SOFTWARE UNDER SIEGE:

VIRUSES AND WORMS

SOFTWARE UNDER SIEGE:

VIRUSES AND WORMS

E.L. Leiss

Department of Computer Science
and Research Computation Laboratory
University of Houston, Texas, USA

Elsevier Advanced Technology
Mayfield House, 256 Banbury Road, Oxford OX2 7DH, UK
Commissioned by
Technical Communications (Publishing) Ltd

Elsevier Science Publishers Ltd
Mayfield House, 256 Banbury Road, Oxford OX2 7DH, England

British Library Cataloguing in Publication Data
Leiss, Ernest L., 1952-
Software under siege.
1. Computer systems. Viruses. Management aspects
I. Title
658.4'78

ISBN 0-946395-58-6

PREFACE

Computer viruses have recently attracted widespread attention, all of it negative. Although they had been known at least since 1983, they were customarily dismissed as insignificant and posing no discernable threat. Subsequent successful attacks by viruses and similar attackers have disproved this position, with the result that parts of the computing community have now embraced the other extreme, a great deal of fear and loathing and a distinct feeling of defenselessness.

This book is based on the premise that a known enemy is easier to defeat than an unknown one. To this end we describe various methods of attacking computer systems and draw conclusions from then about ways of combatting viruses. We feel that there are many who are not interested in all technical intricacies of viruses, but nevertheless want to obtain a reasonably accurate and complete description of the threat posed by them. Thus, we have avoided using too much technical jargon. For those readers who do want to learn more about the technical details, we have included references which should satisfy their interest in more technical points.

This book grew out of an article on computer viruses that I was asked to prepare for the 1990 yearbook of the Encyclopedia of Physical Science and Technology. Subsequently, I felt the need for a more in-depth treatment of this topic, which eventually occasioned the present book. In collecting literature for these works, I was ably assisted by a graduate student, Ms. I-Ling Yen; of course any omissions and errors are exclusively my own fault. By and large, this book reflects the status of virus research and related developments through summer 1989. It is therefore very up-to-date at the time it goes to press and should provide an accurate reflection of developments in this field.

E.L. Leiss

CONTENTS

PART I

INTRODUCTION

The introduction sketches the backdrop against which the threats against data integrity are played out. It discusses changes in the way computing is done and their relationship to the emerging threats by computer viruses and other attackers.

Chapter One

Background: Changes and Concerns

This book is about computer viruses, worms, logical bombs, and other threats to software and data. It will describe the illness, starting with actual attacks, define the causes, and discuss diagnosis as well as possible cures and prevention. However, in order to understand the phenomenon better, one has to start with the overall context within which these problems were able to arise.

Our starting point is an incident, of which certain facts are clear and well known. On the evening of November 2, 1988, a piece of software attacked and successfully invaded an estimated 6000 computer systems world-wide. Within a matter of hours, these systems were inoperable. Reportedly, this was the first time that mainframes were attacked; prior to this incident, only personal computers (PCs) had been affected in several relatively isolated instances. The November 2, 1988, incident exploded the myth that only rather unsophisticated systems such as PCs were vulnerable, thereby creating a great deal of anxiety in commercial computing centers around the world.

Instead of asking how computer virus attacks can occur, it proves much more instructive to ask why there were no extensive attacks earlier. The answer to this question lies in the change that occurred in computing in general in the last five years or so.

1.1 Changes in Computing Milieu

Several major changes occurred in the way users view computing in the past few years. Two of these are of central interest in our exploration of the reasons why

viruses and other attackers threaten the security and integrity of our computer installations. The first relates to the way in which computing is, and has been, administered; it reflects the trend away from monolithic, hierarchical computing centers or centralized computing and towards networks of local workstations or distributed computing. The second change has to do with the distribution of software, in particular the sharing of software at a variety of levels.

1.1.1 Centralized versus Decentralized Computing

Until the beginning of the 1980s, most serious computing was done on mainframes. Together with this came a certain mindset: there was a centralized computing center; all processing power was located there; virtually all data and all programs were stored there; users submitted jobs, typically for batch processing, from (dumb) terminals; and most importantly, all systems programmers were physically located at this center. As a result, any user who needed capabilities that were not ordinarily granted to end users had to submit a request to the computing center which in turn acted upon it. Thus, systems privileges were tightly restricted to a rather small group of systems programmers who could be rather stringently controlled, both physically and organizationally (see Figure 1).

Interactive processing became commercially accepted in the 1970s; in principle it gave the end user more capabilities, but the mindset did not change substantially. However, interactive computing did force the operating systems to become more sophisticated since now several users were active at the same time; consequently safeguards had to be built in to protect users that were running jobs at the same time from interfering with each other.

