Global Information Assurance Certification Paper

How Viruses Attach
Bernard McCargo
December 8, 2000

Introduction

A printed copy of a virus does nothing. Even a copy of the executable code of a virus sitting on a desk does nothing. In order for a virus to do its malicious work and spread itself, it has to be activated by being executed. Fortunately for virus writers, but unfortunately for the rest of us, programs are executed all the time on a running computer.

For example, if the virus code was in a program on the distribution medium, when it was executed, it could install itself on the permanent storage medium (typically hard disk) of the computer, and the virus could also install itself in any executing programs in memory. In the beginning, a human being had to put the virus on the distribution medium, then a human had to execute the program to which the virus was attached. After that point the virus could spread by itself.

Appended Viruses

A program virus attaches itself to a program; then, whenever the program is run, the virus is activated. This kind of attachment is usually easy.

In the simplest case, a virus simply inserts a copy of itself into the executable program file before the first executable instruction, so that all the virus instructions execute first, and after the last virus instruction, control flows naturally to what used to be the first program instruction.

This kind of attachment is simple and usually effective. The virus writer does not need to know anything about the program to which the virus will attach and, often, the attached program simply serves as a carrier for the virus. The virus does it task then transfers to the original program. Typically, the user is unaware of the effect of the virus if the original program still does all that it used to. Most viruses attach in this manner.

Viruses That Surround a Program

An alternative to the attachment is a virus that runs the original program but has control before and after its execution. For example, a virus might want to avoid being detected. If the virus is stored on disk, it will show as a file, or its size will affect the amount of space used on the disk. If it gains control after the listing program has generated the listing but before the listing is displayed or printed, the virus could eliminate its entry from the listing and falsify space counts so that it appears not to exist.

Integrated Viruses and Replacements

A virus might replace some of its target, integrating itself into the original code of the target. Clearly the virus writer had to know the exact structure of the original program to know where to insert which pieces of the virus.

Finally, the virus can replace the entire target, either mimicking the effect of the target or ignoring the expected effect of the target and performing only the virus effect. In this case, the user is most likely to perceive the loss of the original program.

Boot Sector Viruses

A special case of virus attachment, but a fairly popular one, is the so-called **boot sector virus**. When a computer is started, control starts with firmware that determines which hardware components are present, tests them, and transfers control to an operating system. A given hardware platform can run many different operating systems, so the operating system is not coded in firmware but is instead invoked dynamically, perhaps even by a user's choice, after the hardware test.

The operating system is software stored on disk. The operating system has to start with code that copies it from disk to memory and transfers control to it; this copying is called the bootstrap. The firmware does its control transfer by reading a fixed number of bytes from a fixed location on the disk (called the **boot sector**) to a fixed address in memory and then jumping to that address. To run a different operating system, the user jus inserts a disk with the new operating system and a bootstrap loader. When the user reboots from this new disk, the loader there brings in

and runs another operating system. This same scheme is used for personal computers, workstations, and large mainframes.

The boot sector is an especially appealing place to house a virus because the virus gains control very early in the boot process, before most detection tools are active, so that it can avoid, or at least complicate detection. Also, the files in the boot area are crucial parts of the operating system, in order to keep users from accidentally modifying or deleting them with disastrous results, the operating system makes them "invisible" by not showing them as part of a normal listing of stored files, thus preventing their deletion. Thus, the virus code is not readily noticed by users.

The next steps in the boot process are loading and invoking standard parts of the operating system, reading files that personalize this installation, and loading and invoking files called for in the personalization. For MS-DOS/PC, for example, the standard parts of the operating system are files named IO.SYS and MSDOS.SYS, the personalization files are called CONFIG.SYS and AUTOEXEC.BAT. A virus can

- attach itself to either of the system files, IO.SYS or MSDOS.SYS
- attach itself to any other program loaded because of an entry in CONFIG.SYS or AUTOEXEC.BAT or
- add an entry to CONFIG.SYS or AUTOEXEC.BAT to cause it to be loaded

Virus Signatures

A virus cannot be completely invisible. Code must be stored somewhere and code must be in memory to execute. The virus executes in a particular way. And viruses use certain methods to spread. Each of these characteristics is a telltale pattern, called a **signature**, that can be found. The signature of a virus is important for creating a program called a "virus scanner" that can automatically detect and, in some cases, remove viruses. The scanner searches memory and long-term storage and monitors execution, watching for the telltale signatures of viruses. The scanner can then block the virus, inform the user, and deactivate or remove the virus.

