



# Global Information Assurance Certification Paper

Copyright SANS Institute  
Author Retains Full Rights

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

## Interested in learning more?

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

# **GIAC Certified Incident Handler**

Practical Assignment Version 3

**Spoofin, Sniffin, Crackin ..... HACKIN.**

**George T. Hayes III**

© SANS Institute 2004, Author retains full rights.

<b>1.0 Statement of Purpose</b>	.....	<b>3</b>
<b>2.0 The Exploit</b>	.....	<b>4</b>
2.1 The Exploit Name		4
2.2 The Operating Systems Affected		4
2.3 Protocols/Services/Applications Affected		4
2.4 Variants of the Attack		5
2.5 Description of the Vulnerability Exploited		6
2.6 Signatures of the Attack		18
<b>3.0 The Environment</b>	.....	<b>22</b>
3.1 The Victim's PC		22
3.2 The Source Network		22
3.3 The Target Network		23
3.4 Network Diagrams		23
<b>4.0 Stages of the Attack</b>	.....	<b>25</b>
4.1 Reconnaissance		25
4.2 Scanning		25
4.3 Exploiting the System		27
4.4 Keeping Access		32
4.5 Covering the Tracks		32
<b>5.0 The Incident Handling Process</b>	.....	<b>32</b>
5.1 Preparation		32
5.2 Identification		33
5.3 Containment		39
5.4 Eradication and Recovery		40
5.5 Lessons Learned		40
<b>6.0 References and Links</b>	.....	<b>44</b>

## 1.0 Statement of Purpose

This paper will demonstrate an insider attack using a variety of tools designed to circumvent the defenses of a well administrated network. Throughout the exercise we will examine the use of the tools as well as the underlying mechanics behind each stage of the attack. In this scenario, the attacker has learned that the in-house accounting system keeps a database of all salaries. The salaries are sent electronically to an out-sourced payroll company. A salary increase of 10% or less per year, when entered into the database, will go unverified and forwarded on to the payroll company. Only two people can change a salary in the database, the Chief Finance Officer and the Human Resources Director. The attacker intends to compromise the Human Resource Director's computer and give himself a 10% salary increase under the identity of the victim.

This attack will involve a combination of exploits in order to achieve success. The attacker will make use of the following tools:

[Cain and Abel](#) will be used to ARP poison the switch and the victim, sniff the LAN for Windows passwords and then crack those windows passwords.

[The Beast](#) is a Trojan kit which will be used to obtain remote access to the victim as well as deploy a key logger to obtain the password to the accounting application.

[Stealth Tools](#) will be used to hex edit a Trojan and then bind the Trojan to a file the victim routinely uses.

*\*\* Please note that the screen captures of the tools mentioned above will show many functions that these malicious programs provide. To remain within the scope of this paper and to remain focused on this particular exploit we will only observe the functions needed. All are encouraged to further investigate these tools and their capabilities. \*\**

We will now begin dissection of each tool, regarding the functions we intend to use. Let us first examine Cain and Abel. This very powerful tool can be obtained at <http://www.oxid.it/cain.html>. The version we are currently working with is version 2.5 beta47. We are mainly concerned with the sniffer and cracker tabs of this application.

Understand that the main focus of this paper is on ARP Spoofing. I touch lightly on other exploits because it is important to realize that most system compromises require more than a single exploit to successfully gain access to network resources.

## 2.0 The Exploit

### 2.1 The Exploit Name

The name of the exploit addressed by this document is Address Resolution Protocol (ARP) Cache Poisoning also referred to as ARP Spoofing. This exploit is not new and has been around for quite some time, yet it still affects most modern devices and operating systems to date. This vulnerability has been assigned a candidate number, CAN-1999-0667, by the Common Vulnerabilities and Exposures Advisory Council. The CVE website can be found using this link: <http://www.cve.mitre.org>. It is a good resource for researching past and present vulnerabilities.

### 2.2 The Operating Systems Affected

Understand that ARP is a key component used in network communications. ARP is necessary for your network to work properly. Section 2.5 will explain how ARP is exploited in greater detail. Below is a list of SOME of the operating systems and devices affected by this exploit. As you read on you will come to realize that just about any device or operating system which implements ARP will have a good possibility of being vulnerable to this exploit. The application of an ARP attack is broad. However, the fundamentals are the same. The attacker's goal is to pretend to be someone else and gather information that other devices on the network will mistakenly give him. The attack illustrated in this document will affect, but is not limited to, the operating systems and devices listed below:

Microsoft Windows 95/98  
Microsoft Windows NT 4.0 Workstation (including all service packs)  
Microsoft Windows NT 4.0 Servers (including all service packs)  
Microsoft Windows 2000 Professional (including all service packs)  
Microsoft Windows 2000 Servers (including all service packs)  
Netgear WRG614 DSL Router  
Linksys 10/100 switches

### 2.3 Protocols/Services/Applications Affected

The Protocols affected by this attack are the Address Resolution Protocol (ARP) and the Transmission Control Protocol/Internet Protocol (TCP/IP). ARP is the first real interaction between a packet and a Network Interface Card (NIC). Each NIC has, what is known as, a Media Access Control address, commonly referred to as a MAC address. The MAC address is a hexadecimal code, much like a serial number, that is assigned to the device by its manufacturer. Every NIC ever created in the world is supposed to have a unique MAC address. The function of ARP is to correlate each MAC address to the computers IP address. This in turn, assists in the communication process between computers and devices on the network. If you understand the Domain Name System (DNS) then you understand ARP. DNS takes a computer name and maps it to an IP address xxx.xxx.xxx.xxxx. Well ARP does the same thing, only it maps a MAC address xx:xx:xx:xx to an IP address.

TCP/IP is the second protocol that assists in the communication between computers and devices on a network. Where a MAC address is limited to communicate only within the local networking segments, TCP/IP enhances communications by allowing computers and devices to communicate between different network segments via a gateway. ARP is the link between MAC addresses and IP addresses.

It is important to understand that a MAC address is a physical ID for the NIC, while an IP address is a virtual ID for the NIC. A MAC address is, as I mentioned before, much like a serial number that is hard coded into the network card by its manufacturer. The IP address is commonly assigned to a network interface card by the computer operating system via some form of user input. Both protocols serve the same purpose. That purpose is to uniquely identify the computer on a network for the purpose of data communications. MAC addresses and IP addresses allow data to be communicated between computers similar to the way mail is delivered to our homes using zip codes, street names and house numbers.

## 2.4 Variants of the Attack

ARP Spoofing has many variants as it affects just about any communications device using the address resolution protocol. These variants range from simple Denial of Service (DoS) attacks to network intrusion and packet sniffing. Below are a list of links which will provide additional information regarding this exploit and how it can be used. Taking the time to explore these links will show you that this exploits applications are broad. It will also show that this exploit is not limited to software attacks, but hardware compromises as well.

<a href="http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0667">http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0667</a>	CVE-1999-0667	The ARP protocol allows any host to spoof ARP replies and poison the ARP cache to conduct IP address spoofing or a denial of service
<a href="http://www.securityfocus.com/bid/1406">http://www.securityfocus.com/bid/1406</a>	bugtraqid 1406	Microsoft Windows 9x / NT 4.0 ARP Spoofing Vulnerability
<a href="http://www.securityfocus.com/bid/3460">http://www.securityfocus.com/bid/3460</a>	bugtraqid 3460	IEEE 802.11b Arp Cache Poisoning Man-in-the-Middle Vulnerability
<a href="http://www.securityfocus.com/bid/8398">http://www.securityfocus.com/bid/8398</a>	bugtraqid 8398	Cisco 7900 Series VoIP Phone ARP Spoofing Denial Of Service Vulnerability
<a href="http://www.kb.cert.org/vuls/id/399355">http://www.kb.cert.org/vuls/id/399355</a>	VU#399355	Cisco IOS and CatOS fail to properly validate ARP packets thereby overwriting device's MAC address in ARP table
<a href="http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0763">http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0763</a>	CVE-1999-0763	CVE-1999-0763 NetBSD on a multi-homed host allows ARP packets on one network to modify ARP entries on another connected network

<a href="http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0764">http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0764</a>	CVE-1999-0764	NetBSD allows ARP packets to overwrite static ARP entries
<a href="http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0895">http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0895</a>	CVE-2001-0895	Multiple Cisco networking products allow remote attackers to cause a denial of service on the local network via a series of ARP packets sent to the router's interface that contains a different MAC address for the router, which eventually causes the router to overwrite the MAC address in its ARP table.
<a href="http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0612">http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0612</a>	CAN-2000-0612	Windows 95 and Windows 98 do not properly process spoofed ARP packets, which allows remote attackers to overwrite static entries in the cache table

## 2.5 Description of the Vulnerability Exploited

In order to explain why ARP is exploitable, let me first explain a few other tools related to the exploitation outlined in this document. In this section we will explore some very important tools and concepts including: The OSI Model, Packet Sniffing, Password Cracking, Hex Editing and Trojaning. The focus is ARP. The other information is to help you understand how you can bend ARP to your will.

### 2.5.1 The Sniffer

A sniffer is also referred to as a protocol analyzer. Some companies will market their product as a protocol analyzer to avoid being categorized as a hacker tool. However a sniffer and a protocol analyzer are one in the same. Protocol Analyzers capture packets being sent across a network and save them for analysis. Programmers had originally developed sniffers with the intentions of troubleshooting network problems. These analyzers capture the packets being communicated over the network media in their rawest form. In most cases they are displayed in hexadecimal and ASCII format. **Figure 2.5.A** shows a TCP packet captured by a Network Intrusion Detection System (NIDS) called SNORT. On the left you will see a group of numbers (ex: 00 A3 2F). These numbers represent the hexadecimal format of the data portion the packet contains. On the right you will notice the ASCII representation of the packet data. There are a few more components that belong to the packet that the analyzer will show. Some other components are the **source port**, the **destination port**, the **sequence number**, **acknowledgement number** and the **code bits**. The complete breakdown of a TCP packet and all of the header information it contains are beyond the scope of this paper. For the purpose of ARP



spoofing however, we are interested in the source MAC address and the destination MAC address of a TCP packet. Note that the source MAC address and the destination MAC address are highlighted in **Figure 2.5.B**.

**Figure 2.5.A**

```

C:\WINDOWS\System32\cmd.exe

=====
02/13-18:47:51.007598 AA:EF:AA:EF:AA:EF -> 0:4:5A:62:AA:F4 type:0x800 len:0x36
192.168.254.75:3056 -> 192.168.254.100:445 TCP TTL:128 TOS:0x0 ID:8466 IpLen:20 DgmLen:40
*****R** Seq: 0x91D2DC73 Ack: 0x91D2DC73 Win: 0x0 TcpLen: 20

=====
02/13-18:47:51.007853 0:4:5A:62:AA:F4 -> AA:EF:AA:EF:AA:EF type:0x800 len:0x3E
192.168.254.100:139 -> 192.168.254.75:3057 TCP TTL:128 TOS:0x0 ID:36671 IpLen:20 DgmLen:48
***A***S* Seq: 0xEA5412BB Ack: 0x91D36CA2 Win: 0xFFFF TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

=====
02/13-18:47:51.007865 AA:EF:AA:EF:AA:EF -> 0:4:5A:62:AA:F4 type:0x800 len:0x36
192.168.254.75:3057 -> 192.168.254.100:139 TCP TTL:128 TOS:0x0 ID:8468 IpLen:20 DgmLen:40
*****R** Seq: 0x91D36CA2 Ack: 0x91D36CA2 Win: 0x0 TcpLen: 20

=====
02/13-18:47:51.965446 AA:EF:AA:EF:AA:EF -> 0:4:5A:62:AA:F4 type:0x800 len:0x50
192.168.254.75:1026 -> 192.168.254.100:53 UDP TTL:128 TOS:0x0 ID:8968 IpLen:20 DgmLen:66
Len: 38
01 2C 01 00 00 01 00 00 00 00 00 00 07 64 65 73 .....des
74 69 6E 69 08 6C 69 63 68 67 61 74 65 03 6E 65 tini.lichgate.ne
74 00 00 01 00 01 t.....

=====

```

**Figure 2.5.B**

```

Select C:\WINDOWS\System32\cmd.exe

=====
02/13-18:47:51.007598 AA:EF:AA:EF:AA:EF -> 0:4:5A:62:AA:F4 type:0x800 len:0x36
192.168.254.75:3056 -> 192.168.254.100:445 TCP TTL:128 TOS:0x0 ID:8466 IpLen:20 DgmLen:40
*****R** Seq: 0x91D2DC73 Ack: 0x91D2DC73 Win: 0x0 TcpLen: 20

=====

```

Before I begin to explain arp spoofing and sniffing, understand that networking has a layered approach to describing what takes place when a packet leaves and enters a network interface card. This is known as the OSI Model or Open Systems Interconnection reference model. The OSI model is the structure that defines communications tasks. It was adopted by the International Organization for Standardization in 1983 (ISO). **Figure 2.5.C** illustrates the model. The source port of the packet illustrated as AA:EF:AA:EF:AA:EF is the Media Access Control (MAC) address. It is the identity of the originating source or the sender of the packet. Every Network Interface Card (NIC) theoretically has a unique MAC address. This address is similar to a social security number. It identifies the NIC on a network

