Windows Remote Buffer Overflow Vulnerability and the Code Red Worm

Jeremy Baca

Sans Security Essentials
GSEC Practical Assignment Version 1.2d

September 10, 2001

Windows Remote Buffer Overflow Vulnerability and the Code Red Worm.

By: Jeremy D. Baca

Introduction

The following paper is about the Windows Remote Buffer Overflow Vulnerability and the Code Red Worm. I chose this topic because I wanted to inform others of this very real and dangerous threat. I wrote my paper in past tense because Code Red should be all but gone by the time my paper is posted. There will still be a few systems out there that are not patched and are still vulnerable to this worm, however the majority of systems are patched and there should be no imminent threat. To the best of my knowledge the information in my paper is correct as of September 10, 2001.

Discovery of Windows 2000 Buffer Overflow Vulnerability

On May 1, 2001, eEye Digital Security discovered the vulnerability in Windows 2000 running Internet Information Services (IIS) 5.0. Riley Hassell, an employee of eEye Digital Security, first discovered the bug when he was looking into unknown vulnerabilities within the new features of Windows 2000 IIS 5.0. He found the buffer overflow flaw within the printer Internet Server Application Programming Interface (ISAPI) filter that provides support for Internet Printing Protocol (IPP). According to eEye, "the vulnerability arises when a buffer of approximately 420 bytes is sent within the HTTP Host: header for a .printer ISAPI request." (Maiffret). Ryan Permeh, an employee of eEye Digital Security, created a proof-of-concept exploit, which would create a document that would point to eEye's web page if the server was found to be vulnerable to an attack. Microsoft was informed of this vulnerability by eEye and created a patch prior to eEye publishing their research.

This vulnerability caused Microsoft to release Security Bulletin MS01-023. Microsoft had tested and was getting ready to release Service Pack 2 when this vulnerability was discovered. Microsoft decided to delay Service Pack 2 until the patch could be incorporated. There was confusion at first as to whether Microsoft 2000 Professional was affected or if only 2000 Server was affected. Windows 2000 Server installs IIS 5.0 by default but in Windows 2000 Professional you have to choose to install it. Windows 2000 Professional was also found to contain the bug if IIS 5.0 was installed and running.

Discovery of Buffer Overflow Vulnerability in All Versions of IIS

On June 18, 2001, Riley Hassell of eEye Digital Security was again investigating Windows 2000 security. He discovered another buffer overflow vulnerability that affected not only Windows 2000 running IIS 5.0 but also Windows NT 4.0 running IIS 4.0 and Windows XP beta running IIS 6.0 beta. According to eEye, the vulnerability was discovered within the code that allows a Web Server to interact with Microsoft Indexing Service functionality. The Indexing Service ISAPI filter is installed by default in IIS 4.0 and IIS 5.0. The problem lied in the .ida (Indexing Service) ISAPI filter. It did not perform "bounds checking" on user-inputted buffers. This allowed for large packets to be sent and therefore was susceptible to a buffer overflow attack. Attackers could then leverage the vulnerability from a remote location and gain full system level access. This

could be done on any server that was running a default installation of Windows NT 4.0, Windows 2000, or Windows XP and was using Microsoft's IIS 4.0, IIS 5.0, or IIS 6.0 beta Web Server software. System-level access allows an attacker to perform any desired action such as installing and running programs, changing Web Server databases, adding, changing or deleting system files or data, and changing or replacing web pages.

Microsoft released Security Bulletin MS01-033 and a new patch that fixed this new vulnerability. The patch was a small executable and required a reboot of the system after installation. With no known attacks against this vulnerability, I think many administrators failed to install the new patch. This hole seemed to get more attention from the media than other Microsoft security holes. This prompted some system administrators including myself, to find and patch any system in their organization running any version of IIS.

Attack in the Wild

On Friday April 13, 2001 two network administrators sent logs and information to eEye that showed many attacks targeting the .ida vulnerability. From the logs, eEye discovered that someone had released a worm that targeted the .ida buffer overflow vulnerability into the wild. The logs showed five thousand IIS 5 servers attacking other IIS servers with the worm. It was determined by eEye that infected hosts were being used to attack other systems. The worm was named Code Red. Ryan Permeh said "We've designated this the .ida "Code Red" worm, first because part of the worm is designed to deface Web pages with the text "Hacked by Chinese" and second because "Code Red" Mountain Dew was the only thing that kept us awake while we disassembled this exploit." (Permeh). First reports showed that the main purpose of the Code Red worm was to launch a denial-of-service attack against www.whitehouse.gov and to deface Windows NT 4.0 and Windows 2000 systems with a web page that said "Hacked by Chinese!".