Conclusion

The only way to prevent infection by a virus is to not share executable code with an infected source. Because you cannot always know which sources are infected, you should assume that any outside source is infected. Fortunately, you know when you are receiving code from an outside source; unfortunately, it is not feasible to cut off all contact with the outside world.

Techniques for building a reasonably safe community for electronic contact include these:

- Use only commercial software acquired from reliable, well-established vendors.
- Test all new software on an isolated computer.
- Make a bootable diskette and store it safely.
- Make and retain backup copies of executable system files.
- Use virus detectors (often called virus scanners) regularly.

Recent Virus Attacks

While I was writing this paper, I received an email from the University where I am the Head of the CIS Department, stating that they have been affected by a virus. The virus, known as TROJ_Shockwave.a arrives as an email message similar to the infamous Melissa and Love Bug worms. As Security professionals, we must remain current with the Virus Scanning Systems that we use in our day- to-day activities. A steady stream of changes and new challenges is assured as the hacker community continues to seek out vulnerabilities and weaknesses in our systems and our networks.

References

Bridwell, L. and Trippett, P. (2000, October). Malware Mayhem. Information Security Magazine,Vol 3, No. 10, 50-56. [On Line], Available; http://www.infosecuritymag.com

McAuliffe, M. (ZDNet) (2000, December). New Viruses Creates Shockwaves. [On-line], Available: http://www.zdnet.com/zdnn/stories/news/0,4586,2660381,00.html

Norton Anti-Virus. (2000).About Computer Viruses. Phrack Magazine [On Line], Available; http://www.qub.ac.uk/csv/software/pc/nav/v_about.html

Sodani, O. (1999, February). Viruses: Don't Let Your Computer Get Sick. Help2Go. [On-line], Available; http://www.help2go.com/5MinGuide/virus.cfm

Vallabhaneni, S. R. (2000). Telecommunications and Network Security. CISSP Examination Textbooks, Volume 1: Theory, Chapter 2, 44-131.

Last Updated: May 26th, 2017

Upcoming Training

SANS Stockholm 2017	Stockholm, Sweden	May 29, 2017 - Jun 03, 2017	Live Event
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
Community SANS Ottawa SEC401	Ottawa, ON	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS San Francisco Summer 2017	San Francisco, CA	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS Charlotte 2017	Charlotte, NC	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Rocky Mountain 2017 - SEC401: Security Essentials Bootcamp Style	Denver, CO	Jun 12, 2017 - Jun 17, 2017	vLive
Community SANS Portland SEC401	Portland, OR	Jun 12, 2017 - Jun 17, 2017	Community SANS
SANS Secure Europe 2017	Amsterdam, Netherlands	Jun 12, 2017 - Jun 20, 2017	Live Event
SANS Rocky Mountain 2017	Denver, CO	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Minneapolis 2017	Minneapolis, MN	Jun 19, 2017 - Jun 24, 2017	Live Event
SANS Paris 2017	Paris, France	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Cyber Defence Canberra 2017	Canberra, Australia	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, Japan	Jul 05, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, Singapore	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Minneapolis SEC401	Minneapolis, MN	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Los Angeles - Long Beach 2017	Long Beach, CA	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Phoenix SEC401	Phoenix, AZ	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Munich Summer 2017	Munich, Germany	Jul 10, 2017 - Jul 15, 2017	Live Event
Mentor Session - SEC401	Ventura, CA	Jul 12, 2017 - Sep 13, 2017	Mentor
Mentor Session - SEC401	Macon, GA	Jul 12, 2017 - Aug 23, 2017	Mentor
Community SANS Colorado Springs SEC401	Colorado Springs, CO	Jul 17, 2017 - Jul 22, 2017	Community SANS
Community SANS Atlanta SEC401	Atlanta, GA	Jul 17, 2017 - Jul 22, 2017	Community SANS
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Jul 24, 2017 - Jul 29, 2017	Community SANS
SANSFIRE 2017 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Jul 24, 2017 - Jul 29, 2017	vLive
Community SANS Fort Lauderdale SEC401	Fort Lauderdale, FL	Jul 31, 2017 - Aug 05, 2017	Community SANS
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event