Global Information Assurance Certification Paper

Session Hijacking in Windows Networks

GSEC Gold Certification

Author: Paul Jess, paul_jess@urscorp.com

Adviser: Richard Wanner

Accepted: October 12th, 2006

Introduction

Before we can explore the session hijack attack, it is essential that we gain a basic understanding of network communications. The first section of this paper covers some of this background information needed to understand how computers communicate on a network. First we take a look at the TCP/IP protocol (Transmission Control Protocol/Internet Protocol) examining a concept critical to network communication called the three-way-handshake. Once we have a basic understanding of these concepts, we can then work towards understanding how the session hijack attack exploits the design flaws inherent in the TCP/IP protocol.

In section two, the session hijacking attack is defined. The benefits of the attack are closely examined as well as the danger the attack presents to your network.

Section three examines the session hijack attack in detail. The session hijack attack is broken down into five steps including locating a target, finding an active session, sequence number prediction, taking a user offline, and taking over a session.

Detecting the session hijack attack on a network can be very difficult. In section four of the paper, session hijacking detection will be examined. Attack signatures will be examined and real world examples provided.

Section five examines session hijacking software applications that are used by attackers to compromise computers. Windows and Linux/Unix applications will be reviewed, and their features described.

Section six discusses the various ways in which session hijacking can be detected on the network. Using Wireshark, packet captures of a session hijack attack are examined.

Section seven looks at the various countermeasures that can be implemented on your network that will help reduce your exposure to this attack. Microsoft Group Policy Objects, IPSec, IDS and IPS systems, and insecure network protocols and operating system will be examined.

1 TCP/IP Basics

Before we explore the session hijack attack, readers must possess a basic understanding of how computers communicate with one another on a network. In the section that follows, we will look at some basic elements of TCP/IP (Transmission Control Protocol/Internet Protocol) protocol specifically the concepts of the three-way-handshake and random initial sequence number generation.

In order for two machines to communicate on a network they have to negotiate common communication parameters. This is done by transmitting a series of data packets between the two machines in a process known as the three-way-handshake. All computers on the network must complete this process in order to establish a connection with another computer on the network.

1.1 Three-Way-Handshake

When two computers want to communicate with one another, they have to negotiate the technical parameters that they will use to communicate with one another. This is done through a process known as the three-way-handshake. Once the connection is established, the session remains open until one of the machines sends a RST (reset) or FIN (finish)

packet to their communication partner.

The three-way-handshake is comprised of three main processes. Please note that the discussion that follows is based on a scenario where a single workstation computer is attempting to communicate with a file server.

1.1.1 Step 1 - SYN

When a workstation wants to communicate with a server it builds a packet with the SYN or synchronization bit set and then sends the packet to the server. Included in this SYN packet is an initial sequence number (denoted in figure 1 as X) (Lam, 2006).

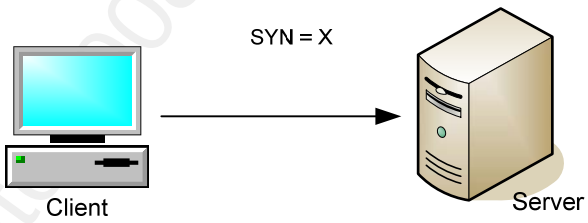

Figure 1 – Step 1 SYN

When the client computer generates the sequence number, it uses a random number generator. Random number generators are used to help prevent communication sessions from being compromised (more on this in the next section). Sequence numbers are critical to network communications as they are used to guarantee packet delivery. Source computers

use sequence numbers for tracking incoming packets and reassembling them as they arrive at their destination. From the attacker's perspective; however, the ability to predict sequence numbers provides the mechanisms needed to successfully hijack a communication session.

1.1.2 Step 2 – SYN/ACK

When the server receives the clients SYN (synchronization) packet, it responds to the workstation computer with a packet containing both the SYN and ACK (Synchronization and Acknowledgement) bits set. The packet includes the server's own randomly generated sequence number (represented in the drawing by the letter P). The server also acknowledges the clients sequence number by adding 1 to the sequence number sent by the client computer (X + 1) (Lamb, 2006).

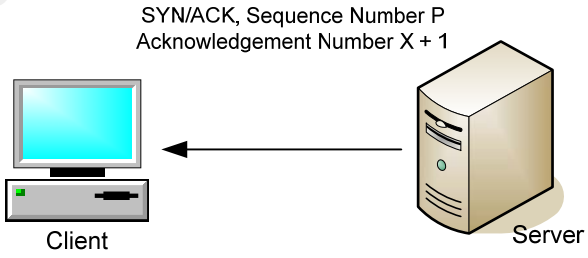

Figure 2 – Step 2 SYN/ACK

1.1.3 Step 3 - ACK

The final phase of the three-way-handshake involves the client sending an ACK packet to the server confirming its desire to communicate. The workstation prepares a packet with the ACK (acknowledgement) bit set and includes an acknowledgement sequence number (X + 1). When the packet arrives at the destination server, the communication session is established and communication can now begin. An active communication session will be maintained until one of the machines sends a RST (Reset) or FIN (Finish) packet to the other computer (Lamb, 2006).

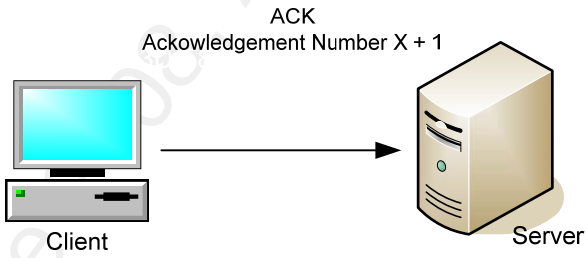

Figure 3 – Step 3 ACK

The following screen output from Wireshark shows what a three way handshake looks like in Wireshark. Packet number three begins the three-way-handshake process by sending a SYN (synchronization) packet to the server. The server then acknowledges the receipt of the SYN packet by sending the workstation computer a SYN/ACK (Synchronization/Acknowledgement) packet (show in packet four). The final step in the three-

way-handshake is an ACK packet sent to the workstation by the server.

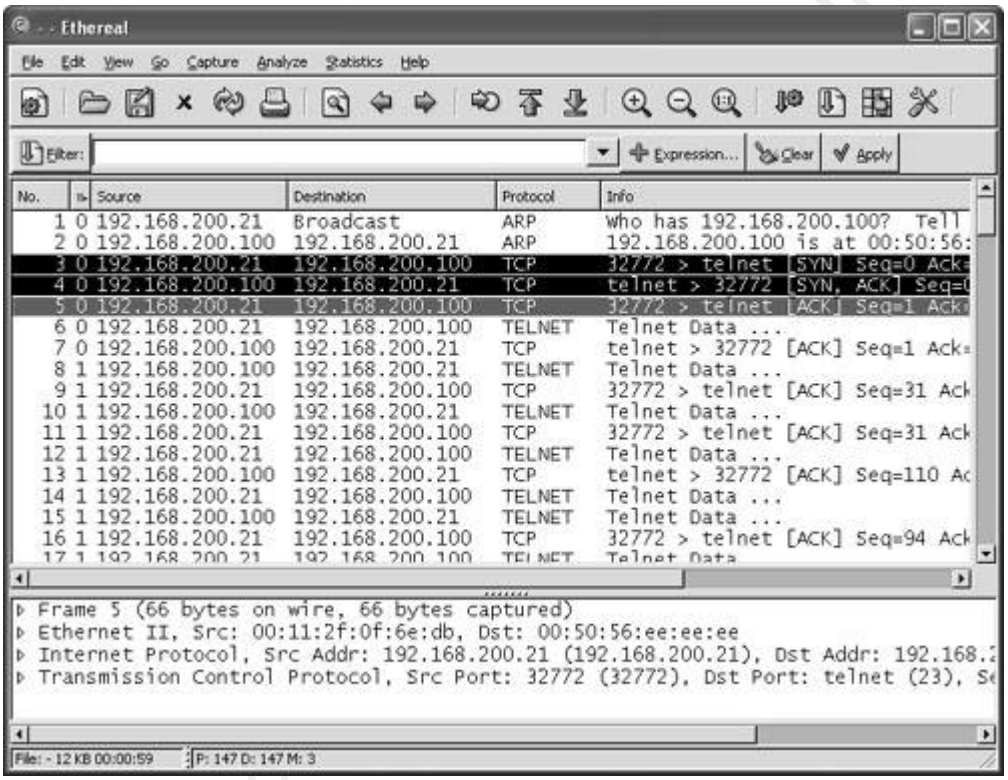

Figure 4 - Example of the Three-Way-Handshake in Wireshark (Resultspk.com, 2006)

1.2 Sequence Numbers

Sequence numbers are an essential component of network communications. It is the sequence number that insures reliable communication on the network. As packets leave the transmitting computer, each packet is assigned a unique sequence number. Sequence numbers provide a mechanism which allows the receiving computer to track incoming packets

and reassemble then into a logical stream of data. Sequence numbers can also be used to detect packets that have not arrived at the destination computer. When packet loss is detected, the destination machine notifies the source computer to resend the missing packet.

TCP/IP sequence numbers are 32-bit numbers, thus providing four million possible number combinations. While this seems like a sufficient quantity of numbers to reduce the chance of sequence number prediction, modern computers make this number arbitrary. Most modern operating systems implement pseudo random number generators that produce complex sequence numbers sufficient enough to make sequence number prediction difficult if not impossible. However, older operating systems, such as Windows NT 4.0, did not provide sufficient random number generation as discussed in Microsoft knowledge base article MS99-046. As you will see in the sections that follow, sequence numbers, and the ability to predict sequence numbers, are a vital component necessary to successfully wage a session hijack attack.

2 Session Hijacking Definition

If you are like most security professionals, session hijacking is not an attack that gets a lot of your attention. In recent years, the session hijack attack has been overshadowed by spyware, root kits, bot networks, and denial of service attacks. Although the session hijack

attack is not in the forefront of everyone's mind, it still remains a commonly used attack. In fact, Kevin Mitnick used many of the underlying principals common to session hijacking in his famous breach of Tsutomu Shimomura's computers (Meriwether, 1995).

Session hijack attacks are defined as taking over an active TCP/IP communication session without their permission or knowledge. When implemented successfully, attackers assume the identity of the compromised user, enjoying the same access to resources as the compromised user.

Session hijack attacks are usually waged against users that are members of large networks containing a substantial number of open sessions. Network protocols like FTP, Telnet, and rlogin are especially attractive to the attacker, because of the session oriented nature of their connections, and the length of their communication sessions. Additionally, FTP, TELNET, and rlogin do not implement any security during logon, authentication, or data transmission. In fact, data sent using these protocols is sent in clear text which can be easily be viewed by anyone monitoring the network.

There are three different types of session hijack attacks; active, passive, and hybrid. The active attack is when the attacker hijacks a session on the network. The attacker will silence one of the machines, usually the client computer, and take over the clients' position in

the communication exchange between the workstation and the server. The active attack also allows the attacker to issue commands on the network making it possible to create new user accounts on the network, which can later be used to gain access to the network without having to perform the session hijack attack.

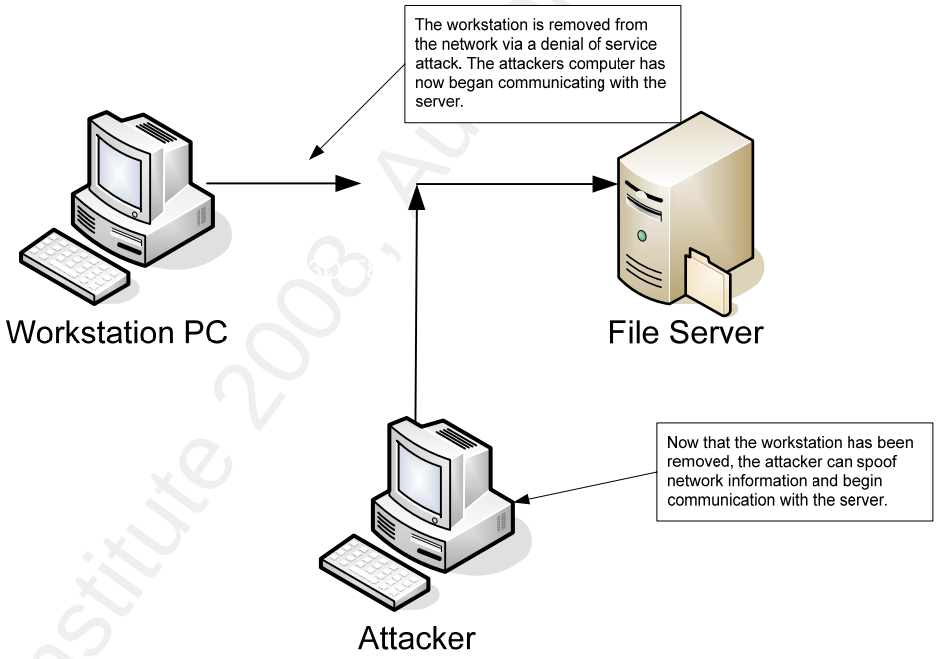

Figure 5 - Aggressive Session Hijack Attack

Passive session hijack attacks are similar to the active attack, but rather than removing the user from the communication session, the attacker monitors the traffic between the

workstation and server. The primary motivation for the passive attack is it provides the attacker with the ability to monitor network traffic and potentially discover valuable data or passwords.

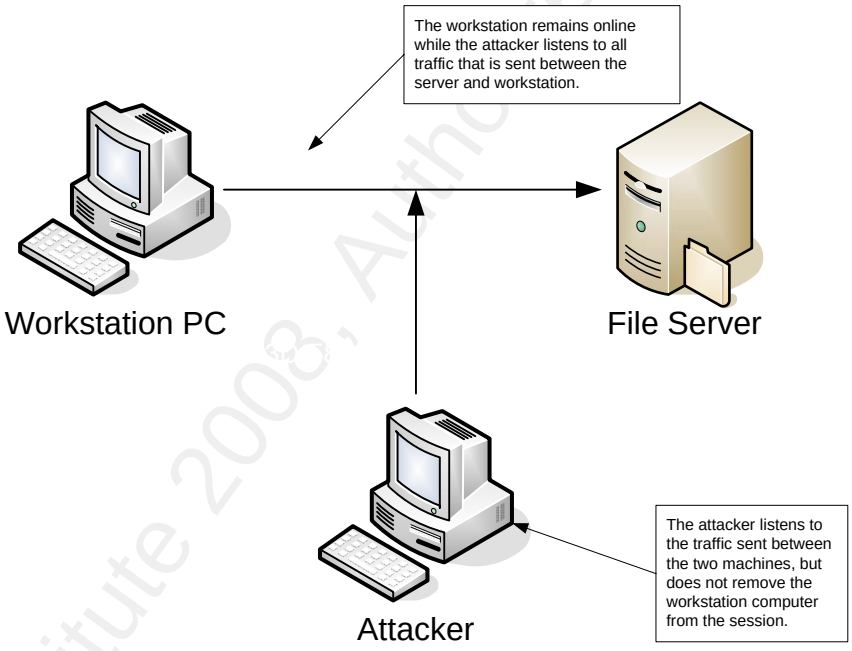

Figure 6 - Passive Session Hijack

The final type of session hijack attack is referred to as the hybrid attack. This attack is a combination of the active and passive attacks, which allow the attacker to listen to network traffic until something of interest is found. The attacker can then modify the attack by removing the workstation computer from the session, and assuming their identity.

2.1 Advantages of Session Hijack for the Attacker

So what makes the session hijack attack worthwhile for the attacker? One of the most valuable byproducts of this type of attack is the ability to gain access to a server without having to authenticate to it. Once the attacker hijacks a session, they no longer have to worry about authenticating to the server as long as the communication session remains active. The attacker also enjoys the same server access as the compromised user because the user has already authenticated to the server prior to the attack.

A successful session hijack attack also allows the attacker to issue commands to servers on the network. This is usually done to create user accounts that can be used to access resources at a later date. The ability to issue commands also provides a way to mask the attacker's presence on the network, by removing or altering the remnants of the attack.

The session hijack attack is very stealthy. Session hijack attacks are usually waged against busy networks with a high number of active communication sessions. The high network utilization not only provides the attacker with a large number of sessions to exploit, but it can also provide the attacker with a shroud of protection due to the large number of active sessions on the server.

Most network attacks depend on software or hardware vulnerabilities as a gateway to

an attack. Having knowledge of specific vulnerabilities in these technologies allow the attacker to scan servers to determine what vulnerabilities exist. However, the session hijack attack does not depend on specific software or hardware vulnerabilities, but rather a design limitation within the TCP/IP protocol that does not guarantee security after the connection is made.

Session Hijacking is also very easy to do, especially on older operating systems! Utilizing commercially available software packages, even a novice computer user has a good chance at successfully waging a session hijack attack.

2.2 What Makes the Attack so Dangerous?

Why is the session hijack attack so dangerous? Should security professionals really be concerned? The answer to both of these questions is yes! As I hinted in the previous paragraph, the risks resulting from session hijack attacks can not be eliminated by software patches, complex passwords, or multi-factor authentication. The root cause of the attack lies with design limitations inherent to the TCP/IP protocol. In addition, all machines regardless of operating system or hardware architecture are vulnerable to the session hijack attack provided they are running TCP/IP.

The attack also exploits all three sides of the CIA triad. The CIA triad is a

representative model of security concepts consisting of three underlying principals. Confidentiality, integrity, and availability make up the triad and A failure on any side of the triad represents a compromise in network security (Cole, E. & Fossen, J. & Northcutt, S. & Pomeranz, H., 2005).

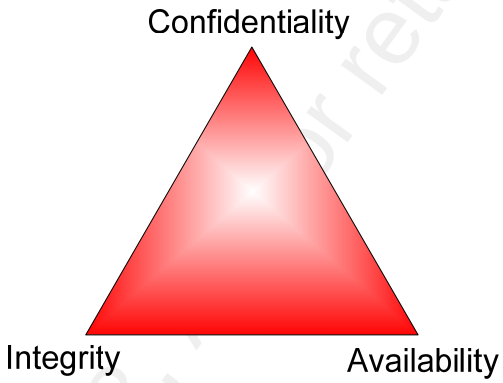

Figure 7 - CIA Triad

The session hijack attack compromises all three sides of the CIA triad. When a successful attack is achieved, the attacker has the ability to read and modify data, violating the confidentiality and integrity portion of the model. Availability is also affected by the session hijack attack due to ARP storms and denial of service conditions that are a byproduct of the attack.

3 The Session Hijack Attack

The session hijack is a process whereby the attacker inserts themselves into an existing communication session between two computers. Generally speaking, session hijack attacks are usually waged against a workstation server type of communication session; however, hijacks can be conducted between a workstation computer communicating with a network based appliance like routers, switches or firewalls.

3.1 Procedural Overview of the Session Hijack Attack

As outlined in the book by Eric Cole, Hackers Beware: The Ultimate Guide to Network Security, the session hijack attack contains the following steps (Cole, 2002).

3.1.1 Step 1 - Locating a Target

The first step in the session hijack attack is locating a target user. Attackers look for two things prior to their attack. First, they look for networks that have a high level of utilization. High volume networks provide a healthy supply of users to choose from, which also helps the attack remain anonymous. Secondly, users who frequently use insecure network protocols such as Telnet, rlogin (remote logon), and FTP (file transfer protocol) are also frequent targets due to their inherently insecure design.

Packet sniffing software can be used to sniff network traffic for the purpose of locating vulnerable protocols like FTP, Telnet, and rlogin. Port scanning software can also be used to identify servers that have FTP, Telent, or rlogin ports open.

3.1.2 Step 2 - Find an Active Session

Session hijack attacks are usually waged against servers with large amounts of activity. The reason is twofold, high network utilization provides an environment containing adequate sessions that can be exploited. Secondly, the high usage on the server helps hide the disruption caused by the attack. Attackers generally target session oriented protocols like FTP, Telnet, and rlogin which provide prolonged connections to other computers.

Attackers who are looking for open sessions generally use software tools like Wireshark or more sophisticated site detection software that is included in some of the popular session hijacking software packages like T-Sight or Juggernaut.

3.1.3 Step 3 - Perform Sequence Number Prediction

Now that a target has been chosen, the next step in the session hijack process is sequence number prediction. This process entails guessing the next sequence number that the server is expecting from the workstation. Sequence number prediction is a critical step, because failing to predict the correct sequence number will result in the server sending reset

packets and terminating the connection attempt. If the attacker guesses the sequence numbers wrong repeatedly, the likelihood of detecting the attack increases.

So how do you accurately predict the next session number? While sequencing number guessing can be done manually by skilled attackers, software tools are available to automate the process. Programs such as Juggernaut (www.packetstorm.securify.com), Hunt (http://fsid.cvut.cz/~kra/index.html), and T-Sight (http://www.engarde.com/software/t-sight/) are very effective tools that can be successfully used by attackers of moderate skill levels.

3.1.4 Step 4 - Take One of the Parties Offline

Once a session is chosen and sequence numbers predicted, you need to silence the workstation computer. This is generally done with a denial of service attack; however, any attack that renders the computer unable to communication on the network would work just as well. The attacker must ensure that the client computer remains offline for the duration of the attack or the client computer will begin transmitting data on the network causing the workstation and the server to repeatedly attempt to synchronize their connections resulting in a condition known as an ACK storm.

Taking the client computer offline is only done in an aggressive session hijack attack. Remember, the passive attack is used to view data as it flows across the network; therefore,

removing the workstation in this scenario would prohibit the attacker from examining the communications between the two machines.

3.1.5 Step 5 - Take over the Session and Maintain the Connection

The final phase of the session hijack attack entails taking over the communication session between the workstation and server. The attacker will spoof their client IP address, to avoid detection, and include a sequence number that was predicted earlier. If the server accepts this information, the attacker has successfully attacked the communication session. Because the attackers' source address has been spoofed, the attacker will not receive any feedback regarding the status of the attack. As a result, the attacker will have to understand what the server is expecting to maintain the attack.

At this point in the attack, full access to the network is limited only by the permissions of the compromised user or computer. Provided that the TCP/IP session is maintained, the attacker will not have to repeat the hijack process for the duration of the connection.

4 Session Hijack Tools

While session hijacking is possible without the assistance of hijacking software, many attackers choose to use software tools due to their ease of use. The session hijacking tools available today provide precision, timing, and session prediction capabilities.

Juggernaut is one of the most popular software packages for session hijacking and it runs only on the LINUX operating system. Juggernaut contains a built in network sniffer which aids in the hijacking process and allows the attacker to watch for keywords as they flow across the network. Juggernaut is frequently used when attackers want to capture passwords as they flow across the network.

Hunt, another UNIX based software application, is primarily used for session hijacking attacks where attackers want to listen and intercept network communications, as well as hijack open sessions on a network. Sequence number prediction, and silencing the workstation computer are all handled internally by the software.

T-Sight, written for the Windows Operating system, is a commercially available product that provides most of the functionality of the UNIX software variants. The application can be purchased from Engarde at the following web site (http://www.engarde.com/software/t-sight). This commercial application was intended to be used by professional security engineers; however, it is very effective in the hands of an attacker. T-Sight automates the selection of open sessions, provides accurate sequence number predication, and is capable of silencing target workstations.

5 Detecting Session Hijack Attacks

There are two primary technologies that assist in session hijack detection. The more manual of the two methods is packet sniffing software which can be used to scan for signatures of an attack. Intrusion detection systems (IDS) and intrusion prevention systems (IPS) provide a more automated method of detection, but they can also create more analysis work for the security administrator.

5.1 Packet Sniffers

Packet sniffers are software applications that possess the ability to capture packets as they flow across the network. Once captured, the contents of the packets can be examined using a variety of filtering tools. One of the most popular packet sniffers on the market is called Wireshark and it is available for free at http://www.wireshark.org/.

Using packet sniffing software to detect session hijack attacks can be very difficult. Doing so requires the user to configure the software to scan the network while displaying the results to the computer screen in real time. After initiating the scan, the operator would have to analyze the data in real time as it is displayed on the screen. Due to the difficultly surrounding this method, packet sniffing software is generally used as an investigative tool rather than a front line detection or defense tool.

The following Wireshark screen shots show us what the various steps of the session hijack attack look like within Wireshark.

5.1.1 Normal Telnet Session

Now that the communication session has been established, the client and server can communicate via the Telnet protocol. This screen shot shows what a normal telnet communication session looks like in Wireshark. You can clearly see data packets being sent to the server and subsequently acknowledged by the server.

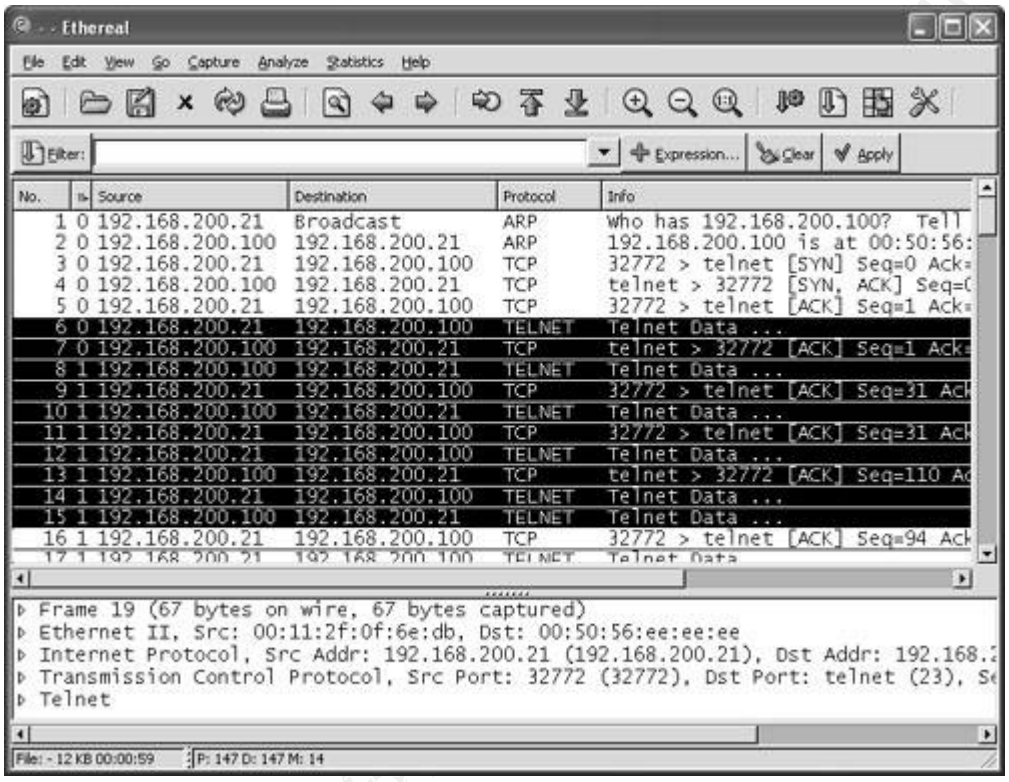

Figure 8 - Normal Telnet Traffic (Resultspk.com, 2006)

5.1.2 The Attack Begins - Forcing an ARP Entry

The next screen shot shows one of first critical steps in a session hijack attack, forcing an ARP cache change on the server. In order to assume the identity of the attacked computer, the attacker must force a MAC to IP address update on the server. Forcing an ARP entry on the server tricks the server into sending subsequent data to the attacker's computer rather than the attacked workstation. The attacker will update the MAC address that resides in the ARP cache of the server substituting an alternate MAC address in place of the

compromised workstation. The primary goal of the ARP entry change is to make sure that no network traffic gets sent to the compromised workstation. If this phase of the attack is not done correctly, it could cause the workstation to respond to the server resulting in the resynchronization of the connection.

MAC to IP address changes on a network can be a strong indicator of a possible session hijack attack and should be investigated closely. The screen shot below shows what a MAC to IP address change looks like in Wireshark.

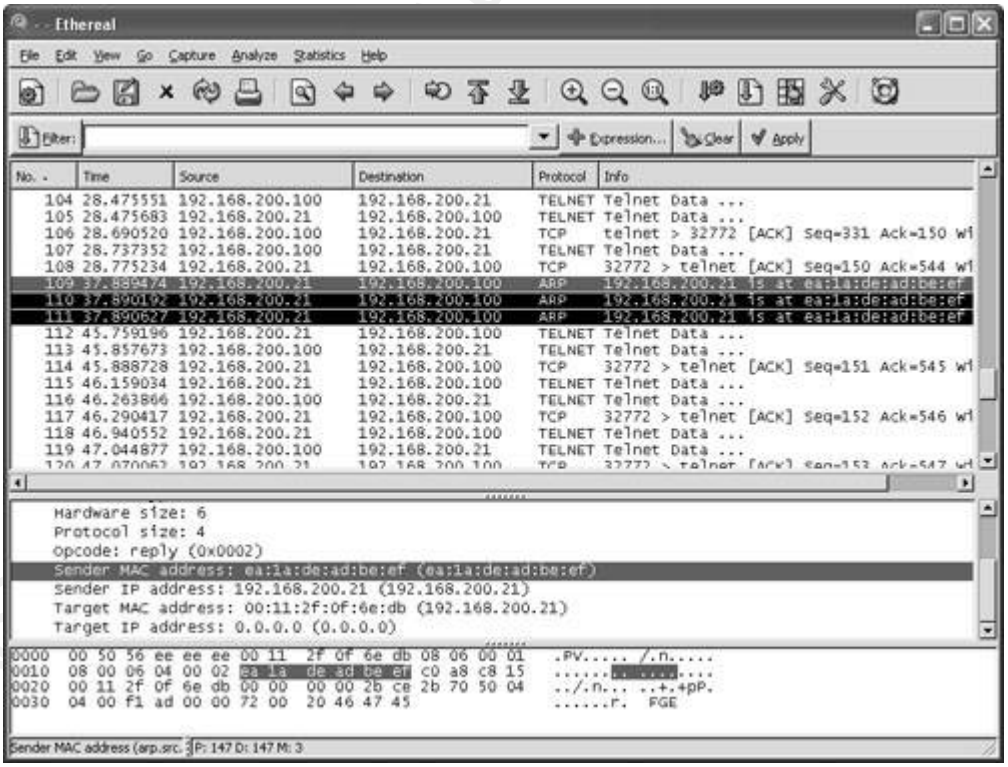

Figure 9 - Forcing ARP Entry (Resultspk.com, 2006)

5.1.3 Hijack Traffic

The next section, we will look what appears to be a normal communication session between and workstation and a server. There is; however, one small anomaly that becomes evident when you closely inspect the details of the packet.

In the packet data below, the source address in frame 112 is the MAC address of the client computer that is sending the data. The dst (destination) address listed in the packet is the MAC address of the server. Up to this point, everything looks normal until we analyze the next packet sent across the network, frame 113.

```
-------------------------------------
Frame 112 sent from hijacker to server
-------------------------------------

Source                Destination
192.168.200.21-----> 192.168.200.100
Ethernet II, Src: 00:11:2f:0f:6e:db, Dst: 00:50:56:ee:ee:ee
Destination: 00:50:56:ee:ee:ee (192.168.200.100)
Source: 00:11:2f:0f:6e:db (192.168.200.21)
Internet Protocol,
Src Addr: 192.168.200.21 (192.168.200.21)
Dst Addr: 192.168.200.100 (192.168.200.100)
Transmission Control Protocol
Src Port: 32772 (32772)
Dst Port: telnet (23)
Seq: 150, Ack: 544, Len: 1
```

MAC address of the attacked client

MAC address of the server

Figure 10 – Normal Telnet Traffic (Resultspk.com, 2006)

In normal network communications, the source and destination MAC addresses address will remain consistent throughout the communication session. If you closely analyze frame 113, you'll notice that the destination MAC address (this should be the MAC address of the client machine) is not the same address that is listed in frame 112. Instead, the MAC address is ea:1a:de:ad:be:ef. Close examination of the MAC address shows that the last four octets of the MAC address spells out "dead beef". This unusual MAC address is a clear indicator that T-Sight has been used on your network.

Frame 113 Response from server to the client (hacker)

Source Destination

192.168.200.100-----> 192.168.200.21

Ethernet II, Src: 00:50:56:ee:ee:ee, Dst: ea:1a:de:ad:be:ef

Destination: ea:1a:de:ad:be:ef (192.168.200.21)

Source: 00:50:56:ee:ee:ee (192.168.200.100)

Internet Protocol,

Src Addr: 192.168.200.100 (192.168.200.100)

Dst Addr: 192.168.200.21 (192.168.200.21)

Transmission Control Protocol

Src Port: telnet (23)

Dst Port: 32772 (32772)

Seq: 544, Ack: 151, Len: 1

This is not the MAC address for the client workstation. Frame 112 shows a MAC address of 00:11:2f:0f:6e:db

This is the MAC address of the server

The MAC address spells out dead beef

Figure 11 - Telnet Packet with Incorrect MAC Address (Resultspk.com, 2006)

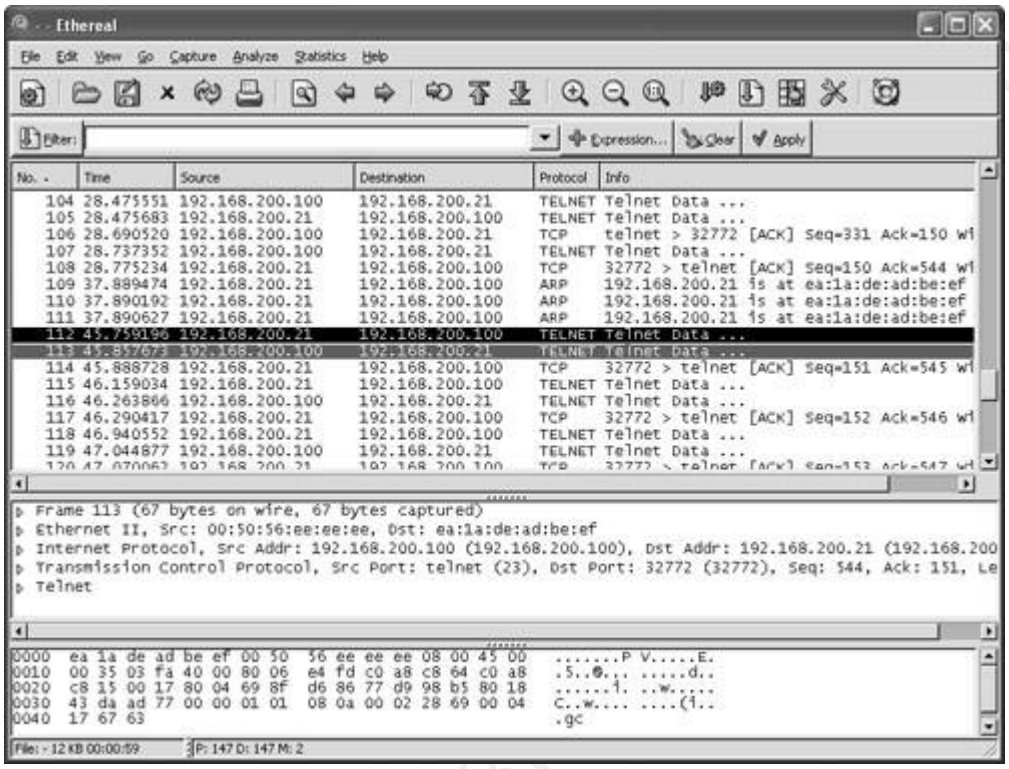

Figure 12 - Complete Session Hijack of a Telnet Session (Resultspk.com, 2006)

The final indicator of a session hijack attack is a phenomenon called an ACK storm. In a session hijack attack, the attacker and the server begin communicating by sending data back and forth. As data is sent, the sequence numbers generated by the workstation and server continue to increase. If the original workstation computer comes back online and sends data to the server, it soon discovers that its sequence numbers are incorrect and attempts to resynchronize them with the server. As a result, the client computer will attempt to synchronize its sequence number with the server, causing a furry of communications between the two machines. The end result is a significant increase in ACK packets on the network,

sometimes to the point where network communications becomes seriously degraded. In Wireshark, the ACK storm will show numerous TCP out-of-order frames. If these packets are discovered on your network, additional research should be conducted to ascertain the root cause.

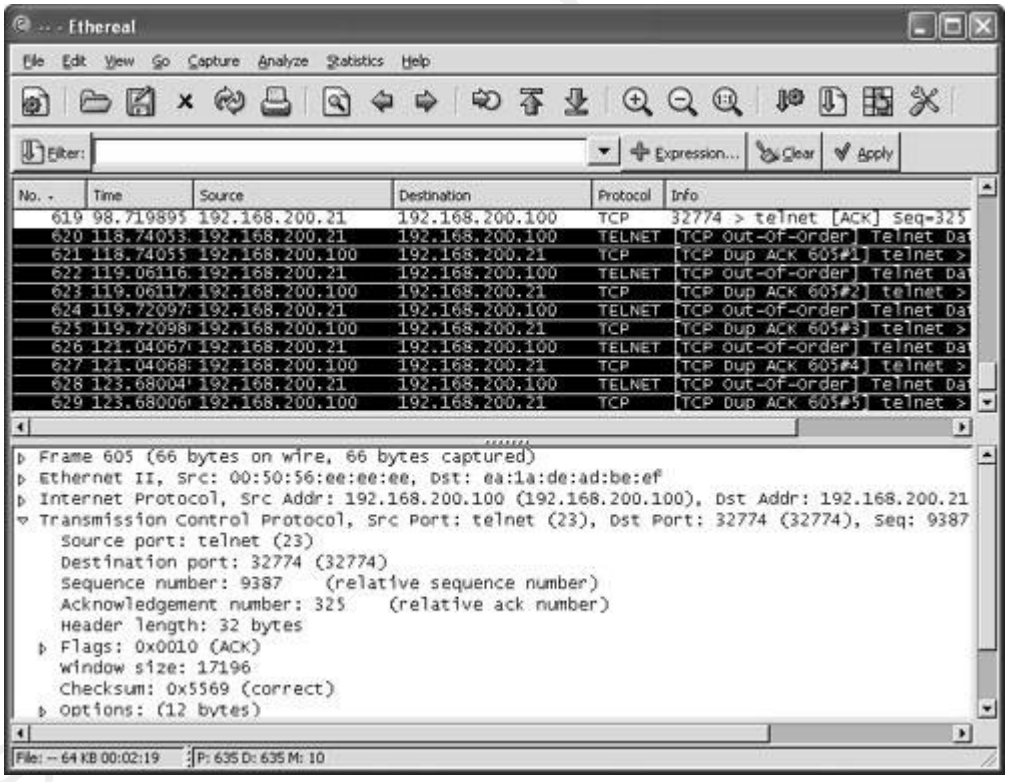

Figure 13 - ACK Storm (Resultspk.com, 2006)

6 Session Hijacking Remediation

Defense in depth is a key component of a comprehensive security plan. Defense in depth is also a key component in protecting a network from session hijack attacks. Defense in depth is defined as the practice of using multiple security systems or technologies to prevent network intrusions. The central idea behind the concept is that if one counter measure fails, there are additional levels of protection remaining to safe guard the network. Defense in depth also requires the attacker to penetrate the numerous layers of security, thus slowing them down allowing additional time for security administrators to detect and defend against the attack.

A good example of the defense in depth strategy can be seen in a new firewall configuration strategy. Many high secure networks implement several firewall types to achieve a defense in depth strategy. In figure 14 below, you can see three different types of firewalls configured in series.

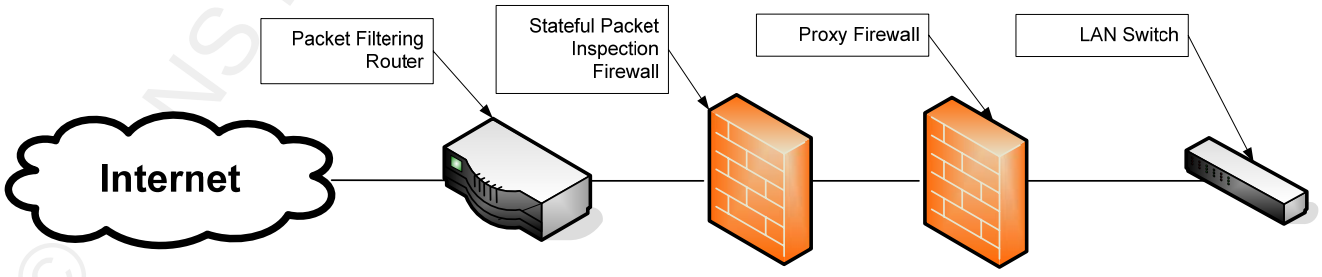

Figure 14 - Defense in Depth Example

The above example utilizes the defense in depth concept by integrating three different types of firewall devices at the network perimeter. These devices all provide the same basic functionality, to protect the network from unauthorized access, but they provide these services in different ways. Packet filter firewall devices are actually built into most modern routers and are used as a first line of defense. These devices filter out a great deal of network “noise” by blocking packets with specific ports. The next line of defense consists of stateful packet inspection firewalls, which possess the ability to inspect the packets in greater detail before allowing them into the network. The final layer of firewall defense is the proxy firewall providing the ability to analyze the entire packet before allowing it on the internal network. The above referenced firewall design is a good example of the defense in depth strategy.

Session hijack attacks are very difficult to detect on busy networks. There are tell tale signs, like computers getting disconnected from the network or periodic network congestion, but these signs usually get ignored by users as “typical network problems”. There are several steps a network administrator can take to preemptively protect their network. Remember, defense in depth is critical to an effective security plan, and when possible, multiple layers of protection should be implemented.

6.1 Protect Against Spoofing

Many of the same security principals that protect the network from a spoofing attack are effective in preventing a session hijack attack. By definition spoofing is defined as slang word that refers to the act of fooling; that is, presenting a false truth in a credible way (Hassell, 2006). In the context of information technology, spoofing is defined as a user who transmits packets on a network with an IP address other than their own. Spoofing is generally used to conceal the identity of the attacker and is frequently used in network based attacks. Session hijack attacks have many of the same attributes as a spoofing attack; therefore, assembling a comprehensive security plan aimed at preventing spoofing provides some protection against the session hijack attack.

One of the easiest methods to combat spoofing is to remove administrative rights on all workstation computers. Doing so prohibits internal spoofing attacks by preventing users from modifying network card settings on their computer.

Ingress and Egress filtering is an excellent way to prevent spoofing. Ingress filters examine the packets as they arrive on the router interface, and egress filtering examines packets as they leave the network. If the source address matches the address scheme of the internal network, the packets are dropped. All routers on your network should be configured with ingress and egress routing filters to prohibit this type of data from being allowed in or out

of your network.

In normal network communications, packets are routed across a network via a series of routers that make the decisions on how the packets are routed to their final destination. Source routing allows users to bypass the traditional routing process and specify the path they want their packets to take. Source routing is valuable tool for the attacker because it allows the attacker to route traffic through their computer making it possible to see the all communications occurring between these computers. The defense against source routing includes ingress and egress filter policies prohibiting source routing packet to be processed by the router.

To defend against source routing, ingress and egress filters should be configured to prohibit source routing packets from traversing through the router.

6.2 IPSec and Encryption

Currently, TCP/IP (Transmission Control Protocol/Internet Protocol) does not provide a mechanism for confidentiality or integrity of the packets sent across a public or private network. However, security can be achieved by implementing IPSec (Internet Protocol Security).

IPSec is a mature, state-of-the-art, Engineering Task Force (IETF) designed security protocol that provides defense in depth against network based attacks from untrusted computers. IPSec protects data by processing data at levels below the application network layer, thus making the IPSec process transparent to most applications.

IPSec includes two different types of IPSec headers including ESP (Encapsulated Security Payload) and AH (Authentication Header). Encapsulated Security Protocol is commonly used when sending data across untrusted external networks. ESP headers provide data integrity, data confidentiality, and data origin authentication. As depicted below, ESP has the ability to encrypt data on the network, thus providing confidentiality to data transverses the network. This is especially useful when connecting to remote servers via FTP and Telnet.

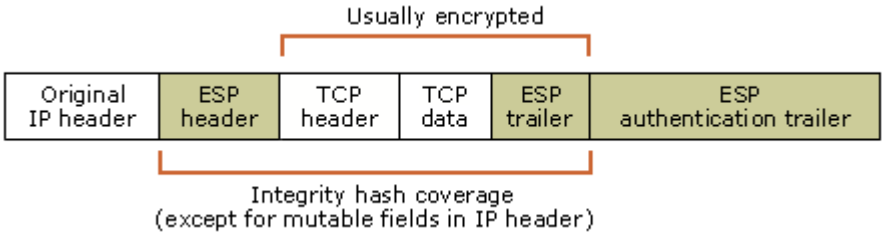

Figure 15 – IPSec ESP Header (Dixon, 2003)

The second IPSec header is Authentication Header (AH). The AH header provides for data integrity, anti-replay, and origin authentication. AH is similar to ESP, but it does not provide the ability to encrypt data. Due to its origin authentication and data integrity capabilities, AH is commonly used on local area networks to defend against spoofing or session hijack attacks.

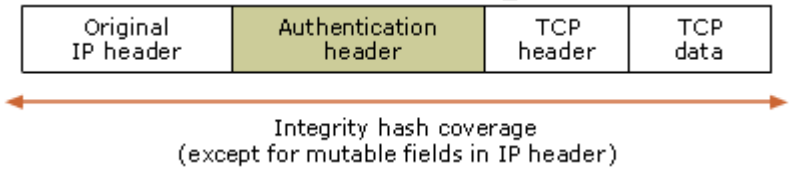

Figure 16 IPSec AH Header (Dixon, 2003)

6.3 Intrusion Detection Systems and IPS Intrusion Prevention Systems

Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) can be very useful in defending your network from session hijack attacks. While implementing these devices can be difficult, the benefits far out weight the steep implementation costs. IDS/IPS systems look at the data that enters the network and compares it to an internal database of known attack signatures. If the packet is matched against an entry in the IDS/IPS database, the IDS will generate an alert, and the IPS will block the traffic from entering the database.

IDS/IPS devices have been successful in preventing session hijack attacks early in the

attack by detecting some of the early components of the attack. These devices possess the ability to detect ports scans used to locate possible targets, packet anomalies created by early phases of the attack, as well as ARP storms that are byproducts of the attack itself.

6.4 Eliminating Insecure Network Protocols and Operating Systems

Insecure protocols such as FTP (file transfer protocol) rlogin (Remote Login) and Telnet present a significant risk to networks. From the context of the session hijack attacker, these protocols are good targets because the communication sessions tend to be persistent, making it easier to wage a successful attack. The authentication mechanisms of FTP and Telnet also present a significant risk to the server. Both of these protocols use unsecured authentication mechanisms and transmit passwords across a network in clear text.

Both of these protocols should be replaced with secure equivalents such as Secure FTP (http://security.sdsc.edu/software/secureftp/) and Secure Telnet (http://216.147.98.109/support_cp_ssh.html). If you are responsible for managing switches or routers within your network, contact your hardware vendor to see they have integrated secure remote access technologies built into their devices. If devices on your network do not support secure remote management, then remote administration functionality should be disabled.

Operating systems that lack the ability to generate complex initial sequence numbers

should also be avoided. Based on a study conducted by Michal Zalewski, the following operating systems do not provide suitable random number generators and should be avoided in a secure computing environment (Zalewski, 2002).

- Netware 6 (Netware 6 SP3 does provide adequate sequence number generation)
- IRIX 6.5.16
- Windows NT 4.0 server and workstation
- Windows 9x (95/98/ME)
- Tru64
- OS400
- AIX 5.1 or lower
- Open VMS v7.2

6.5 GPO - Group Policy Objects

Beginning with the introduction of Windows 2000 Server, Microsoft introduced a

technology called group policy that allows administrators to distribute a variety of computer settings to all machines in the domain from a central point of management.

