Overview of Code Red
or
What is this "NNNNNNNNNNNNNNNNNNNNNNNNNNN" thing?

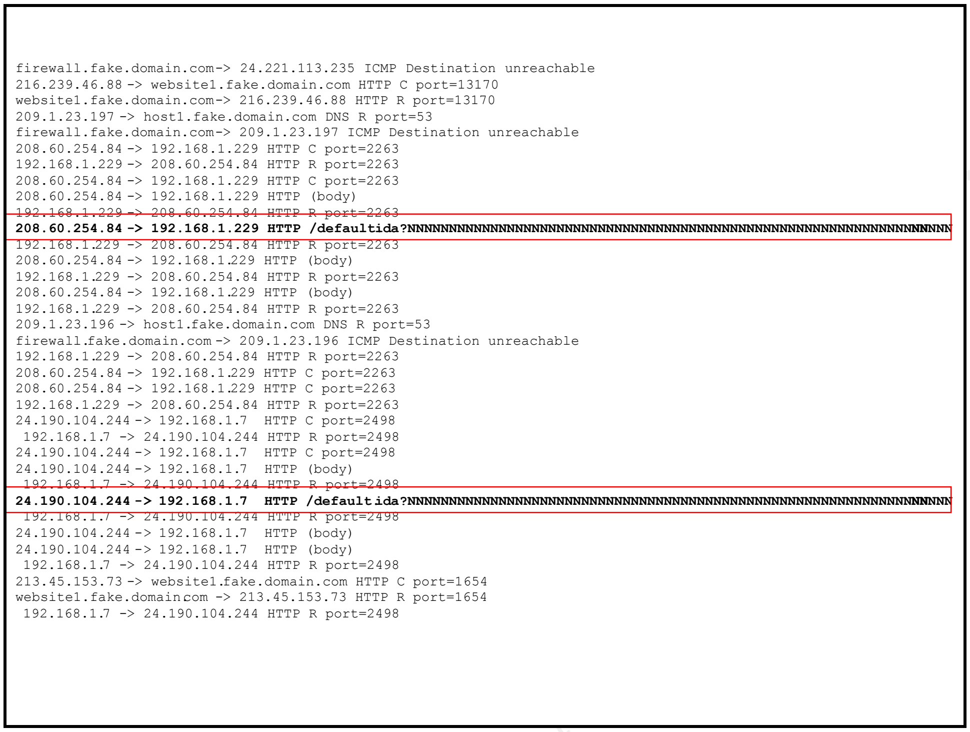
By Stephen T. Kelly
For SANS Security Essentials

Abstract: The Code Red worm has made the headlines recently. Its impact to date seems not to have been severe. This paper presents a Network Administrators perspective on gaining preliminary information as to the nature and details of Code Red as well as collecting a number of valuable references for additional information for others hoping to learn more about the worm, how it works, and buffer overflow vulnerabilities in general.

