Hal Berghel

The Code Red Worm

Malicious software knows no bounds.

The concept of combining the "new soft drink flavor of the summer" with "worms" seems to suggest a non-alcoholic variation of tequila rather than an major Internet security breach. However, this past August, the Code Red worm took on an ominous significance. In this column, I'll discuss how this latest incarnation of "malware" drilled itself into cyberspace.

THE FOLLOWING FBI ALERT provides a useful measure of the potential threat of the Code Red worm to the Internet Community:

For Immediate Release: 3:00 PM (EDT) July 29, 2001
"A very real and present threat to the Internet: July 31 deadline for action.

"Summary: The Code Red worm and mutations of the worm pose a continued and serious threat to Internet users. Immediate action is required to combat this threat. Users who have deployed software that is vulnerable to the worm (Microsoft IIS Versions 4.0 and 5.0) must install … a vital security patch.

"How big is the problem? On July 19, the Code Red worm infected more than 250,000 systems in just nine hours. The worm scans the Internet, identifies vulnerable systems, and infects these systems by installing itself. Each newly installed worm joins all the others causing the rate of scanning to grow rapidly. This uncontrolled growth in scanning directly decreases the speed of the Internet and can cause sporadic but widespread outages among all types of systems. Code Red is likely to start spreading again on July 31st, 2001, 8:00 P.M. EDT and has mutated so that it may be even more dangerous. This spread has the potential to disrupt business and personal use of the Internet for applications such as e-commerce, email and entertainment."

This alert was produced jointly by Microsoft and the FBI National Infrastructure Protection Center, the Information Technology Association of America, The CERT Coordination Center, the SANS Institute, Internet Security Systems, and the Internet Security Alliance—a "Who's Who" of major agencies and organizations concerned with Internet security. But what is Code Red, and how did it happen?

A Worm by Any Other Name

Code Red began as just another piece of malicious software ("malware" in modern techno-jargon). The two most common forms of malware are viruses and worms, and combinations thereof.

Computer viruses attach themselves to otherwise "healthy" host programs from which they launch their attack and infection of computer systems. Viruses with nicknames like "Jerusalem," "Christmas," "Michelangelo," "Chernobyl," and other sundry mutations, have been widespread since the 1980s, initially spread by disk sharing, thereafter by means of digital networks. Modern

PETER HOEY

After the initial infection and incubation periods, Code Red was programmed to unleash a denial-of-service attack on the Whitehouse.gov Web site by targeting the actual Whitehouse.gov IP address.

viruses frequently appear as executable content embedded in distributed data files (email attachments, spreadsheet, and word processing macros). As of September 3, 2001, Symantec's Norton Antivirus software checked for 52,911 known viruses.

Worms, on the other hand, run as autonomous, standalone programs. Worms achieve their malevolence without need of unsuspecting host programs for either infection or propagation. Passive worms propagate with such data transmissions as email, (as in the case of the VBS/AnnaKournikova spamming worm that used Visual Basic to exploit a hole in Microsoft Outlook to replicate itself to everyone in the host computer's email address book). Melissa and the love letter worms were passive.

Active worms, on the other hand, exploit security weaknesses in networking and operating system software to aggressively gain entry into computer systems. The association of the terms "tapeworm" and "worm" with digital networks is derived from John Brunner's 1975 science fiction novel, *The Shockwave Rider*, while the initial discussion of the early experiences with worm programs was provided by John Schoch and Jon Hupp in 1982 (*Communications*, March, 1982).

Today, worm technology is so widespread that a lexicon could be (and perhaps has been) developed to describe each variety. Code Red is an active worm, as was the 1987 Morris Cornell Internet worm and the Linux Raymen worm. As of March, 2001, CNET reported that worms accounted for 80% of the invasive malware on the Internet (see news.cnet.com/news/0-1003-201-5125673-0.html).

