In Berkshire Encyclopedia of Human Computer Interaction,
ed. W. Bainbridge, Berkshire Publishing, 2004, pp. 788-791

VIRUSES

Viruses and worms are malware: malicious computer programs that electronically spread through a network. A virus must be executed by a user action to spread, whereas a worm spreads simply by direct infection. (The terms *virus* and *worm* are often respectively attributed to a 1974 science-fiction novel by David Gerrold, *When H.A.R.L.I.E. Was One,* and a 1975 science-fiction novel by John Brunner, *Shockwave Rider.*) The first uncontrolled virus, Elk Cloner, appeared in 1982 and infected Apple II computers; today, during the height of recent computer virus attacks, e-mails containing computer viruses have accounted for up to 8 percent of e-mail being transmitted worldwide.

Vectors for Viruses

Computer viruses predate today's nearly universal use of the Internet. The earliest computer viruses were spread through floppy disks that were used to exchange messages. In many cases of early viruses on operating systems such as MS-DOS, the virus would be stored in the boot sector of a hard disk and installed in memory when the computer booted up. As use of the Internet has grown, virus transmission has largely moved to macro viruses and viruses that are spread by e-mail.

Macro viruses install "macros" in applications that allow users to execute code when they access certain document types. For example, popular applications such as Microsoft Word allow users to write macros—short programs that perform operations—and then store those programs both in the applications and in documents produced by the applications. A macro virus for Word installs itself in a Word application and spreads its code in Word documents produced by that application. When a Word document is sent from a user to a recipient, and the recipient opens the infected document, her Word application is infected. As she produces subsequent Word documents, they in turn carry the virus with them. Because most computers run Microsoft operating systems and Microsoft office applications, Microsoft document types are frequent targets of virus writers.

E-mail viruses spread through attachments to e-mail messages. These attachments contain executable code (often the fact that the code is executable is hidden) that causes them to spread when they are executed. Often the virus contains some tantalizing message (a recent virus contained the subject line "I love you"). In many cases e-mail viruses themselves are macro viruses using scripting programs built into popular e-mail programs such as Microsoft Outlook or Qualcomm Eudora. These scripting pro-

grams will search through a user's records to find e-mail correspondents and then transmit themselves to those correspondents. In many cases the e-mail virus will forge a return address, making location of the source of the virus difficult (and frustrating uninfected users as they sort through numerous messages that incorrectly report them as a transmitter of the e-mail virus).

Vectors for Worms

Worms spread directly from computer to computer. The first large-scale worm infection occurred on the UNIX operating system in November 1988 and was written by Robert T. Morris, then a graduate student at Cornell University. The Morris worm demonstrated a number of techniques that later became standard for worm authors. In particular, it was the first large-scale worm to exploit a weakness called "buffer overflow." The worm sent messages to other computers requesting information about users on those computers using the UNIX "finger" service. The messages exceeded the expected maximum length of the finger request. Because the receiving computers did not have provisions for overly long messages, data were corrupted. By careful exploitation of this weakness, the author of the worm was able to cause the worm to reproduce itself. Buffer overflow problems turned out to be common in many applications and operating systems, and by 1998 two-thirds of all advisories issued by the U.S. Department of Defense-sponsored Computer Emergency Response Team involved the buffer overflow weakness.

The Morris worm also demonstrated that worms can use multiple methods to spread themselves. The Morris worm attempted to guess passwords using a list of potential passwords; when a username/password infection was found on a given computer, the worm would then try that same username/password combination on remote computers likely to be used by the user (exploiting the fact that users often use a single password on multiple computers).

More recent worms spread at frighteningly fast rates. For example, the "Slammer" (also called "Sapphire") worm of 25 January 2003, infected systems that run Microsoft SQL Server software. The majority of Slammer infections occurred within ten minutes of the worm's release, and during the initial phases of infection, the infected population doubled in size every 8.5 seconds. (As the worm saturated virtually all vulnerable computers on the Internet, the exponential growth of infected computers abated.) The rapid spread of the worm made real-time human response virtually impossible. As a result, Slammer's spread led to serious real-world consequences, including a failure of major automatic teller machine networks, failure of the 911 emergency phone number system in Seattle, Washington, and failure of airline flight reservation systems and subsequent flight cancellations.

Worms, Viruses, and Denial-of-Service Attacks

In many cases worms and viruses are designed to carry out specific malicious attacks, such as destruction of data on infected computers. Perhaps the most serious attack is the backdoor attack, which allows the infected computer to be controlled at will by an outsider. A particularly serious type of backdoor attack is the denial-of-service attack. Denial-of-service attacks (often called "DoS" attacks) attempt to infect a large number of "zombie" computers. On a signal from the attacker (or on a particular date/time), the infected zombie computers start transmitting messages to a victim computer, overwhelming the capability of the victim computer to process the incoming messages. For example, the Code Red worm exploited a buffer overflow weakness in Microsoft software in an attempt to target the www.whitehouse.gov website run by the Office of the President of the United States.

Because denial-of-service attacks involve numerous infected computers, stopping them when they occur is difficult if not impossible—identifying all the infected computers and removing the worm or virus from them are simply unfeasible. Instead, the target of the attack must work with its Internet service provider to block certain types of incoming messages or rename the targeted site. Because these defenses can take hours to implement, waves

of denial-of-service attacks can make targeted sites unavailable for extended periods of time.