According to eEye's report there were six steps that the worm took once it had infected a system. First, the worm sets the initial worm environment on the infected system. The initial environment was in the memory of the computer and never wrote any data to the hard disk. If the computer was rebooted the worm would be destroyed and not come back unless the computer was re-infected.

Second, it set up 100 threads of the worm. The worm counted the number of active threads running on the system. If the count was less than 100 it created a new thread until it reached 100 active threads. Each thread was an exact replica of the original worm, using the same code base. When the worm reached a count of 100 active threads then control was directed back to the main program.

Third, it assigned tasks to the threads. The first 99 threads were used to spread the worm (infecting other Web servers). It spread itself by creating a sequence of random IP addresses, but the worm's list of IP addresses to attack was not altogether random. There seemed to be a static seed, either a beginning IP address that was always the same or a

beginning number. The worm used this particular number when generating new IP addresses to attack, which meant every computer infected by the worm was going to go through the same list of IP addresses, causing the worm to duplicate its effort. This feature of the worm would end up re-infecting the same systems multiple times. It also caused traffic to cross back and forth between hosts, ultimately creating a denial-of-service type effect. This denial-of-service effect was due to the amount of data being transferred between the IP addresses in the list of not so random IP addresses. If the worm had performed truly random IP generation then it would have allowed the infection of many more systems at a much faster rate. There was speculation as to why this might have been done. One person did pose an interesting idea. He said, "if the person who wrote this worm owned an IP address that was one of the first hundred or thousand to be scanned, then they could setup a "sniffer" and anytime and IP address tried to connect to port 80 on their server they would get confirmation that the IP address that connected to them was infected with the worm. With this knowledge, they would be able to create a list of the majority of systems that were infected by this worm." (Permeh)

Fourth, the 100th thread checked to see what language of IIS was running on the system. If it was on an English Windows NT or 2000 system then the worm would proceed to deface the infected system's website. The local Web Server's Web page would be changed to a message that said: "Welcome to http://www.worm.com!, Hacked By Chinese!". The hacked Web page message would stay on the Web Server for ten hours and then disappear. The message would not change again unless the system was re-infected by another copy of the worm. If the system was found to be running any language other than English the 100th worm thread was then also used to infect other systems.

Fifth, each worm thread checked for c:\notworm. If it found the file, then the worm went dormant. The system would then remain dormant and would stop any further execution. Ryan Permeh, called this the "lysine deficiency", and says it was a built in check to keep malicious code from spreading too far. If it did not find the c:\notworm file, then each thread would continue to attempt to infect other systems.

Finally, each worm thread checked the infected computer's system time. If the date was past the 20th of the month, then the threads would stop searching for systems to infect. The threads would instead attack www.whitehouse.gov. The attack consisted of the infected system sending 100k bytes of data at a rate of one byte at a time plus 40 bytes of overhead for the actual TCP/IP packet. It would send this data to port 80 of www.whitehouse.gov. This flood of data would amount to 410 megabytes of data every four and a half hours per instance of the worm. This would amount to a denial-of-service attack against www.whitehouse.gov. If the date was between the 1st and the 19th of the month, then the worm thread would not attack www.whitehouse.gov but instead it would continue to try to find and infect new un-patched web servers.

Even with the IP generation technique used, the worm was still able to spread extremely fast. It was estimated to have infected over 200 thousand unique hosts in six days. Administrators that had applied the MS01-033 path were not vulnerable to the

attack. Infected computers needed to install the patch and re-boot because Code Red stored itself in memory. On July 19, 2001, the Code Red worm was spreading so rapidly and caused such a treat that the Internet's INFOCON Alert Status was changed to yellow. The worm was set to cause a denial-of-service attack on www.whitehouse.gov on the 20th of the month but failed to do so because the site was moved to a new IP address. Government Computer News reported "With plenty of warning about the coming attack, the site's IP address was moved from 198.137.240.91 to 198.137.240.92. The worm code directed traffic to the former address. Legitimate traffic to the whitehouse.gov domain was redirected to the new address." (GCN). There was a lot of media publicity about this worm and I think this helped to get infected computers patched relatively quickly, but it was very clear that some system administrators did not keep up to date on patches. The Code Red worm also affected Cisco devices running IIS 4.0 or IIS 5.0. Cisco 600-series DSL routers would stall and crash upon receiving a probe sent by the worms. Cisco Systems also stated "The nature of the "Code Red" worm's scan of random IP addresses and the resulting sharp increase in network traffic can noticeably affect Cisco routers running Cisco IOS software, depending on the device, its current configuration, and the topology of the network." (Cisco).