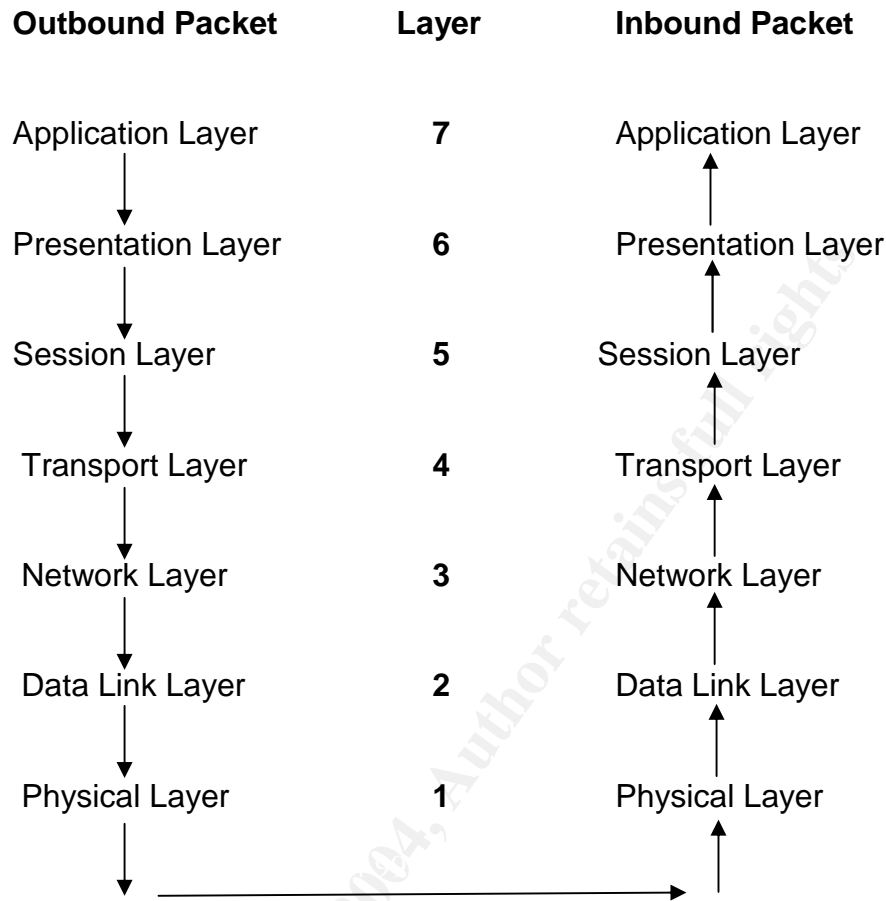


so that other network devices can communicate with it and even keep track of who it belongs to. If two NIC's have the same MAC address then communication errors will occur. It would be like two houses having the same exact mailing address. This brings us to the destination port, represented as 0:4:5A:62:AA:F4. For the sake of simplicity, when a packet traverses a network it visits each network interface card. This is what we call the physical layer of networking. The NIC senses the packet and passes it up to the data link layer. The data link layer provides error detection and control. If the packet is error free and it contains a matching MAC address, it will pass the packet along to the network layer. The data link layer compares the destination address, 0:4:5A:62:AA:F4, with its own address. If the two are a match the NIC will say "Hey this is for me!" and pass it up to the network layer. If the two do not match the NIC will say "Nope, not mine." and pass it along or drop it without any further action. Again, this is the simple version of how packets flow. It is important to know that each layer of the OSI model can only communicate with the layer above it or below it. At this point we can explain how a sniffer works. A sniffer enables a NIC to capture packets and read packet data regardless of who it belongs to and where it is going. No match on the MAC address is needed. When a NIC is passing all packets it receives above the data layer it is referred to, as being in **promiscuous mode**. Picture it as an opening of the flood gates. Instead of the network layer discarding any packets that don't contain a matching MAC address it will pass all packets to the next layer of the OSI model. The danger here is that these packets can contain sensitive information, even passwords. Some sniffers need an additional program in order to capture packets in promiscuous mode. For example, to use the windows version of SNORT you will need to also install a packet capture program called Winpcap. Snort can be downloaded for free at <http://www.snort.org> and Winpcap can be obtained for free as well at <http://winpcap.polito.it>.

The NIC is responsible for the physical and data link layers. The operating system will usually handle the network and transport layers. The application is basically responsible for the session, presentation and application layers. Remember the OSI model is a structure for data communications. It is both an abstract concept and a physical occurrence. It was established to help programmers write communications applications without having to know how the engineers designed communications equipment and visa-versa. Below is a quick example to illustrate how the OSI model is applied in **Figure 2.5.C**.

© SANS

**Figure 2.5.C**



The NIC is responsible for the physical and data link layers. The operating system will usually handle the network and transport layers. The application is basically responsible for the session, presentation and application layers. Remember the OSI model is a structure for data communications. It is both an abstract concept and a physical occurrence. It was established to help programmers write communications applications without having to know how the engineers designed communications equipment and visa-versa. Below is a quick example to illustrate how the OSI model is applied in **Figure 2.5.C**.

Outbound Application Layer - A terminal server client tries to connect to a terminal server. The user enters the IP address of the server and clicks connect.

Outbound Presentation Layer - The client application prepares the data in a way that the terminal server will recognize.

Outbound Session Layer – The terminal server client prepares the data for TCP/IP communication. This is where port 3389 will be opened for communication with the Terminal server, considering that the client is using the default port. Sniffing is also

taking place at this layer or between this layer and the inbound session layer. This is because authentication mostly occurs here and that connections are established, managed and disconnected here, by the application.

Outbound Transport Layer – The transport layer takes over the information preparation for reliable transfer of data between the end points (client and server in this case).

Outbound Network Layer – Before data is sent a TCP Handshake has occurred to make sure the connection is available. The data is sent on its way to the destination MAC address. This layer is directing packets where to go. Connections are established, maintained and terminated here by the operating system.

Outbound Data Link Layer – This layer sends blocks of data called frames. Hardware error detection and correction is maintained here.

Outbound Physical Layer – This is where the data is sent out the physical media access point. An example would be Ethernet cable.

Inbound Physical Layer – The packet is sensed and passed up to the data link layer.

Inbound Data Link Layer – The packet is checked for errors and passed up to the network layer.

Inbound Network Layer – The destination address of the packet is matched to the MAC address of the NIC, a match will cause the packet to be moved up to the transport layer.

Inbound Transport Layer – The packet is checked for errors, flow control is established.

Inbound Session Layer - The packet is received on the listening, the terminal server opens the port 3389, establishes a connection.

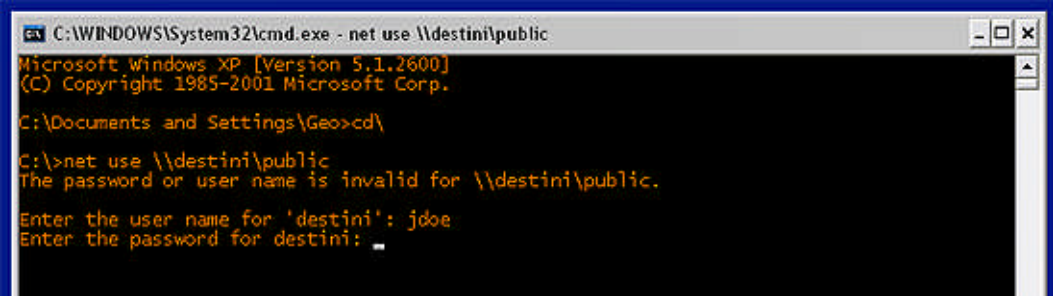
Inbound Presentation Layer – The terminal server prepares the data to be read by the application layer.

Inbound Application Layer – The terminal server process the information as a request to connect and sends a login request to the client, which then begins the process of transmission all over again, from the server side to the client side. Thank goodness electricity travels fast!

Okay, let's get back to the point. The sniffer will take all this traffic and show it to the user. If a sniffer is placed between the terminal server client and the terminal server, I will capture sensitive data and even the login account and password used to access the server.

Now that we know how the sniffer works, let's examine what we are looking for. Basically we are looking for passwords. They can be collected in many forms. The easiest is clear text passwords. They show up in the sniffer ASCII output as plain text. In addition to these passwords, an attacker will also be looking for FTP, HTTP, IMAP, POP3, SMB, Telnet, VNC, TDS, SMTP, Kerberos, well.... Just about any authentication information he/she can find. For the purpose of this document we will be looking for Server Message Block (SMB) passwords or kerberos5 pre-authentication hashes sniffed from the LAN. Let's take a look at both. Server Message Blocks are what LAN Manager and NT clients use to communicate with each other. It is important to know that SMBs are used by windows for accessing resources like shared directories remotely. The SMB protocol was originally developed by IBM, and then jointly developed by Microsoft and IBM. SMBs are a higher level protocol that can be transported over NETBEUI, NETBIOS over IPX, and NETBIOS over TCP/IP (or NBT). A password sniffer can obtain a password from an SMB when a user attempts to access a shared resource and is challenged to authenticate. **Figure 2.5.D** shows this dangerous procedure in action. As you can see a user is trying to map a drive to a shared folder called public on the computer Destini. The user is putting in their username and password. As soon as the user completes the password and hits enter, the username is sent in plain text and the password is hashed with the MD5 algorithm (based on operating system). It then traverses the network to reach the computer Destini for validation. An attacker that is ARP spoofing will obtain this information. This type of file sharing is usually only necessary when the user is accessing a resource and is not a part of the Windows domain. Server Message Blocks can be categorized into four types: Session Control, File, Printer and Message. In most cases the File and Print types will be the ones that are exploited. A method used in the past to extract password hashes, instead of waiting for them to appear was known as SMB relaying. I have not had any success with SMB relay tools in an all Windows 2000/XP environment. In fact, the reason why SMBs still exist is so Microsoft can maintain backwards compatibility with Windows operating systems pre-dating Windows 2000. If you have legacy operating systems on your network, keep a sharp eye, SMBs may be running rampant.

**Figure 2.5.D**



```
C:\WINDOWS\System32\cmd.exe - net use \\destini\public
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Geo>cd\

C:\>net use \\destini\public
The password or user name is invalid for \\destini\public.

Enter the user name for 'destini': jdoe
Enter the password for destini: _
```

Kerberos is a network authentication protocol. It is designed to provide strong authentication for client/server applications by using secret-key cryptography.

Kerberos was developed at the Massachusetts Institute of Technology and the source is available with copyrights similar to FreeBSD. Windows implemented Kerberos with the release of NT 5.0. There are only two choices for network authentication within Windows 2000 domains: Kerberos5 and Windows NT LAN Manager (NTLM). The Kerberos version 5 authentication protocol is the default for network authentication on computers with Windows 2000. The NTLM protocol was the default for network authentication in the Windows NT 4.0 operating system. It is retained in Windows 2000 for compatibility with down-level clients and servers. NTLM is also used to authenticate logons to standalone computers with Windows 2000, hence reconfirming our previous discussion about Server Message Blocks. This information on Kerberos integration with Microsoft Windows 2000 platforms was derived from and can be verified on Microsoft's website at

<http://www.microsoft.com/windows2000/techinfo/howitworks/security/kerberos.asp>

There is good reason the attacker chose Cain and Abel as one of the attack tools used in this exercise. It has the ability to sniff both authentication types mentioned above, including NTLMv2, as well as the ability to crack them offline.

### **2.5.2 The Cracker**

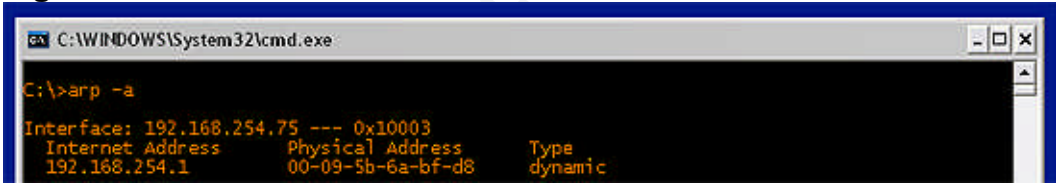
Cain is nicely laid out so that the sniffer can forward sniffed encrypted passwords to its built in password cracker. If Cain picks up a password from the LAN, the attack is usually all downhill from here. Cain has three password cracking methods. The first is a dictionary attack. This is where the application is supplied a database of words commonly used by most people. Some examples of commonly used passwords are the names of sports teams, the names of people and pets and easy to remember keyboard sequences like "qwerty" and "123456". The application will try every word in the database against the hashed password until it is either successful or the database is exhausted. The second is a brute-force attack where the application will attack the hash with every possible combination of numbers, alphabet characters and special characters (ex: !@#&%\*). The success rate of a brute-force attack is usually above 95%. The problem with brute-force attacks is that as a password increases in size and complexity, the amount of time required to crack it increases exponentially. In other words, a password, "mike", is cracked in less than 5 minutes where a password of "m1Ke123" can take significantly longer to crack. The speed at which a password is cracked by brute-force also depends on the hardware configuration of the computer performing the attack. The third method of password cracking is called cryptanalysis. Cryptanalysis works by building a large sample of encrypted information, trying to find patterns and repetition within the sample and also comparing it with other samples. A simple explanation is if A = q then all q's in the analysis can be replaced with A. Of course, with hashes and what we are trying to accomplish it is much more complex than that. A cryptanalysis attack usually requires tables in which it can reference and compare information. These tables can usually be quite large. It is tedious to setup a cryptanalysis attack. However, once you have

compiled some tables for the job, one can crack passwords significantly faster with cryptanalysis than with brute-force attacks. We are aiming for a simple brute-force attack, so this is all I will include about cryptanalysis. For more information on cryptanalysis I would recommend using an internet search engine, like Google.

### 2.5.3 ARP Spoofing

Let's explore the world of ARP spoofing and why it is significant to sniffing and cracking passwords, but first, let's explore ARP. ARP stands for Address Resolution Protocol. ARP is a key ingredient to networking. ARP is responsible for managing the relationship between IP addresses and MAC addresses, much like a Domain Name Service manages the relationship between Host Names and IP addresses. Most people believe that TCP/IP is the core of internet functionality when in fact, it is ARP. Without ARP network communications would fail to work. Why is this? Like I just mentioned, ARP is responsible for the association between IP address and MAC address. It is the bridge between the physical and the virtual addresses of a computer. Without this bridge, you simply cannot cross the gap. Remember we talked about the Data Link Layer in the OSI Model in section 1.1. ARP is the bridge to the Network Layer above. MAC addresses and IP addresses that are related are stored in an ARP table. **Figure 2.5.E** shows an example of an ARP table which is accessed by typing "arp -a" at a command prompt.

