

The Year of the Worm

Bill Hayes 2001-08-16

The Year of the Worm

by *Bill Hayes*

last updated August 16, 2001

If the first eight months are any indication, 2001 should be remembered the "Year of the Worm". Never before have we seen so many Internet worms spreading at such a dizzying pace. In examining the Internet worms fielding thus far, two facts stand out. First, these pieces of malware are more complex than previous instances and, second, their writers have taken them beyond being proof-of-concept attacks. Some now contain payloads designed to destroy computer files and configuration information. Their complexity and lethality make this year's crop of worms place special challenges on anti-virus vendors. This article will take a brief look at some of the developments in malware, specifically in the development and proliferation of worms, thus far in 2001.

Internet Worm	Discovery Date	Comments
VBSWG series (Anna Kournikova, Homepage, Neue Tarife)	Beginning 12 February, 2001.	VisualBasic Script worms attacking Windows Outlook users. Products of script kiddie toolkit.
Magistr	12 March, 2001	Windows 32-bit worm attacking Windows users through email and network enumeration.
Sadmind/IIS	Early May, 2001	Solaris-based Worm attacking IIS hosts through well-known vulnerabilities in both systems.
SirCam	17 July, 2001	Windows 32-bit worm attacking Windows users through email and network enumeration.
Code Red	17 July, 2001	Windows 32-bit worm attacking Windows IIS web servers though indexing vulnerability

Figure 1: An overview of worms discovered in the wild in the first seven months of 2001.

Anna Kournikova: The Socially-Engineered Worm

On Feb 12, 2001 a worm with the attachment name of "AnnaKournikova.jpg.vbs" made the rounds. Capitalizing on the fame and alluring looks of international tennis star Anna Kournikova, the virus writer earned worldwide infamy. The worm quickly circled the globe infecting over 200,000 desktops. Within a few days of its introduction, a young Dutchman turned himself in to his hometown's police force after F-Secure and Excite successfully identified him.

The young man, who called himself OntheFly, claimed that he was far from being a virus writer. The young wannabe claimed he had merely used the VBSWG construction kit to write a worm as an experiment. This kit had been developed by the Argentinean virus writer (K)alamar. Other variants produced by this worm construction kit included the Homepage and Neu Tarife worms. Along with the Anna Kournikova worm, this class of malware represents the Molotov Cocktail of the virus world.

Far from being the most sophisticated coding, these worms illustrate that the compelling nature of an attachment file name can inflame a user's curiosity, compelling them to make very bad choices. In an obvious, if not necessarily ingenious, bit of social engineering, victims were enticed to ignore all the common sense precautions offered by countless virus commentators, security administrators and other security watchdogs, instead choosing to open an unsolicited attachment for the chance to feast their eyes on the Russian athlete's physical charms.

[K]alamar's VBSWG worm construction kit (versions 1.50b and 2 beta) does a workmanlike job of creating a worm. It is the first advanced worm creation kit to be written in Visual Basic. Using a GUI-based interface, the virus writer wannabe is guided through the worm creation process.

The VBSWG kit user can choose a number of configuration options. This kit allows a user to select e-mail and IRC as infection vectors. Among the additional choices a would-be worm creator is given are where the worm will start, how e-mail messages will be formatted, and what type of payload to use. The Startup dialog page gives the user the options to choose a name for the worm, the name of the virus writer credited with the creation of the worm, the worm file name and extension. This screen also assigns which folder the worm is copied during the host infection process.

New features with version 2 include the ability of a worm to send itself as html. This makes it much easier to hide the worm encapsulated in html-formatted e-mail. The creator is also given the option of joining up to a 350 Kilobyte executable (.exe) file to the worm. This could allow a backdoor or sniffer program to be installed on the host as part of the infection process. Version 2 also has anti-deletion features that allow the worm to recreate itself while in memory if its source file was deleted, or restore registry entries if they were removed while the worm was active.

While [K]alamar has stated he plans no further versions of VGSWG, features in the VBSWG virus construction kit point the way other more sophisticated kits with more construction and payload options.

Magistr: The Well-Engineered Worm

The Magistr Internet worm was supposedly developed in Malmo, Sweden and discovered in the wild on 12 March 2001. Magistr is a very complex worm with multiple infection strategies, detection countermeasures, and multiple payloads. In order to cram all of this into an easily transportable size, Magistr's virus writer chose to write the worm in Assembler. The size of the worm, about 30 Kilobytes, makes this a coding challenge.

Magistr spreads via mass e-mailings with multiple message bodies. Text may be randomly selected from text documents from the infected system, or legal-sounding courtroom phrases written in English, French, and Spanish may be included. This worm selects a 32-bit Windows program to use as its infected attachment. For Windows 98 systems, this is often sulfnbk.exe. In order to downplay the importance of the attachment, the virus writer designed Magistr to select one or more text documents from the infected system.

