Global Information Assurance Certification Paper

“Using Social Engineering and Sniffing Techniques To Change An Employees Salary – A True Story”

GCIH – Certification Practical
Version 2.1a – Option 1

By: Jonathan Zerden

Submitted June 3, 2003

Attended: SANS New York
March 24, 2003 – March 28, 2003

Table Of Contents

Beep beep, beep beep is never a pleasant sound especially on vacation. Unlucky as I was, this was the exact sound that awoke me on the morning of September 26th 2002, while vacationing in California. I looked down at my pager expecting to see the phone number for the Network Operations Center (NOC) that I had grown accustomed to seeing. Instead I saw a phone number from my company's main office that I did not recognize. I got out of bed, searched for my cell phone, stumbling around in the dark I called the number back. On the second ring a pleasant woman on answered "Human Resources, how can I help you?" In a whisper, as to not wake my girlfriend who was still sleeping I responded "this is Jarod from Networking, did someone page me?" The woman who later identified herself as Maria, answered "oh yes there is a major problem. Benny needs to speak with you." My heart sunk, Benny was the VP of Human Resources for our company, Company M. In apparent haste, Benny picked up the phone and grumbled "Jarod we need you in my office immediately." I explained I was on vacation across the country and inquired about the situation. Benny began to explain that it appeared as if six of the non-exempt (hourly) employees pay rates had doubled without reason (at the time there were anywhere between sixty and one hundred non-exempt works on a given day.) Benny further explained that after the issue was identified, a call had been placed to the payroll services company that issues the paychecks for these employees. The payroll company had confirmed the issue and reported that the file from our company started to come through with the new amounts five weeks prior. Benny had confronted the internal payroll staff who vehemently denied the allegations that they had intentionally changed the worker's rates. No one was able was to identify what had happened, though a computer security breech was the only plausible explanation. I explained to Benny that I would be back to Texas immediately to assist in the investigation.

Abstract:

After a financial reconciliation identified improper payroll being paid to a number of non-exempt works at Company M, the internal security task force was called into action to identify the source of the over payments. After approximately four days of technical research, in-depth analysis and interviews with a number of different employees and contractors; technical and social engineering exploits were identified that illegally led to the manipulation of the payroll data. Included in the exploits identified were: the unauthorized use of a sniffer on company premises to detect usernames and passwords sent in clear text, unauthorized use of external computers plugged into the company's network and the attempted social engineering of a staff member.

The Exploit:

Name

Upon arrival back in Texas the only thing known was that unauthorized data had been sent to the payroll company. After further analysis it was discovered that a number of exploits and tricks were used in order gain access to the systems that controlled the payroll data. The following well known exploits were used: ARP spoofing (CVE # CAN-1999-0667) in order to sniff passwords on internal web applications and a Web spoofing of SSL pages. Also included in the incident was an unsuccessful attempt at social engineering.

Operating Systems:

Company M uses a number of operating systems in their daily operations. These operating systems include: Z/OS (IBM mainframe), Windows 2000 Server, Windows 2000 Workstation, Windows NT 4.0, HPUX, SGI Irix 6.5, and Novell Netware 5.0. For the particular exploits described in this paper Windows 2000 server and Windows 2000 workstation were used. Both versions of Windows 2000 were patched to their fullest during the attack. Company M has a stringent and strictly enforced patch management policy, especially in regards to Microsoft Windows machines. All patches released from Microsoft must be applied with in 48 hours of release.

Protocols / Services / Applications affected by the exploit:

This incident used a number of different protocols, services and applications. HTTP, HTTPS, ARP, DHCP and DNS were all used to varying degrees. In addition IIS was used along with a purchased application to manage web security.

Brief Exploit Description:

Two methods were attempted (one of them being successful) in this incident:

Unsuccessful method: The first method failed, however it was shear luck that this method was not successful. When this method was uncovered during the identification phase of the investigation we decided that it was extremely important to educate our staff companywide of this potential exploit. Therefore I think it is important to discuss this method in this paper. Badh (the attacker) had gone under the assumption that most users have the same username and password across systems internally and externally to the company. Thus if an external system was established it would be likely that the intended victim would easily give their username and password to the thief. In addition, a visual trick to pretend that a site was SSL encrypted, offering a false sense of security to the victim, called SSL spoofing was attempted.

Successful method: After the first attack failed Badh researched the ability to take usernames and passwords off the Ethernet wire using a sniffer. Since the network was switched Badh had to learn how to sniff a switched network (in this

actual attack Badh used the ARP spoofing technique to sniff the switch.) He then had to understand how to sniff the particular computers in question and find the appropriate usernames and passwords for his attack. In summary Badh was able to trick Goodpayroll's (the internal company payroll employee who was the victim) machine into believing that Badh's machine was the default gateway and thus all traffic destined for the other networks was sent through Badh's machine. Freely downloadable software was used to capture the traffic and pull out the pertinent usernames and passwords to perform data change.

Variants:

The method employed for this sniffing technique was ARP spoofing. However there are a number of other possible sniffing techniques that could have been used:

1. Non-switched environment: In a non-switched environment a sniffer could have been used with out the necessity of ARP spoofing and subsequent routing of traffic through Badh's machine.
2. MAC Flooding: Depending on the switch type another exploit could have been used to sniff the switch: MAC Flooding. This exploit would have sent a large number of MAC addresses to the switch. Eventually, some switches become overwhelmed by the flood of MAC addresses and open themselves up into a "hub-like" mode. (http://www.sans.org/resources/idfaq/switched_network.php)
3. MAC Duplicating: Assuming Badh knew the MAC address of the victim's machine, Badh could have modified his MAC address to match that of the victim's machine. When this is done, the switch would forward the intended packets to both machines on the switch – thus allowing sniffing.

Since Badh did not have a large amount of network experience it is unlikely that he would have been able to use any of the variants of this attack (the fact that he only knows the Windows operating system and that the only hacking tool Badh used was Cain, which only supports ARP spoofing.) However these other variants were all possible in specific segments of Company M's network.

