Global Information Assurance Certification Paper

Jeffrey Bollinger GSEC Version 1.4 Option #2 (revised)

De-Worming Nimda Without Pulling the Plug

Abstract: The Nimda worm of September 2001 spread rapidly across the Internet disabling Microsoft Internet Information Services and forcing many organizations to block all web traffic at the ingress of their networks. The ITS-Security office at the University of North Carolina at Chapel Hill, along with the UNC networking group, the UNC IT Control Center, and the UNC help desk worked together with an assortment of tools from Intrusion Detection Systems to virtual networking to eliminate Nimda from the campus network without the use of a firewall or router filtering, and with no network interruption. This paper describes the day Nimda hit and the response of my incident response team, including myself, to the situation.

Early morning September 18th, 2001: Scans to TCP port 80 trickled into campus quietly attacking and compromising computer systems running unpatched installations of Microsoft's Internet Information Services.

At the same time the East Coast woke up to find web sites defaced and router CPU loads increasing to dangerous and unstable levels. The HTTP port scans looked very similar to Code Red I and II, which had hit earlier in the year causing major damage to numerous systems across the globe and on campus. By mid-morning, sites were struggling to deal with a worm that, unlike the Code Red worms before it, attacks and spreads through multiple vectors. Spreading at epidemic rates, Nimda proliferated through open Microsoft Windows NetBIOS shares, e-mail, Code Red infected web (IIS) servers, and infected web sites (JavaScript)[1]. Organizations were reeling from the attacks and finding the infection difficult to contain and eradicate. Some state governments were forced to shut down their systems in an effort to clean up the mess. According to Jason Miller's article from the September 20, 2001 edition of the Government Computer News, "Connecticut, Idaho, Mississippi, Montana, Ohio, North Carolina and Rhode Island reported infections as early as Tuesday that forced IT managers to shut down networks and Internet and e-mail access for at least part of the next two days."[2] State governments were not the only institutions that experienced problems. Many universities faced difficulties as well. Ellen

[1] http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/isa/deploy/isanimda.asp
[2] http://www.gcn.com/vol1_no1/daily-updates/17148-1.html

Messmer and Jason Meserve mention in their October 15th, 2001 article in Network World Fusion, "The Code Red and Nimda computer worms continue to plague networks, particularly at universities, where a tradition of openness is making it hard for IT managers to stamp out this wildfire of malicious code."[3]

9:59 am: Sixteen systems on campus were identified as compromised, and were rapidly scanning internal and external hosts for open NetBIOS shares and vulnerable IIS servers.

The Control Center was able to capture some packet traces from Snort, one of our Intrusion Detection Systems (IDS), which gave me a clearer idea as to what this worm is looking for. They captured a few of the commands Nimda executes on any and every instance of an open port 80:

GET /scripts/root.exe?/c+dir
GET /MSADC/root.exe?/c+dir
GET /c/winnt/system32/cmd.exe?/c+dir
GET /d/winnt/system32/cmd.exe?/c+dir
GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir
GET /_vti_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir
GET /_mem_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir
GET
msadc/..%5c../..%5c../..%5c/..\xc1\x1c../..\xc1\x1c../..\xc1\x1c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc1\x1c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc0/../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc0\xaf../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc1\x9c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%35c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%35c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%2f../winnt/system32/cmd.exe?/c+dir

Through these packet captures I can tell that Nimda is using the "GET" HTTP verb to access and execute cmd.exe (a Windows root level prompt) by searching in the /scripts, /_vti_bin, /MSADC, /_mem_bin, and other folders that are installed in the /Inetpub directory by default on IIS web servers. This attack falls under two primary attack categories: directory traversal and Unicode exploits. Both Code Red and Nimda take advantage of IIS's inability to process Unicode characters[4] (%2f for example). According to Unicode.org (http://www.unicode.org), "Unicode enables a single software product or a single website to be targeted across multiple platforms, languages and countries

[3] http://www.nwfusion.com/archive/2001/126291_10-15-2001.html
[4] CVE-2000-0884

without re-engineering. It allows data to be transported through many different systems without corruption."[5] Nimda's attempts to move through different directories classify it as a directory traversal type attacker. The Unicode string tells the system to move up a directory from the document root (typically /Inetpub/wwwroot) into the /Inetpub/scripts directory, where Code Red and Nimda store their cmd.exe, or the equivalent root.exe. Once the worm has successfully accessed cmd.exe or root.exe, administrator privileges are given at the root level of the system. At this point, the infected system immediately begins to scan out and attack other vulnerable systems. Since there was no evidence other than what the IDS showed, and since the switches and routers were reaching very high utilization rates (both throughput and processor), my primary concern was to combat the spread by isolating and removing the infected hosts from the network, regardless of their infection vector, or how they may have been attacking.