At the end of the 1970s, PCs made their appearance. Initially these were very primitive systems. Designed for a single user, they were slow, with insufficient data storage

and very rudimentary operating systems. By and large, these were home computers, as opposed to business computers, and they remained just that, even when individual systems became more powerful. In advertisements and computer magazines, PCs were quite early touted as superior to terminals connected to central mainframes; however the business community remained largely unconvinced. For good reason: most business activities tend to be interrelated; different units of the same enterprise must operate on a common database. Consequently, computers that were unable to share data and programs efficiently and consistently did not meet the requirements of business data processing in the 1980s. (Copying data onto floppy disks and sharing disks should be viewed akin to corresponding by pigeon carriers.) Moreover, secondary data storage facilities of PCs were simply inadequate for business purposes.

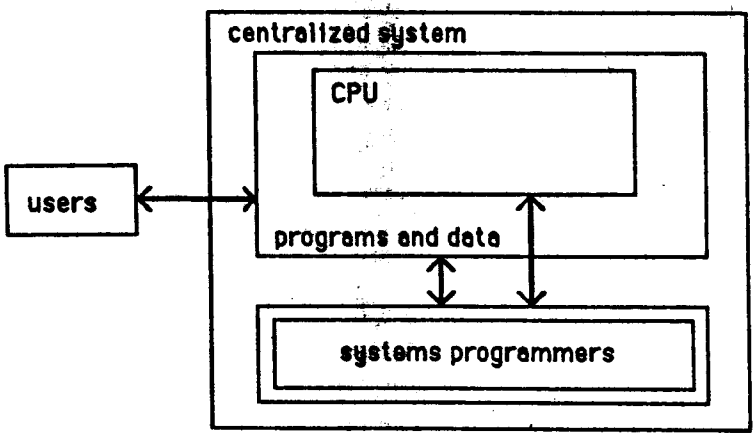

Figure 1. Centralized processing

This situation changed substantially with the advent of networking, especially local area networks (LANs), in the second half of the 1980s. Instead of having isolated islands (read PCs) floating independently in a sea of information, accessing data independently and producing possibly inconsistent results, all computers of an enterprise could now be connected to each other, including PCs and mainframes. Even more importantly, through the use of networks data storage could be organized so that different processors could access the same databases and operate on them in a consistent manner.

With networking came a decentralization of power, as the computing center personnel were no longer the undisputed high priests of computing in a company. Most other units in a commercial enterprise would have fought such a development. For example, accounting functions are typically reasonably centralized, reporting to a controller, who would strenuously oppose any move to distribute these functions to the smallest operational units (departments, project teams) of a company. However, since personal work stations (as PCs came to be known, once marketing divisions of PC manufacturers realized that businesses did not want to buy PCs - read home computers - for their employees) and networking were considered the cutting edge of computing technology, computing centers either acquiesced to, or even became active champions of, the decentralization of computing.

This development implied a rather subtle but from our point of view crucial change: every employee who had a work station on his or her desk now acquired, to some extent, functions that previously were carried out by systems programmers. While in many cases this change was not obvious to the user, since it rarely went beyond the loading of the operating system which for most employees was not much more than the switching on of the terminal, at the operational level this introduced a qualitative and most important difference: it was now the end user that was in control of the computer, and not the computing center personnel (see Figure 2).

The significance of this shift did not become immediately apparent to the users, nor for that matter to most computing center directors. However, at the level of the operating system, it had far reaching consequences. As we pointed out, the systems software (file systems, operating systems, etc.) of PCs was basically designed with a single user in mind. As soon as PCs were connected to each other (and to mainframes), this single-user world view was shattered; yet the operating systems of many personal work stations did not change drastically: on the one hand, networking was added to enhance the functionality of the older systems, on the other hand, even newer systems sometimes ran older systems software, usually for reasons of compatibility. As a result, safeguards against (accidental or malicious) undesired changes of data and software were, and still are, substantially inadequate. Since most networks are not safe (and are really not designed to be safe), it is the processors on which the burden of safeguarding data and software rests. For historical reasons as well as for reasons of personnel management, mainframes are somewhat more resilient to such changes than personal work stations (but by no means impervious; see Section 4.4).