**Figure 2.5.E**



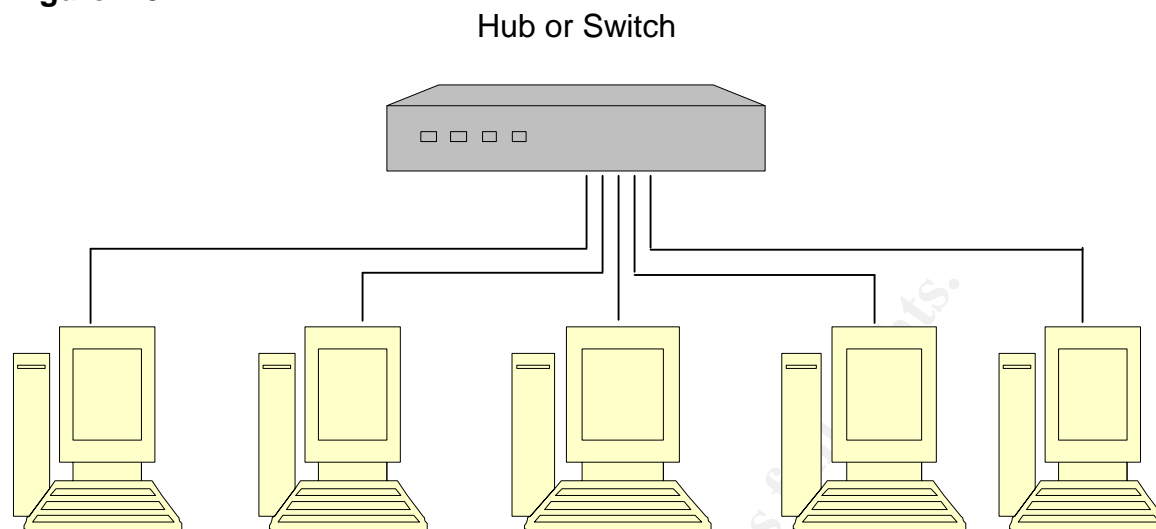
```
C:\WINDOWS\System32\cmd.exe
C:\>arp -a
Interface: 192.168.254.75 --- 0x10003
Internet Address Physical Address Type
192.168.254.1 00-09-5b-6a-bf-d8 dynamic
```

You should be able to determine the IP address and the MAC address from the information provided in this document thus far. Remember, the MAC address is an address that is hard coded, by the manufacturer, into the networking device, in hexadecimal format. Thusly, it is identified as the "Physical Address". The ARP table is telling this computer that the IP address 192.168.254.1 and the MAC address 00-09-5b-6a-bf-d8 belong to the same device. Pay special attention to the Type in this illustration. The type is dynamic, dynamic addresses are assigned when booting up and can be changed. The other type is Static and Static addresses do not be change. They are coded into the table so that they cannot be lost or forgotten. Static addresses also provide easy reference. If a person never moves or changes their address, you can always find them right? Static ARP addresses are rarely implemented because from a maintenance and administrative point of view, it is a burden to statically code every IP and MAC address into every NIC for every other NIC that exists.

So now we understand ARP. What is ARP spoofing? ARP spoofing is when malicious software alters the ARP table of one or more devices by falsely identifying itself as another device. This is also referred to as ARP poisoning. Why would an attacker want to do this? Let me illustrate using the scenario outlined in Section 1.0 of this paper. The first thing we need to understand is that when attempting to sniff passwords off a LAN we must be receiving packets that are circulated throughout the LAN. Two things need to be done in order to assure we are capturing the necessary packets. Our Network Interface card must be in promiscuous mode as discussed in section 2.1 and we must be connected to a hub with other computers. A HUB is a device that all computers will connect to in order to communicate with each other (see **Figure 2.5.F**). Unfortunately, for attackers, hubs are antiquated and are not used in secure, modern networks. Instead of hubs, switches are deployed. The difference between a hub and a switch is that traffic is controlled on a switch. A switch will know what MAC address is connected to which of its ports (via ARP table) and forward packets to that corresponding port, whereas, a hub will forward the packets to all of its ports. The switch tackles two issues with its enhanced functionality. It decreases traffic and packet collisions, which increases performance and it creates a secure environment against packet sniffing by not allowing packets to be seen by every computer, on every port of the device. Now pay close attention as I explain why the attacker must spoof an ARP address. In section 1.0 we have a victim, the The victim. Since we are on a switched network, and we want to obtain the The victims password in transit to the server, we must be able to capture all packets sent out by the The victims computer. We know that the switch will not pass these packets to us, because of the nature of how a switch operates. We must trick either the switch into thinking we are the The victim or trick the The victim into thinking we are the server and visa-versa. In order to trick the switch into passing us the information we would have to flood the switch with ARP information until it became overwhelmed with traffic and changed itself into a hub, broadcasting all packets to all ports. This is a protection mechanism the switch uses to keep data flowing. To the attacker it is an open door. However, this method will cause network performance to dramatically decrease and may also cause resources on the network to become unavailable. In turn, this may tip off an administrator that a break-in is taking place. The best way to ARP spoof is to trick the client into thinking that our MAC address belongs to the server's IP address. By doing this, the client will send information to the attacker. The attacker will read it and then forward it to the server. As you can see, the attacker is acting as a liaison between the server and the client. Some ARP spoofing tactics will actually place the attacker as a router in the network. At this point it will only be a matter of time before the attacker sniffs out a password. That is ARP spoofing in a nutshell, pretending to be someone or something that you are not by way of altering a victims ARP table. **Figure 2.5.G** shows the ARP table of the client. **Figure 2.5.H** shows that the IP addresses in the ARP tables of the client and server are actually mapped to the MAC address of the attacker by running `ipconfig /all` at the command prompt of the attacker's computer.



**Figure 2.5.F**



**Figure 2.5.G The victim's ARP Table**

```
Select C:\WINDOWS\System32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\jdoe>ARP -a

Interface: 192.168.254.33 --- 0x10003
Internet Address      Physical Address      Type
192.168.254.1         00-09-5b-6a-bf-d8    dynamic
192.168.254.100      00-0d-56-37-5e-0c    dynamic
```

**Figure 2.5.H Attacker's IP configuration**

```
Ethernet adapter Local Area Connection:

Connection-specific DNS Suffix  . : 
Description . . . . . : Broadcom 440x 10/100 Integrated Controller
Physical Address. . . . . : 00-0D-56-37-5E-0C
Dhcp Enabled. . . . . : No
IP Address. . . . . : 192.168.254.83
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.254.1
DNS Servers . . . . . : 192.168.254.100
                          151.198.0.38

C:\>
```

Notice between **Figure 2.5.G** and **2.5.H** the IP addresses are different but the Physical addresses are the same. The client ARP Table has been poisoned and the The victim will now mistakenly send packets to the attacker. The switch will now forward packets destined for the server IP Address to the attacker. Why? The physical address is the first point of contact for any networking device. Refer to the OSI Model discussed in section 2.1 of this document. The physical address will be acted upon before the IP address because of the nature of the way packet flow is established in the OSI model. The IP address is technically not acted upon until reaching the network layer of the OSI model. Remember a layer can only interact with another layer above it or below it.

### **2.5.4 The Trojan**

A Trojan is an application that bypasses a systems security by creating an open connection from inside the the victims computer. This is called a back door, and is essentially opened from the inside allowing the attacker to get in. The term comes from Homer's *Iliad*. In the Trojan War the Greeks presented the citizens of Troy a gift known as the Trojan Horse. Secretly hidden inside the horse were a group of warriors. Once the horse was wheeled into the city and passed the protective barriers, the warriors waited until nightfall, leapt out and open the gates. The army waiting outside was then able to penetrate the city. In computing today, Trojans are used to create backdoors, by opening outbound connections, allowing attackers to penetrate firewalls.

For the purpose of gaining control of our victim and achieving the goal stated in section 1.0 of this paper, we will use a Trojan kit known as The Beast v2.05. The Beast gives us a robust console for creating and customizing a Trojan server application that will allow us to key log, disable virus protection, disable the Microsoft XP Professional firewall and remotely control our victim.

### **2.5.5 Hex Editing**

Virus protection products do a good job of detecting viruses and hack tools. With the help of the internet, virus production products are able to respond to new threats very quickly. By practicing good virus protection management, an administrator, can keep these rapidly spreading threats at bay. A common technique for an attacker to exploit virus protection products is to change the physical characteristics of the malicious software. When I say "physical characteristic" of a virus or other malicious code, what I really mean is the way the file is structured in bits and bytes. This structure, like a popcorn string, has a beginning and an end. Along the string are the bits (BInary digiT'S), each bit being a 1 or 0. Eight bits make up a byte. When we view these bits and bytes with a hex editor, they are represented in hexadecimal format and sometimes ASCII format.

The hexadecimal number system is much like the "normal" decimal system with just one difference. In a decimal number system we begin at 0 and end at 9 so that we count like this: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

When we reach the 10<sup>th</sup> position ("deci") we add a digit and begin again with 0.

Ex: 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19

So here comes the difference. In a hexadecimal system, we carry the counting to 16 places instead of 10 places. In hex we count like this: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F and when we reach the 16<sup>th</sup> place we add a digit and begin again, the same way as the decimal system.

Ex: F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F

When speaking in terms of Hex we can use two methods 0xA8 or A8H. We would describe a decimal number as 17 or d17.

You can find more detailed information on hex editing at the follow link:

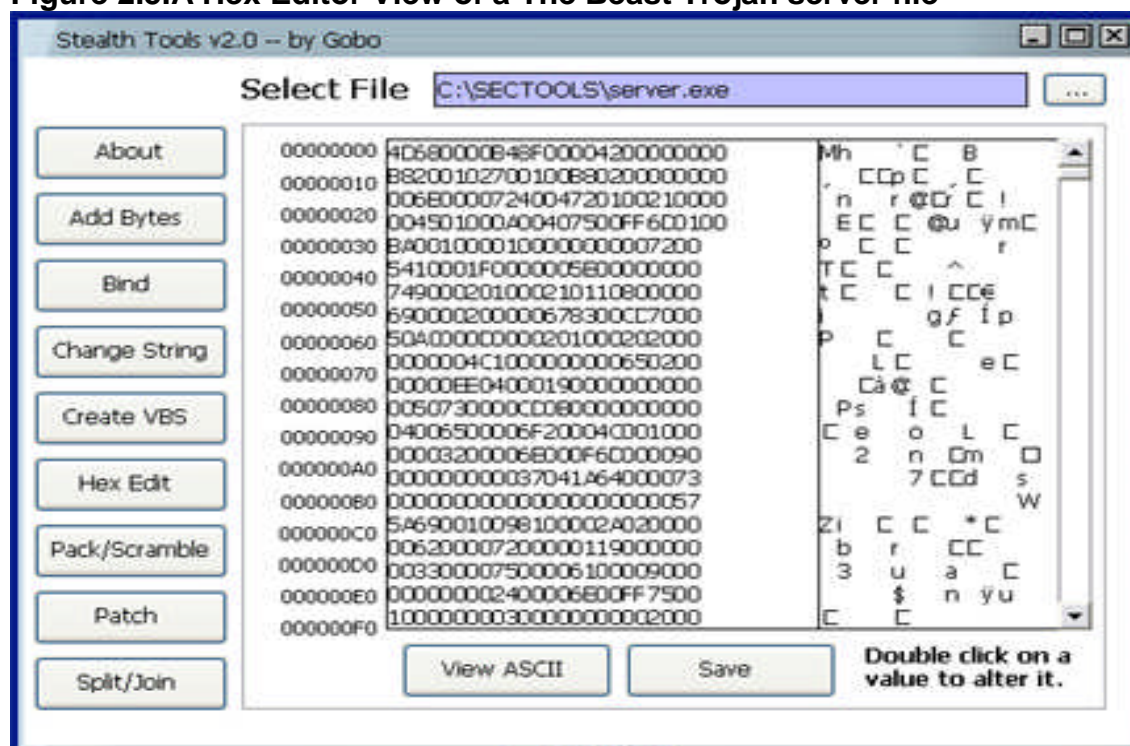
<http://www.teambg.com/iesdp/General/HexGlossary.htm>

We won't explore the depth of hex editing that this article does, because we are looking to change code, not data. Hex editing is also used to change values in database files. This is beneficial because it allows you to change values without having to decompile code or have editing rights within the database. Hex editing is a popular method for exploiting game software, where a player gives himself unlimited lives and other such advantages that other players would not have.

**Figure 2.5.A** shows an example of what we see when viewing an application with a hex editor. The physical characteristic of the file is significant because virus protection products identify malicious code by taking a portion of the popcorn string, usually 8-32 bytes long, as an identifier of the harmful code. The piece of string that is kept in the database is known as the virus signature for that virus or potentially harmful software. Finding an 8 byte signature in a 500 kilobyte file can be an arduous task. A technique known as file splitting can be used to locate the general area of a virus signature. Stealth Tools provides us with the ability to split files. Simply split up the infected file and scan each split piece until the virus protection detects a virus, then find that section of code in the original un-split file and make edits until you have successfully evaded detection. Sometimes this method doesn't always work. Another way to find a virus signature is to change the hex values to 0's a few lines at a time and keep rescanning the file until the virus is undetected. As you can see in **Figure 2.5.A**, the Stealth Tools application has a number of tricks for trying to change malicious code in a way that it goes undetected by a systems antivirus product. I have found that most forums suggest learning ASM before trying to hex edit anything. I have also found that hex editing is the best way to circumvent detection of a known virus.

© SANS Institute

**Figure 2.5.A Hex Editor View of a The Beast Trojan server file**



## 2.6 Signatures of the Attack

The attack took place over a period of 10 Days. Below is a breakdown of the attack over that time period.

<b>Day 1</b>	<b>Recon and Network scanning</b>
<b>Day 2</b>	<b>ARP Poisoning and Sniffing</b>
<b>Days 3 to 8</b>	<b>Password Cracking</b>
<b>Day 9</b>	<b>Gaining Access and Infection</b>
<b>Day 10</b>	<b>Trojan Active, Read Key Logs, Commit offense, Cover Tracks</b>

There are several windows of opportunity to detect this attack. These windows however are small. The first indicator is on day 1 when the attacker has performed a MAC Address scan. A network intrusion detection system will show a flooding of packets from the same IP address. The following capture from snort shows us the scan in progress used for this attack.