10:06 am: Campus-wide alert issued regarding the severity of the new worm, the potential for compromise, and some initial removal instructions.

During the initial stages of the attack I rapidly searched the web for developing information about the new worm. I also checked the many security mailing lists to which I am subscribed. The UNISOG[6] list helped me gather information about the worm and confirm that other sites are seeing the problems as well. Many other campus security officials and network administrators were seeing the same problems at their campuses. The Nimda worm had the potential to affect at least 10,000 different systems on our campus, including over 300 servers. Because of the damage potential during these attacks, UNC had to organize and take immediate and effective action against this worm.

The ITS-Security office consists of a full-time security analyst, a student employee, a director, and me. We work closely with the Network Operations Center (Control Center) and the UNC networking group. I have to be able to make recommendations to these groups when problems arise. While I do not have access to their specific tools I understand their functions, and thus can quickly dispatch mandatory requests to other groups. During the major Nimda attack, the Security Office also worked with the help desk and their call center by giving the phone consultants information about the worm and having them explain to administrators why their machines did not have network connectivity. Dividing the tasks among these groups easily allows each individual group to concentrate on one or two steps of solving the problem, rather than having one or two groups, or even one or two people handle the whole situation. This was particularly important during the Nimda attacks, as this proved to be one of the largest attacks in our history. In my experience it has been nearly impossible to

[5] http://www.unicode.org/unicode/standard/WhatIsUnicode.html
[6] http://www.theorygroup.com/Archive/Unisog/

accomplish the task of successfully handling a major incident without a few small teams of incredibly talented professionals. Without the support of my team and our affiliates, we would not be able to manage major problems efficiently, due in part to the size and diversity of our institution.

10:45 am: Eighteen more systems affected.

Armed with information from mailing lists, trusted web sites, and information from the IDS, I immediately began the incident handling process. Everyone's attention was focused on removing Nimda, and protecting the campus network from further attack. I advise the student employee to put up a web page, linked from the main security website with information about the new worm. We referred people with questions about Nimda and those who had been infected to the Security News site, http://www.unc.edu/security. [7] The site was updated every time that new and verifiable information was received. Initially, thirty systems begin scanning out on port 80 across both the LAN and the Internet. Snort detected these scans and the Control Center's network monitoring tools saw high CPU utilization spikes on the border router. The Domino server also recorded high bandwidth usage coming from these hosts, though we determined that the traffic was only simple port scanning. The scanning took place at such a rapid pace that bandwidth utilization began to skyrocket. The infected systems had to be removed from the network to prevent further compromise and to maintain the network's stability. We helped create custom IDS rules to watch for this new traffic, modifying the earlier directory traversal and Unicode signatures. When a machine crossed over the new Snort signature, I could see the traffic traversing the wire and could begin to take action with the tools I already had in place.

Virtual Networking As a Solution

The network switches have the functionality to create a separate VLAN that has no inter-VLAN connectivity and no Internet connectivity. This VLAN is affectionately called the "Penalty Box", though it is also referred to as "protective custody." The Penalty Box was not created to solve security problems. It was first created to hold devices that were not functioning in an accepted manner. However, during the Nimda incident I took full advantage of this virtual networking technology to isolate infected machines within the Penalty Box, so that they could not call out beyond their VLAN. This effectively contained the infection until the administrator could get to the machine for clean up and/or forensics.

After I discovered an infected machine, I contacted the networking group who used a tool called the VLAN Manager to create a special hardware address

[7] http://www.unc.edu/security/arch_ju01.html#nimda

based VLAN that helped our network contain devices that did not need to speak on the network. The VLAN Manager is proprietary software that controls the VLAN behaviors of all the layer three switches on campus. The only exceptions are the experimental switches, which use a different and incompatible architecture. A different policy manager governs the experimental switches, though they still exist in a test environment. The production layer three switches act under two VLAN policies, Open and Secure. The Penalty Box is in a secure configuration which means devices cannot pass traffic through their secured VLAN, whereas the rest of our organization's switch fabric is set to Open (the Open VLAN). In the Open VLAN, each VLAN can communicate with each other without the need of a router. Switches learn the MAC addresses and ports of all users connected behind them and share that information with other switches by intercepting and resolving ARP requests. It is a hardware address based VLAN which works on both unicast, and non-unicast traffic. Communication within the VLAN is not allowed in the secure policy. ARP requests remain in the VLAN in a "flood state" and never get resolved. ARP flooding can be enabled (set to bi-directional), meaning machines in the secured VLAN can resolve each other's IP addresses and communicate, but ARP flooding is typically switched off unless I need to intentionally place a system in the Penalty Box to monitor the traffic passing around in the VLAN. This also gives me the ability to get packet captures (typically from Ethereal), port scans (from Nmap), and vulnerability scans (Nessus) of machines that have been isolated. It is also a good testing bed for remote detection or repair tools, which are occasionally released for trojans like Trin00 or other problems like the SQL "snake". When a hardware address is placed in the Penalty Box, that address cannot call beyond its port or switch – regardless of its location on the main network, or "Base VLAN".