Its mildest payload lets Magistr scramble desktops, making users chase icons around the screen. In Windows NT or Windows 2000 PCs, its final payload lets Magistr replace the contents of files with an obscene phrase. In Windows 9x and ME boxes, this payload erases CMOS memory, flash BIOS and corrupts data on the hard drive in much the same manner as the Windows 95 CIH virus.

The complexity of design and the Magistr worm's ability to deflect conventional debugging techniques used in virus analysis produced some glaring differences of analysis. There appeared

to be some difference of opinion about the payload counters among anti-virus vendors. Commercial virus encyclopedias disagree on the payload activation date: some state the final payload would go off in two months, others claim only a month.

With the Magistr's ability to randomly mass mail selected documents from an infected machine, its sting is felt immediately in a victim organization. Sensitive documents can be distributed widely, creating potential legal and financial repercussions.

Although Magistr initially spread slowly, in a little over two months it was widespread enough to spawn the "sulfnbk.exe" virus hoax. Someone received a Magistr-infected "sulfnbk.exe" from a compromised Windows 98 PC and began a well-intentioned campaign to warn everyone that the "sulfnbk.exe" could be found in the Windows/Command folder. The sulfnbk.exe utility, which backs up long file names, is supposed to be in the Windows 98 Windows/Command folder. Many users who received the warning note wound up deleting this obscure utility, thinking it to be a worm that their AV products couldn't detect. Thus, Magistr wound up causing collateral damage as users were duped into erasing operating system files instead of recognizing the real threat.

Sadmind: The Cross-Platform Worm

The Sadmind/IIS worm appeared in May, 2001, and was thought to be connected with a rumored seven-day "war" of web site defacement between American and Chinese hackers. (The existence of such a 'cyberwar' is still highly contentious.) This Solaris-based worm propagates through a buffer overflow attack against the Sadmind daemon found in the x86 Solaris 2.6, SPARC Solaris 2.6, X86 Solaris 7.0, and SPARC Solaris 7.0 operating systems, if these systems are running versions of Solstice AdminSuite earlier than 2.3. With a powerful UNIX system subverted, the Sadmind/IIS worm proceeds to reproduce itself on another vulnerable Solaris system and at the same time deface Microsoft IIS servers through 12 variations of the web folder traversal (Unicode) vulnerability.

The worm uses a mixture of shell scripts and binaries. This code is used to randomize IP subnet searching patterns and to identify and attack vulnerable systems. This approach allows for easy modification of the worm's code. Lethal payloads could be introduced by other virus writers through code additions to the original worm's shell scripts and binary executables.

While the AV vendors acknowledged the worm through virus encyclopedia entries, it was largely

ignored. This was most likely because the worm was hosted on a UNIX platform that is not a large market for AV products.

SirCam: The Mass-Mailing worm

The SirCam worm, introduced into the wild in mid-July, 2001, incorporated features found in the Magistr and Navidad Internet worms. The virus writer used Delphi, a Windows application language, to create a sophisticated worm that sends out a barrage of e-mail from infected machines. Borrowing the concept for spawning the worm every time an application ending in .exe is started from the Navidad worm, the SirCam worm sends e-mail to a list generated from Windows address book files and from the system cache. On Windows 9x and Me machines, this cache is usually the "**Windows/Temporary Internet Files**" folder. The mailing list is stored in a file called "sc*1.dll" where the third character of the file name is a letter. Samples seen in the wild have included "sct1.dll", "scd1.dll", and "scw1.dll".

The file attachment is created by including a file found in the "**My Documents**" folder inside the infected attachment. The resulting file is named after the original file with another file extension, such as .bat, .com, .lnk, and .pif. These latter two file extensions might not be present in the default settings of older AV programs. The message body of the worm's e-mail includes text found on the host machine, or pleas for help written in either English or Spanish. These can be in plain text or found in an attachment called "**C.txt**".

SirCam has several infection schemes. While infecting a host, SirCam hides a copy of itself in the Recycle bin (C:\RECYCLED) and in the Windows system folder under different names with different keys, ensuring that the worm will run each time the system is started or a .exe program is executed. Since the default settings of older AV products do not usually allow the scanning of the Recycle Bin, this ensures that the worm can survive a casual attempt to remove it. The worm has a 1 in 33 chance of placing itself in the Windows Startup folder as "**Microsoft Internet Office.exe**" and in the Windows folder as another name. This scheme makes it harder for users to find it and delete it. In addition to spreading itself via e-mail, the worms can also enumerate the local area network, looking for open shares to infect.