Varieties

Another type of malware is adware. Adware causes advertising messages to be generated on a user's computer. Although adware can be spread as part of a virus or worm, it is often installed on users' computers by more pedestrian methods: as part of an installed application. A number of free or low-cost applications on the World Wide Web include adware. Adware often is difficult to remove from a computer.

Conventional viruses and worms infect computers with exact copies of themselves. However, more powerful polymorphic viruses and worms modify themselves with each new infection. This modification makes detection of such viruses and worms more difficult because virus-scanning applications must scan for all possible forms. A virus designer can build a polymorphic virus by using a random number generator to generate a random cryptographic key and then using that key to encrypt the virus as it spreads onto a new computer. Because most of the virus will be encrypted, and each encryption will be different, detecting the presence of the virus is difficult.

Computer scientists have described a variety of methods by which backdoors can be installed in computer systems. One method deserves special mention; it was presented by Ken Thompson (one of the original developers of the UNIX operating system) in his acceptance speech for the 1983 Turing Award (the highest academic award in computer science). Thompson explained how one could use a compiler, a program that transforms source code into machine executable object code, to spread a backdoor. Thompson described how a compiler could recognize a login program. The compiler would add a backdoor to the login program, allowing any user who knew a particular secret string to gain access to a computer. Then the compiler would be modified to add the backdoor to any compiler program. After this stage was reached, it would no longer be necessary to include any evidence of the backdoor in source code—whenever a compiler program was itself recompiled, the backdoor would be automatically included.

This method is not merely a theoretical attack but rather was actually implemented by Thompson and by others. Because this kind of backdoor remains benign until actually triggered, detecting it even with full analysis of the source code of a system is impossible.

Defenses against Viruses and Worms

As Ken Thompson demonstrated, no method of virus and worm defense can be completely satisfactory—a clever virus author can defeat any form of defense. As a result, most defenses are reactive—rather than try to anticipate all possible viruses and worms, they try to detect and defend against viruses and worms that are actually observed in practice. After a virus or worm is identified, engineers characterize it by a set of particular features—its signature—that uniquely identifies it. (In the case of polymorphic viruses, finding a virus signature can be difficult.) A virus scanner is programmed to search for the signature, and whenever the signature is found, the virus scanner attempts to isolate or remove the infection. As new viruses are found, updated lists of signatures and removal (or isolation) techniques are published. Commercial virus scanners often use regular updates of these lists from a vendor-provided source.

Similarly, many operating system vendors (including Microsoft and Apple) provide systems for automatically and quickly distributing updates to vulnerable software. Although these systems are certainly useful, their effectiveness is limited if few people use them.

Experts often cite using better computer security (especially turning off unused software features) as a method for defending against viruses and worms. For example, many computer users do not need to use macro (scripting) features in e-mail programs, and if these features are turned off, viruses spread by e-mail will be far less effective. Unfortunately, turning off features is often complicated, limiting this technique's effectiveness for typical users. To maximize the capabilities of their software, vendors are usually motivated to ship software with most (or all) features turned on.

An alternative defense is a firewall, which is software or hardware that isolates a computer on a network. Firewalls vary from complex programs to relatively simple systems that are included in many home network routers. Firewalls can defend against many, but not all, attacks.

Many researchers attribute virus vulnerability to the widespread distribution of a relatively small number of operating systems and application software. For example, a recent report by a team led by computer security researcher Dan Geer described Microsoft software as a monoculture that presents a serious computer security threat. However, viruses and worms have been observed even in software (such as the Linux operating system) that is used by only a small fraction of users.

One proactive (acting in anticipation of future problems) method of virus protection involves statistical techniques. For example, a common characteristic of many worm attacks is that an infected computer will try to contact a large number of other computers. Some researchers have proposed automatically checking for this pattern of communication and temporarily disabling computers that demonstrate it. However, such a technique at best can only slow down an attack—as long as the virus author keeps the attack level below the triggering threshold, the attack will evade detection.

The number of viruses and worms has increased each year since they first became common in the mid-1980s. Although computer security specialists continue to find methods to defeat many viruses and worms, their growing presence suggests that we will continue to struggle with them for many years.

J. D. Tygar

See also Hackers; Spamming

FURTHER READING

Cheswick, W., Bellovin, S., & Rubin, A. (2003). *Firewalls and Internet security: Repelling the wily hacker* (2nd ed.). Boston: Addison-Wesley.

Computer Science and Telecommunications Board. (1999). *Trust in cyberspace*. Washington, DC: National Academy Press.

Geer, D., Pfleeger, P., Schneier, B., Quarterman, J., Metzger, P., Bace, R., & Gutmann, P. (2003). *CyberInsecurity: The cost of monopoly*. Retrieved April 13, 2004, from http://www.ccianet.org/papers/cyberinsecurity.pdf

Moore, D., Paxson, V., Savage, S., Shannon, C., Staniford, S., & Weaver, N. (2003). Inside the Slammer worm. *IEEE Security & Privacy, 1*(4), 33–39.

Spafford, E. (1988). *The Internet worm program: An analysis (Purdue Technical Report CSD-TR-823)*. Retrieved April 22, 2004, from www.protovision.textfiles.com/100/tr823.txt

Spafford, E. (1989). Crisis and aftermath. *Communications of the ACM, 32*(6), 678–687.

Staniford, S., Paxson, V., & Weaver, N. (2002). How to own the Internet in your spare time. *Proceedings of the 11th USENIX Security Symposium* (pp. 149–162).

Thompson, K. (1984). Reflections on trusting trust. *Communications of the ACM, 27*(8), 761–763.