05/11-10:08:53.761151 ARP who-has 192.168.254.155 tell 192.168.254.75

05/11-10:08:53.771162 ARP who-has 192.168.254.156 tell 192.168.254.75

05/11-10:08:53.781312 ARP who-has 192.168.254.157 tell 192.168.254.75

05/11-10:08:53.791242 ARP who-has 192.168.254.158 tell 192.168.254.75  
05/11-10:08:53.801208 ARP who-has 192.168.254.159 tell 192.168.254.75  
05/11-10:08:53.811351 ARP who-has 192.168.254.160 tell 192.168.254.75  
05/11-10:08:53.821233 ARP who-has 192.168.254.161 tell 192.168.254.75  
05/11-10:08:53.831252 ARP who-has 192.168.254.162 tell 192.168.254.75  
05/11-10:08:53.841409 ARP who-has 192.168.254.163 tell 192.168.254.75  
05/11-10:08:53.851278 ARP who-has 192.168.254.164 tell 192.168.254.75  
05/11-10:08:53.861299 ARP who-has 192.168.254.165 tell 192.168.254.75  
05/11-10:08:53.871514 ARP who-has 192.168.254.166 tell 192.168.254.75  
05/11-10:08:53.881323 ARP who-has 192.168.254.167 tell 192.168.254.75  
05/11-10:08:53.891386 ARP who-has 192.168.254.168 tell 192.168.254.75  
05/11-10:08:53.901518 ARP who-has 192.168.254.169 tell 192.168.254.75  
05/11-10:08:53.911367 ARP who-has 192.168.254.170 tell 192.168.254.75  
05/11-10:08:53.921379 ARP who-has 192.168.254.171 tell 192.168.254.75  
05/11-10:08:53.931536 ARP who-has 192.168.254.172 tell 192.168.254.75  
05/11-10:08:53.941427 ARP who-has 192.168.254.173 tell 192.168.254.75  
05/11-10:08:53.951422 ARP who-has 192.168.254.174 tell 192.168.254.75  
05/11-10:08:53.961691 ARP who-has 192.168.254.175 tell 192.168.254.75  
05/11-10:08:53.971453 ARP who-has 192.168.254.176 tell 192.168.254.75  
05/11-10:08:53.981468 ARP who-has 192.168.254.177 tell 192.168.254.75  
05/11-10:08:53.991803 ARP who-has 192.168.254.178 tell 192.168.254.75  
05/11-10:08:54.001496 ARP who-has 192.168.254.179 tell 192.168.254.75  
05/11-10:08:54.011516 ARP who-has 192.168.254.180 tell 192.168.254.75

05/11-10:08:54.021663 ARP who-has 192.168.254.181 tell 192.168.254.75  
05/11-10:08:54.031539 ARP who-has 192.168.254.182 tell 192.168.254.75  
05/11-10:08:54.042233 ARP who-has 192.168.254.183 tell 192.168.254.75  
05/11-10:08:54.141865 ARP who-has 192.168.254.184 tell 192.168.254.75  
05/11-10:08:54.151794 ARP who-has 192.168.254.185 tell 192.168.254.75  
05/11-10:08:54.161728 ARP who-has 192.168.254.186 tell 192.168.254.75  
05/11-10:08:54.171883 ARP who-has 192.168.254.187 tell 192.168.254.75  
05/11-10:08:54.181779 ARP who-has 192.168.254.188 tell 192.168.254.75  
05/11-10:08:54.191805 ARP who-has 192.168.254.189 tell 192.168.254.75  
05/11-10:08:54.201915 ARP who-has 192.168.254.190 tell 192.168.254.75  
05/11-10:08:54.211799 ARP who-has 192.168.254.191 tell 192.168.254.75  
05/11-10:08:54.221811 ARP who-has 192.168.254.192 tell 192.168.254.75  
05/11-10:08:54.462159 ARP who-has 192.168.254.216 tell 192.168.254.75  
05/11-10:08:54.472319 ARP who-has 192.168.254.217 tell 192.168.254.75  
05/11-10:08:54.482187 ARP who-has 192.168.254.218 tell 192.168.254.75  
05/11-10:08:54.492419 ARP who-has 192.168.254.219 tell 192.168.254.75  
05/11-10:08:54.502357 ARP who-has 192.168.254.220 tell 192.168.254.75  
05/11-10:08:54.512230 ARP who-has 192.168.254.221 tell 192.168.254.75  
05/11-10:08:54.522253 ARP who-has 192.168.254.222 tell 192.168.254.75

Another dead give away in this incident is the fact that the ARP requests are incremental, inquiring for MAC addresses for an entire segment in less than a few seconds. This should certainly be considered suspicious activity. This however is only a sign of trouble. How do we detect the actual trouble? How do we find an ARP spoof in progress?

On day 2 the window of detection is very small, but our NIDS application has picked up an ARP request for IP address 192.168.254.33 from 192.168.254.33. This should not happen unless 2 computers have the same IP address or 1 computer is claiming to have IP

address 192.168.254.33 when it does not (spoofing). On day 2 we may also be able to find that the attacker was sniffing packets on the LAN if we used a promiscuous mode detection utility. This utility can be used to identify cards that have promiscuous mode on. NICs that are in promiscuous mode are most likely sniffing, and if they are not, they should be dealt with anyhow.

On days 3 through 8 there is relatively no chance of detection since Cain and Abel has the ability to perform an offline crack of the password using the NTLMv2 hash that it sniffed out. This means not even one failed logon attempt is logged to the server event logs during the crack phase.

On day 9 we can detect both the mapping of the network drive from the attacker to the victim as well as the copying of the Trojan to the victims PC. SNORT logs will show this activity as well as Ethereal.

On day 10 we can detect a problem by viewing the open ports on the workstation. The initial port used is 8888 (see **figures 2.6.A and 2.6.B**). By typing **netstat -a**, at a command prompt we will see that the port is listening for a connection. We know the ports 0 – 1024 are the well-known ports and that any ports above this range are either registered or private. When we see a port above the well known range it should raise concern and be investigated. When an attacker is connected to the port, Beast will open the next 8 consecutive ports. That leaves the victim with 9 open ports with established connections. Another detection mechanism would be the keylogger file. This file by default has a .blf file extension. If you have these 2 pieces of evidence, you know for sure you are dealing with The Beast Trojan. Another clue that we have been infected with the Trojan is that the local virus protection is disabled.

**Fig. 2.6.A Beast actively listening on port 8888**

```

Select C:\WINDOWS\System32\cmd.exe
TCP        HRDIRECTOR:8888        HRDIRECTOR.LICHGATE.NET:0 LISTENING
TCP        HRDIRECTOR:8889        HRDIRECTOR.LICHGATE.NET:0 LISTENING
TCP        HRDIRECTOR:8890        HRDIRECTOR.LICHGATE.NET:0 LISTENING
TCP        HRDIRECTOR:8891        HRDIRECTOR.LICHGATE.NET:0 LISTENING
TCP        HRDIRECTOR:8892        HRDIRECTOR.LICHGATE.NET:0 LISTENING
TCP        HRDIRECTOR:8893        HRDIRECTOR.LICHGATE.NET:0 LISTENING
TCP        HRDIRECTOR:8894        HRDIRECTOR.LICHGATE.NET:0 LISTENING
TCP        HRDIRECTOR:8895        HRDIRECTOR.LICHGATE.NET:0 LISTENING
TCP        HRDIRECTOR:8896        HRDIRECTOR.LICHGATE.NET:0 LISTENING
TCP        HRDIRECTOR:nethios-ssn  HRDIRECTOR.LICHGATE.NET:0 LISTENING
TCP        HRDIRECTOR:1068        HRDIRECTOR.LICHGATE.NET:0 LISTENING
TCP        HRDIRECTOR:1068        192.168.254.100:nethios-ssn ESTABLISHED
TCP        HRDIRECTOR:1125        192.168.254.100:microsoft-ds TIME WAIT
TCP        HRDIRECTOR:8888        192.168.254.75:1776 ESTABLISHED
TCP        HRDIRECTOR:8889        192.168.254.75:1777 ESTABLISHED
TCP        HRDIRECTOR:8890        192.168.254.75:1778 ESTABLISHED
TCP        HRDIRECTOR:8891        192.168.254.75:1779 ESTABLISHED
TCP        HRDIRECTOR:8892        192.168.254.75:1780 ESTABLISHED
TCP        HRDIRECTOR:8893        192.168.254.75:1781 ESTABLISHED
TCP        HRDIRECTOR:8894        192.168.254.75:1782 ESTABLISHED
TCP        HRDIRECTOR:8895        192.168.254.75:1783 ESTABLISHED
TCP        HRDIRECTOR:8896        192.168.254.75:1784 ESTABLISHED
UDP        HRDIRECTOR:microsoft-ds  *:*
UDP        HRDIRECTOR:isaknp      *:*
UDP        HRDIRECTOR:1025        *:*
UDP        HRDIRECTOR:1026        *:*
UDP        HRDIRECTOR:1042        *:*
UDP        HRDIRECTOR:ntp         *:*
UDP        HRDIRECTOR:ntp         *:*
UDP        HRDIRECTOR:nethios-ns  *:*
UDP        HRDIRECTOR:nethios-dgm *:*

C:\Documents and Settings\jdoe>
```



Fig 2.6.B Beast with established connection on several open ports

```
C:\WINDOWS\System32\cmd.exe
C:\Documents and Settings\jdoe>netstat -a

Active Connections

Proto Local Address           Foreign Address         State
TCP   HRDIRECTOR:epmap        HRDIRECTOR.LICHGATE.NET:0 LISTENING
TCP   HRDIRECTOR:microsoft-ds HRDIRECTOR.LICHGATE.NET:0 LISTENING
TCP   HRDIRECTOR:1062        HRDIRECTOR.LICHGATE.NET:0 LISTENING
TCP   HRDIRECTOR:2882        HRDIRECTOR.LICHGATE.NET:0 LISTENING
TCP   HRDIRECTOR:8888        HRDIRECTOR.LICHGATE.NET:0 LISTENING
TCP   HRDIRECTOR:nethbios-ssn HRDIRECTOR.LICHGATE.NET:0 LISTENING
TCP   HRDIRECTOR:nethbios-ssn 192.168.254.100:16143 ESTABLISHED
TCP   HRDIRECTOR:1039        192.168.254.100:1026 TIME_WAIT
TCP   HRDIRECTOR:1046        192.168.254.100:microsoft-ds TIME_WAIT
TCP   HRDIRECTOR:1051        192.168.254.100:epmap TIME_WAIT
TCP   HRDIRECTOR:1052        192.168.254.100:1026 TIME_WAIT
TCP   HRDIRECTOR:1056        192.168.254.100:ldap TIME_WAIT
TCP   HRDIRECTOR:1058        192.168.254.100:ldap TIME_WAIT
TCP   HRDIRECTOR:1061        192.168.254.100:ldap TIME_WAIT
TCP   HRDIRECTOR:1068        HRDIRECTOR.LICHGATE.NET:0 LISTENING
TCP   HRDIRECTOR:1068        192.168.254.100:nethbios-ssn ESTABLISHED
UDP   HRDIRECTOR:microsoft-ds *:*
UDP   HRDIRECTOR:isakmp      *:*
UDP   HRDIRECTOR:1025        *:*
UDP   HRDIRECTOR:1026        *:*
UDP   HRDIRECTOR:1042        *:*
UDP   HRDIRECTOR:ntp         *:*
UDP   HRDIRECTOR:ntp         *:*
UDP   HRDIRECTOR:nethbios-ns *:*
UDP   HRDIRECTOR:nethbios-dgm *:*

C:\Documents and Settings\jdoe>
```

## 3.0 The Environment

The network environment is a simple LAN with a private addressing scheme of 192.168.254.0 with a subnet mask of 255.255.255.0, making it a class C network. This is an inside attack involving 3 computers (including the attacker's) residing on the same network. We will refer to the relevant computers from this point on as the attacker, the server and the victim.

### 3.1 The Victim's Platform

The victim is running an XP Professional workstation with SP1 installed. The victim is also running Symantec Corporate Edition 8.0 Anti-Virus software. The computer is a member of a Windows 2000 domain. The IP address of the victim is 192.168.254.33.

### 3.2 The Source Network

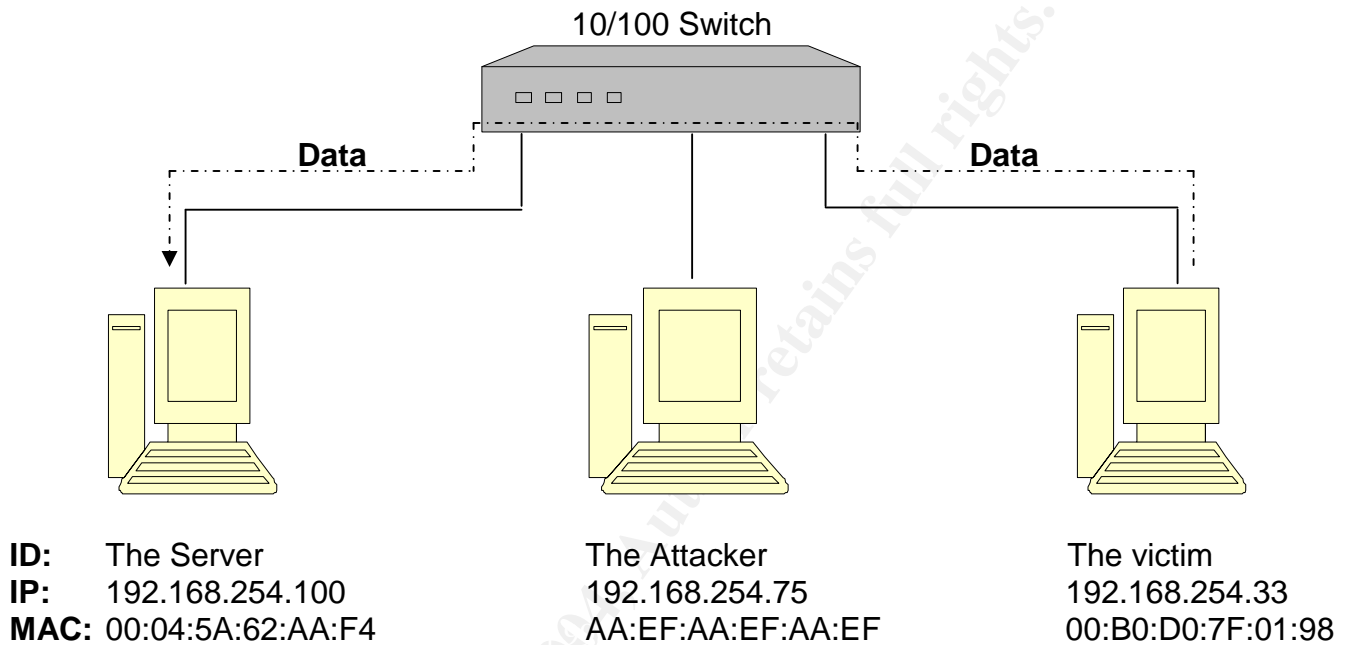
The source network is a single server environment with a network address of 192.168.254.0. It is a Microsoft Windows 2000 platform and the server is running service pack for and all current security updates. The server is running DNS services. The server IP address is 192.168.254.100