The worm has a number of payload options, ranging from a 1-in-10 chance of immediate payload activation, to activation upon a specific date - October 16. When the primary payload is activated, the worm erases files in the system disk. The worm will also immediately activate its

primary payload if any of the worm's executables are renamed and then run. This might happen during a botched attempt to manually remove the worm. The worm makes another use of the Recycle Bin for a secondary payload activation that causes the file called Syrcam.sys to grow until the system runs out disk space. Because of a glitch in the original code, this payload does not activate.

The worm also has its own registry key for counters and e-mail settings at **HKEY_LOCAL_MACHINE\Software\SirCam**. Settings include how many times the worm has been activated, and the name of the worm's active file name.

Code Red: The Ghost-in-the-Machine Worm

On July 17th, researchers for [eEye Digital Security](#) discovered an unusual class of worm that existed entirely in memory. The worm attacks Windows NT and Windows 2000 Microsoft IIS web sites through a buffer overflow in the web site indexing service. The worm exhibited a cycle of rapid propagation and defacing infected web sites, followed by attack and dormancy. In honor of the quantities of soft drink downed in the analysis of this complex worm, and because the deface web pages contain the phrase "Hacked by the Chinese", the researchers decided to name it the "Code Red" worm.

Not a great deal is known about the origin of this worm. Eeye researchers Marc Maiffret and Ryan Permeh site sources in the computer underground who believe that the Code Red worm was derived from an earlier IIS worm exploiting the .htr vulnerability. This earlier worm was not greatly noticed because the vulnerability was too old.

At least two variants of the Code Redworm exist: the original, which advertises its presence by defacing web sites, and a variant that does not deface web sites. Both variants follow the same pattern of rapid propagation. According to Maiffret and Permeh, a single host can infect up to a half million IIS web sites per day.

When Code Red infects a vulnerable IIS site, the worm opens up 100 threads. Ninety-nine of the threads are used for nineteen days of scanning and infecting other IIS web sites. On the twentieth day of the month, each of the 99 threads launches a denial of service (DoS) attack against a White House web server at 198.137.240.91. The 100th thread has a limited purpose. If the code page for the operating system indicates that it uses the English language, then the thread is used to deface html output from the web server for 10 hours. The original web pages

are unmodified. The defacing occurs in memory. After that time period, the thread goes dormant.

The worm has a check for a file called "c:\notworm". If this file is present, then the worm does not infect the system. Maiffret and Ryan compared this behavior to the "Lysine deficiency" mentioned in Jurassic Park . In the novel by Michael Crichton, the lack of this enzyme elsewhere kept the dinosaurs from spreading off their island sanctuary.

The Code Red worm is a harbinger of more complex worms that exist entirely in memory. Their detection will have to be handled largely by intrusion detection systems and host-based defenses like personal firewalls such as SecureIIS and BlackICE Defender for servers, and through on-access anti-virus monitors.

The initial reaction by IIS administrators to the Code Red worm did worry Maiffret.

"I am not surprised that IIS administrators handled installed the patch," Maiffret said. "I was surprised though at the very large percentage that did not install the patch. I hope that CodeRed has worked as a wakeup call - however I doubt it - to let administrators know that patching systems is very important. Code Red could have been written to do much worse things, like truly bringing down large parts of the Internet."

The Linux-Preferred Worm

Though it hasn't gained the limelight like its Microsoft-munching counterparts, the Lion worm has made inroads into the Linux-based DNS servers. Attacking a buffer overflow existing in Bind, this worm sends system data back to a Chinese e-mail address, presumably for follow-up hacker attention. To counter this worm, someone unleashed the Cheese worm to search out vulnerable systems and patch the Bind vulnerability.

This worm looks for a root shell listening on port 1008/tcp. The Cheese worm uses a Ramen-style attack, by downloading itself from an attacking host and then starting the attack cycle. All files are downloaded into the working directory names "/tmp/.cheese". To prevent accidental re-infections, the worm checks for a file named ADL in the working directory. if it exists, the worm stops.

The worm has a limited scanning Class-B scanning pattern with the first octet ranging from 193

to 218. After infecting the host, the worm attempts to shut down port 1008 by modifying inetd.conf and then restarting inetd service.

While some may characterize the Cheese worm as a "Little Dutch Boy" attempt to plug holes in compromised system, it is not up to the chore of patching the varying number of ports left open by the Lion worm and its variants. It is incapable of patching the BIND vulnerability. All the Cheese worm really does is cover up an existing vulnerability. Far from being a blessing, the success of the Cheese worm points out the same level of inaction by Linux system administrators in patching their system vulnerabilities. The success of these worms also show that Linux has reached a critical mass where worm-based attacks can propagate widely. This can only mean further attacks against this operating system as well.

[Privacy Statement](#)

Copyright 2006, SecurityFocus