References:

- "Why A Switched Network Isn't Secure." http://www.sans.org/resources/idfaq/switched_network.php
- "SwitchSniff" - http://www.linuxjournal.com/article.php?sid=5869
- The actual tool used in the sniffing exploit: http://www.oxid.it/cain.html
- The software used to provide the website in the unsuccessful attack with authentication capture: http://www.flicks.com/authentix_isp/
- A demonstration of SSL spoofing: http://www.cs.dartmouth.edu/~pkilab/demos/spoofing/
- "Web Spoofing Revisited: SSL and Beyond": http://www.cs.dartmouth.edu/~pkilab/papers/tr417.pdf

The Attack:

The Network:

The network upon which this particular attack took place was a standard 10/100 auto-sense based switched network. The network was connected to the Internet via four combined T1's providing a total of approximately six mbps of traffic to the Internet. The T1's were connected to a Cisco 3500 series router that was fully managed by the ISP (the ISP in this case was UUNet.) On the internal side of the router was a Checkpoint NG Firewall (FP1.) The hardware was configured in an active-passive failover configuration using Nokia's proprietary implementation of the VRRP protocol. The primary hardware was a Nokia 530 while the secondary firewall was a Nokia 440. Both firewalls featured two quad Ethernet cards each for a total of eight available ports (at the time of the incident only five ports were in use.) One port in each firewall was used as the sync port for failover between the two. The other ports were used for: the external internet interface, the internal segment, the internal server segment and an external DMZ. Behind each firewall segment was an unmanaged D-link switch.

The external interface port of each machine was plugged into a D-link unmanaged switch with a single uplink cable connected to the managed Cisco router to provide Internet access.

The internal segment featured a number of switches plugged directly into the D-link switch. Each switch then supported between 9 and 36 internal workstations. The internal segment featured one DHCP server with an address scheme of 10.1.1.X with a subnet of 255.255.255.0. The default gateway (and the internal VRRP address of the firewall) was 10.1.1.1. Each of the machines on the internal subnet were Windows 2000 workstations patched to their highest levels at the time of the incident. The DHCP server was Windows 2000 Server Standard edition. This server had also been patched to its highest level.

The internal server segment consisted of a number of different servers plugged in directly to the D-link switch. The purpose of this was to create a zone and associated rule set to segment the servers which performed internal functions only, from those that needed access to the Internet. Each server in this zone was hard configured with an IP scheme of 10.1.2.X and a subnet of 255.255.255.0. Each machine had a default gateway of 10.1.2.1 which was the VRRP address of the firewall interface. Each of the machines on the internal server segment were running Windows 2000 Server, standard edition. Each of the machines was also running IIS 5.0. All current patches had been applied to these servers.

The DMZ consisted of an SMTP and web server. Each machine was hard coded with an IP scheme of 10.1.3.X and a subnet of 255.255.255.0. The default gateway (and the internal VRRP address of the firewall) was 10.1.3.1. Each machine was running Windows 2000 Server, standard edition. The SMTP server was running Exchange 2000 and the web server was running IIS 5.0. Each of these servers had all possible patches applied to them.

The firewall rules were as follows:

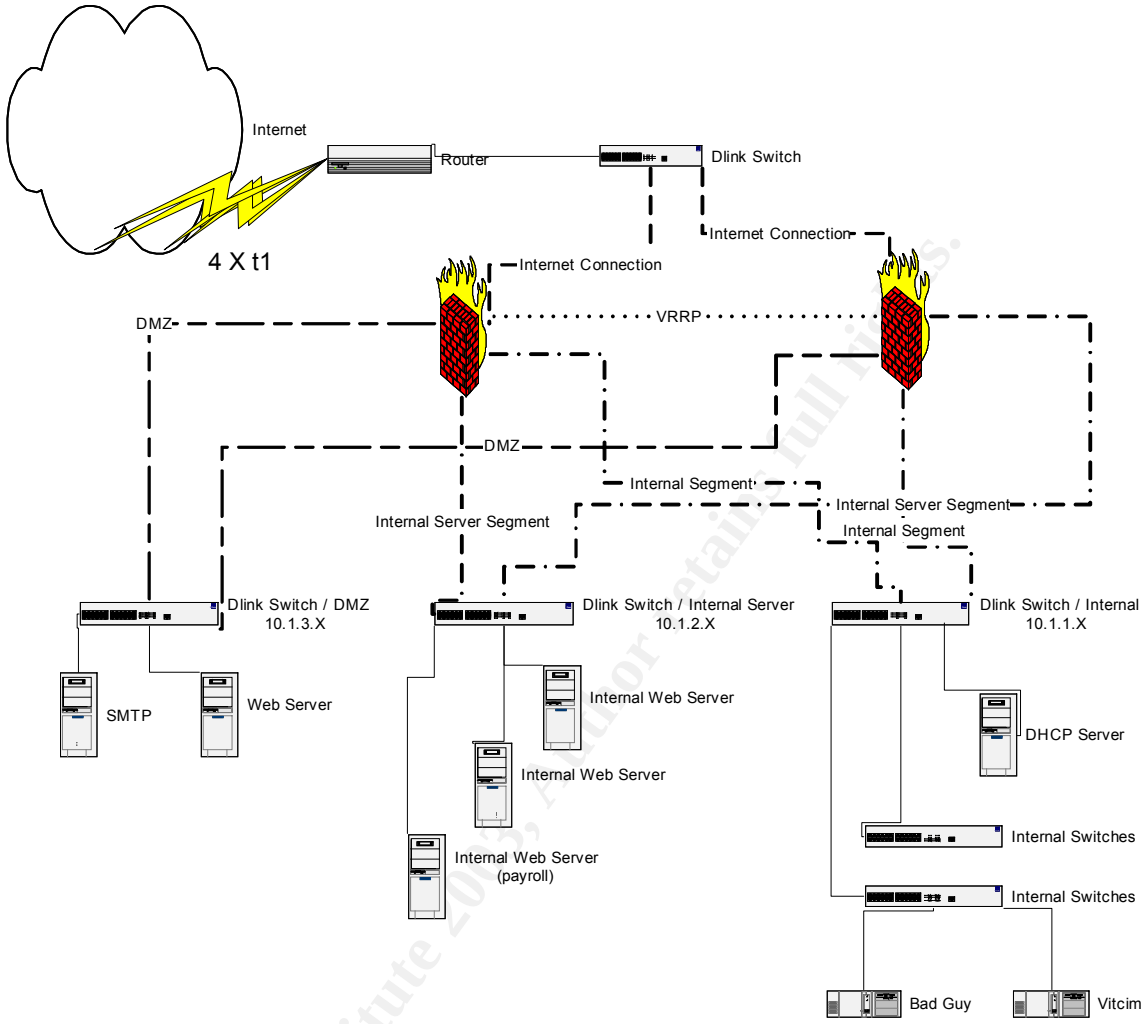
Rule #	Source	Destination	Port	Action
1	Firewall 0 and Firewall 1	Firewall 0 and Firewall 1	VRRP	Allow
2	Internal Segment	Anything Except Internal Segment, DMZ, Server Segment	Any	Allow
3	Internal Segment	Dmz.Mail.Ip	Exchange Ports	Allow
4	ANY	Dmz.server.ip	HTTP, HTTPS	Allow
5	Internal Segment	Server Segment.IP's	HTTP	Allow
6	DMZ.Mail.ip	Anything Except Internal Segment, DMZ, Server Segment	SMTP, DNS	Allow
7	ANY	DMZ.Mail.IP	SMTP	Allow
8	ANY	ANY	ANY	Drop