Not surprisingly, modern malware has become hybridized. Melissa, for example, is not only a virus and a worm, but also a Trojan Horse. In addition, worms, unlike viruses, typically reside in primary memory rather than disk memory, and as such are immune to detection from most virus scanners.

Code Red also incorporated a few unique twists:

1. It propagated through TCP/IP Web port 80;
2. It identified itself by defacing English language Web sites with "Welcome to www.worm.com!—Hacked by Chinese!";
3. Self-propagation was controlled by means of a "random" IP address generator—that had a bug in it; and
4. After the initial infection and incubation periods, Code Red was programmed to unleash a denial-of-service attack on the Whitehouse.gov Web site by targeting the actual Whitehouse.gov IP address.

As it turned out, the latter were of far-reaching significance.

The Discovery of Code Red

The first indication a new worm had been unleashed on the Internet occurred on Thursday, July 12, 2001 as Ken Eichman, senior security engineer for the Chemical Abstract Services, noticed 611 attacks on the CAS Web servers by 27 different computers. By the following Saturday, Eichman noticed the number of attack sources exceeded 1,000. By Sunday, the presence of a worm was confirmed by Dshield.org, and by Monday, July 16, eEye Digital Security programmers began reverse-engineering the malicious code.

With caffeine-induced fury from the new soft drink, Code Red (hence the honorary nom de plume), the eEye team lead by Marc Maiffret, eEye's Chief Hacking Officer, determined by Tuesday, July 17 (one week after Code Red was first deployed from a university in China) the new worm exploited a security hole in Microsoft's Internet Information

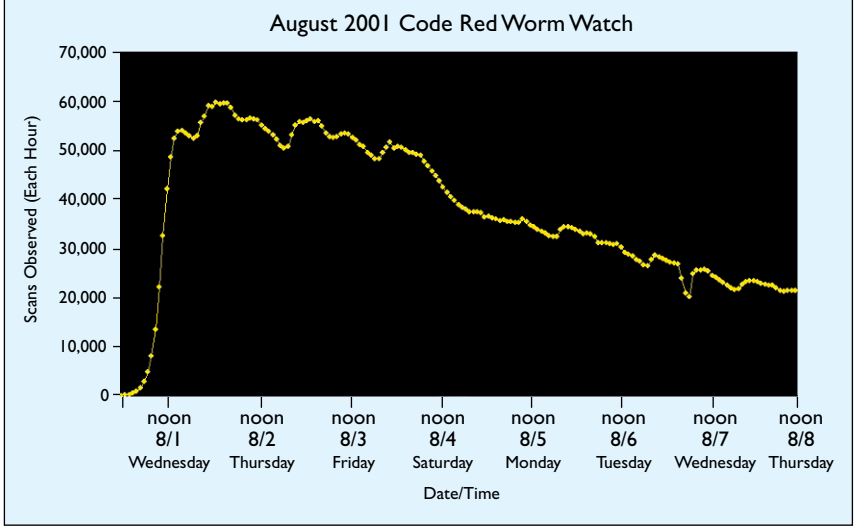
Services deployed on the millions of Microsoft Windows NT, Windows 2000, and beta versions of Windows XP servers. Technically, the security hole is called an "index-server flaw." The flaw is that versions 4 and 5 of Microsoft's Internet Information Services, or IIS for short, uses an indexing tool called an "ISAPI" filter that assigns data files to executable program environments automatically. However, this tool did not check for buffer overflow, and the undetected overflow was the access point for the Code Red worm.

By Wednesday, July 18, the eEye team determined Code Red was designed to terminate propagation and launch a denial-of-service (d-o-s) attack on the Whitehouse. gov server at midnight GMT, July 19. D-o-s attacks overwhelm Internet servers with so much useless data they are unable to function properly. The eEye team also discovered the d-o-s targeted the White House site by IP address rather than URL. This is a pivotal moment in the discovery process, for the definitive repair involved nothing more than having the Whitehouse.gov server relocated to another IP address. As predicted, the d-o-s attack began on time, but without significant result to the White House Web server. According to CNET, by the time of the d-o-s launch, each of the more than 359,000 infected computers were set to unload 400MB of useless data on the White House sever after 4.5 hours.