A Windows 2003 Active Directory domain contains a numerous group policy objects that can provide a greater level of security against a session hijack attack. As part of Microsoft's trusted computing initiative, a great deal of security information has been created and posted on Microsoft's security web site which can be found at http://www.microsoft.com/security/default.mspx and http://www.microsoft.com/technet/security/default.mspx. The information outlined in the following sections came from two very important security documents that are a must read for all Windows Server administrators. The first document is called the Windows 2003 Server Security Guide (http://www.microsoft.com/technet/security/prodtech/windowsserver2003/w2003hg/sgch00.mspx) and the Microsoft Threats and Countermeasures Guide (http://www.microsoft.com/technet/security/topics/serversecurity/tcg/tcgch00.mspx). The Microsoft Windows 2003 Security Guide provides detailed information on how to configure your Active Directory domain structure and defines the security policies that should be set. The Microsoft Threats and Countermeasures guide digs deeper into the technical elements of each GPO setting as well as potential problems that may be encountered as you deploy these

settings.

The group policy material outlined below is based on a Windows 2003 Native domain structure with client computers running Windows 2000 Professional or Windows XP Professional workstations. Networks consisting of older operating systems like Windows 95/98/ME or Windows NT 4.0 can not be secured using GPO's due to their lack of support for Active Directory.

Name	Description	Setting
Domain Controller: LDAP Server Signing Requirements	This policy determines whether the Lightweight Directory Access Protocol (LDAP) server requires LDAP clients to negotiate data signing. This GPO prevents an intruder from capturing packets sent between the LDAP server and the client computer. Prevents an attacker from modifying packets sent between the domain controller and workstation computer.	Enable
Domain Member: Digitally Encrypt or Sign Secure Channel Data (Always)	When enabled, a secure channel cannot be created with any domain controller that cannot sign or encrypt secure channel data. This GPO prevents man-in-the-middle attacks between a member machine and the domain controller.	Enable
Domain Member: Digitally Encrypt Secure Channel Data (When Possible)		Enable
Domain Member: Digitally Sign Secure Channel Data (When Possible)		Enable
Domain Member: Digitally Encrypt or Sign Secure Channel Data (Always)	A secure connection cannot be established with any domain controller that cannot sign or encrypt all secure channel data. This setting encrypts all of the data that flows between the computer account and the domain controller.	Enable
Domain Member: Digitally Encrypt Secure Channel Data (When Possible)		Enable
Domain Member: Digitally Sign Secure Channel Data (When Possible)		Enable
Domain Member: Require Strong (Windows 2000 or Later) Session Key	Determines if the computer can establish a secure channel with a domain controller that cannot encrypt secure channel traffic with strong 128 bit session key. When enabled, all outgoing secure channel traffic will require a strong Windows 2000 or later session key.	Enable
Microsoft Network Client: Digitally Sign Communications (Always)	Requires digitally signed communications on SMB (server message block) communications. This policy will prevent impersonation of clients and servers.	Disable
Microsoft Network Server: Digitally Sign Communications (Always)		Disable

Microsoft Network Client: Digitally Sign Communications (If Server Agrees)		Enable
Microsoft Network Server: Digitally Sign Communications (If Client Agrees)		Enable
Network Security: LDAP Client Signing Requirements	Determines the level of data signing that is requested on behalf of the clients that issue LDAP BIND requests. The secure recommendation is to set this setting to “Require Signature”. This policy will prevent man-in-the-middle attacks and prevent altered data from being processed by the LDAP server.	Enabled
Network Security: Minimum Session Security for NTLM SSP Based Clients	Allows a client computer to require the negotiation of message confidentiality (encryption), message integrity, 128 bit encryption, or NTLMv2 session security.	Enable

7 Summary

In the era of viruses, worms, malware, buffer overflows and alike, the session hijack attack is still alive and well. The attack is very effective and can provide the attacker with unlimited access to server resources. The session hijack attacks allows the attacker to monitor the network for password information which can later be used to create access accounts on the compromised machine, or intercept data flowing between the client and server.

Defending against the session hijack attack is very difficult because the attack is not dependant on software vulnerabilities, but rather, protocol limitations within the TCP/IP protocol. Some of the byproducts of the attack are subtle, and are usually dismissed by users and network administrators as normal network events.

A variety of methods can be used to reduce your exposure to the attack including intrusion detection and intrusion prevention systems, firewall configuration, IPSec, secure FTP and Telnet, and Windows 2000/2003 Group policy objects. These technologies implemented together for form a defense in depth strategy, can provide a great deal of protection against the session hijack attack.

8 References

Laswon, L., July (2005). Session Hijacking Packet Analysis. Retrieved from the web 10/9/2006. http://www.securitydocs.com/link.php?action=detail&id=3479&headerfooter=no

Lam, K. & Leblanc, D. & Smith, K., (2006). Theft on The Web: Prevent Session Hijacking. Retrieved from the web 07/06/2006. http://www.microsoft.com/technet/technetmag/issues/2005/01/Sessionhijacking/default.aspx?pf=true

Microsoft Server 2003 Threats and Counter Measures Guide. Retrieved from the web 06/20/2006. http://www.microsoft.com/downloads/details.aspx?FamilyID=1b6acf93-147a-4481-9346-f93a4081eea8&displaylang=en

Cole, E. (2002). Hackers Beware: The Ultimate Guide to Network Security. (pp 147-152). Indianapolis, Indiana: New Riders.

Detecting Session Hijacking. Retrieved from the web August 10, 2006. http://www.resultspk.net/penetration_testing_and_network_defense/ch06lev1sec5.html

Meriwether, D. (2006). Retrieved from the web July 20, 2006. http://www.takedown.com/

Hassell, J. (2006). The Top 5 Ways to Prevent IP Spoofing. Retrieved from the web July 15, 2006. Http://masc2279.no-ip.org/gadgets-toys/internet/the-top-five-ways-to-prevent-ip-spoofing/

San Diego Super Computer Center (2006). Retrieved from the web June 23, 2006. http://security.sdsc.edu/software/secureftp/

Using Microsoft Windows IPSec to Help Secure an Internal Corporate Network Server. Retrieved from the web July 20, 2006, http://www.microsoft.com/downloads/details.aspx?familyid=a774012a-ac25-4a1d-8851-b7a09e3f1dc9&displaylang=en

Zalewski, M, (2002). Strange Attractors and TCP/IP Sequence Number Analysis – One Year Later. Retrieved from the web October 9, 2006. http://lcamtuf.coredump.cx/newtcp/

Cole, E. & Fossen, J. & Northcutt, S. & Pomeranz, H. (2005). Security 401 Security Essentials Defense in Depth. SANS Institute.

Dixon, W. & Wong, D. & Scambray, J. (2003). Using Microsoft Windows IPSec to Help Secure an Internal Corporate Network Server. Retreived from the web October 5, 2006. http://www.microsoft.com/downloads/details.aspx?familyid=a774012a-ac25-4a1d-8851-

b7a09e3f1dc9&displaylang=en