Note: Anti-spoof was turned on for all segments both incoming and outgoing. Also NAT was used for the two servers requiring externally facing IP's (dmz.mail.ip and dmz.server.ip)

Explanation:

Rule #1 – Allowed VRRP traffic between the two firewalls.
Rule #2 – Allowed the internal segment to go out to anything, except the other segments behind the firewall.
Rule #3 – Allowed the internal segment to get to the mail server on the DMZ with Exchange/Outlook.
Rule #4 – Allowed anyone to HTTP and HTTPS to the web server on the DMZ segment.
Rule #5 – Allowed the internal segment to send HTTP traffic to the server segment (it is this rule that was ultimately abused.)
Rule #6 – Allowed the DMZ mail server to send out DNS and SMTP traffic.
Rule #7 – Allowed anyone to send mail to the mail server.
Rule #8 – The drop rule for all traffic not excepted above.

The following is the portion of the network diagram relating to this exploit:



Protocol Description:

Two protocols were used in this attack, though only one was successful. The unsuccessful attack used SSL spoofing, which is more of a "training" flaw then a protocol flaw. The successful attack used the standard IP protocol in conjunction with ARP and http basic authentication.

SSL: Secure Socket Layer is a protocol that was developed by Netscape to facilitate secure communications between a client (web browser) and a server in addition to identity authentication. A clear technical explanation of how SSL accomplishes both secure communication and identity authentication can be found at http://developer.netscape.com/tech/security/ssl/howitworks.html. The flaw exploited with SSL in this case (and in my opinion the largest flaw having to do with SSL) is not actually a protocol flaw but rather a user training issue. Most web users assume that as long as the "locked" icon is displayed on their browser,

all information is secured. The two most common issues that can arrive from this naivety is that this lock box can be spoofed and that just because information is secured over the wire in transit (via encryption) it is not necessarily stored securely once it has reached the intended server.

ARP: Address Resolution Protocol is the protocol that allows for the mapping of Media Address Control (MAC) addresses to TCP/IP addresses. Every network device is shipped from its manufacturer with a unique twelve HEX character MAC address. When computer A attempts to connect to computer B on the same subnet over IP, an ARP request is sent to all machines on that network. This is done so that computer A knows how to physically direct its traffic. The network device (computer B) that has the IP that computer A is trying to reach, responds back to computer A with the MAC address of its NIC. At the same time computer B caches the MAC address of the requesting device (computer A.) The ARP requester (computer A) receives this information back and stores the MAC address of the requested IP in it's ARP cache. The requestor now sends all future communications to the MAC address of the machine in question. To view a machines ARP table on Window, go to the dos prompt and type arp –a. You will see something similar to:

```
Microsoft Windows XP [Version 5.0.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\>arp -a

Interface: 192.168.1.150 --- 0x2
  Internet Address      Physical Address      Type
  192.168.1.1           00-04-5a-2d-d2-6b     dynamic
  192.168.1.100         00-01-02-33-aa-b4     dynamic

C:\Documents and Settings\>
```

This table explains that I have been communicating with two machines 192.168.1.1 and 192.168.1.100. Both of their unique MAC addresses are displayed under the Physical Address heading.

Note: the only MAC addresses listed in my cache are those for which I have had direct communicate with. All other machines on my subnet were not stored.

The major flaw with ARP is what is known as "Gratuitous ARP's." A Gratuitous ARP occurs when a device sends an ARP response, without an actual ARP request being sent. Gratuitous ARP's can be used to fool a device into believing that a third-party machine is the "owner" of the IP address the device is trying to communicate with.

For example look at the following summary from a sniffer:

1	Victim	Broadcast	ARP	Who has 192.168.1.1? Tell 192.168.1.100
2	Router	Broadcast	ARP	Who has 192.168.1.100? Tell 192.168.1.1
3	Victim	Router	ARP	192.168.1.100 is at 00:01:02:33:aa:b4
4	Attacker	Broadcast	ARP	Who has 192.168.1.100? Tell 192.168.1.154
5	Victim	Attacker	ARP	192.168.1.100 is at 00:01:02:33:aa:b4
6	Attacker	Victim	ARP	192.168.1.1 is at 00:10:a4:b0:44:74
7	Attacker	Victim	ARP	192.168.1.1 is at 00:10:a4:b0:44:74
8	Attacker	Victim	ARP	192.168.1.1 is at 00:10:a4:b0:44:74
9	Attacker	Victim	ARP	192.168.1.1 is at 00:10:a4:b0:44:74

What occurred is as follows:

1. The victim machine is turned on and asks who the router is.
2. The router asks who the computer is
3. The victim replies to the router with its own MAC

At this point the router and the victim machine are talking directly.

4. The attacker machine is booted up and set to poison .100, the victim machine – it asks who is the victim machine?
5. The victim machine tells the attacker its MAC address.
6. The attacker then begins to send out requests to the victims machine saying that the correct MAC for the router is the attackers machine.
7. 7-9 are sent frequently so that the ARP stays poisoned on the victim machine.

A good technical overview of ARP can be found at: http://www.microsoft.com/windows2000/en/datacenter/help/default.asp?url=/windows2000/en/datacenter/help/sag_tcpip_und_arp.htm

HTTP Authentication: The HTTP basic authentication mechanism was also exploited during this security incident. The HTTP 1.1 protocol provides for two common authentication schemes for access to “protected” (ie. Any page that the web administrator requires a username and password to access) web pages: basic and digest authentication. Basic authentication was the particular protection exploited in this incident. When a user attempts to access a portion of a website protected by basic authentication the web server returns a 401 code requesting a username and password. When the browser receives the 401 code, it prompts the user for a username and password. The browser captures this information, BASE-64 encodes it (changing it to a text representation of binary data) and sends it back to the web server. The web server decodes the

password and compares it to the expected passwords allowing the user to see the protected document if they are correct. The major weakness with Basic authentication is that all of the data is sent in clear text.

How the exploit works:

The first exploit attempted during this incident was a combination of SSL spoofing and social engineering. Badh (the attacker) attempted to gain access to a protected system by assuming that users often use the same username and password for all systems. Badh then set up a website on his home PC. This home site was configured in a way to require user authentication to enter portions of his site. Not being an extremely strong programmer, Badh used the Authentix authentication software from Flicks software. The attacker sent the victim a link to his home website: http://website.url.com with the promise of pictures from a recent company outing. Upon arrival on the web site the victim was presented with the requirements of registering for the site. A popup windows appeared requiring a username and password along with additional personal information. The popup window had a forged (spoofed) browser window, appearing to the user as if they were in an SSL (secure) session. Assuming that the session was secure the victim then registered for the site. Upon completion of the registration the user was granted access to the content. Having configured the flicks software to write the registration information to an Access database (without field level encryption) the attacker was now presented with the password used to register on the system. The attacker then attempted to access the internal web based payroll system using the victims username and the password entered on the attackers personal site. In this case, the users password did not resemble or closely match that which they used when registering on the attackers site.

The second exploit used was to sniff the username and password of the victim in real time. The system the attacker was trying to gain access to was a system that used basic authentication for its access control. The attacker believed that if they could gain administrator access to the system (via the victim's username and password) he could alter employees hourly pay rate. After significant research the attacker performed a Man-In-The-Middle attack to sniff the username and password from the wire while the victim was accessing the payroll web site. To sniff the traffic in question the attacker need to know the internal IP address of the victims machine. Once this was attained (detailed below on how this was attained in this particular case) the attacker is able to send a gratuitous ARP with the attackers MAC address to both the default gateway and the victims' machine. This intern confuses the victims' machine and the default gateway into believing that attackers machine is the proper next hop. All traffic is subsequently forwarded through the attackers' machine. Assuming that the attacker has turned on IP Forwarding, all information is then passed from the attackers machine along to its intended location thus not alerting the victim to any problems.

The particular exploit used in this incident was the manipulation of the ARP protocol. The RFC regarding the ARP protocol (RFC 826) was developed in November of 1982, and thus the developers of the protocol were unlikely to be overly concerned with security. The flaw stems from the fact that a device can send an answer to an ARP request that was never asked. Since most (if not all) operating systems cache the ARP answers (so as not to use too much bandwidth for a routine task) it is possible to trick a computers ARP cache into believing a different MAC address is the correct MAC address for a particular IP.

How the exploit works:

1. A victim boots his computer and is assigned the ip address of 10.1.1.9, with a subnet mask of 255.255.255.0 and a default gateway of 10.1.1.1.
2. Victim attempts to go to the Internet site www.aol.com by entering it in their browser.
3. Victim computer looks up from the TCP stack the location of the DNS server to resolve www.aol.com – it finds the answer of 10.1.1.2.
4. Not knowing how to reach the physical machine of 10.1.1.2 an ARP request is sent to the network. This request is looking for the MAC address of 10.1.1.2.
5. The machine whose ip address has been configured as 10.1.1.2 receives the request and responds with its MAC address of 00-01-03-4D-A4-F9.
6. In the meantime, the responder to the ARP request (10.1.1.2) also caches the requesters MAC.
7. Upon receipt of the ARP answer, the requester stores the MAC in its cache table for future use.
8. The requestor then queries the DNS server to receive the IP of www.aol.com using the MAC address of the DNS server to communicate with it.
9. The answer to the DNS request is answered with the IP of 64.12.149.18.
10. The user's machine realizes that it can not communicate directly with 64.12.149.18. It then requests the MAC address of the default gateway.
11. The default gateway responds with its MAC address and caches the requestors address.

    At this point in the exploit the following is known:
    a. The default gateway has in it's cache the MAC for the victim's computer
    b. The victim's computer has the cached MAC address for the DNS server and the default gateway.

12. The attackers computer enters the network (the machine is on the same subnet as the victims computer and has IP forwarding turned on.)
13. The attackers computer sends a gratuitous ARP (the answer to an ARP request, without actually having been asked for one) to the default gateway. This ARP answer states that the attackers MAC address is the

correct MAC address for the victims IP. Due to the implementation of the RFC the newest ARP response overwrites the any old response in the cache.

14. The same process is used to poison the cache on the users victim's machine.
15. Fake ARP answers are regularly sent to both the default gateway and the victim's machine so as to not allow the real ARP Reply to overwrite the fake one.

Once the ARP caches are poisoned, all packets destined for the default gateway and victims machine are sent through the attackers computer. The attacker is then able to use standard sniffing tools to capture the data.

There are a number of freely available tools that help to exploit this vulnerability. The most popular ARP spoofing tool is Arpspoof available from the Dsniff suite of utilities (http://www.monkey.org/~dugsong/dsniff/.) By running Arpspoof you are able to easily send fake (gratuitous) ARP responses on a subnet. To do so, simply run the following command after installing Dsniff:

arpsoof –t ipaddressofvictim ipaddressofgateway

Running Dsniff with its various flags will allow the sniffing of pertinent and informative packets such as usernames and passwords.

There are numerous other freely available tools that perform the same functions as the Arpspoof and Dsniff utilities. However if no tools were available a custom tool could be designed. This tool would need to have the ability to:

1. Determine the MAC address of the attackers computer (or allow it to be inputted.)
2. The ability to send "fake" ARP response to designated machines with the attackers MAC address.
3. The ability to continue to send these ARP requests so as to not loose the poisoning.
4. The ability to turn a network card into promiscuous mode and to listen to all traffic coming across its network card and to write it to a file.
5. Nice to have: the knowledge of different protocols and the ability to grep through the raw packets to find the important data (ie. Usernames and passwords.)

Once all of the pertinent data was sniffed from the line, the user would then need to un-encode the username and password from the BASE64 encoded version used with basic authentication.

Below is the raw data sniffed between a client and server requiring basic authentication :

GET /members/listserver/listserver.shtml HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, application/x-shockwave-flash, */*
Referer: http://www.whatever.com/public/member_benefits/listserver.shtml
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.0.3705)
Host: www.whatever.com
Connection: Keep-Alive
Cookie: cookie=set
Authorization: Basic dGVzdDpwYXNzd29yZA==

The key to this particular portion of the exploit is the decoding of the BASE64 data. The username and password was sent in the following form:

Authorization: Basic dGVzdDpwYXNzd29yZA==

This portion of the transmission identifies the authentication type (Basic) and displays the encoded username and password. Most current sniffing applications have the ability to decode this information on the fly, however if one was not available you would need to manually convert the string "dGVzdDpwYXNzd29yZA==" to readable data. This could be done by either using the BASE 64 "algorithm" (as explained in RFC 1521, page 21) or using freely available tools. My favorite BASE64 decoder can be found at: http://www.opinionatedgeek.com/dotnet/tools/Base64Decode/Default.aspx. By pasting the string as it was sniffed (dGVzdDpwYXNzd29yZA==) and pressing submit, the user is returned with: test:password. In this example 'test' was the username entered and 'password' was the password entered.

Description and Diagram of the Attack:

This section will focus only on the successful attack used in this incident.

Step 1 – Identifying the IP address of the victims machine:
For this incident the attacker acquired this information by doing the following tasks:

A. The attacker brought his home web site computer in to the office and connected it to a network drop. Since the network was configured to use DHCP, the machine was assigned an IP address.
B. The attacker identified the IP address of his personal computer by using the ipconfig command from the command prompt. It was identified as 10.1.1.98 (an internal IP address.)

C. The attacker modified the DNS record for website.url.com to point to 10.1.1.98. This change caused any user who attempted to gain HTTP access to http://website.url.com to be directed to 10.1.1.98. Thus in this case the victim would access the site directly without going through the firewall, and therefore NAT would not be invoked.
D. The attacker sent an e-mail to the victim inviting them back to his personal website of http://website.url.com. Once the victim accessed this URL the attacker reviewed the IIS logs to determine the users internal IP address

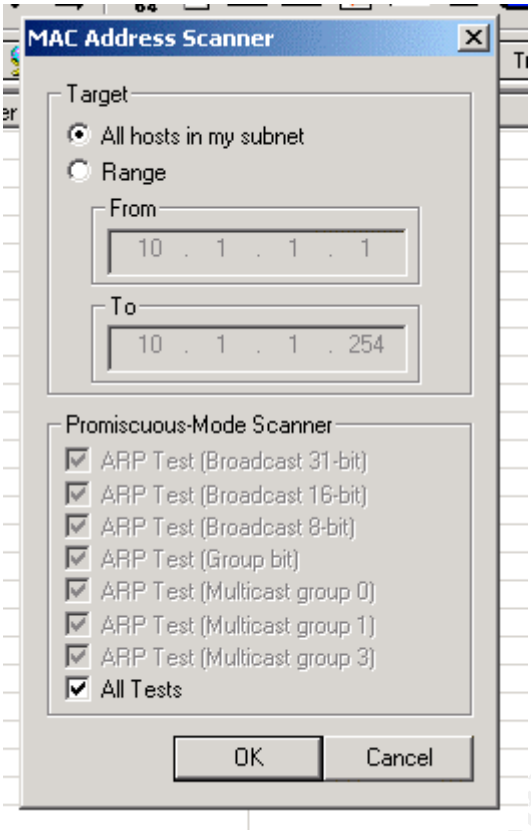
The following is an example log file from which the attacker was able to identify the victim's IP address:

2003-05-04 20:22:22 10.1.1.209 - 10.1.1.98 80 GET /index.htm - 304 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0;+.NET+CLR+1.0.3705)
2003-05-04 20:22:22 10.1.1.209 - 10.1.1.98 80 GET /Default.htm - 304 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0;+.NET+CLR+1.0.3705)
2003-05-04 20:22:22 10.1.1.209 - 10.1.1.98 80 GET /main.htm - 304 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0;+.NET+CLR+1.0.3705)
2003-05-04 20:22:22 10.1.1.209 - 10.1.1.98 80 GET /jon.JPG - 304 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0;+.NET+CLR+1.0.3705)
2003-05-04 20:22:22 10.1.1.209 - 10.1.1.98 80 GET /about1.JPG - 304 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0;+.NET+CLR+1.0.3705)
2003-05-04 20:22:22 10.1.1.209 - 10.1.1.98 80 GET /res1.JPG - 304 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0;+.NET+CLR+1.0.3705)
2003-05-04 20:22:22 10.1.1.209 - 10.1.1.98 80 GET /con1.JPG - 304 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0;+.NET+CLR+1.0.3705)
2003-05-04 20:22:22 10.1.1.209 - 10.1.1.98 80 GET /fr1.JPG - 304 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0;+.NET+CLR+1.0.3705)
2003-05-04 20:22:22 10.1.1.209 - 10.1.1.98 80 GET /link1.JPG - 304 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0;+.NET+CLR+1.0.3705)
2003-05-04 20:22:22 10.1.1.209 - 10.1.1.98 80 GET /main_h1.jpg - 304 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0;+.NET+CLR+1.0.3705)

From these logs, the attacker could identify the victims IP address of 10.1.1.209.

Step 2 – ARP Poisoning and Sniffing
For this incident the attacker used the Cain tool (http://www.oxid.it/cain.html) to sniff, locate and decode the username and passwords. The following were the steps used to accomplish this:

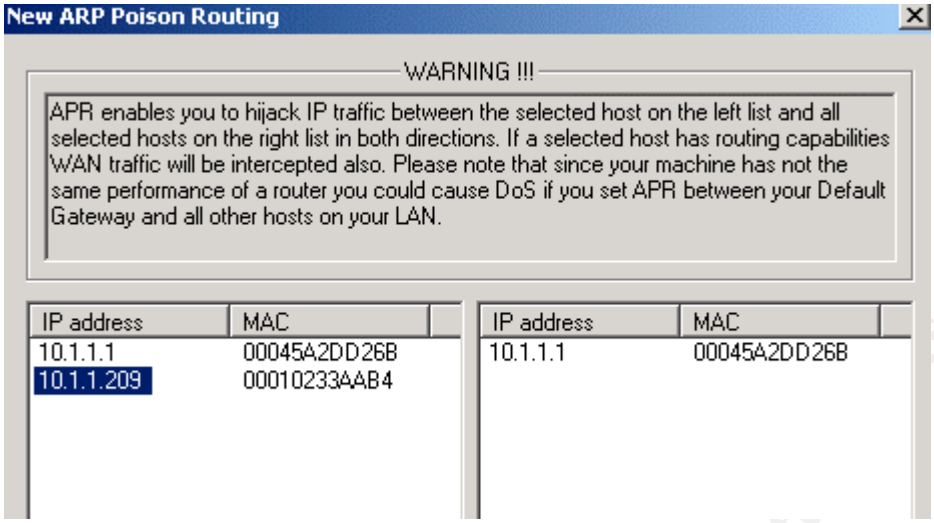
a. Attacker uses Cain to scan all IP's on the network



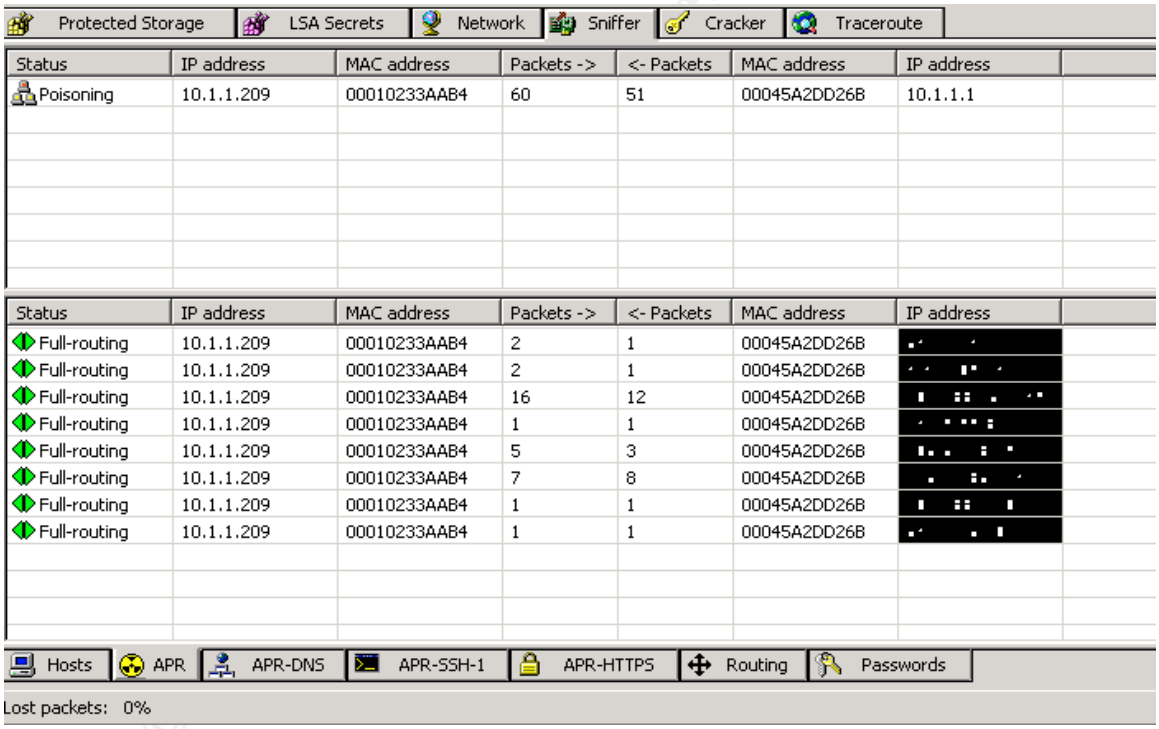
b. After the IP's are scanned you are presented with the following (summarized) results:

IP address	MAC address	OUI fingerprint	Host name	B
10.1.1.1	00045A2DD26B	[redacted]		*
10.1.1.209	00010233AAB4	[redacted]		

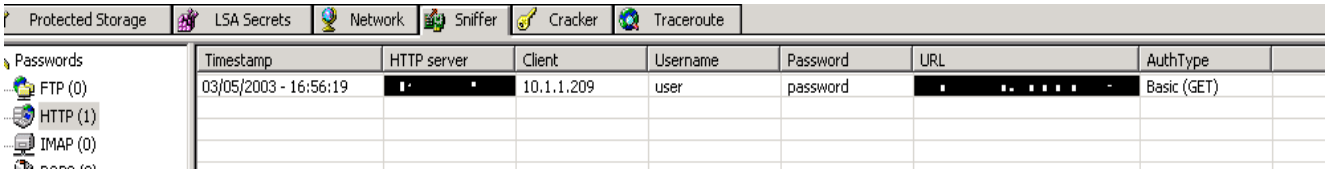
c. The attacker then selects which IP's to sniff and poison:



d. Attacker receives confirmation of sniffing and sees a representation of each packet moving through the attacking machine:



e. Attacker waits for the victim to access the correct web server and logon. After this is completed, Cain will display the username and password used to access this system:



f. Once this information is received, the attacker logs on to the system in question with the identification that was sniffed from the victim.

During the attack changes in the ARP cache can be identified by either the victims machine and/or the default gateway. Below are examples of the changes that could have been viewed on the victim's machine:

```
C:\Documents and Settings>ipconfig /all

Windows IP Configuration

        Host Name . . . . . . . . . . . . : [redacted]
        Primary Dns Suffix  . . . . . . . :
        Node Type . . . . . . . . . . . . : Unknown
        IP Routing Enabled. . . . . . . . : No
        WINS Proxy Enabled. . . . . . . . : No
        DNS Suffix Search List. . . . . . : [redacted]

Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . :
        Description . . . . . . . . . . . : 3Com EtherLink 10/100 PCI For Comple
te PC Management NIC (3C905C-TX)
        Physical Address. . . . . . . . . : 00-01-02-33-AA-B4
        Dhcp Enabled. . . . . . . . . . . : No
        IP Address. . . . . . . . . . . . : 10.1.1.209
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 10.1.1.1
        DNS Servers . . . . . . . . . . . : 24.29.99.81
                                            24.29.99.82
                                            [redacted]

C:\Documents and Settings>arp -a

Interface: 10.1.1.209 --- 0x2
  Internet Address      Physical Address      Type
  10.1.1.1              00-04-5a-2d-d2-6b     dynamic

C:\Documents and Settings>_
```

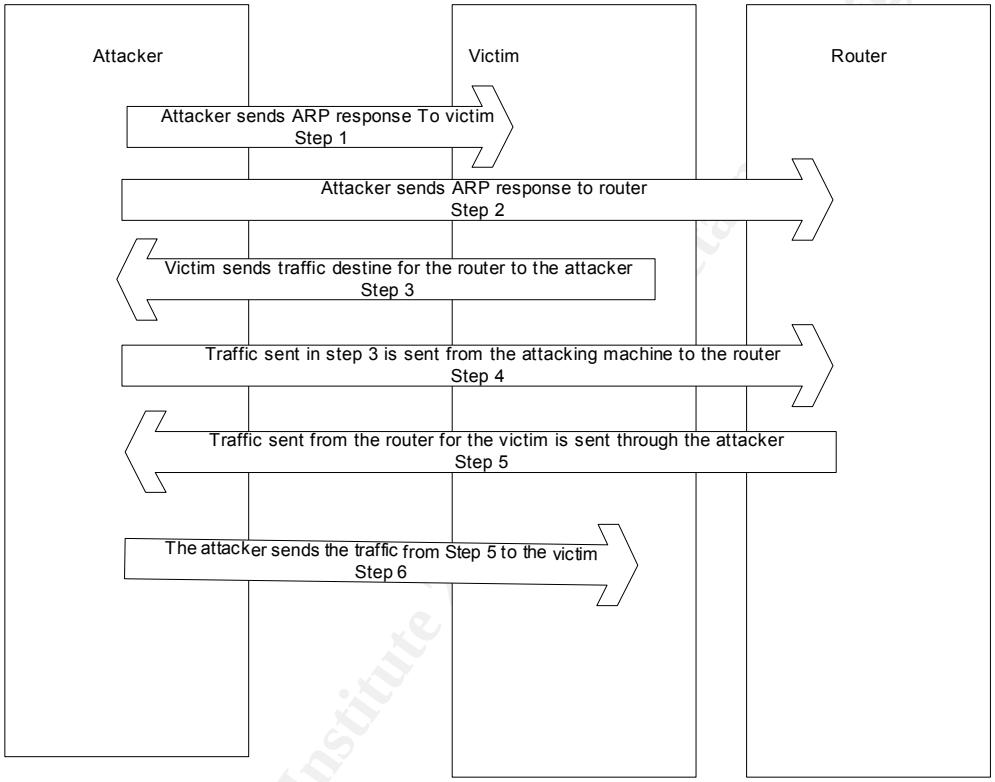
The screen shot above depicts the user's IP and ARP cache prior to the exploit. As seen above, there is only one ARP entry in the machine – the default gateway. This is because the only machine this computer is talking to directly on the subnet is the default gateway.

```
C:\Documents and Settings>arp -a

Interface: 10.1.1.209 --- 0x2
  Internet Address      Physical Address      Type
  10.1.1.1              00-0c-29-53-80-bf     dynamic
  10.1.1.152            00-0c-29-53-80-bf     dynamic

C:\Documents and Settings>
```

This screenshot depicts the ARP cache after the poisoning has begun. Three features have changes in this screenshot:

1. The MAC for the default gateway has changed.
2. The MAC for the default gateway is the same as another machine on the network.
3. The ARP cache now depicts a machine that the victim's machine should not be talking directly to.

The following is a diagram of the attack in action starting after the identification phase:



Signature of the Attack:

There are two different signatures associated with this attack:

1. The log files on the web server could be analyzed to detect the wrong combination of users from a particular IP. If the victim always accesses the payroll system from the same IP address, a tool could be written that would look for a mismatch of such information and alert the administrators.

For example, in the following log files snippet you can see that a user came in from 10.1.1.98 named "usera." The log file reader could be programmed in such a way to grep through the log files to alert when "usera" comes from any other machine then their own:

2003-05-17 17:15:51 10.1.1.98 usera 10.1.3.160 80 GET /PPPP/ - 200 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1;+.NET+CLR+1.0.3705)

2. The signature of the ARP poisoning is the flip-flop of IP to MAC address. In addition to ARP responses without requests.

To monitor for ARP changes the tool must be able to look at the packets as they come off-the wire, decode them and store an IP to MAC address table. For example:

Frame 28 (42 bytes on wire, 42 bytes captured)
Arrival Time: May 17, 2003 17:45:28.252493000
Time delta from previous packet: 0.000013000 seconds
Time relative to first packet: 5.846920000 seconds
Frame Number: 28
Packet Length: 42 bytes
Capture Length: 42 bytes
Ethernet II, Src: 00:01:02:33:aa:b4, Dst: 00:04:5a:2d:d2:6b
Destination: 00:04:5a:2d:d2:6b (Router)
Source: 00:01:02:33:aa:b4 (Victim)
Type: ARP (0x0806)
Address Resolution Protocol (reply)
Hardware type: Ethernet (0x0001)
Protocol type: IP (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: reply (0x0002)
Sender MAC address: 00:01:02:33:aa:b4 (Victim)
Sender IP address: 192.168.1.100 (192.168.1.100)
Target MAC address: 00:04:5a:2d:d2:6b (Router)
Target IP address: 192.168.1.1 (192.168.1.1)

In this frame, the tool would need to identify and store the data in bold. It would need to continually compare this data to the current packets coming across the wire. When a packet that looks like the following occurs:

Frame 584 (60 bytes on wire, 60 bytes captured)
Arrival Time: May 17, 2003 17:45:56.993882000
Time delta from previous packet: 0.048594000 seconds
Time relative to first packet: 34.588309000 seconds
Frame Number: 584
Packet Length: 60 bytes
Capture Length: 60 bytes
Ethernet II, Src: 00:10:a4:b0:44:74, Dst: 00:01:02:33:aa:b4
Destination: 00:01:02:33:aa:b4 (Victim)

Source: 00:10:a4:b0:44:74 (Attacker)
Type: ARP (0x0806)
Trailer: 00000000000000000000000000000000...
Address Resolution Protocol (reply)
Hardware type: Ethernet (0x0001)
Protocol type: IP (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: reply (0x0002)
Sender MAC address: 00:10:a4:b0:44:74 (Attacker)
Sender IP address: 192.168.1.1 (192.168.1.1)
Target MAC address: 00:01:02:33:aa:b4 (Victim)
Target IP address: 192.168.1.100 (192.168.1.100)

It would need to compare the bolded data to the data found in its tables. If the MAC-IP has changed (as it has in this case from the router to the attacker machine) it would need to alert an administrator. Also the tool should look at the volume or ARP replies to the same IP Address and responses when a request has not been asked for.

