Vaccine, Worldwide Data Corporation, New York, New York
10. ViruSafe, ComNETco, Incorporated, Bernardsville, New Jersey

Detective

1. Antigen, Digital Dispatch, Incorporated, Minneapolis, Minnesota
2. Cryptographic Checksum, Dr. Fred Cohen, Cincinnati, Ohio
3. Data Physician, Digital Dispatch, Incorporated, Minneapolis, Minnesota
4. Vaccine, Sophos Limited, Kidlington, Oxford, England
5. Vaccinate, Sophco, Incorporated, Boulder, Colorado
6. VI-Raid, Prime Factors, Eugene, Oregon
7. Viralarm 2000, Lasertrieve, Incorporated, Metuchen, New Jersey

Virus-specific

1. Antidote, Quaid Software Limited, Toronto, Ontario, Canada
2. C-4, Interpath Corporation, Santa Clara, California

[2] A less orthodx source is "A Software Bestiary" by Corinne Cullen Hawkins on Pages 107-112 of the Fall 1988 issue of *Whole Earth Review.*

References

(1) Anderson, James P., "Computer Security Technology Planning Study," ESD-TR-73-51, Volumes I and II, USAF Electronic Systems Division, Bedford, Massachusetts, October 1972. (The concept of a "Trojan horse" is Dan Edwards'.)

(2) Arbouw, Peter, "Security in Multi-Company Networks," *Proceedings of the Second European Conference on Computer Audit, Control and Security*, Amsterdam, the Netherlands, November 4-6, 1987, Paper 49 (8 pages).

(3) Chess, David, Private communication.

(4) Clark, David D., and David R. Wilson, "A Comparison of Commercial and Military Computer Security Policies," *Proceedings of the 1987 IEEE Symposium on Security and Privacy*, IEEE, Pages 184-194.

(5) Clegg, Frederick, et al. "Forum on Computer Virus (sic) Attacks," /usr/group, Baltimore, Maryland, December 2, 1988.

(6) Cohen, Fred, "Computer Viruses," *Proceedings of the 7th DoD/NBS Computer Security Conference* 1984, Pages 240-263. The paper was first presented in August 1984 at the IFIP Sec/'84 Congress and Exhibition in Toronto, Canada, but the paper did not appear until the proceedings of the September conference in Gaithersburg, Maryland, were published.

(7) Cohen, Fred, "Computer Viruses: Theory and Experiments," *Computers and Security* Volume 6, Number 1 (January 1987), Pages 22-35.

(8) Courtney, Robert H., "Contemporary Data Security: A Leadership Vacuum," *Computer Security Journal*, Volume 4, Number 2 (1987), Pages 7-16.

(9) Davis, Frank G. F., and Rex E. Gantenbein, "Recovering from a Computer Virus Attack," *Journal of Systems and Software* Volume 7, Pages 253-258.

(10) Denning, Peter J., "Computer Viruses," *American Scientist*, Volume 766 (May-June 1988), Pages 236-238.

(11) Eichlin, Mark W., and Jon A. Rochlis, "With Microscope and Tweezers: An Analysis of the Internet Virus (sic) of November 1988," *Proceedings of the 1989 IEEE Symposium on Security and Privacy*, IEEE, forthcoming.

(12) Fak, Viiveke, "Are We Vulnerable to a Virus Attack, A Report from Sweden," *Computers and Security* Volume 7, Number 2 (April 1988), Pages 151-155.

(13) Gunn, J. B., "Use of Virus Functions to Provide a Virtual APL Interpreter Under User Control," *APL QuoteQuad* Volume 14, Number 4, ACM SIGAPL, Pages 163-168.

(14) Helfant, Robert, and Glenn J. McLaughlin, "Computer Viruses: Overview and Policy Considerations," Congressional Research Service, Library of Congress, December 15, 1988. (Refers to Herger, Rep. Wally (R-California), "Computer Virus Eradication Act of 1988," House Resolution 5061, July 14, 1988.)