The end? Not quite. A brief review of the FBI alert at the beginning of the column will indicate that it is dated 10 days later. By then, a new-and-improved Code Red had appeared, some have argued because of the Code Red security advisory posted by eEye. To revisit this issue is counterproductive, but the reincarnation of an improved version of Code Red scheduled for a midnight GMT. July 31 deployment is beyond dispute. This new-and-improved version was the subject of the Code Red FBI alert. Fortunately, by the time the new Code Red triggered, the Windows patches were widely enough deployed to lesson the damage to specific targets. However, the spread of the second version was considerably wider than the original (see www.digitalisland.net/codered). The figure portrays the spread of the second Code Red worm through its first week of life. This second version eschewed defacing Web pages and put in a fix for its faulty random IP address generator.

One last point. I mentioned that "twists" (3) and (4) in the previous section had far-reaching significance. I haven't yet touched on the significance of (3). An interesting byproduct of the bug in the original Code Red was that instead of creating random paths to each infected server, Code Red infected each new server via the same path as its predecessor, thereby leaving a log of infected servers behind with each new infection. A failure of epidemic proportions in hiding one's tracks (should the perpetrator be identified, perhaps we should give him or her an honorary "F" in software design). In any event, bugs of this nature are the holy grail of Internet Forensics.

Figure. The Spread of Code Red, v2—August 1–8, 2001. Source: The Digital Island (www.digitalisland.net/codered)



Index-Server Flaws in a Nutshell

I would be remiss to not provide at least a simple overview of the technical aspect of the Code Red problem.

The essence of the vulnerability was the way Windows handled buffer overflows with its dynamic link libraries (DLLs). Incoming data with predefined filename extents (for example, .html) was automatically assigned to DLLs for interpretation and processing (such as ssinc.dll). If all went well, the input string was "understood" with the help of the DLL file, and the appropriate action was taken on the data string currently in the buffer. But what happens if the data string is too long to fit in the buffer? If the DLL is executing within the system context, any anomalies can take on life-threatening proportions for the computer system.

Code Red accesses servers through the primary Web port (#80), regardless of the host operating system. In the case of Microsoft's IIS servers, the invasion is more problematic, although invasion of other servers can cause unpredictable results as well. The worm itself was actually sent to the server as a chunk of data that follows one input buffer's worth of data in a Web query. The idea is pretty simple.

Assume you're sending data to a Perl program on some Web server somewhere. One way to accomplish this is to enclose the program name and data in a query string entered through your browser. To illustrate, the Web query string "http://www.website.net/greeting.cgi?data=25nnn" will race through port 80 on website.net, which will cause the server to execute the Perl program named "greeting.cgi" accepting "25nnn" as its input.

In order to accelerate processing, operating systems are designed to automatically associate files with certain extents with built-in programs. This happens on a workstation when one clicks on a .jpeg image file from a directory index, and Photoshop or some other program automatically launches to render the graphic. On Microsoft servers, files with extensions such as .ida and .idq are automatically handled by IIS through the ISAPI extension idq.dll, which is running within a core system program, 'inetinfo.exe'.

Therein lies the rub. Files like .ida and .idq are expected to be scripting files containing indexing information. The bug in the Windows IIS resulted from the fact that when the data assumed to be associated with an .ida file was too long for the buffer, the data that overran the buffer will cause "infected" code to be executed by the operating system. It would be as if 'greeting.cgi' were some rogue code designed to only accept two- integer input (say, "25"), and transfer control to any

Number of Internet attacks by port and frequency. Source: The SANS Institute (www.incidents.org)

Port	Number of Source IPs	Count
80	444166	22940784
137	220721	5008225
111	7554	2948379
0	49997	1377248
53	15356	993163
21	3338	708226
515	2017	285052
2049	1686	266446
6346	7900	233192
1	30389	227748
1999	1349	219405
25	5487	208526
2301	1051	152714
113	13618	133946
161	494	133930
13	13513	126009
4329	1248	100025
139	5952	85778
23	785	84351
8000	3439	70370

binary executable, or pointer hereto, that followed (such as "nnn").

The temporary solution put in place that got us through the first iteration of the Code Red worm, and others of its ilk, simply disassociated the offending DLL from the offending file extents. However, this is at most a patch unless the underlying logic has been changed.

Deja Vu All Over Again

To say that Code Red represented a serious threat is an understatement. The fact that a security hole as simple as the one described allowed the infection of hundreds of thousands of Internet computers betrays a fundamental flaw in the way we handle the standards for data exchange. This is confirmed by the fact that within the first week after the attack of the revised version of Code Red, a second-generation, Code Red II, emerged, targeting cable and DLS ISP networks. Unlike Code Red v1 and v2, Code Red II opens backdoors to infected servers through which subsequent attackers may pass (see sidebar for relevant links). We obviously haven't heard the last of Code Red.

What's more important is the economic impact of all of this derring-do. The table illustrates the volume of computer attacks broken out by the number of computers engaged in the attack and the total number of incidents for the 20 leading TCP/IP ports. These numbers are staggering. But what is more staggering are the estimates of economic impact caused.

Links to Understanding the Code Red Worm

Additional information on the Code Red alert may be found in David Becker's CNET column at news.cnet.com/news/0-1003-200-6718987.html?tag=rltdnws. More information on Marc Maiffret's discovery of IIS vulnerability, along with an extensive product line of protective software, is available on the eEye Web site (eeye.com/html/index.html).

The original June 18, 2001 security advisory explaining "buffer overflow" vulnerability from eEye is at www.eeye.com/html/press/PR20010618.html. This caused a considerable stir, as Microsoft claimed this advisory was directly responsible for the second, improved version of Code Red. For comparison, another variant from NSFOCUS is at www.nsfocus.com/ english/homepage/sa01-06.htm.

Microsoft's description of the problem of IIS 5 can be found at www.microsoft.com/technet/itsolutions/security/tools/iis5chk.asp. Detailed instructions on accessing patches to Windows environments, including links to Microsoft's own download sites, are available on Digital Island (www.digitalisland.net/codered). In addition, Digital Island includes an audio-enhanced slide presentation on Code Red by Jason Fossen of the SANS Institute (www.sans.org) that provides a good overview of the underlying technology issues.

For an overview of Code Red II, see the SANS Institute Emergency Incident Handler site (www.incidents.org/react/code_redII.php).

Michael Erbschloe's analytical approach to quantifying the costs of information attacks is explained in his new book, *Information Warfare: How to Survive Cyber Attacks* (Osborne/McGraw-Hill, 2001). C

Michael Erbschloe, vice president of research at Computer Economics, estimates the Code Red worm cost society about $2.6 billion in July and August 2001 alone. Add to that $8.7 billion for the Love Bug, $1.2 billion for Melissa, $1 billion for Explorer, another $1 billion for Sir Cam, and we're talking serious money. Erbschloe's estimates account for approximately equal losses resulting from returning the computer systems to pre-infection operating status and lost productivity. These losses are so considerable that Erbschloe has developed a rigorous model for measuring the economic impact of 10 different types of information warfare (see the sidebar).

What is wrong with this picture? Perhaps this is the time to add emphasis to courses in social issues in computing in our curriculum models while we simultaneously ratchet up the software standards of and disclosure requirements for software connected to the Internet. C

Hal Berghel (www.acm.org/~hlb) is a professor and chair of computer science at the University of Nevada at Las Vegas and a frequent contributor to the literature on cyberspace.

© 2001 ACM 0002-0782/01/1200 $5.00