Early reports showed that the worm would come back on the 1st of the month. Further evaluation showed that the rumor was false. The worm actually went dormant after the 27th of the month and would not wake back up, although the computer could still be re-infected with a new copy of the worm if they had not installed the patch. By this time there was a lot of publicity about the worm but there were still systems out there that were not patched. Some system administrators did not believe that they were vulnerable for a verity of reasons. The worm also spread to Windows 2000 computers running IIS on high-speed modems. This included cable modems and DSL modems on computers in people's homes. Unfortunately, the average home user does not think that they will ever become the target of an attack over the Internet.

Code Red Variants in the Wild

On July 20, 2001, Ryan Permeh wrote a submission to the NT Bug Traq mail list detailing what he called CRv2. There were two major differences between CRv1 and CRv2. First CRv2 had true IP randomness added by using the system clock as a seed and second it no longer contained the web page hack. Ryan described the differences like this:

> It has 13 or so pertinent bytes changed, adding a time based randomness factor and disabling page defacement. The code had been there all along. It had intentially (we must assume) been disabled in CRv1, then re-enabled near the end of the cycle. There has been discussion that this was a natural progression of the worm code, however, we do not believe this is the case. From analysis of CRv1, there seems to be no distinct way to shift the necessary bytes to generate CRv2. Hence, it is my belief that this is a modified worm, re-released. It has been posited that the CRv1 was a target

> acquisition mechanism, gathering data on infectable hosts to gain a high initial base for the following CRv2 infection. (NTBugTraq)

According to Incidents.org there were three variants of the worm in the wild called CRv1, CRv2a, and CRv2b. Incidents.org described the differences like this: CRv2a had no web defacement and random target selection. CRv2b had no web defacement, and optimum per thread random target selection.

Attack of Code Red II

On August 4, 2001, a new and much more destructive worm was discovered in the wild, it was named Code Red II. The new worm not only propagated three times faster it also created a back door on infected systems leaving their hard drives wide open. Russ Cooper, the Surgeon General of TruSecure Corporation and NTBugTraq Editor, described the four steps of Code Red II in an e-mail to the NTBugTraq list. First he said the worm made a copy of CMD.EXE called ROOT.EXE in the \inetpub\scripts and \program files\common files\system\msadc directories. It did this on both drives C: and D: and it didn't fail if D: didn't exist.

Second, it then ran its attack program code. It started 300 threads to infect other computers. If the system was found to be a Chinese language system it would start 600 threads to infect itself on other computers. Although this was done randomly there was a tendency to attack computers that were part of the same class A network as the infected attacker. This would cause the worm to it hit internal systems sooner rather than later. The attack code ran for 24 hours unless it was a Chinese language system, then it would run for 48 hours. It seemed to be based on clock ticks and not date.

Third, after the attack code ran it would then write out a Trojan file, Explorer.exe. The new file was 8192bytes or 7K as displayed by Windows. This was dropped from the code in the original attacking URL to the root of drive C: and D: and again it would not fail if D: didn't exist. The system was then rebooted by a forced reboot.

Finally, when the system restarted, it loaded the Trojan Explorer.exe from the root directory on the boot drive. The new Explorer.exe code then did several things, beginning with launching the real Explorer.exe so the system looked normal. Then it added an undocumented value of SFCDisable into registry key hklm\software\microsoft\windows nt\currentversion\winlogon. It was thought that this disabled Windows File Protection allowing critical files to be overwritten. Next, it created two virtual directories by changing the registry key hklm\system\currentcontrolset\services\w3svc\parameters\virtual roots. This created a "C" and "D" drive mapping. They were mapped to the root directories of the two drives and permissions were established in the virtual directory. This allowed script, read, and write access as well as setting execute permissions to scripts and executables. Finally it goes into an endless sleep loop. Russ Copper described the final result as follows:

> The end result of all of this action is to leave your box wide open to

> remote connection and total compromise. Unlike "Code Red", this worm doesn't attack any single target at any point, although its attack strength seems to be much higher (it launches 300 threads right off, although some may only launch 100), so its propagation seems much higher. (Cooper)

Code Red II only worked on Windows 2000 systems running IIS 5.0. The Trojan in Code Red II did not affect NT4 system with IIS 4.0 and NT4 systems running Personal Web Server. It had been suggested by many IT Professionals that the only true way to clean a computer that was infected was to format and rebuild the system. There was no way of telling what someone who used the back boor to access data could have changed on the system. Code Red II also caused a flooding side effect because of the way it targeted systems. Incedants.org described this effect like this:

> Because of the worm's preference to target its closest neighbors in IP space, combined with the enormous amount of scanning traffic generated by 300/600 threads running in parallel, a huge number of broadcast ARP requests will be generated by each infected host. If several machines on a local segment are infected and attempting to propagate the infection to their neighbors simultaneously, ARP broadcasts can be generated at "flooding" rates. Systems on the receiving end of the effective "ARP flood" may experience the effects of a Denial of Service attack.
> Note that the problem is amplified on ISP networks that use routers to propagate ARP requests throughout a very large customer base. Reports indicate that some cable network providers, for example, propagate ARP broadcasts to a very large span of their IP space, resulting in an enormous number of requests being received by every customer. (SANS)

Code Red II delivered a powerful blow to many networked computers. It created a backdoor into the system and allowed hackers access around any security software they had in place. There was so much publicity and so many published articles by the time that Code Red II hit, that any competent server manager would have had ample opportunity to patch their systems in time. Companies that were hit by this worm should look strongly at in-depth security training for their current server managers.

Microsoft Releases MS01-044

On August 15, 2001, Microsoft released security bulletin MS01-044 and a cumulative IIS patch. This was a very good step for security. Microsoft rolled up all IIS hot fixes since NT 4.0 SP5 and Windows 2000 gold release and put them in one easy to install patch. The patch included fixes for five vulnerabilities, two privilege escalation issues and three denial-of-service issues. Microsoft recommends that anyone running NT 4.0 and IIS 4.0 first install Service Pack 6a and then install MS01-044 cumulative patch for NT 4.0. If you are running a Windows 2000 system, it is recommended that you install Service Pack 2 and then install MS01-044 cumulative patch for Windows 2000. Keeping on top of all these patches and hot fixes can be quite a job, but it is a very important part of the job. I manage 22 servers and one of the biggest responsibilities I

have is making sure our systems are fully patched and checking security logs. As software gets more complex, new ways for hackers to gain access are introduced. There will always be a battle between system administrators and hackers, but informed systems administrators should always prevail.

Lessons Learned

These worms taught system administrators to keep up with their patches. The Code Red worm, Code Red II worm, and their variants could not infect a system that had the MS01-033 patch installed. The fact that so many systems were infiltrated by these worms and are still potentially being infected shows a real need for current security training. There are many Microsoft Certified Professionals (MCP) and Microsoft Certified System Engineers (MCSE) that have no concept of computer security. In my option Microsoft should evaluate their certificate programs and add a mandatory security test. The hole in IIS 5.0 was found on May 1 and a patch was released. The hole in all versions of IIS was found on June 18 and a patch was released. Code Red did not strike until July 13. This gave even the slowest system administrators time to patch their systems before getting hit, but this was not the case. It was estimated that over three hundred thousand systems were infected. I feel that anyone who administers a system should keep up with patches and hot fixes. I would hope in the future that attacks like these, where patches were released in advance would be stopped dead in their tracks.

Works Cited

Maiffret, Marc. "Windows 2000 IIS Remote buffer overflow vulnerability." eEye Digital Security. 01 May 2001. URL: http://www.eeye.com/html/Research/Advisories/AD20010501.html (29 July 2001)

Hassell, Riley. "All versions of Microsoft Internet Information Services Remote buffer overflow." eEye Digital Security. 18 June 2001. URL: http://www.eeye.com/html/Research/Advisories/AD20010618.html (29 July 2001)

Permeh, Ryan. ".ida "Code Red" Worm." eEye Digital Security. 17 July 2001. URL: http://www.eeye.com/html/Research/Advisories/AL20010717.html (2 August 2001)

GCN Staff. "White House sidesteps Code Red Worm" Government Computer News. 20 July 2001. URL: http://www.gcn.com/vol1_no1/daily-updates/4690-1.html (10 August 2001)

Cisco. "Cisco Security Advisory: "Code Red" Worm - Customer Impact" Version 2.2. Cisco Systems. 11 August 2001. URL: http://www.cisco.com/warp/public/707/cisco-code-red-worm-pub.shtml (20 August 2001)

Permeh, Ryan. "Code Red: the next generation" NT Bug Traq Mail List. 20 July 2001. URL: http://www.ntbugtraq.com/default.asp?pid=36&sid=1&A2=ind0107&L=ntbugtraq&F=P&S=&P=2797 (20 July 2001)