(15) Highland, Harold Joseph, "An Overview of 18 Virus Protection Products," *Computers and Security* Volume 7, Number 2 (April 1988), Pages 157-161.

(16) Highland, Harold Joseph, "The BRAIN Virus: Fact and Fantasy," *Computers and Security* Volume 7, Number 4 (August 1988), Pages 367-371.

(17) Hoffmeister, Frank, "An Approach to Defend Computers Against Computer Viruses," *Proceedings of the IASTED International Symposium of Applied Informatics--AI '87* Pages 176-179.

(18) International Business Machines Corporation, "AS-400 Programming: Security Concepts and Planning," Form Number SC21-8083, 1988.

(19) Israel, Howard, "Computer Viruses: Myth or Reality?," *Proceedings of the 10th National Computer Security Conference*, September 21-24, 1987, Pages 226-230.

(20) Kurzban, Stanley A., "Antiviral Measures--Which Ones Pay?," in *Computer Viruses, Proceedings of an Invitational Symposium*, Deloitte Haskins & Sells, New York, New York, October 10-11, 1988, Pages 56-57.

(21) McAfee, John, "The Virus Cure," *Datamation*, Volume 35, Number 4, February 15, 1989, Pages 29ff.

(22) McLellan, Vin, "Computer Systems Under Siege," *The New York Times*, January 17, 1988, Business Section Pages 1 and 8. Also published on Pages 33-38 of Volume 3 (1988) of the *The EDP Auditor Journal*

(23) Martin, James, *Security, Accuracy, and Privacy in Computer Systems*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1973.

(24) Murray, William H., "The Application of Epidemiology to Computer Viruses," *Computers and Security* Volume 7, Number 2 (April 1988), Pages 139-155.

(25) Murray, William H., and Peter Capek, "Can we, or Should we, Trust Data We Get From Others?," *Computers and Security*, forthcoming.

(26) National Computer Security Center, "Proceedings of the Virus (sic) Post-Mortem Meeting," Fort George G. Meade, Maryland, November 8, 1988.

(27) Office of the Secretary of Defense, "DARPA Establishes Computer Emergency Response Team," News Release, December 6, 1988.

(28) Parker, Donn, "Computer Crimes, Viruses and Other Crimoids," speech presented to the Commonwealth Club, January 13, 1989.

(29) Parker, Donn, *Crime by Computer*, Charles Scribner's Sons, New York, New York, 1976, Page 107.

(30) Pinnow, K. W., J. G. Ranweiler, and J. F. Miller, "System/38 Object-Oriented Architecture," *IBM System/38 Technical Developments*, International Business Machines Corporation, 1978, Pages 55-58.

(31) Pozzo, Maria M., and Terence E. Gray, "An Approach to Containing Computer Viruses," *Computers and Security* Volume 6, Number 4, Pages 321-331.

(32) "Revised List of Virus Filters," *Computers and Security* Volume 7, Number 3 (June 1988), Pages 259-260.

(33) Seeley, Donn, "A Tour of the Worm," University of Utah Technical Report, November 1988.

(34) Shoch, John F., and Jon A. Hupp, "The Worm Programs--Early Experience with a Distributed Computation," *Communications of the ACM* Volume 25, Number 3 (March 1982), Pages 172-180.

(35) Spafford, Eugene H., "The Internet Worm Program: An Analysis," CSD-TR-823, November 1988, Purdue University.

(36) Troxell, Peter J. and Eugene F. Troy, "An Isolation Mechanism for the Evaluation of Software for Malicious Code,' in *Computer Viruses, Proceedings of an Invitational Symposium*, Deloitte Haskins & Sells, New York, New York, October 10-11, 1988, Pages 59-63.

(37) White, Steve R., David M. Chess, and Chengi Jimmy Kuo, "Coping with Computer Viruses and Related Problems," Research Report Number RC 14405, International Business Machines Corporation, Yorktown Heights, New York, 1989.

(38) Young, Catherine L., "Taxonomy of Computer Virus Defense Mechanisms," *Proceedings of the 10th National Computer Security Conference*, September 21-24, 1987, Pages 220-225.