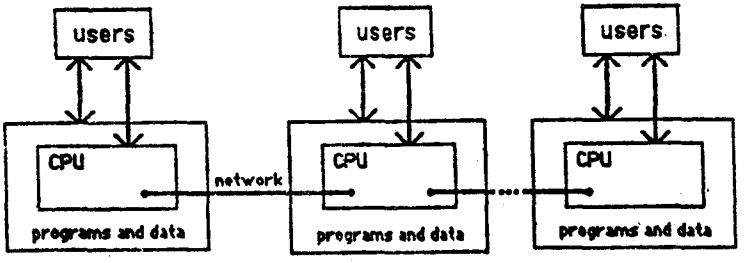

Figure 2. Processing using networked workstations.

1.1.2 Sharing of Software

With the advent of significantly expanded programmer communitites, the question arose whether it made sense to write programs for essentially the same problems again and again. This question was (obviously) answered negatively, and this gave rise to the problem of how to share software. Sharing here has a variety of meanings. It could be attempting to copy (usually illegally) a major piece of code supplied by a vendor with the objective of avoiding purchase; it could result from identifying an interesting and desirable technique for solving a particular common problem in a colleague's software; or it could be the consequence of a carefully designed "toolbox" of modules which a programmer established in order to simplify the task of solving complex problems. In all these cases, it is clear that both the person having the software and the person desiring it must agree to exchange the code. This is true regardless of whether the sharing involved is legal, i.e., if the owner has the right to make copies and distribute them, or not. Thus, in the situations described a certain personal relationship had to be established, a certain amount of trust was required.

A subtle change in this occurred with the advent of bulletin boards and program exchanges: the sharing became far less personal. In many cases, the person copying did not know the original owner of the software and guarantees were usually explicitly denied. However, these disclaimers were universally disregarded by the programmers who acquired at little or no cost interesting and attractive software that allowed them to do things that were either beyond their capabilities or would have required a substantial investment in time which they were unwilling to make.

In this way, trust was established that was misplaced, since it could be, and was, abused by more or less vicious pranksters. Programs were placed on electronic bulletin boards under the pretext of making them available to the general public whose primary function was the infection of the computer system on which they were installed. A particularly vicious (or ingenious) example is the program

Flu-Shot 4 which masqueraded as an anti-virus product but in fact was itself a damage-causing virus. Bulletin boards in this way provided an attacker with a trusted means of entry to a system, which was clearly in its ultimate result the antithesis to the purpose for which they were created; because they facilitated the distribution of attacks they undermined the very purpose for which they were created, namely the distribution and sharing of information and software.

Decentralized processing and software sharing provide the background against which the drama is played out. Many problems stemming from virus attacks can ultimately be traced back to this change in the computing milieu.

A related issue is that of convenience of use, which in most cases is compounded by the decentralization of control.

1.2 Protection versus Convenience of Use

Many security risks can be eliminated by using certain precautions. One problem with these precautions is that they almost invariably reduce the ease of use of the resulting system. For example, some of the threats to data integrity could be eliminated by encrypting data together with some redundancy (for more details, see the Appendix). The major drawback of this approach is that the data file must be decrypted before each use.

A more mundane example is the use of nontrivial passwords. As we will see, among others, the November 2 incident referred to above was substantially facilitated by the fact that users tend to choose rather simple passwords, which can often be guessed by an attacker. The obvious advantage of simple passwords is that they can be remembered easily. More complicated passwords tend to be written down and stored in an easily accessible location - thereby again compromising the security of the system they are supposed to protect. Note however that in the case of easily guessable passwords anybody with access to the network can become a threat, while in the

case of passwords posted in a drawer or on the side of the work station, physical access to the employee's office is required before a threat may materialize. Since physical access to an office is traditionally more restricted than access to a general purpose data network, complicated passwords, even if they are posted on the terminal, may be more secure than simple passwords which can be easily remembered but also guessed.

The problem of convenience of use is aggravated by the decentralization of control; previously a computing center might have unilaterally assigned passwords and changed them periodically to other prespecified ones; now the owner of a work station is in control of this. While it would of course be possible to assign preselected passwords to all participants in a network (and to change them periodically), this clearly goes against the notion of a "personal work station" – and is therefore often not done.

Another variation on this theme is provided by short-cuts. These are methods that circumvent ordinary controls, designed to ensure privacy and integrity. In most cases these short-cuts are used for convenience, and again they provide attackers with a convenient means of penetrating systems. For example, in the November 2 incident the perpetrator took advantage of such a short-cut.

Taken together the changes in computing milieu and the related emphasis on ease of use of computing equipment provide an answer to our original question, namely why computer virus attacks had not occurred earlier: while the programming techniques used to design viruses were certainly known for quite some time, it was widespread networking together with the decentralization of control that provided the fertile soil in which computer viruses could flourish.

1.3 Potential for Damage

Before 1988, several minor incidents had hinted at a potential for major problems related to the organized and widespread subversion of computer systems, and most