### 3.3 The Target Network

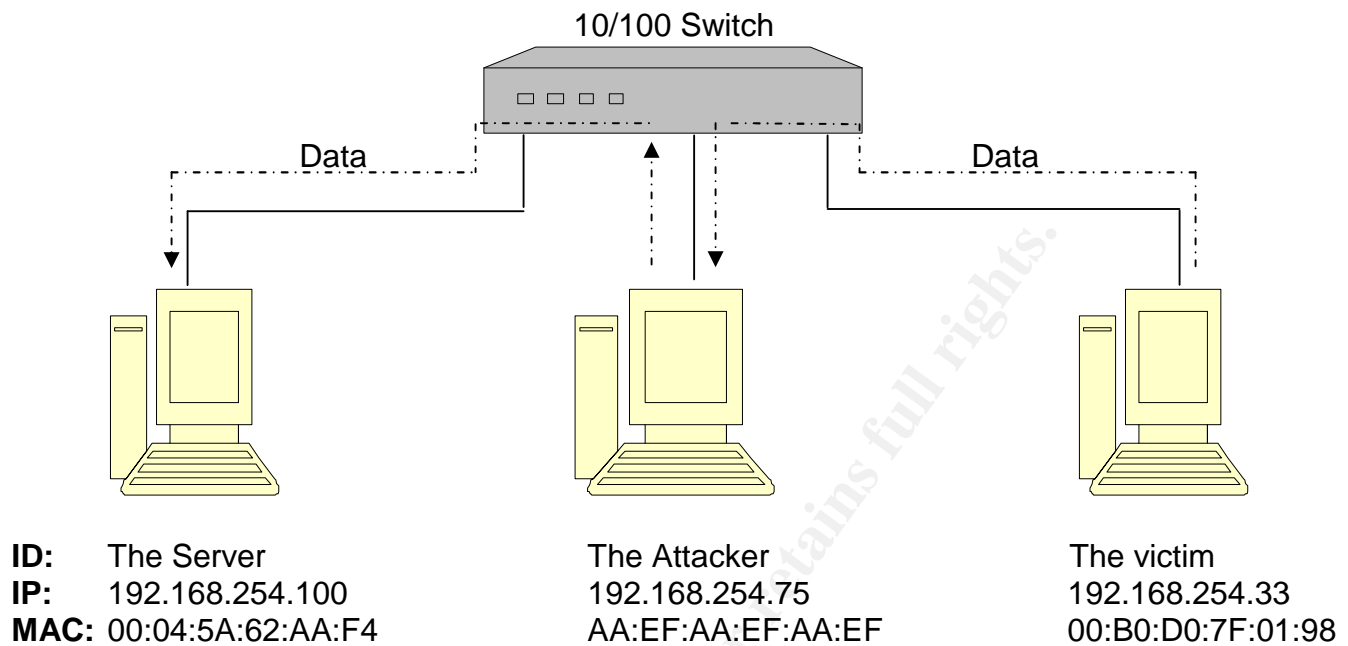
The attacker and the victim are on the same network segment. Therefore the target network is the same as the source network outlined above.

### 3.4 The Network Diagram

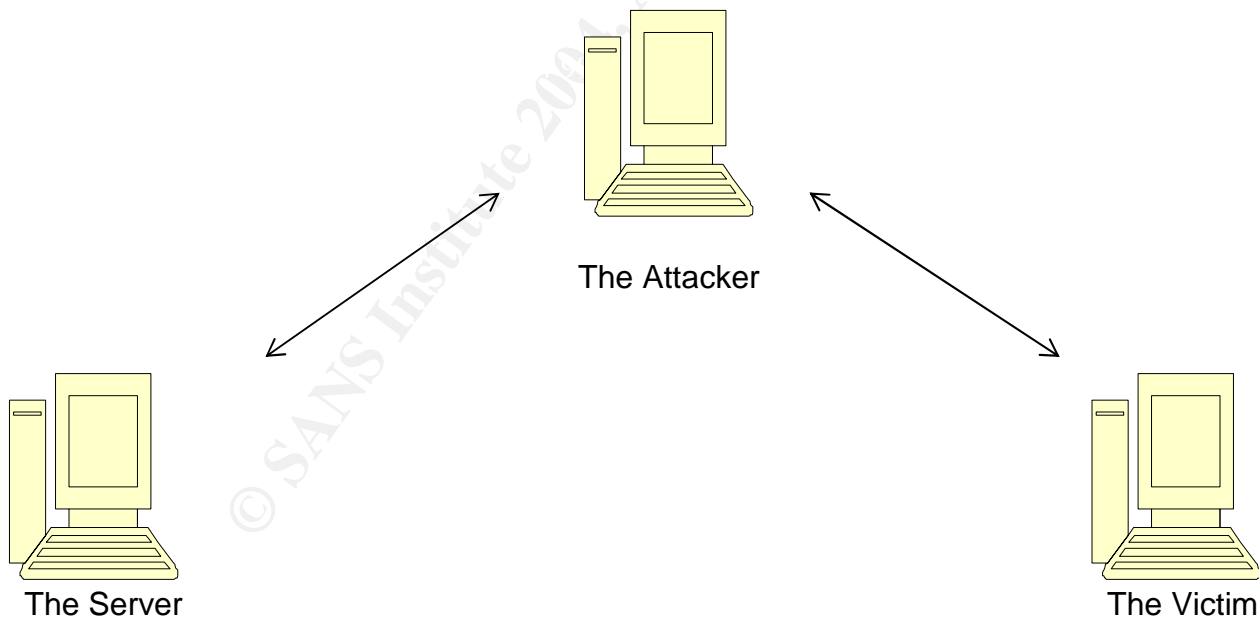
#### 3.4.A BEFORE THE ATTACK



### 3.4.B During the Attack



**The server believes that AA:EF:AA:EF:AA:EF has IP address 192.168.254.33**  
**The victim believes that AA:EF:AA:EF:AA:EF has IP address 192.168.254.100**



The attacker is acting as a router reading the information and then passing it on. Communication between the victim and the server appears to be normal. This is because once the workstation sends out a packet it doesn't care how it gets there, it only cares that it did in fact get there.

## 4.0 Stages of the Attack

### 4.1 Reconnaissance

The reconnaissance for this attack is fairly simple. Since we are on the same network as the victim and we have the same network administrator, we can gather valuable information about the victim by inspecting our own system. As part of the reconnaissance we have created a problem with our system by intentionally corrupting the registry. We did this to observe the level of technical expertise that the network administrator demonstrates. We also see how organized the administrator is when dealing with disaster recovery situations. To know the administrator is to know the network. We see **by looking at our own system** that the virus definitions are updated daily at 11pm. **By visiting the windows update website** we find that there are no critical updates to install, which means all known vulnerabilities are un-exploitable on our workstation. **By opening the command prompt window we run the ipconfig /all** utility and switch to identify our hostname and tcp/ip configuration.

### 4.2 Scanning

Here we will use the built in scanning tool provided by Cain and Abel to find our victim. **By clicking on the Network Tab** and **expanding the Microsoft Windows Network object** we will see a list of computers within the windows domain (*see fig. 4.2.A*). A domain is a logical boundary for windows networking where computers reside, much like houses reside in a neighborhood. This area of Cain resembles the windows explorer. The domain in this example is called Lichgate. We are not exploiting any code or doing any harm to the network at this point. The nature of windows networking and TCP/IP communications is to share this information about the system via NetBIOS. We can get similar information by using the nbtstat and net view utilities from a command prompt as well as using windows explorer. NetBIOS is technically obsolete although it still serves a purpose in some windows networking functions; it is mainly used for backwards compatibility with legacy systems like Win9x. We do not see our victim in this list, this is primarily because the victim is using Microsoft Windows XP. For security reasons, Microsoft Windows 2000 and later versions implement DNS, in place of NetBIOS. We are going to have to take the scan to the next level by using Cain and Abel's MAC address scanning tool. **By clicking on the sniffer tab** up top and then **clicking the hosts tab** below we can access the MAC address scanning tool (*see fig. 4.2.B*). We must first **enable the sniffer by clicking the sniffer icon** in the top Left corner of the toolbar. Right click any white space area and choose the scan all MAC addresses. After clicking the scan all MAC addresses option we are presented with some options in a pop-up window. Since this is a relatively small network and we have no idea what our victim's IP address is we will scan the entire range of network addresses and run all tests. When the scan completes our table will populate with IP addresses, MAC addresses and manufacturer information. We can now **select the records** in this table, **right click the selection** and choose to **resolve host names**. When Cain is finished resolving names in the table, we will see a computer with the host name HR Director with a matching IP address of 192.168.254.33. We have now successfully located our victim.

Figure 4.2.A

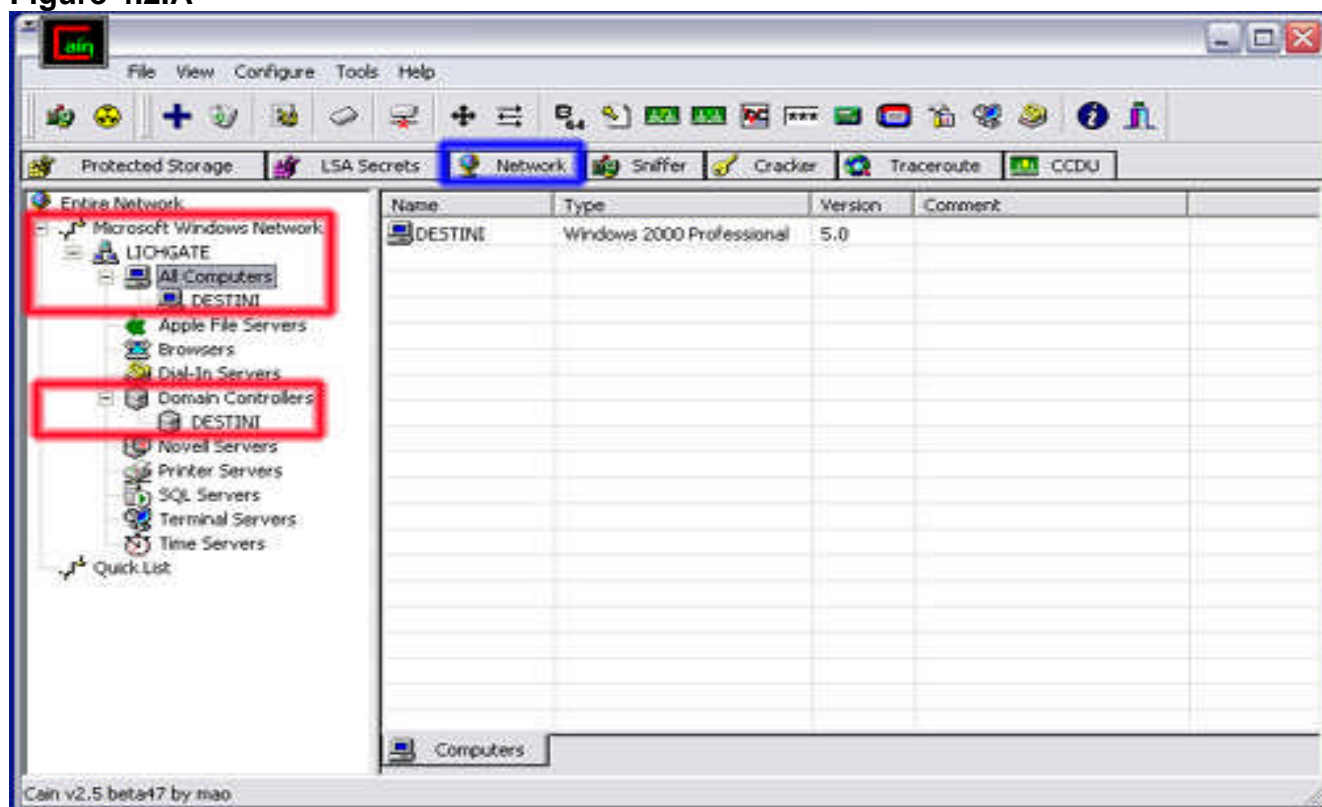


Figure 4.2.B

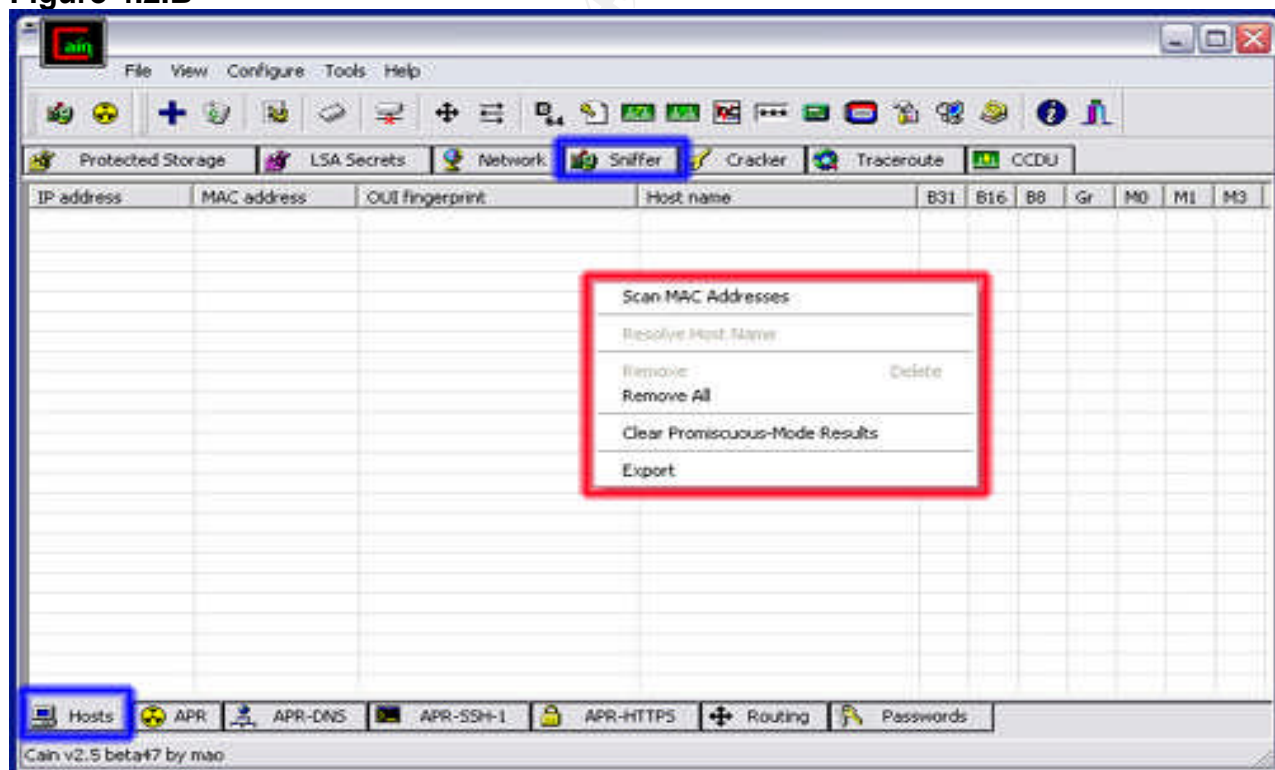
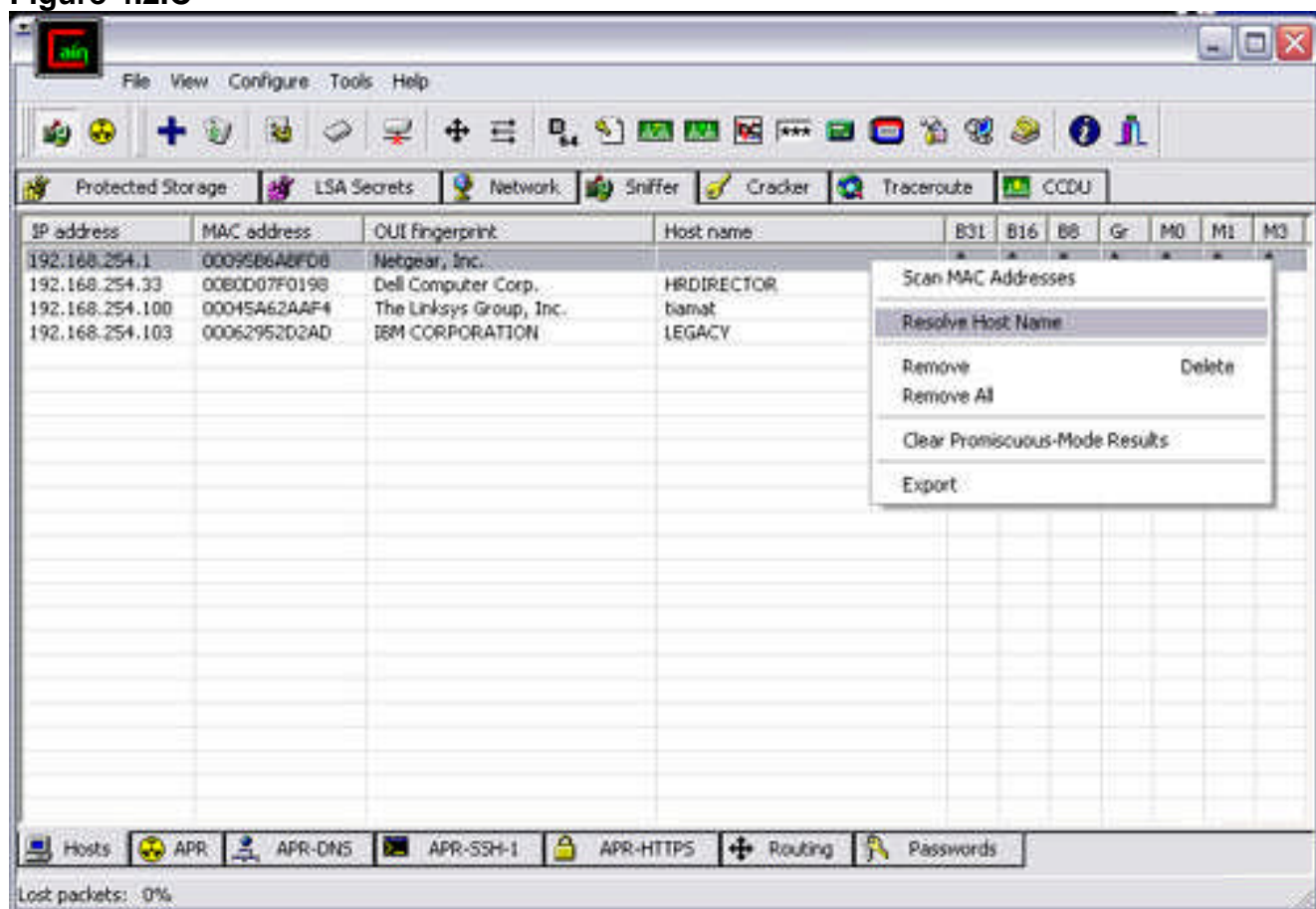


Figure 4.2.C



By clicking **My Computer**, tools, **disconnect network drives** we can find some additional host names, most likely the server host name where shared files and applications are stored. Our workstation has the letter “P” mapped to a network share <\\destini\\public>. If we refer to Figure 4.2.A we know that destini is our server because it has been identified as a domain controller by Cain and Abel. The hostname of a mapped drive immediately follows the “\\”. The host name of the shared folder we just identified is destini. This entire mapping is known as a UNC path where the letters UNC stand for Universal Naming Convention. Cain shows the IP address of Destini to be 192.168.254.100.

### **4.3 Exploiting the System**

Now that we have identified the target’s IP address as 192.168.254.33 and the server’s IP address as 192.168.254.100, it’s time to begin poisoning the ARP tables. Below are the click by click instructions for using Cain and Abel to perform this exploit.

- Click the SNIFFER tab. (TOP)**
- Click the APR TAB. (BOTTOM)**
- Click the + Symbol. (TOP)**
- Click the Victim IP address on the Left.**
- Click the Server IP address on the Right.**

**Click start APR (TOP LEFT).**

Poisoning is now taking place and could take several minutes. After poisoning has successfully occurred, you will begin to see increasing packet counts between the server and the victim. Now click on the passwords TAB and wait for passwords to show up. If Cain has intercepted a password, it will increase the count shown in parenthesis next to the password type that was sniffed out. **Figure 4.3.A** shows that we sniffed an MSKerb5-Preauth hash and 3 SMB Hashes. We explained Kerberos5 in a previous section. It is the Algorithm windows uses to encrypt network passwords before transmission between client and server. Now that we have a password hash we can send it to Cain's password crack utility **by right clicking the password and left clicking send to cracker**. Then proceed to crack the password with the following instructions:

**Click the Cracker TAB. (TOP)**

**Click the Kerb5-PreAuth object. (Left Pane)**

**Right click the captured password.**

**Left click start brute force attack.**

While the password is being cracked we can now prepare our Trojan. We will work with a Trojan kit known as The Beast. We will want to turn off our real-time virus protection scanning in order to build the Trojan and not have our kit or Trojan file quarantined.

After Launching The Beast we begin building our customized Trojan by clicking the Build Server button. The following is a guide that walks thru each option of the Trojan builder and the settings we will require for the specific task at hand.

#### **4.3.1 Basic Menu**

Leave the name as server.exe

Create a password (this is for the initial remote connection to the server we are creating)

Define the listening port (you can make this whatever you want between 1024 and 65,535) the default port is usually the best choice unless it is already used by another application on the system. TCP ports fall into three categories

0-1023 are the well known ports

1024 – 49151 are the registered ports

49152-65535 are for private use

Note that the above usage is a guideline. You can ultimately define whatever port you like, as long as it is not in use already.

We will not use the SIN port here, but it is a very useful option for bypassing a firewall. You define the SIN port again as any port you desire. The reverse connection option must be used when defining a SIN port. The reverse option connection is exactly that. Instead of you initiating the connection with the victim, the victim will initiate the connection to you, on the port you specified.



This bypasses firewall security because most firewalls will allow outgoing connections with less restriction than incoming connections.

Inject into IE is an option that injects the server into internet explorer. Injection is actually file binding. The server.exe file is bound to the internet explorer. This means that every time the victim surfs the web with internet explorer our Trojan is re-established.

#### **4.3.2 Notifications Menu**

Choose email notification and make sure the enable SIN box is checked. You can use any of the methods to be notified that the server has been activated. I find that an anonymous or fictitious email account is the safest when covering your tracks.

#### **4.3.3 Startup Menu**

Use the default settings. These are the method that will continue to start the server after a user reboots their system. This is a method of keeping access.

#### **4.3.4 Kill AV and Firewall Menu**

Check all options here. It doesn't hurt to keep The Beast from being detected. Also, you will need The Beast to kill any firewall products that may be installed. Otherwise, you will not be able to connect to the port that you defined earlier.

#### **4.3.5 Miscellaneous**

The miscellaneous section has some good features. The default features are melt server on install and enable key logger. These two are important for this exercise. The melt server option will delete the server.exe file once it is injected into IE. The key logger is going to help us obtain the password to the accounting system.

After creating the Trojan executable we need to hex edit the file because it is a known virus to Symantec corporate edition 8.0. If you check the file size you will see that the executable is approximately 28kb. The files size will vary depending on the options you chose when building the Trojan. The smaller the file, the better our chances of avoiding detection. We hex edit our Trojan with the following steps:

**Open the Stealth Tools application.**

**Click Hex Edit.**

**Select the Trojan we created (server.exe)**

**Double click the capital B in the offset line 000000B0. (I have identified this as part of the virus signature by trial and error)**

**Click Save.**

By passing over the file with our virus scanner we will see that the virus is no longer detected. When the cracker has finished cracking the password, we will use it to access the victim's

system and drop our Trojan in the startup folder of the victim's pc. The cracker shows that the password for the victims account (jdoe) is emma04.

The next step is to attach to the victims system and plant the bug. We will first try to attach to an administrative share, like C\$. The command line utility for this is:

**C:\ NET USE Z: \\hrdirector\c\$ \* /user:Lichgate\jdoe**

This is the best choice when trying to attach to a victim's computer. One reason is that the C\$ share is created by default on windows operating systems. Another reason is that you can navigate to any other directory from this point. The directory we want to navigate to is the startup folder of the user's profile. The default path for a user's startup directory on windows XP is

**C:\windows\documents and settings\<user name>\start menu\programs\startup**

After placing the Trojan file server.exe into the startup directory, we can compromise the system the next time it is rebooted. The Beast will notify us that it is active and will remind us what port it is listening on and the other parameters we specified. By tomorrow the system will be rebooted, it is company policy to turn off computers after leaving for the day.

We are now at the final stage of our attack. It is time to launch the beast control console and connect to the victim. Once the console is launched we need to enter the following information:

**The IP address of the victim. (192.168.254.33)**

**The Port the Trojan is listening on. (8888)**

**The password that we assigned.**

After the information is supplied, simply click connect and the beast should show a connected status at the bottom left of the GUI. We finally have full control of the workstation and can use any of the robust features this Trojan offers. It is a good idea at this point to create some new backdoors. It would be a good idea to use the file manager to drop another back door or use the remote desktop feature to create a backdoor user account. To accomplish our goal however, we only need to retrieve the password used for entering the accounting system. We'll do this by retrieving the file from the key logger out Trojan has initiated. The following steps show you how this is done.

**Click Misc. Button**

**Click Key logger**

**Click Get Log**

**Click Yes to decrypt the log.**

The following text is a sample of the the information the key logger provides... as you can see it is very easy to track what the user is doing and will only be a matter of time before the accounting application is opened and the password is recorded.

\*\*\*\*\* Boot:[5/29/2004]-[19:4:28]

[Program Manager]-[19:4:28]

[Shortcut to rpcapd]-[19:4:52]

[Start Menu]-[19:4:54]

[Run]-[19:4:56]

[Program Manager]-[19:4:57]

[C:\WINDOWS\System32\cmd.exe]-[19:4:57]

snort -a{Undo}e -l logs -

vcd \snortcd bin

[C:\WINDOWS\System32\cmd.exe - cd bin]-[19:5:21]

sn

[C:\WINDOWS\System32\cmd.exe]-[19:5:21]

ort -e -l logs -v

[C:\WINDOWS\System32\cmd.exe - snort -e -l logs -v]-[19:5:34]

[Start Menu]-[19:5:43]

[Run]-[19:5:44]

[C:\WINDOWS\System32\cmd.exe]-[19:5:46]

netstat 0-{Undo}{Undo}-a

[C:\WINDOWS\System32\cmd.exe - netstat -a]-[19:5:49]

[C:\WINDOWS\System32\cmd.exe]-[19:5:51]

[C:\WINDOWS\System32\cmd.exe - snort -e -l logs -v]-[19:6:18]

[C:\WINDOWS\System32\cmd.exe]-[19:6:18]

[Shortcut to rpcapd]-[19:6:19]

[C:\WINDOWS\System32\cmd.exe]-[19:6:19]

[Start Menu]-[19:6:23]

[Adobe Photoshop]-[19:6:31]

[Adobe Registration - Registration Choice]-[19:7:19]

[Adobe Photoshop]-[19:7:23]

[New]-[19:7:32]

[Adobe Photoshop]-[19:7:33]

[Adobe Photoshop - [Untitled-1 @ 66.7% (Layer 1, RGB)]]-[19:7:46]

[Adobe Photoshop - [Untitled-1 @ 100% (Layer 1, RGB)]]-[19:7:47]

If you take notice of the line highlighted in the sample capture log you can see that The Beast KeyLogger records the application that is opened, in this case cmd.exe. It also records the keyboard input as we see the user initiated the SNORT application with a few switches. At the end of the line is a timestamp in military format. The key logger will eventually capture the accounting db username and password when the HR Director uses it. At this point the attacker has the necessary information to complete his objective. He remotely accesses the

accounting information under the guise of the HR Director and gives himself a 10% pay increase, while the victim is out to lunch.

#### **4.4 Keeping Access**

There are several ways for us to keep access at this point. We have compromised the systems user password. This gives us the ability to do anything the user is allowed to do. We can install other Trojans. We can install tools like netcat. We can create a new user account in case the current user changes their password. What is especially nice about The Beast Trojan Kit is that we can tell the server.exe file to inject itself into Internet Explorer or Notepad.exe, so that if the virus is cleaned it will be re-installed the next time the injected executable is run.

#### **4.5 Covering the Tracks**

There are only 2 simple things we need to do to cover our tracks. In the key logger file we saw that the user turned on snort. We saw that they defined the log files to be placed in the "logs" directory of the current path which equates to c:\snort\bin\logs. We will want to delete the files inside this directory. Second we will use a function in beast that will kill its own server and clean itself from the system. We do this by clicking the server button and the kill server button. If the tools were used on our own system they should be cleaned up as well.

### **5.0 The Incident Handling Process**

#### **5.1 Preparation**

The network administrator follows standard operating procedures regarding network administration. By incorporating virus protection and routinely applying windows critical updates, he has fought half the battle. Although preventative measures were in place the network was lacking in any form of intrusion detection and countering. This lack of preparation is mainly due to the fact that the company has never suffered an obvious or debilitating intrusion incident. The incident handling consists of the IT administrator and the office manager. The office manager's role is only to be the liaison between the staff and the IT administrator. This is a filtering process so that the network administrator will only be alerted of serious issues as perceived by the office manager. The network administrator is available by cell phone and pager 24/7. The preventative measures in place include real-time virus protection, strict password policies, acceptable computer usage policies and internet usage policies.

## **5.2 Identification**

Before we jump into the identification of this attack I would like to describe my jump kit contents that I keep with me all the time. The following is a list of items I need to be an effective troubleshooter and incident handler.

**Screwdrivers** – all types and sizes, Phillips head, Straight Head and Star Head.

**USB Hard Drive** – good for backing up data and installing software when you can't access the network or the internet. Good for keeping admin tools and such in an accessible place.

**10/100 4 port switch** – Always good to have for providing additional ports to an area or isolating a workstation with a troubleshooting device

**Blank CD's** – good for burning tools when a USB port is not available for the USB HD.

**Pen and Pad** – For keeping notes

**Extra Cables** – Video, Mouse and Keyboard extension cables, IDE HDD cables, SCSI HDD cables, 50' ethernet patch cable, 3' ethernet crossover cable.

**Applications** – AATOOLS, STINGER, SuperScan4, TCPView, Netcat, NMAP, WINPCAP, SNORT are the main apps. However I write some helpful scripts and also constantly look for new tools that aid me in the endeavor to keep my networks safe.

I also consider my laptop a key component of my jump kit.

The network administrator knows that the most common method of intrusion is through open TCP/UDP ports on a system. Therefore, the administrator will scan his own network daily with a program called superscan4 which can be downloaded for free at <http://www.foundstone.com/index.htm?subnav=resources/navigation.htm&subcontent=/resources/scanning.htm>. SuperScan is a particularly nice tool to use because of its HTML report feature which organizes the information and is an easy read. Knowing what ports your workstations and servers require to be open is crucial to this identification process. By knowing what is supposed to be open you will know what is not supposed to be opened. Any open ports found that your system does not require should receive immediate attention. Here is a sample report from the workstation HRDIRECTOR. This is where our administrator notices that port 8888 is open on the computer and investigates the issue. An investigation is now underway.

### **SuperScan Report - 07/05/04 18:21:54**

IP	192.168.254.33
Hostname	[Unknown]
Netbios Name	HRDIRECTOR
Workgroup/Domain	LICHGATE
TCP Ports (4)	
135	DCE endpoint resolution
139	NETBIOS Session Service
445	Microsoft-DS
<a href="#">8888</a>	NewsEDGE server TCP / AnswerBook2
UDP Ports (2)	

123	Network Time Protocol	
137	NETBIOS Name Service	
TCP Port		Banner
UDP Port		Banner
137	MAC Address: 00:B0:D0:7F:01:98 NIC Vendor : 3Com / Computer Products International	
NETBIOS Name Service	Netbios Name Table (6 names)	
	HRDIRECTOR 00 UNIQUE Workstation service name	
	LICHGATE 00 GROUP Workstation service name	
	HRDIRECTOR 03 UNIQUE Messenger name	
	HRDIRECTOR 20 UNIQUE Server services name	
	LICHGATE 1E GROUP Group name	
	GEO 03 UNIQUE Messenger name	

Total hosts discovered	1
Total open TCP ports	4
Total open UDP ports	2

After finding a suspicious open port the administrator quickly checks all computers scanned for the same open port. After finding no other computers with this open port, he quickly moves to the location of the workstation and begins to investigate. The first thing the admin will do is check for any established connections to the workstation. He does this by opening a command prompt and typing netstat -a on the command line.

**Figure 5.2.A Using netstat**

```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\geo.LICHGATE>netstat -a

Active Connections

Proto Local Address           Foreign Address         State
TCP   HRDIRECTOR:135          HADIRECTOR.LICHGATE.NET:0 LISTENING
TCP   HRDIRECTOR:445          HADIRECTOR.LICHGATE.NET:0 LISTENING
TCP   HRDIRECTOR:1037         HADIRECTOR.LICHGATE.NET:0 LISTENING
TCP   HRDIRECTOR:8888         HADIRECTOR.LICHGATE.NET:0 LISTENING
TCP   HRDIRECTOR:139          HADIRECTOR.LICHGATE.NET:0 LISTENING
UDP   HRDIRECTOR:445          *:*
UDP   HRDIRECTOR:500          *:*
UDP   HRDIRECTOR:1025         *:*
UDP   HRDIRECTOR:1026         *:*
UDP   HRDIRECTOR:123          *:*
UDP   HRDIRECTOR:123          *:*
UDP   HRDIRECTOR:137          *:*
UDP   HRDIRECTOR:138          *:*

C:\Documents and Settings\geo.LICHGATE>

```

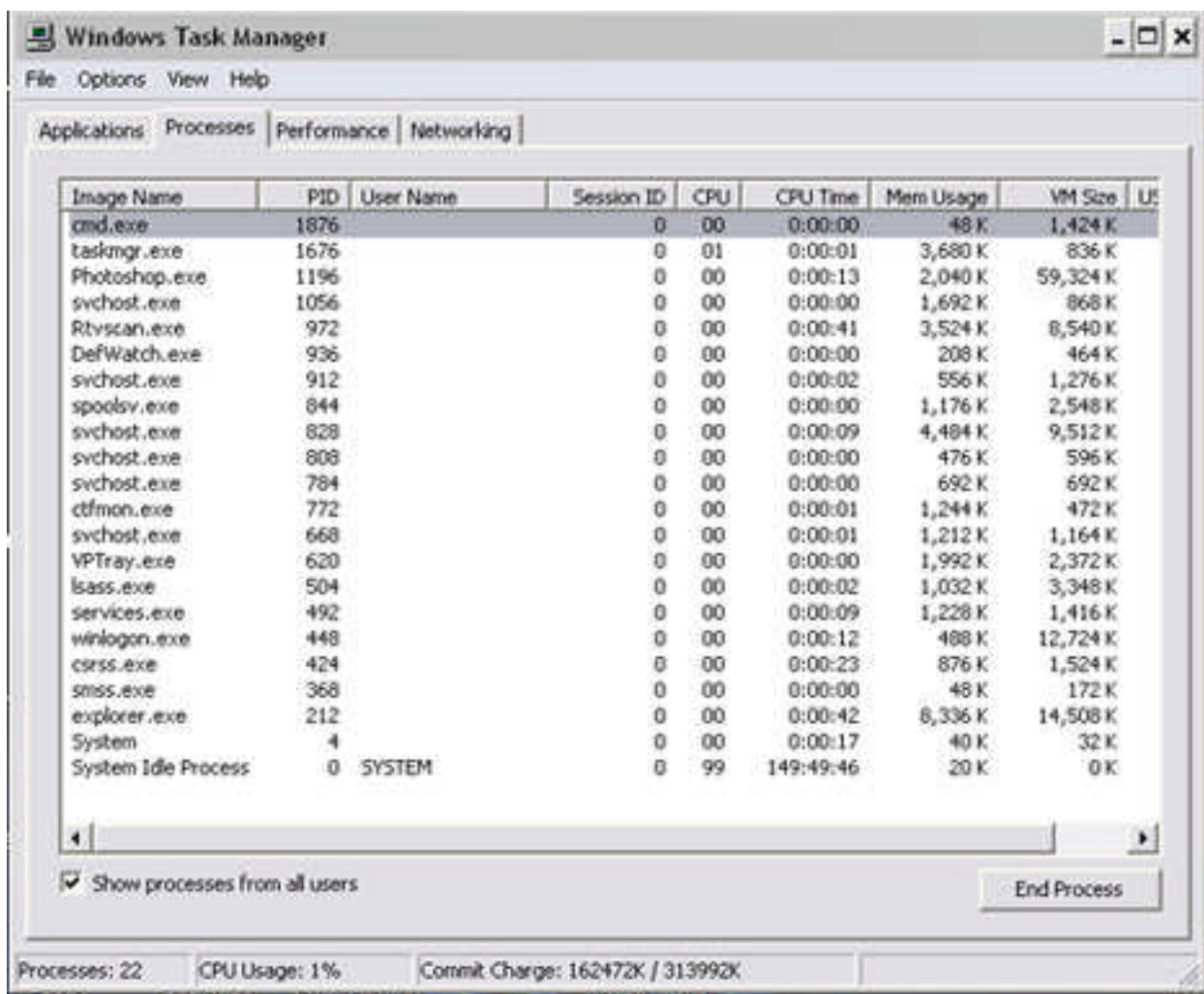
Sure enough we see that the computer is listening on tcp port 8888, but currently there is no connection. If a connection to the port had been made the state would change from

LISTENING to ESTABLISHED. At the moment there seems to be no apparent threat but the administrator needs to find out what is going on with this computer. The next thing we want to check for is any unfamiliar processes. If we find an unfamiliar process running on the workstation we may be able to identify this mysterious open port. Below is a capture of the windows task manager showing all active processes. You can call this up by simultaneously pressing CTRL + ALT + DEL. Then click on the task manager button and then the processes tab. Our example here in **Figure 5.2.B** shows all the processes currently running on the victims computer.

**Figure 5.2.B**

© SANS Institute 2004, Author retains full rights.





Everything so far appears to be normal. Sometimes when a system is compromised you cannot open the task manager by design of the attacker. It is good to have a command line utility for windows that will list and kill processes. Unix based operating systems have a built in command line utility for viewing processes. This next capture shows a command line utility for viewing windows processes.

It was downloaded from <http://www.beyondlogic.org/consulting/processutil/processutil.htm>.  
(WARNING! This site has pop-up ads)

**Figure 5.2.C** shows the process utility in action. What can you see here that may be suspicious? Well before I answer here is a site that will help you identify processes that are running on your computer. [http://www.answerthatwork.com/Tasklist\\_pages/tasklist.htm](http://www.answerthatwork.com/Tasklist_pages/tasklist.htm). It is helpful to know what processes are needed by Windows and what processes are used by viruses and other malicious code. At a glance, all the processes in **Figure 5.2.C** are valid. The only thing the administrator was able to see here is that all of the SVCHOST.EXE processes have a priority of 8 save one. The last SVCHOST.EXE is showing a priority of 13. This is just a shot in the dark but sometimes, that's all you have. You need to really explore all of your options and find that slightest difference. By killing the process, as we did in **Figure 5.2.D**, we see that it was the process responsible for the opening of port 8888 on the

victim's computer, as the port is no longer open. This is verified by running netstat -a again as shown in **Figure 5.2.E**

**Figure 5.2.C**

```

C:\WINDOWS\System32\cmd.exe
Command Line Process Viewer/Killer/Suspender for Windows NT/2000/XP V2.03
Copyright(C) 2002-2003 Craig.Peacock@beyondlogic.org

ImageName      PID  Threads  Priority  CPU  Owner
-----
Idle           0         1         0    100 Error 0x6 : The handle is invalid.
System         4         41         8         0 Error 0x5 : Access is denied.
smss.exe      368         3        11         0 NT AUTHORITY\SYSTEM
csrss.exe     424        10        13         0 NT AUTHORITY\SYSTEM
winlogon.exe  448        20        13         0 NT AUTHORITY\SYSTEM
services.exe  492        15         9         0 NT AUTHORITY\SYSTEM
lsass.exe     504        20         9         0 NT AUTHORITY\SYSTEM
svchost.exe   668         8         8         0 NT AUTHORITY\SYSTEM
svchost.exe   776         6         8         0 Error 0x5 : Access is denied.
svchost.exe   800         4         8         0 Error 0x5 : Access is denied.
svchost.exe   820        34         8         0 NT AUTHORITY\SYSTEM
spoolsv.exe   836         9         8         0 NT AUTHORITY\SYSTEM
DefWatch.exe  932         3         8         0 NT AUTHORITY\SYSTEM
RtvsScan.exe  992        35         8         0 NT AUTHORITY\SYSTEM
svchost.exe  1056         5         8         0 NT AUTHORITY\SYSTEM
explorer.exe  1780        17         8         0 HARDIRECTOR\Administrator
UPTray.exe    1884         2         8         0 HARDIRECTOR\Administrator
opware32.exe  1892         2         8         0 HARDIRECTOR\Administrator
ctfmon.exe    1900         1         8         0 HARDIRECTOR\Administrator
Photoshop.exe 428        10         8         0 HARDIRECTOR\Administrator
AAtools.exe   540         1         8         0 HARDIRECTOR\Administrator
svchost.exe  1740         5        13         0 HARDIRECTOR\Administrator
cmd.exe       2800         1         8         0 HARDIRECTOR\Administrator
Process.exe   1748         1        13         0 HARDIRECTOR\Administrator
  
```

**Figure 5.2.D Showing the process utility commands and the killed process**

```

C:\WINDOWS\System32\cmd.exe

C:\>process /?

Command Line Process Viewer/Killer/Suspender for Windows NT/2000/XP U2.03
Copyright(C) 2002-2003 Craig.Peacock@beyondlogic.org
Usage: process [-v] [-t] [-c]
        process [-q] [Process Name/PID] [timeout sec(optional)]
        process [-k] [-s] [-r] [Process Name/PID]
        process [-p] [Process Name/PID] <RealTime|High|AboveNormal|
        Normal|BelowNormal|Low>
        process [-a] [Process Name/PID] [Mask<To Set>]
        -v    View Processes.
        -t    View Kernel and User CPU Times.
        -c    View Process Creation Times.
        -q    Send WM_CLOSE Message. Default timeout is 60 Sec
        -k    Kill Process. (Terminate)
        -s    Suspend Process.
        -r    Resume Suspended Process.
        -p    Set Process Priority.
        -a    Get/Set Affinity Mask of Process.

C:\>process -k 1740

Command Line Process Viewer/Killer/Suspender for Windows NT/2000/XP U2.03
Copyright(C) 2002-2003 Craig.Peacock@beyondlogic.org
Killing PID 1740 'svchost.exe'

C:\>

```

Figure 5.2.E Showing the port 8888 no longer open after killing process svchost.exe

```

C:\>netstat -a

Active Connections

Proto Local Address           Foreign Address         State
TCP   HRDIRECTOR:135          HRDIRECTOR.LICHGATE.NET:0 LISTENING
TCP   HRDIRECTOR:445          HRDIRECTOR.LICHGATE.NET:0 LISTENING
TCP   HRDIRECTOR:1037         HRDIRECTOR.LICHGATE.NET:0 LISTENING
TCP   HRDIRECTOR:139          HRDIRECTOR.LICHGATE.NET:0 LISTENING
UDP   HRDIRECTOR:445          *:*
UDP   HRDIRECTOR:500          *:*
UDP   HRDIRECTOR:1025         *:*
UDP   HRDIRECTOR:1026         *:*
UDP   HRDIRECTOR:123          *:*
UDP   HRDIRECTOR:123          *:*
UDP   HRDIRECTOR:137          *:*
UDP   HRDIRECTOR:138          *:*

C:\>_

```

Although it is common for a Trojan or Virus to hide itself as a valid windows process like SVCHOST.EXE, you should understand that killing these processes can crash the operating system or cause it to malfunction. There are good programs available than can show more detailed information than we have previously mentioned. One application known as AATOOLS can be downloaded from the web at the following website address:

<http://www.glocksoft.com/aatools.htm>.

**Figure 5.2.F** shows the network monitor portion AATOOLS listing all open ports. The great thing about this tool is we can actually see that port 8888 is open and identify which application it is linked to. Closer inspection to this application shows that all of the other svchost.exe files reside in the c:\windows\system32 folder while the process in question is found in the c:\windows folder. At this point we have enough information to suspect foul play and should remove the computer from the network, to preserve any evidence. The computer should be further investigated for any traces of the attacker and any signs of the attacker's intention.

**Figure 5.2.F A program that maps ports to the relevant application**

Prot.	Local IP	Local Port	Remo...	Rem...	Remote...	State	PID	Process	Path
TCP	192.168.254.33	123				LISTEN	820	svchost.exe	C:\WINDOWS\System32\...
TCP	127.0.0.1	123				LISTEN	820	svchost.exe	C:\WINDOWS\System32\...
TCP	192.168.254.33	138				LISTEN	4	System	
TCP	192.168.254.33	137				LISTEN	4	System	
TCP	0.0.0.0	500				LISTEN	504	lsass.exe	C:\WINDOWS\system32\ls...
TCP	0.0.0.0	445				LISTEN	4	System	
TCP	0.0.0.0	1026				LISTEN	776	svchost.exe	C:\WINDOWS\System32\...
TCP	0.0.0.0	1025				LISTEN	776	svchost.exe	C:\WINDOWS\System32\...
TCP	0.0.0.0	8888				LISTEN	1740	svchost.exe	C:\WINDOWS\svchost.exe
TCP	192.168.254.33	139				LISTEN	4	System	
TCP	0.0.0.0	1037				LISTEN	4	System	
TCP	0.0.0.0	135				LISTEN	668	svchost.exe	C:\WINDOWS\system32\sv...
TCP	0.0.0.0	445				LISTEN	4	System	

## 5.3 Containment

This company happens to have a computer lab that is isolated from the network. It is used for testing and research. The first step the administrator wants to take in the containment process is to preserve any evidence the computer may have on it. He connects a special cloning device used for hard drive duplication and creates an exact copy of the hard drive from the compromised workstation. The device used, The Forensic SF-5000, was purchased from [http://www.logicube.com/products/hd\\_duplication/](http://www.logicube.com/products/hd_duplication/) and is an acceptable means of data seizure because of the non-tampering, bit-by-bit imaging process with very low margin of error. After completion, he shuts down the computer and takes it to the lab where it



can be studied. He then repeats the scanning process of all the computers on the LAN for open ports, especially port 8888. He checks his Symantec Corp Edition Console Center to be sure that all workstations have received updates, and that all computers are running real-time virus protection. The console is set to lock features so that user can not tamper with the virus protection. If the virus protection has been turned off the application is set to restart every 5 minutes. All of this information can be monitored and configured at the Symantec Primary Virus Protection server. He finds no other instances of the current problem and decides not to take down internet connectivity. In a case where 3 or more workstations were showing similar activity this would be an appropriate measure.

#### **5.4 Eradication and Recovery**

The hard drive is removed from the computer at the lab and a spare blank hard drive is re-imaged with image used for setting up new employees on the network. Any applications specific to the user are also installed. This is a fairly quick process because all user files are stored on a network server and each workstation is within 90% of being a carbon copy of the other. In this instance the only thing the HR Director needed after re-imaging was the proper email settings and the accounting client application re-installed. The workstation is then returned to the user. The user is up and running and we are now free to investigate the compromised system further. If this occurrence took place on multiple machines across the LAN then the recovery process would take significantly longer. The internet connection would have to be cut, but the administrator would continue to let people work, while investigating the problem on one of the compromised machines. In this situation, he would manually close the port by killing the svchost.exe file process using AATools on each workstation. He would delete that file from the c:\windows directory. This would prevent connections to the port (8888) that was opened by the application. The next step would be to re-image each user to ensure the integrity of the operating system. The final recovery step is to force each user to create a new password at their next logon session.

#### **5.5 Lessons Learned**

This section is a breakdown of what the attacker did, why it was successful and how it can be prevented in the future.

##### **The attacker was able to install a password sniffing program on his workstation.**

Most platforms are able to deny users from installing applications, unless they have local administrative rights. Windows platforms like the one we discussed in this paper can prevent users from installing applications via the Group Policy for Domains and also by configuring the local policy on each workstation. It is more effective to lockdown a computer through its local policy, but more efficient to lockdown many workstations through the Domain Group Policy.

### **The attacker was able to ARP cache poison 2 computers**

The attacker was able to do this because of the relationship of ARP to TCP/IP and the nature of the how the two protocols are used in network communications. Preventing ARP cache poisoning is nearly impossible. Using the right equipment, like a switch instead of a hub, can make ARP spoofing a more difficult task for the attacker to accomplish. Other steps to prevent ARP cache poisoning include LAN segmentation. Remember ARP is a local segment only communications protocol. Some switches have what is called Virtual LAN capabilities. This is a device that will actually block traffic to a port based on ARP information. It will also let you statically assign MAC addresses to each port that cannot be changed except through the application interface for that switch (meaning an attacker couldn't change the ARP table with remote commands and spoofed addresses, it must be done manually). However this method is not effective with DHCP services on the network and can be an administrative burden. The use of static IP addresses is recommended for smaller networks and whenever possible. Static IP's will give the administrator much more control of the network as far as monitoring is concerned. It will also allow the administrator to deploy third party applications like ARPWatch. These types of applications keep a database of MAC and IP addresses. When it detects that an IP address and MAC address of a packet do not match the entry in the database it will alert the administrator. This solution however will create false positives in a DHCP environment. The focus will have to be on both detection and prevention.

### **The attacker sniffed out a username and an NTLMv2 encrypted password and cracked it using a password cracker**

One of the most important mechanisms for protecting against password cracking is to ensure that you are implementing strict password policies. Ensure that all passwords are a minimum of 8 characters, using alpha, numeric and special characters (!@#\$%). It is also wise to have a set period for password expiration where a user is forced to change his/her password. The longer the period, the more time an attacker has to crack the password. Also, the shorter and less complex the password, the faster it can be cracked. As for sniffing, there are several methods and programs that can detect a sniffer and/or a network card that is in promiscuous mode. The administrator needs to learn how to detect when his network is being sniffed. The following link provides information about sniffer detection.

[http://www.securiteam.com/unixfocus/Detecting\\_sniffers\\_on\\_your\\_network.html](http://www.securiteam.com/unixfocus/Detecting_sniffers_on_your_network.html)

VLANs may also prevent password information from traversing unnecessary areas of the network. It is also wise to use as few resource shares as possible because it shared resource access may require that password information be sent before allowing the resource to be accessed. Upgrading all legacy operating systems and implementing a windows 2000 native mode platform will decrease the chances of a weak password hash being transmitted over the wire. Try to centralize shared resources and keep the number of shared resources that require authentication as low as possible. Always avoid root level shares.

**The attacker was able to create a drive mapping to an administrative share (C\$) on the victim's workstation and drop a Trojan in the startup directory**

The attacker was able to access the C\$ share of the victim because that share is installed by default on every windows operating system. Any sharename followed by a "\$" is referred to as an administrative share. The "\$" prevents the share from being displayed when browsing computers through the windows explorer. Unless you are using Microsoft Operations Manager (MOM) or Microsoft Systems Management Server (SMS), you may want to remove administrative shares from your workstations. Microsoft knowledge base article ID 314984 shows you how to do this. If you can get away with removing default admin shares from you servers then knowledge base article ID 318751 will show you how. I would be careful of this as some applications and services may require the admin shares be present. MS KB article ID 318755 tells you how to restore those shares in the event removing them has caused you grief. The following web address allows you to search for Microsoft knowledge base articles by article ID number.

[HTTP://support.microsoft.com](http://support.microsoft.com) (click on the Knowledge Base Article ID Number Search link)

**The attacker was able to install a hexed version of a Trojan called The Beast into the victim's startup folder.**

The attacker accessed the startup folder through the root share. Following the information previously mentioned, about shared resources and root shares may have prevented access to this directory which would in turn prevent the attacker from being able to execute the trojan's server file. However, where it is necessary to retain the admin shares one should monitor the startup folders and prevent new files from being written to them by non-administrators. Two other ways for an attacker to have malicious code executed is through the task scheduler and the registry. You should disable the task scheduler service using the control panel\administrative tools\services interface. You deny registry edits to non-administrative users and also check the following keys for any suspicious file paths.

HKEY\_Local\_Machine\Software\Microsoft\Windows\CurrentVersion\Run

and

HKEY\_Local\_Machine\Software\Microsoft\Windows\CurrentVersion\RunOnce

\*\*\* Always backup your registry before making changes

**The attacker was able to remotely control the victim pc and use a keylogger to get username and password information for the accounting application**

Whenever possible I always try to design my networks so that clients talk to servers and servers talk to clients. In this manner it is easier to detect an internal attack by monitoring for client to client communications. I assign my servers lower range IP addresses and my clients receive higher range IP addresses. This is very effective for networks with less than 250 nodes. I am able to use my NIDS to alert on any traffic high range IP addresses outside of the normal windows activity. There are also several 3<sup>rd</sup> party applications that detect key logging on a workstation, when installed locally. In this particular



attack the log file of a beast Trojan by default ends with the extension .blf. However this extension can be changed to anything the attacker wants. The attacker does have to download the log file before he can read it. I have not tried to identify a signature for this log file as my skills in writing rules for SNORT are at a novice level. I can however provide the following link which is a tutorial for writing SNORT rules. The link is:

<http://www.cert.org/security-improvement/implementations/i042.14.html>

**The attacker was able to logon to the accounting application and give himself a salary increase under the guise of the HR Director**

Unfortunately, at this point, there is no way to prevent this, except by preventing the steps that lead up to this point. However, the company should change its policy to review all salary increases and possibly implement alerts in the application itself.

**The attacker was able to remove the Trojan, the snort logs and the event logs from the victim's workstation**

We can't do much about the Trojan file being deleted as it was done so by design. The best practice for the log files is two them in two places. This is good for redundancy as well as criminal investigation. Have the evidence in duplicate from two different sources makes for a stronger argument in court. Certain 3<sup>rd</sup> party applications like Tripwire, can alert the administrator when a file is changed, moved or deleted.

© SANS Institute 2004, Author retains full rights.

## 6.0 References and Links

### **6.1 References**

Stallings, William. Data and Computer Communications, Fourth Edition. Upper Saddle River: Prentice-Hall, Inc, 1994 436-463

Vaskez. Hex-Editing Guide by Vaskez.

<http://www.teambg.com/iesdp/General/HexGlossary.htm>

Unknown. Windows 2000 Kerberos Authentication. Microsoft,

<http://www.microsoft.com/windows2000/techinfo/howitworks/security/kerberos.asp>

July 9, 1999

Sharpe, Richard. What is SMB? <http://samba.anu.edu.au/cifs/docs/what-is-smb.html>,

October 8, 2002

### **6.2 Links**

<http://www.areyoufearless.com>

<http://www.oxid.it/cain.html>

<http://www.snort.org>

<http://winpcap.polito.it>

The Beast, Stealth Tools

Cain and Abel

Snort

Winpcap