The ability to remove systems from the network without having to pull the plug physically (the layer one solution) is vital to the interworkings of the networking group and my office. Every step of the incident handling process can be handled remotely except for the clean-up of systems, and there are some instances where I can clean machines for the administrators without them logging in. During Nimda the networking group created a script that would put multiple devices in the Penalty Box at once instead of the original method of entering them individually. This script not only saved lots of time during Nimda, but it also helped me in future incidents that involved multiple compromised machines.

Source Blocking

Another incredible tool that we implemented during the Nimda attacks was a switch function called "Source Blocking". The Penalty Box is a highly effective tool for isolating compromised machines; however, with that approach there is a delay because the device must manually be entered into the VLAN manager. Source Blocking is an automatic process which blocks traffic after it reaches a certain threshold of attempts to unknown or irresolvable destination

addresses. We set two thresholds to prevent legitimate requests, or a temporary problem with a device from going into the "Source Blocker" too often. Source Blocking basically puts a filter in the switches' connection table for the source/destination hardware address pair that overreached one of the thresholds. During Nimda scanning, the machine's MAC address will be matched with the MAC of the unresolved ARP requests; typically ff:ff:ff:ff:ff:ff. Once the threshold has been crossed, all broadcasts will be blocked from that source until it has either been manually removed from the filter table, or the source MAC address has been given "block immunity". Certain machines on the network need to send out large numbers of ARP requests or broadcast certain information. Some of my security systems need to have block immunity because I frequently portscan all of our Class B networks, and inevitably trigger some threshold of unresolved ARPs.

Once Source Blocking was implemented, the effects of Nimda quickly dwindled, as any infected box could not speak past its switch; this approach effectively set up multiple "penalty boxes" all around campus. Because I had tools that helped to monitor when a system goes into the Source Blocker, I was able to identify the location of the source blocked machine and the time at which it was initially blocked. At this point I observed the attacks coming through the IDS, and know exactly which machines were infected because of the source block table. Using the trouble ticketing system, I could pass tickets back and forth between UNC networking and my office. Thanks to an existing list containing departmental subnet administrator contact information, I helped to set up a triage and began calling the affected administrators, to notify them of which systems in their respective subnets are getting hacked, and penalty boxed. At the same time I helped develop a Web page with removal and clean-up instructions for the administrators to follow. When I received calls and e-mails from the administrators saying their machines were clean, I updated the trouble tickets and assigned them back to the networking group who subsequently removed them from the Penalty Box with my permission. After the machines were removed from the Penalty Box, I carefully monitored the outgoing traffic on the IDS to see if a host that the administrator claimed was clean continued to scan out on port 80.

12:06pm: Hourly updates to the campus began.

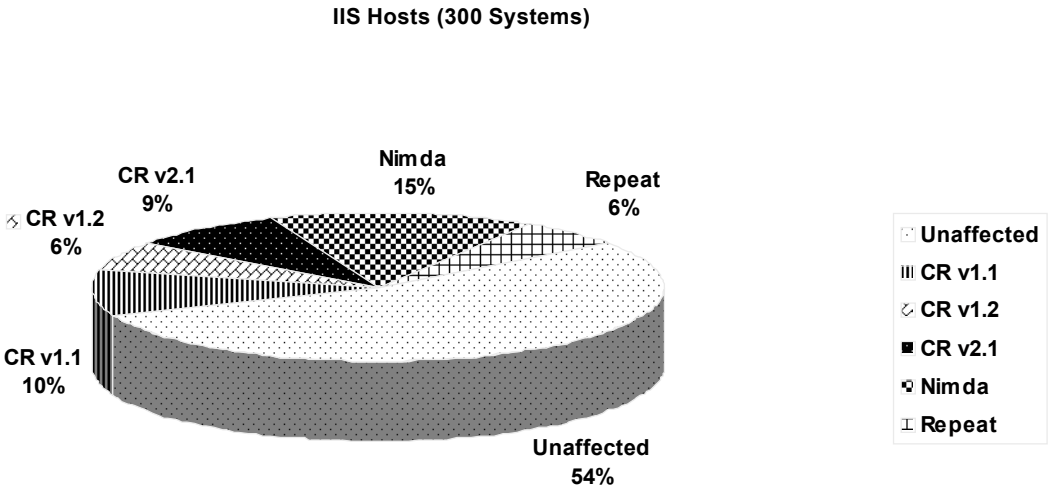
Once the majority of infected systems on campus were under control, I took advantage of a very large internal mailing list, to which most subnet administrators, and other system administrators belong. Every time new information was uncovered about Nimda regarding the clean up or its effects, we e-mailed the list informing all those who could potentially be affected. The ability to post vital information to such a large group of people substantially lowered the chances of infection. It also lowered the call volume to both the Security Office and the help desk's call center. We instructed the help desk to issue a recorded statement at the beginning of every call that describes the

problems around campus, and directs callers to the News and Alerts page to resolve any Nimda related activity.

1:26 pm: Information about removal from Symantec is finally available.

I still had no information from Symantec, my primary virus/worm/trojan information source, about Nimda until late in the day after we had already contained the majority of the infections. Until they released a fix, I had to tell users how to manually remove Nimda by deleting *.eml files and removing cmd.exe or root.exe from their Inetpub directories. After Symantec released virus definitions[8], I told the affected administrators to update their systems, and run full-system scans. This of course only worked on those servers that had Norton Anti-Virus installed. The ITS-Security office in conjunction with the Software Acquisition office had previously licensed Norton Anti-Virus for both workstations and servers for use by university faculty, staff, and students; however many administrators did not have Norton installed, and were forced to either clean up the systems manually or re-install the OS.

I observed repeat infections on several systems, at which point I simply penalty boxed them again, until the systems could be verified as clean. What was more worrisome was the fact that some of the Nimda infected machines had been previously infected with Code Red earlier in the year. Based on information gathered from the trouble ticketing system, I saw how many systems did not get sufficiently locked down after their Code Red infections. The chart below shows how many repeat offenders showed up after Nimda. The results were taken from the approximate 300 IIS systems that were running on campus at the beginning of the Nimda attacks. Fortunately, 54% of all the IIS servers on campus remained unaffected by Code Red and by Nimda, though there were at least 6% that were infected by both.



The fact that 6% of the 300 IIS hosts on campus were infected by two separate but similar attacks led me to believe that the administrators responsible for the systems had not recognized the importance of hardening their servers. As a result, the other Security Analyst and myself offered several classes for administrators on securing Windows 2000 and securing IIS. Since the classes, and the addition of my two Windows security documents[9] on the website, we have not seen any repeat infections from Nimda or Code Red crop up, even though UNC is continually scanned by external hosts looking for compromised machines. Nimda is still floating around the Internet attacking unpatched, and insecure systems.

The goal of the ITS-Security office is to educate the campus administrators so that their system administration follows the best practices outlined by both vendor specific recommendations and the IT Security community. It is more cost effective for both the system administrators and myself to spend time on how to harden their systems and applications than to spend time cleaning up after a major incident and dealing with lost data and down time.

Lessons Learned

After Nimda I learned several valuable lessons, which left me much more prepared for a large-scale attack. I have put our new measures to the test after more recent attacks, and have survived them with only a few cuts and bruises. New Code Red variants and other Unicode/Directory Traversal attacks will not go unnoticed as we now have solid Snort signatures that will immediately detect an intrusion from these methods. We are constantly monitoring and updating the Snort signatures to catch attacks as they occur. Unfortunately, because of our open network, our massive bandwidth, and the high concentration of multiple platform servers and workstations, the University is a prime target for attackers; particularly those interested in causing Distributed Denial of Service Attacks, or DDoSes. Because we are such a large target[10], we are often some of the first networks to see new attacks. We were the first[11] to notice the original SQL worm back in November of 2001[12] and have also seen directed attacks hours after exploit code is released. This makes it more difficult to prepare for such "Zero Day" attacks, though with the lessons learned from Nimda, we are now tightly organized to handle large incidents in the future.

Other than Code Red I, this was a huge incident that really tested my incident handling team. Since Nimda, I have consolidated our incident handling and can quickly triage wide-scale attacks momentarily in our office. We have gained recognition from the campus community and people feel more

[9] http://www.unc.edu/security/securing_windows2000.html, http://www.unc.edu/security/securing_iis.html
[10] http://news.com.com/2100-1023-236933.html?legacy=cnet
[11] http://archives.neohapsis.com/archives/ntbugtraq/2001-q4/0153.html
[12] http://archives.neohapsis.com/archives/incidents/2001-11/0102.html

comfortable being able to check our News and Alerts page. Nimda gave us the opportunity to try out Source Blocking, which has been tweaked to give maximum network performance, while still giving the networking group the granularity to troubleshoot network problems and my group the ability to catch a compromise at the earliest stages.