Cooper, Russ. "Re: Alert: New version of Code Red, XXXX" NT Bug Traq Mail List. 5 August 2001. URL: http://www.ntbugtraq.com/default.asp?pid=36&sid=1&A2=ind0108&L=ntbugtraq&F=P&S=&P=279 (22 July 2001)

SANS Staff. "Code Red (II)" Version 0.7. SANS Emergency Incident Handler. 7 August 2001. URL: http://www.incidents.org/react/code_redII.php (28 August 2001)

Last Updated: February 20th, 2019

Upcoming SANS Training

Click here to view a list of all SANS Courses

SANS Reno Tahoe 2019	Reno, NVUS	Feb 25, 2019 - Mar 02, 2019	Live Event
SANS Brussels February 2019	Brussels, BE	Feb 25, 2019 - Mar 02, 2019	Live Event
Open-Source Intelligence Summit & Training 2019	Alexandria, VAUS	Feb 25, 2019 - Mar 03, 2019	Live Event
SANS Baltimore Spring 2019	Baltimore, MDUS	Mar 02, 2019 - Mar 09, 2019	Live Event
SANS Training at RSA Conference 2019	San Francisco, CAUS	Mar 03, 2019 - Mar 04, 2019	Live Event
SANS Secure India 2019	Bangalore, IN	Mar 04, 2019 - Mar 09, 2019	Live Event
SANS London March 2019	London, GB	Mar 11, 2019 - Mar 16, 2019	Live Event
SANS St. Louis 2019	St. Louis, MOUS	Mar 11, 2019 - Mar 16, 2019	Live Event
SANS Secure Singapore 2019	Singapore, SG	Mar 11, 2019 - Mar 23, 2019	Live Event
SANS San Francisco Spring 2019	San Francisco, CAUS	Mar 11, 2019 - Mar 16, 2019	Live Event
SANS Secure Canberra 2019	Canberra, AU	Mar 18, 2019 - Mar 29, 2019	Live Event
ICS Security Summit & Training 2019	Orlando, FLUS	Mar 18, 2019 - Mar 25, 2019	Live Event
SANS SEC504 Paris March 2019 (in French)	Paris, FR	Mar 18, 2019 - Mar 23, 2019	Live Event
SANS Norfolk 2019	Norfolk, VAUS	Mar 18, 2019 - Mar 23, 2019	Live Event
SANS Munich March 2019	Munich, DE	Mar 18, 2019 - Mar 23, 2019	Live Event
SANS Doha March 2019	Doha, QA	Mar 23, 2019 - Mar 28, 2019	Live Event
SANS Jeddah March 2019	Jeddah, SA	Mar 23, 2019 - Mar 28, 2019	Live Event
SANS SEC560 Paris March 2019 (in French)	Paris, FR	Mar 25, 2019 - Mar 30, 2019	Live Event
SANS Madrid March 2019	Madrid, ES	Mar 25, 2019 - Mar 30, 2019	Live Event
SANS 2019	Orlando, FLUS	Apr 01, 2019 - Apr 08, 2019	Live Event
SANS Cyber Security Middle East Summit	Abu Dhabi, AE	Apr 04, 2019 - Apr 11, 2019	Live Event
SANS London April 2019	London, GB	Apr 08, 2019 - Apr 13, 2019	Live Event
Blue Team Summit & Training 2019	Louisville, KYUS	Apr 11, 2019 - Apr 18, 2019	Live Event
SANS Riyadh April 2019	Riyadh, SA	Apr 13, 2019 - Apr 18, 2019	Live Event
SANS Seattle Spring 2019	Seattle, WAUS	Apr 14, 2019 - Apr 19, 2019	Live Event
SANS Boston Spring 2019	Boston, MAUS	Apr 14, 2019 - Apr 19, 2019	Live Event
FOR498 Battlefield Forensics Beta 1	Arlington, VAUS	Apr 15, 2019 - Apr 20, 2019	Live Event
SANS FOR585 Madrid April 2019 (in Spanish)	Madrid, ES	Apr 22, 2019 - Apr 27, 2019	Live Event
SANS Northern Virginia- Alexandria 2019	Alexandria, VAUS	Apr 23, 2019 - Apr 28, 2019	Live Event
SANS Muscat April 2019	Muscat, OM	Apr 27, 2019 - May 02, 2019	Live Event
Cloud Security Summit & Training 2019	San Jose, CAUS	Apr 29, 2019 - May 06, 2019	Live Event
SANS Pen Test Austin 2019	Austin, TXUS	Apr 29, 2019 - May 04, 2019	Live Event
SANS Riyadh February 2019	OnlineSA	Feb 23, 2019 - Feb 28, 2019	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced