COMPUTER VIRUSES

Matt Bishop
Department of Computer Science, University of California, Davis, CA, USA

Keywords: policy, mechanism, trust, computer virus, Trojan horse, infection, prevention, detection, principle of least privilege, signature, specification, anomaly

Contents

Summary

Computer viruses attack systems by tricking an authorized user into executing them. The virus infects other programs, and spreads throughout the system. They can perform any action that the user can, including deleting files, altering data in files, and transmitting confidential information over a network.

Computer viruses are related to other types of programs such as Trojan horses and computer worms. These programs are forms of malicious logic. Users trust programs to perform a set of functions. The important characteristic of malicious logic is that it executes unknown, unwanted instructions as well as the known set of functions. The unknown, malicious instructions violate system security. But the operating system cannot determine this, because it does not know that the user is unaware of these instructions being executed (as the user ran the program containing them).

There is no generic technique to determine if an arbitrary program contains a computer virus. Hence specific characteristics of particular computer viruses, or of particular types of computer viruses, must be used.

Prevention methods are designed to stop the computer virus from infecting other programs and from doing collateral damage. One set of methods prevents instructions from being interpreted as data, and vice versa. Another set limits the amount of sharing. A third reduces the access rights of programs as much as possible. The rights allowed depend on user settings, file names, or guards associated with files.

Detection methods are designed to detect computer viruses, either directly or through their actions. Signature analysis methods look for sequences known to be contained in computer viruses. Integrity checking methods examine files for unexpected changes that may be a result of infection. Specification-based methods look for actions that a program should not take. Anomaly detection looks for unexpected characteristics or actions of programs.

1. Introduction

Computer viruses are the bane of modern computing. An estimated 50,000 computer viruses provide a variety of effects ranging from the merely unpleasant to the catastrophic. They attack all platforms and are written in all popular computer languages. As Internet connectivity grows, the ease with which computer viruses can spread also grows. In 1984, the first computer viruses were contained at a few sites. In 2000, the ILOVEYOU program spread worldwide within hours.

Understanding the threat of computer viruses requires understanding what computer security is. A security policy states what is, and is not, allowed. Such a policy may refer to actions (for example, users are not allowed to alter a set of configuration files), to configurations (for example, a system must require a password to authenticate a user) or both. A security mechanism enforces some aspect of a security policy. The threat that viruses pose is the ability to evade the restrictions that the security mechanisms impose. The computer virus thereby violates the security policy, threatening data and the operation of the system.

The term computer virus is widely misused and, worse, misunderstood. The next section places computer viruses in context. It presents other forms of rogue programs that are similar to computer viruses. It also discusses trust, a fundamental component of any security procedure or mechanism. The third section discusses theoretical considerations in the analysis of computer viruses. Computer viruses affect the way users and administrators trust systems. Technical measures against computer viruses are presented in the next section, followed by procedures and policies that limit exposure. We conclude with a discussion of the future of computer viruses and protections against them.

2. What is a Computer Virus?

Understanding computer viruses requires examining other, related threats. A computer virus is a special case of malicious logic (programs that act in violation of the security policy). The Trojan horse is the most general form of malicious logic. It is an appropriate starting point.

2.1. Trojan Horses

In 1972, the Anderson Report first identified the Trojan horse as a threat to computer systems. A Trojan horse is a program with an overt function and a covert function. The overt function is a documented result or effect that the user expects the program to

perform. The covert function is an undocumented result or effect that the user does not intend to occur.

As an example, a simple Trojan horse might present a login banner on a computer monitor. The user would enter a name and a password. The Trojan horse would copy both into a file and then pass both to the real login mechanism. The overt purpose of this Trojan horse is to log the new user in, which it does by passing the information to the login mechanism. The covert purpose is to save the user name and password in a file that an attacker can access at a later time. The user logging in is unaware of this function, which violates the site's security policy.

In 1974, Karger and Schell described a Trojan horse with a covert function of copying itself. This replicating Trojan horse (an example of which is presented in Thompson's Turing Award lecture) laid the groundwork for experiments with malicious reproducing programs.

2.2. Computer Viruses

According to Ferbrache, the first virus-like programs ran on Apple II computers. One wrote itself to the boot sectors whenever the catalog command was executed. A second infected a game program. The game stopped working. The author rewrote the game to locate, and delete, the infected copies of the game.

In 1983, Fred Cohen described a Trojan horse contained in a small segment of code placed into another (apparently benign) program. When executed, this segment of code inserts itself into another program (the infection phase) and then performs some action (the execution phase). The execution phase may be benign (for example, compressing files) or malicious (for example, deleting files). Figure 1 shows pseudo-code for a computer virus. In practice, benign computer viruses are exceptionally rare. For this reason, the term "computer virus" in this article refers to a malicious computer virus, unless otherwise stated.

```
vbegin:
        if condition is true then begin
        insert instructions from vbegin to vend into target file(s)
        patch target file(s) to invoke these instructions
        end
        perform some action
        goto infected program
vend
```

Figure 1. Anatomy of a computer virus.

The phases may require certain conditions to be satisfied. For example, the Lehigh computer virus would determine if the boot file of the disk were infected. If not, it inserted itself into the file (the infection). The virus then incremented a counter and,

when the counter reached 4, would erase the disk. However, if the boot disk were infected, the virus took no further action.

Figure 2 illustrates how a computer virus spreads. Priam has execute access to some of Odysseus' files. One of them contains the computer virus. After Priam executes that file, some of his files are infected. The key observation is that only those files to which Priam can write are infected. As the computer virus spreads by inserting code into other files, and the computer virus is running as Priam, it cannot write to any file that Priam cannot write to.

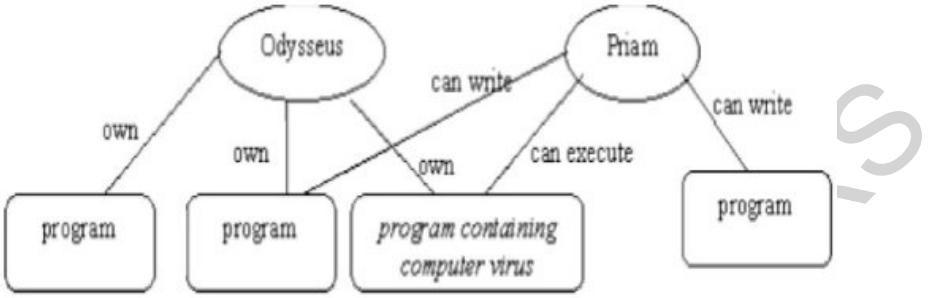

Figure 2. Example of a computer virus spreading. The infected files are italicized. The top diagram shows the permissions Odysseus and Priam have on a set of files. The bottom diagram shows the newly infected files after Priam has executed the infected program.

Because the computer virus is a form of Trojan horse, defenses against Trojan horses also work against computer viruses. This will be the starting point of the defenses discussed in section 5.

Computer viruses can be classified by type of targeted file, longevity, self-concealment, and type of virus.

2.2.1. Type of Targeted File

The Lehigh virus is of the type boot sector infector because it inserts itself into the boot sector. Boot sector viruses must ensure the system is bootable after infection. They can either rewrite the boot sector to insert themselves in any unused space, or copy the original boot sector to another location and arrange for it to be loaded after the virus runs.

An executable infector is a virus that infects an application. The MacMag Peace Virus infected an application, and when executed printed a message expressing hope for universal peace. The virus can either prepend itself to the program, or append itself. Typically, the virus code is inserted into the executable in empty areas (such as at the end of a page) or appended, and the executable modified to transfer control to the added code at the appropriate time.

Some computer viruses can infect either boot sectors or executables. These viruses are called multipartite viruses.

2.2.2. Longevity

A terminate and stay resident (TSR) computer virus deposits code either in memory or on disk that can be activated without the virus executing again. For example, the Jerusalem virus sets up code to intercept all service interrupt calls. Those that request a file to be executed trigger the virus. The file may not be affected, may be deleted, or may be infected (depending on the name of the file and the setting of flags). Even after the executable containing the Jerusalem virus terminates, the service interrupt code continues to intercept service requests. The stealth virus is a special type of TSR virus. Stealth viruses intercept file access requests. If the request is to obtain attributes of an infected file, or read an infected file, the data is sanitized to represent the uninfected file. But if the request is to execute the file, the infected file is returned. The virus conceals itself from programs that scan file systems looking for viruses (hence the name).

-
-
-

Bibliography

Cohen F. (1984). Computer Viruses: Theory and Experiments. Seventh DOD/NBS Computer Security Conference Proceedings 240–263. [This seminal paper introduced computer viruses and discussed many possible defenses].

Cohen F. (1987). Computer Viruses: Theory and Experiments. Computers and Security 6(1) 22–35. [This refined Cohen's results and proved the generic virus detection problem undecidable].

Denning P. (1990). Computers Under Attack: Intruders, Worms, and Viruses. Reading, MA, USA: Addison-Wesley Publishing Company. [This collection of papers discusses aspects of computer viruses and related programs].

Duff T. (1989). Experiences with Viruses on UNIX Systems. Computing Systems 2(2) 155–172. [In addition to an illuminating discussion of computer viruses in multiuser environments, this paper presents a portable machine-independent computer virus].

Eichen M. and Rochlis J. (1989). With Microscope and Tweezers: An Analysis of the Internet Virus of November 1988. Proceedings of the 1989 IEEE Symposium on Security and Privacy 326–343. [This is the seminal paper on the Internet worm of 1988].

Ferbrache D. (1991). A Pathology of Computer Viruses. New York, NY USA: Springer-Verlag. [This book presents a good overview of the history and types of computer viruses].

Hoffman L. (1990). Rogue Programs: Viruses, Worms, and Trojan Horses. New York, NY, USA: Van Nostrand Reinhold. [This is another good collection of papers on malicious logic].

Wack J. and Carnahan L. (1989). Computer Viruses and Related Threats: a Management Guide. NIST Special Publication 500-166. Washington DC, USA: National Institute of Science and Technology. [This guide presents procedures and management techniques for minimizing exposure to computer viruses, and managing situations in which computer viruses are found].