The Blox Monitor

Since Nimda, we have also developed a mechanism for tracking all the MAC and IP addresses of machines in the Penalty Box, and their associated trouble ticket number. The monitor is constantly refreshed every time a system is either placed into or removed from the Penalty Box. The "Blox Monitor" also has the capability to indicate that a device is source blocked. The Blox Monitor not only allows networking, the Control Center, and ITS-Security to see what systems are in the Penalty Box, but it also allows system administrators to see if one of their systems may be boxed. The administrators can locate their system by either looking for their MAC address or by their associated trouble ticket number. I always notify the administrators when I place a machine in the Penalty Box; however sometimes the administrators may not read that information until after they realize their machine is off the network. The help desk call center also utilizes the Blox Monitor when they are troubleshooting a network problem over the phone. If a user calls in and indicates that they have no connectivity, the help desk consultants can check the Blox Monitor, and the corresponding trouble ticket number, to see if the user's device may be in the Penalty Box or in the source blocker.

Since Nimda, I have helped to create a substantial honeynet with over 15 machines that allow us to actively monitor potential attack traffic towards machines that have intentional security holes. I have the ability to track the attacker from the moment they cross the border router all the way down to the system level. The honeynet rack contains several machines with different operating systems that all pass through a switch. I setup a hardened box that sniffs all the traffic coming into and out of the switch. The information gathered from sniffing the attacks helps me to suggest new Snort signatures that match the contents of potential future attacks, and also gives me an idea of what kind of new attack traffic we may be seeing. The honeynet still constantly picks up Nimda and Code Red type scanning, even though Nimda was released in September of 2001. Whereas the honeynet was once just a small project, it has grown to become a valuable part of the security infrastructure. We can now more accurately predict large-scale attacks, and from some packet contents, can even see exactly what the payload affects.

Trying to create and maintain a secure networked environment for a research-focused university with over 40,000 hosts remains a major challenge. We must operate in an environment that does not permit border firewalls. It would be nearly impossible to install effective firewalls capable of processing all the traffic that passes in and out of our university. In addition to a gigabit link to

commodity Internet, UNC is a member of Internet2, which provides a 2.4Gb/s link to the other member sites.

It is my responsibility and the responsibility of UNC networking and the ITS-Security Office to provide a safe network for academics and researchers, while minimally affecting their work. I was able to use existing tools like the Penalty Box, and source blocking to combat Nimda and the sources of the attacks, rather than having port 80 blocked at the border and trying to contain the damage done within. This allowed me to only remove the compromised machines, rather than forcing unaffected hosts to lose connectivity. While the network performance suffered because of the massive amounts of port scanning from infected hosts, both egress and ingress, the network never went down, port 80 was never blocked, and no major interruptions occurred. The combination of an efficient division of labor, the Intrusion Detection System, and virtual networking tools allowed me to successfully and effectively identify, contain, and eradicate the Nimda worm. As a result, I am now better prepared to handle other major incidents using the infrastructure and policies that we developed during and after the Nimda attacks of September 2001.

References:

Borland, John. "Universities likely to remain Net security risks." 15 Feb 2000. URL: http://news.com.com/2100-1023-236933.html?legacy=cnet

Cooper, Russ. "Alert: MS SQL Worm." 23 Nov 2001. URL: http://archives.neohapsis.com/archives/ntbugtraq/2001-q4/0153.html

Gutt, Zachary. "How ISA Server Can Be Configured to Help Prevent the Nimda Worm." URL: http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/isa/deploy/isanimda.asp

Messmer, Ellen and Jason Meserve. "Universities struggle to eliminate worms." 15 Oct 2001. URL: http://www.nwfusion.com/archive/2001/126291_10-15-

2001.html

Miller, Jason. “States are Reeling from Nimda Infections.” 20 Sep 2001. URL: http://www.gcn.com/vol1_no1/daily-updates/17148-1.html

http://www.cve.mitre.org/. 22 Jan 2001. CVE-2000-0884.

http://www.symantec.com/avcenter/venc/data/w32.nimda.a@mm.html. Updated 24 Jun 2002.

“What is Unicode?” Unicode Consortium. Updated 25 Jun 2002. URL: http://www.unicode.org/unicode/standard/WhatIsUnicode.html