Introduction

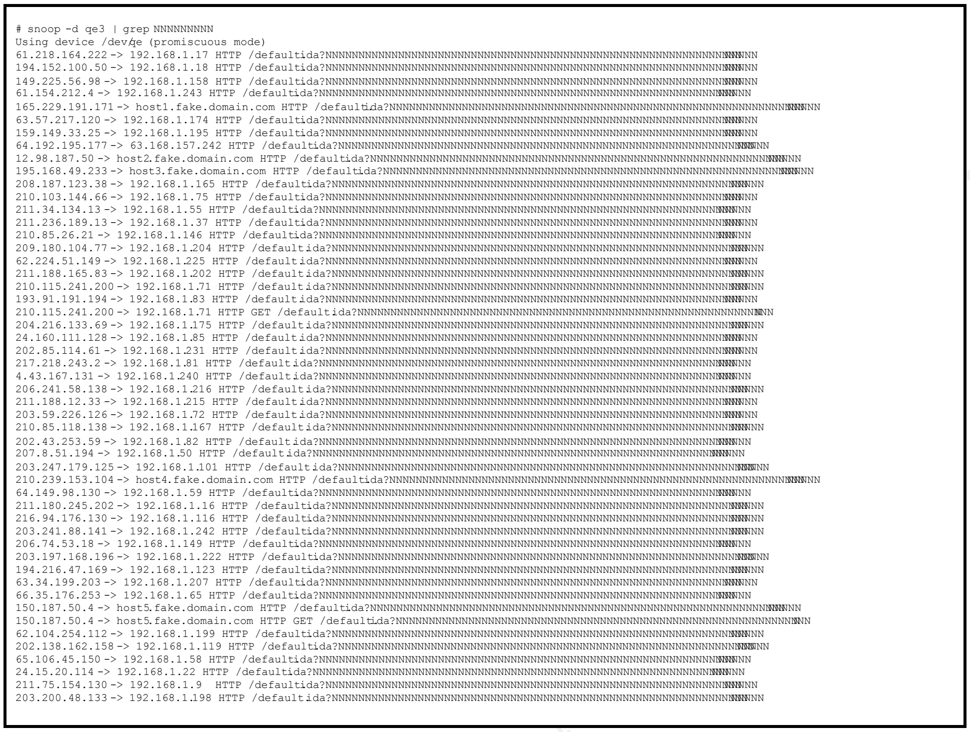
As an administrator of a computer network and the connected computer systems I was debugging a network routing issue on the 1st of August. In the course of this, I was examining the internal and external interface of our perimeter firewall trying to trace the problem. Because the computers are primarily UNIX based I wasn't watching the CERT advisories applicable to Windows platforms. I'd heard about the "Code Red" virus a couple of weeks before and knew it was targeted at Microsoft IIS Web servers. I didn't have any of these exposed to the Internet so I wasn't worried. But that wasn't what caught my eye.

Watching the snoop (a TCPdump like utility for Sun Microsystems platforms) I kept seeing similar lines



```
firewall.fake.domain.com-> 24.221.113.235 ICMP Destination unreachable
216.239.46.88 -> website1.fake.domain.com HTTP C port=13170
website1.fake.domain.com-> 216.239.46.88 HTTP R port=13170
209.1.23.197 -> host1.fake.domain.com DNS R port=53
firewall.fake.domain.com-> 209.1.23.197 ICMP Destination unreachable
208.60.254.84 -> 192.168.1.229 HTTP C port=2263
192.168.1.229 -> 208.60.254.84 HTTP R port=2263
208.60.254.84 -> 192.168.1.229 HTTP C port=2263
208.60.254.84 -> 192.168.1.229 HTTP (body)
192.168.1.229 -> 208.60.254.84 HTTP R port=2263
208.60.254.84 -> 192.168.1.229 HTTP /defaultida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
192.168.1.229 -> 208.60.254.84 HTTP R port=2263
208.60.254.84 -> 192.168.1.229 HTTP (body)
192.168.1.229 -> 208.60.254.84 HTTP R port=2263
208.60.254.84 -> 192.168.1.229 HTTP (body)
192.168.1.229 -> 208.60.254.84 HTTP R port=2263
209.1.23.196 -> host1.fake.domain.com DNS R port=53
firewall.fake.domain.com -> 209.1.23.196 ICMP Destination unreachable
192.168.1.229 -> 208.60.254.84 HTTP R port=2263
208.60.254.84 -> 192.168.1.229 HTTP C port=2263
208.60.254.84 -> 192.168.1.229 HTTP C port=2263
192.168.1.229 -> 208.60.254.84 HTTP R port=2263
24.190.104.244 -> 192.168.1.7  HTTP C port=2498
 192.168.1.7 -> 24.190.104.244 HTTP R port=2498
24.190.104.244 -> 192.168.1.7  HTTP C port=2498
24.190.104.244 -> 192.168.1.7  HTTP (body)
 192.168.1.7 -> 24.190.104.244 HTTP R port=2498
24.190.104.244 -> 192.168.1.7  HTTP /default ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
 192.168.1.7 -> 24.190.104.244 HTTP R port=2498
24.190.104.244 -> 192.168.1.7  HTTP (body)
24.190.104.244 -> 192.168.1.7  HTTP (body)
 192.168.1.7 -> 24.190.104.244 HTTP R port=2498
213.45.153.73 -> website1.fake.domain.com HTTP C port=1654
website1.fake.domaincom -> 213.45.153.73 HTTP R port=1654
 192.168.1.7 -> 24.190.104.244 HTTP R port=2498
```

I noted that they were from different source IPs and were targeting different hosts on my network. Was this some exploit being tried against my network? Was it a distributed denial of service attack? What bothered me was the number of them that I'd seen scroll by. Since I'd seen so many of them I thought I'd pipe the output of snoop to "grep" to see only this interesting traffic. Within a minute or two I'd already captured the following data.



```
# snoop -d qe3 | grep NNNNNNNNN
Using device /dev/qe (promiscuous mode)
61.218.164.222 -> 192.168.1.17 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
194.152.100.50 -> 192.168.1.18 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
149.225.56.98 -> 192.168.1.158 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
61.154.212.4 -> 192.168.1.243 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
165.229.191.171 -> host1.fake.domain.com HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
63.57.217.120 -> 192.168.1.174 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
159.149.33.25 -> 192.168.1.195 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
64.192.195.177 -> 63.168.157.242 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
12.98.187.50 -> host2.fake.domain.com HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
195.168.49.233 -> host3.fake.domain.com HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
208.187.123.38 -> 192.168.1.165 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
210.103.144.66 -> 192.168.1.75 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
211.34.134.13 -> 192.168.1.55 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
211.236.189.13 -> 192.168.1.37 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
210.85.26.21 -> 192.168.1.146 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
209.180.104.77 -> 192.168.1.204 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
62.224.51.149 -> 192.168.1.225 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
211.188.165.83 -> 192.168.1.202 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
210.115.241.200 -> 192.168.1.71 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
193.91.191.194 -> 192.168.1.83 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
210.115.241.200 -> 192.168.1.71 HTTP GET /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
204.216.133.69 -> 192.168.1.175 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
24.160.111.128 -> 192.168.1.85 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
202.85.114.61 -> 192.168.1.231 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
217.218.243.2 -> 192.168.1.81 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
4.43.167.131 -> 192.168.1.240 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
206.241.58.138 -> 192.168.1.216 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
211.188.12.33 -> 192.168.1.215 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
203.59.226.126 -> 192.168.1.72 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
210.85.118.138 -> 192.168.1.167 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
202.43.253.59 -> 192.168.1.82 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
207.8.51.194 -> 192.168.1.50 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
203.247.179.125 -> 192.168.1.101 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
210.239.153.104 -> host4.fake.domain.com HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
64.149.98.130 -> 192.168.1.59 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
211.180.245.202 -> 192.168.1.16 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
216.94.176.130 -> 192.168.1.116 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
203.241.88.141 -> 192.168.1.242 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
206.74.53.18 -> 192.168.1.149 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
203.197.168.196 -> 192.168.1.222 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
194.216.47.169 -> 192.168.1.123 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
63.34.199.203 -> 192.168.1.207 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
66.35.176.253 -> 192.168.1.65 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
150.187.50.4 -> host5.fake.domain.com HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
150.187.50.4 -> host5.fake.domain.com HTTP GET /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
62.104.254.112 -> 192.168.1.199 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
202.138.162.158 -> 192.168.1.119 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
65.106.45.150 -> 192.168.1.58 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
24.15.20.114 -> 192.168.1.22 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
211.75.154.130 -> 192.168.1.9  HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
203.200.48.133 -> 192.168.1.198 HTTP /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
```

What was I seeing that would cause this much traffic to my network? Why was it coming from so many different hosts (assuming the IP address were not spoofed)?

A quick trip to the CERT website http://www.cert.org/ and entering the string "default.ida" into the search engine lead me to http://www.cert.org/incident_notes/IN-2001-08.html. So this was "Code Red". OK – I had some Windows NT servers on my network – perhaps someone had not configured the system correctly or installed a Webserver I was not aware of. I decided to take a closer look at "Code Red".

What is a Code Red?

"Code Red" (partially named after a soda[1]) is a name that was given to an Internet worm or virus that first appeared on the Internet about 13 July 2001. According to an analysis downloaded from eEye Digital Security:

> On Friday July 13th we received packet logs and information from 2 network administrators that were experiencing large amounts of attacks targeting the recent .ida vulnerability that eEye Digital Security discovered (http://www.eeye.com/html/Research/Advisories/AD20010618.html) on June 18, 2001. After reviewing the logs sent to us we determined that in fact someone had released a worm into the Internet that was spreading rapidly through IIS web servers.

What's the difference between a Virus and a worm? According to XXXXX, "A

[1] According to eEye Digital Security, they've designated this the .ida "Code Red" worm, because "part of the worm is designed to deface web pages with the text "Hacked by Chinese" and also because code red mountain dew was the only thing that kept us awake all last night to be able to disassemble this exploit even further."

computer virus is a computer program that can spread across computer networks by making copies of itself, usually without the user's knowledge". Worms on the other hand, according to XXXXX "…are similar to viruses but do not need a carrier (like a macro or boot sector). Worms simply create exact copies of them selves and use communications between computers to spread. Many viruses, such as Kakworm (VBS/Kakworm) or Love Bug (VBS/LoveLet-A), behave like worms and use email to forward themselves to other users." So in a nutshell, a virus tends to get passed by some end-user activity and worms do the propagation themselves with email attachment viruses some sort of middle ground. I like to think of a worm as a virus with its own legs. But I wasn't doing anything to cause this attack against my network – this was a worm. But was it going to do?

Much like a biological virus this definition above doesn't describe the impact. In both worlds the impact can range from benign to terminal. Viruses and worms can spawn messages (WM97/Jerk), perform pranks (Yankee), limit access (WM97/NightShade), steal data (Troj/LovLet-A), corrupt data (XM/Compatible), delete data (Michelangelo), or even disable the hardware (Chernobyl).

Similar Problems

This wasn't the first worm on the Internet. Perhaps the most famous worm is the Morris Worm[2]. On November 2, 1988, Robert Morris, Jr., a graduate student in Computer Science at Cornell, released what is generally acknowledged as the first Internet worm. His worm exploited a bug in the debug mode of the Unix sendmail program used for delivering email, and a bug in the finger daemon fingerd, which provides information about systems and users on those systems. Once released the worm spread faster than may have been intended. The result was crashed systems and networks overloaded with traffic. It appeared that Morris tried to inform the community of the problem but the word didn't get out due to the traffic the worm was generating. By the time it was identified and the correction well known the damage was done. "Damage" in this case was caused by systems being "crashed" and networks being overloaded. It should be noted this worm had no destructive intent but that the damage was a result of the worm's rapid unchecked propagation.

Then there was Melissa that hit the Internet in late March of 1999. Melissa was a Microsoft Word 97 and Word 2000 macro virus[3]. Melissa was a virus – it didn't propagate itself. But why I mention it here, and what made Melissa both wormlike and severe was that it was propagated in email attachments. The problem being in that anyone opening and reading the email propagated the virus giving it the worm-like appearance. Remember, this was one of the first email spreading virus. At final toll CERT indicated over 100,000 hosts were affected. Again, Melissa itself didn't do anything destructive to systems.

[2] One of the most detailed sources on the Morris worm I found was http://world.std.com/~franl/worm.html if you're interested in the background of other worms.

[3] CERT provides information on Melissa at http://www.cert.org/advisories/CA-1999-04.html .

One of the most recent big worms that occurred in early May of 2000 might be the Love Letter Worm. Similar to Melissa in some ways, the "Love Letter" worm is a malicious VBScript program. In the true sense of a worm – it not only spreads by email but can actively spread by file sharing, Internet Relay Chat (IRC), and USENET News. Something different from Melissa, however was the destructive aspect (like Code Red) where files on the infected system were destroyed. Also, initial indications showed that the scope of systems affected was in excess of 500,000. Another lesson learned here was the release of variants or copycats which hit the Internet shortly after the initial attacks.

This opened the door to a whole series of virus/worms like PrettyPark, VBS,Satges.A, and BubbleBoy. It should be noted that other non-email and non-Windows based worms had been around like Lion, sadmind, or cheese worm.

So this looked like it was the newest worm. But what exactly was going on here. I was seeing a HTTP request on port 80. It looked like a normal valid HTTPD request in syntax at least; the argument was just a long string with some octal at the end. In fact if it guessed one of the IPs on my network where a valid Webserver was located, this traffic would be blindly passed by my firewall to the webserver. I had Apache servers on UNIX platforms. The GET /default.ida wouldn't do anything to my servers but merely the traffic was beginning to concern me that a DDOS attack would at the very least effect my bandwidth to the internet. I needed more information. Was I vulnerable and if I was what would the impact be?

Who's Affected & Impact

A trip back to the CERT website[4] showed that the problem was with Microsoft Windows NT 4.0 with IIS 4.0 or IIS 5.0 enabled and Index Server 2.0 installed and Windows 2000 with IIS 4.0 or IIS 5.0 enabled and Indexing services installed[5]. I didn't have any of these but what was the problem I was seeing. I wasn't seeing this traffic before the 1st. What happened then? The CERT material describes the worm as operating in three phases. From the 1st to the 19th of the month it propagates, from the 20th to 27th it attacks a specific IP with a Distributed Denial-Of-Service (DDOS) attack, and from the 27th until the 1st it is inactive. What I saw was the propagation of a new cycle of infected machines.

OK – so except for the DDOS side effect of the worm I didn't seem to be vulnerable. But what would the impact be if I had a Microsoft IIS server running? CERT further stated that some variants of the Code Red worm deface web pages supported by the compromised servers. This was one of the "payloads". Any requests to the compromised server would get "HELLO! Welcome to http://www.worm.com! Hacked

[4] The actual URL is http://www.cert.org/advisories/CA-2001-23.html .

[5] Additionally there were problems with Cisco Routers also caused by the worm but these were likely a side effect and not an intent of the worm.

By Chinese!". In attack mode, compromised servers would perform a DDOS against a defined IP. Initially one of the IP's was the White House website, www.whitehouse.gov.

The Underlying Problem

The specific problem was first raised in a discovery by eEye Digital Security on 18 June 2001. They initially released some information at their Website[6] that identified a potential security issue with Microsoft's Internet Information Services (known as IIS). Their research indicated that various versions (4.0, 5.0, and 6.0) running on various operating systems (NT 4.0, Windows 2000, and Windows XP Beta) were effected. As stated on the referenced page:

> The vulnerability lies within the code that allows a Web server to interact with Microsoft Indexing Service functionality. The vulnerable Indexing Service ISAPI filter is installed by default on all versions of IIS. The problem lies in the fact that the .ida (Indexing Service) ISAPI filter does not perform proper "bounds checking" on user inputted buffers and therefore is susceptible to a buffer overflow attack.

So what we had was a Webserver subject to a "buffer overflow attack". They went on to say that this attack could be used to gain full control of the system and execute any arbitrary code or perform any function. They also offered an explanation of how worked as well as noting it was not the first SYSTEM level IIS vulnerability they'd identified.

Code Red Detailed Infection

Saying "it's a buffer overflow issue" does little for me to explain what was happening to all those unpatched IIS webservers. Attacks are often specific to an operating system (this worm to Windows NT & 2000) , service implementation (recent BSD telnetd vulnerability), or specific application (Microsoft IIS Webserver).

I'm not neither a Windows Guru nor Assembly Programmer, much less both- but what you read in many sources[7] does little to tell you how the "buffer overflow" actually works. For this specific case eEye Digital Security explains it very well (Hey, they discovered it). You can download an example of the disassembled code as well a technical analysis. For those of us who are not experts in such matters, in nutshell here's how it works.

The attack relies on the state of the instruction stack on the machine being compromised at the time the exploit is applied. Since it is known that the attack works on certain machines running certain operating systems with a given application the stack will be fairly predictable. By attempting to insert more data into the buffer than it will handle the additional data ends up on the stack. By crafting this data to be known assembly

[6] The specific page is http://www.eeye.com/html/Research/Advisories/AD20010618.html

[7] One of the best books I've read, "Hacking Exposed" references the "buffer overflow attack" but provides no real detail on how it works.

instructions (and not junk which the processor will error on) the control of the program can be diverted from the expected execution to the execution of the code forced onto the stack. Two things are key to this. First, the knowledge as to where pointers to functions are on the stack; and second is the ability to use parts of the code already loaded (part of the operating system or other dynamically loaded libraries (DLLs) to do work such that the code required to "do the dirty work" or deliver the payload can remain small.

Is that a sufficient explanation? Probably not; so who better to consult than those who do this sort of thing for fun. Cult of the Dead Cow explains buffer overflow (specifically on Microsoft Windows Platforms no less) in a very detailed fashion[8]. Briefly, according to them, it works like this:

> When a procedure returns, it moves EBP (buffer pointer) back into ESP (stack pointer), and POP's the return address off the stack. When the above line of code (generally a read) executes with too much data it overflows the buffer, writing the contents of the read over the old value of EBP and over the return address. By overwriting the return address, you can seriously alter the course of program flow. All you have to do is change the return address to point to a memory location of your choice, and the code you want to execute will be reached when this procedure decides to 'return'. If you stuff the buffer with code bytes, you can then reroute the EIP to them on the next RET, since the stack is considered executable memory in Windows 9x/NT on the Intel architecture.

They continue to explain in detail how to utilize this behavior to perform any actions desired with a specific example.

Please refer to both the eEye Digital Security information as well as the Cult of the Dead Cow pages as they both explain in far greater detail than I could here the inner workings of exploiting a buffer overflow and how Code Red implemented the buffer overflow exploit.

Code Red Detailed Action

As for what the worm does once the overflow has been preformed and the inserted code is in place? Here's what the Code Red worm does step by step. Using the buffer overflow described above the worm initializes an environment on the compromised system. Then it sets up a 100 threads of the worm. The bulk (99 threads) are used to spread the worm (infect other web servers). The worm spreads itself by creating a sequence of random IP addresses[9] The 100th thread checks to see if its running

[8] You must read the information at http://www.cultdeadcow.com/cDc_files/cDc-351/index.html ; it's one of the clearest explanations of buffer overflow on Windows that I've found.

[9] As stated in eEye Digital Security analysis, the IPs attacked were not entirely random. The sequence seemed to be generated form the same seed value yielding the same string of "random" IP's for each attack. Every compromised system will try to compromise the same list of IPs. This "cross-traffic" can create it's own DDOS attack in compromised networks. It could have been more random but one supposition was that the

on a English (US) Windows NT/2000 system; if it's not it acts like the other 99 threads; if it is it defaces the web pages served by the compromised system. In initial versions of the worm, if the infected system is found to be a English (US) system, then the worm defaces the infected systems website as noted above[10] but in later versions of the worm the pages are not defaced.

Each thread first checks for a file c:\notworm. If it's present there is no action else it will continue. A thread next checks the clock. For 4 hours of the day the thread attacks an address sending 100 bytes of data to port 80. For the other 16 hours it tries to find and compromise new servers.

The Solution that would have rendered Code Red harmless

What I also found at CERT as well as eEye Digital Security found was a reference to a Patch available from Microsoft[11]. The patch was made available on the 18th of June 2001. Microsoft explained the problem like this:

> A security vulnerability results because idq.dll contains an unchecked buffer in a section of code that handles input URLs. An attacker who could establish a web session with a server on which idq.dll is installed could conduct a buffer overrun attack and execute code on the web server. Idq.dll runs in the System context, so exploiting the vulnerability would give the attacker complete control of the server and allow him to take any desired action on it.
>
> The buffer overrun occurs before any indexing functionality is requested. As a result, even though idq.dll is a component of Index Server/Indexing Service, the service would not need to be running in order for an attacker to exploit the vulnerability. As long as the script mapping for .idq or .ida files were present, and the attacker were able to establish a web session, he could exploit the vulnerability.

If there was a patch available for a month why was I getting all this traffic now? The patch was free. If you read anything about the incident[12] you probably got enough information to find the patch or enough to know you ought to go looking for more about this worm.

Correcting the Problem

developers of the worm watched an IP in the initial part of the series and as connections came from already compromised systems they could develop a list of compromised system

[10] The web page message only appears on the server for 10 hours.

[11] The patch is available from http://www.microsoft.com/technet/security/bulletin/MS01-033.asp

[12] A large number of sites mentioned the worm. A search into www.cnn.com or www.abcnews.com or www.msnbc.com with the keywords "code" and "red" yields a large number of articles at each site. Typical technical references included: http://news.excite.com/news/r/010802/17/news-tech-codered-dc , http://www.zdnet.com/zdnn/stories/news/0,4586,5095076,00.html ,

Correcting the problem is as simple as applying the patch. In fact the vulnerability was identified in Microsoft Security Bulletin MS01-033 first posted on 18 June 2001. The Microsoft website at http://www.microsoft.com/technet/security has several links to the patch. The machine will also have to be shut down (which will in itself stop the worm but not stop reinfection). In fact creation of the file c:\notworm will stop the original Code Red worm but this is not a suggested solution.

Code Red II and beyond

On the 4th of August I was seeing something slightly different. Now the string of "N"s was replaced with a string of "X"s. This was a variant of the original worm. Code Red II brings a whole new dimension to the worm. It exploits the same vulnerability in the Microsoft IIS software, however the "payload" now leaves behind a "back door" that allows anyone else access to the computer. Recently there seems to be yet a third major variant Code Red III although at the time of this writing CERT had no information on it.

System Administration Sidebar

In the course of dealing with the excess traffic to our network I had our ISP block all the traffic to our network except to the handful of known webservers I specified. What I didn't know at the time was that what they would implement was an "allow port 80 to server 1, server 2, server 3, server 4, and server 5" followed immediately by "Deny all port 80". Further they were just going to block all traffic, which included traffic to our network addresses both inside and outside our firewall.

I was stopped in the hall on the 8th of August by someone saying that they couldn't complete their demonstration. This was apparently because the remote site couldn't access the demo web pages. I questioned what server and what address these web pages were being served from. "From this demo machine here" I was told as he pointed to a machine that was suppose to be connected to our internal network and not running a web server. What I ultimately discovered was a NT 4.0 machine connected to our external network outside the firewall. What web server was it running? Was it patched?

As it turned out it wasn't running IIS. It also turned out that it wasn't patched. But the point of this sidebar is further identify one of the bigger problems faced when dealing with worms like Code Red: Having appropriate system administration.

Conclusion

In the final analysis I was left asking the question: So why was a vulnerability known about for months affecting me now?

In a nutshell it was vigilance. As a Network and System Administrator I fight a constant battle in getting the required resources to do the job that needs to be done. That job isn't filling the paper try or replacing a broken mouse. It's just staying on top of security issues. I have to read and investigate all the information and alerts for all the different types of systems and network hardware I support. I have to apply the necessary patches and upgrades as they are released. I have to watch the logs for suspicious or unusual activity that might be on the frontline of the battle to protect my systems and network from compromise. Apparently I was not alone. Data posted by CERT stated "Data reported to the CERT/CC indicates that the "Code Red" worm infected more than 250,000 systems in just 9 hours." That's 250,000 unpatched systems. Some number of those probably weren't even aware of the problem. Some number of those are probably still unpatched – I know I still see Code Red & Code Red II traffic.

The keys to system security such that this won't happen again?

Awareness – Keep up with the information on security issues for hardware/software/networks you administer. Subscribe to security notfiers and newsgroups. Watch CERT and the other media for information about new security threats. Be aware of the past – there will be copycat worms of the next worm that hits. Know what systems you have and what they are running.

Patches/Update – Keep your systems and applications current. Apply the necessary patches (sometimes they will cause you some extra work but it's the price of security). Keep

Vigilance – Be ever watchful. Look at your logs. Watch your traffic. Know what's going on with all your systems. Poke around sometime and look – often if you can. Stay aware. Sat up with patches.

At the Microsoft Website under the TechNet, under Security, under Tools & Checklists[13], there is a Screen Savers link as well as a link to "The Ten Immutable Laws of Security Administration". Many of them echo the three points above. Others underlie various issues that have come to light with Code Red. They're worth reading. The screen saver constantly reminds me what I have to do to be secure.

So whom did I write this for? Myself - because I now know a great deal more about worms and how they work. For other folks who want to know more about Code Red – there is some useful information collected here. And for a few managers who don't know yet – who might read this and realize that: the threat is real, I do need to spend the resources on patches and updates, and that those dollars spent on vigilance will really be well spent.

[13] http://www.microsoft.com/technet/treeview/default.asp?url=/technet/itsolutions/security/tools/tools.asp

References

Abreu, Elinor Mills, and Zabarenko, Deborah. “Code Red Worm Attack Blunted, Internet Watchers Say”. Excite News. Reuters Limited. August 2, 2001. URL: http://news.excite.com/news/r/010802/17/news-tech-codered-dc .

CERT Coordination Center. CERT Advisory CA-2000-04 Love Letter Worm Web Page. URL: http://www.cert.org/advisories/CA-2000-04.html .

CERT Coordination Center. CERT Advisory CA-2001-19 ‘Code Red’ Worm exploiting buffer overflow in IIS Indexing Service DLL Web Page. URL: http://www.cert.org/advisories/CA-2001-19.html

CERT Coordination Center. CERT Advisory IN-2001-09 Web Page. URL: CERT Incident Note IN-2001-09: "Code Red II:" Another Worm Exploiting Buffer Overflow In IIS Indexing Service DLL

DilDog. Cult of the Dead Cow. “The Tao of Windows Buffer Overflow”. URL: http://www.cultdeadcow.com/cDc_files/cDc-351/index.html .

eEye Digital Security. Downloaded Analysis, Commented disassembly, Full IDA database, and binary of the worm. URL: http://www.eeye.com/html/advisories/codered.zip.

eEye Digital Security. “.ida ‘Code Red’ worm” URL: http://www.eeye.com/html/Research/Advisories/AL20010717.html .

Excite News. Reuters Limited Article “’Code Red’ Internet Worm Could Re-Emerge, Say Officials”.July 30, 2001. URL: http://news.excite.com/news/r/010730/00/news-tech-codered-dc .

Excite News. Reuters Limited Article “Code Red Hits Servers, Seen Remaining a Menace ”. August 1, 2001. URL: http://news.excite.com/news/r/010801/22/news-tech-codered-dc .

Junnarkar, Sandeep, and Fried, Ian.“News: Code Red II: A double whammy” ZD Net News. Reuters Limited Article.August 6, 2001. URL: http://netscape.zdnet.com/zdnn/stories/news/0,4586,5095260,00.html?chkpt=zdnnp1tp02 .

Mcclure, Stuart, and Scambray, Joel, and Kurtz George. Hacking Exposed: Network Security Secrets & Solutions. Copyright 1999 McGraw-Hill.

Microsoft Corporation. Microsoft Security Bulletin MS01-033. URL: http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS

01-033.asp .

Sophos Inc..”Computer Viruses Demystified”.

Zabarenko, Deborah. “Code Red Internet Worm Disturbs Pentagon Networks” Excite News. Reuters Limited August 1, 2001. URL: http://news.excite.com/news/r/010801/15/news-tech-codered-dc .

ZD Net News. Reuters Limited Article “Code Red worm wriggles into the Pentagon”.August 1, 2001. URL: http://www.zdnet.com/zdnn/stories/news/0,4586,5095076,00.html .

“Zen and the Art of the Internet”. URL: http://sunland.gsfc.nasa.gov/info/guide/The_Internet_Worm.html .