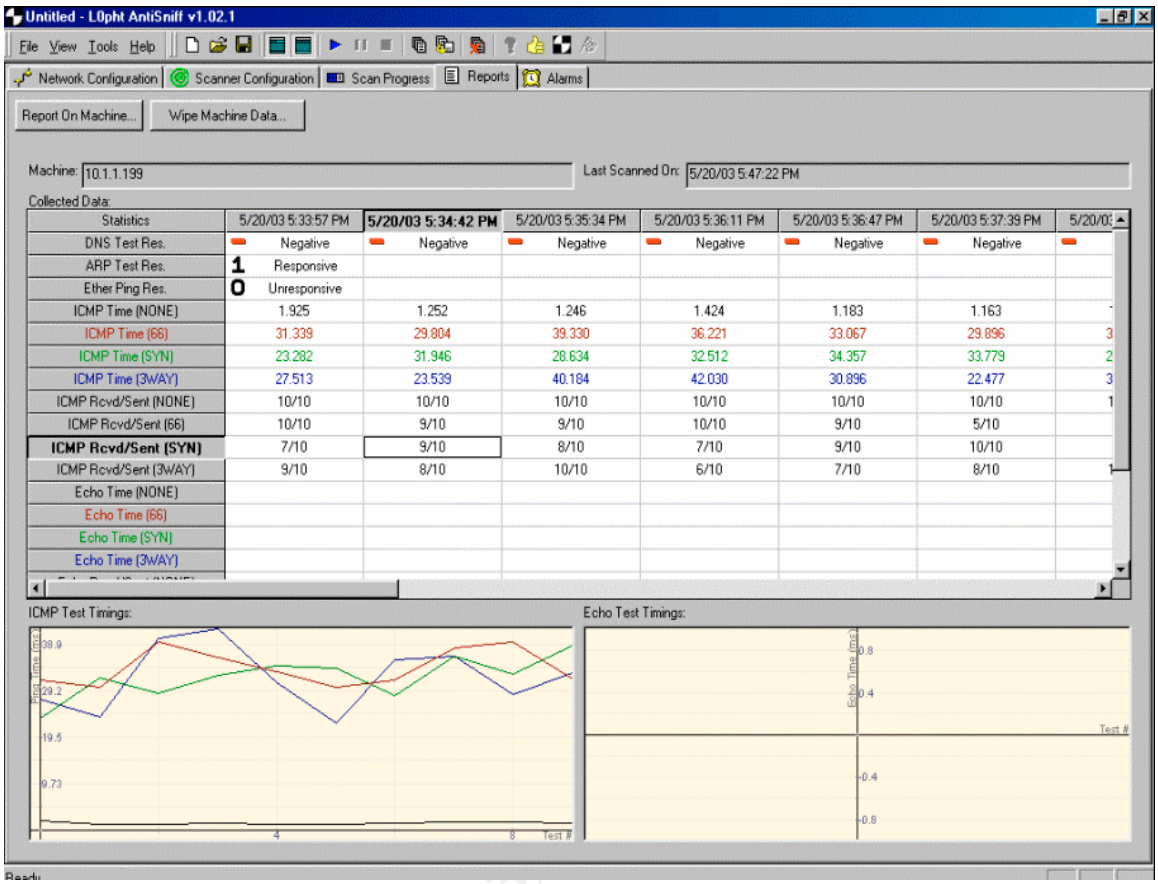
How to Protect Against the Attack:

To prevent the unsuccessful portion of the attack there is only one solution: training. Users must be taught how secure (or insecure) data on the Internet is and understand the level of security (when possible) a site uses.

There are a number of different potential technological solutions to prevent and/or warn administrators to the sniffing / basic authentication attack:

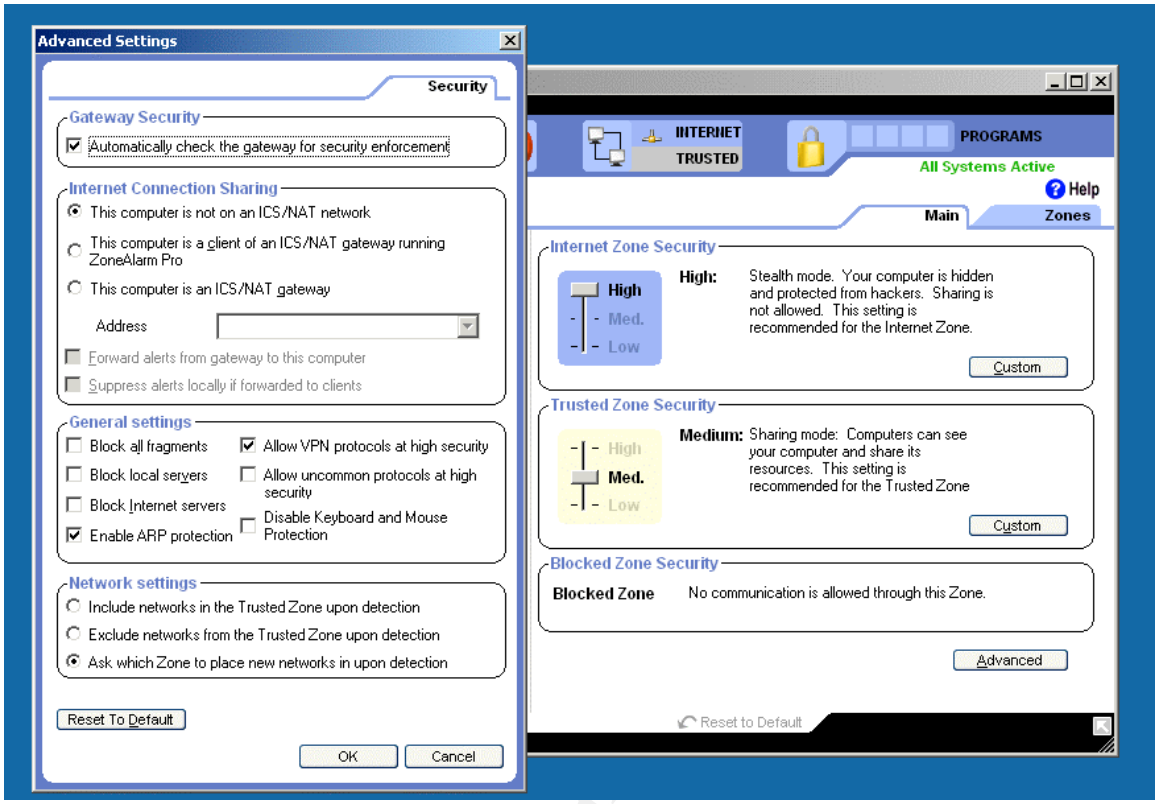
1. A different type of web authentication: Basic authentication by its design is not a secure way to authenticate users to a web server. Administrators whose websites require authentication should use other methods including SSL and Digest authentication to have the data encrypted across the wire and/or one-time hashed.
2. Static ARP Cache's: Desktop systems can be programmed with static ARP entries. If this is done the gratuitous ARP used in ARP Cache poisoning should not work. (note: Windows 2000 machines as are used in this current environment, will overwrite their static ARP entries when a gratuitous ARP is sent. This has been "corrected" in Windows XP.)
3. Antisniff: Antisniff or similar tool will allow for administrators to scan all machines connected to a network and identify those which have their network cards running in promiscuous mode. Network cards in promiscuous mode are a very strong indication that someone is attempting to sniff the network traffic.

Antisniff in use:

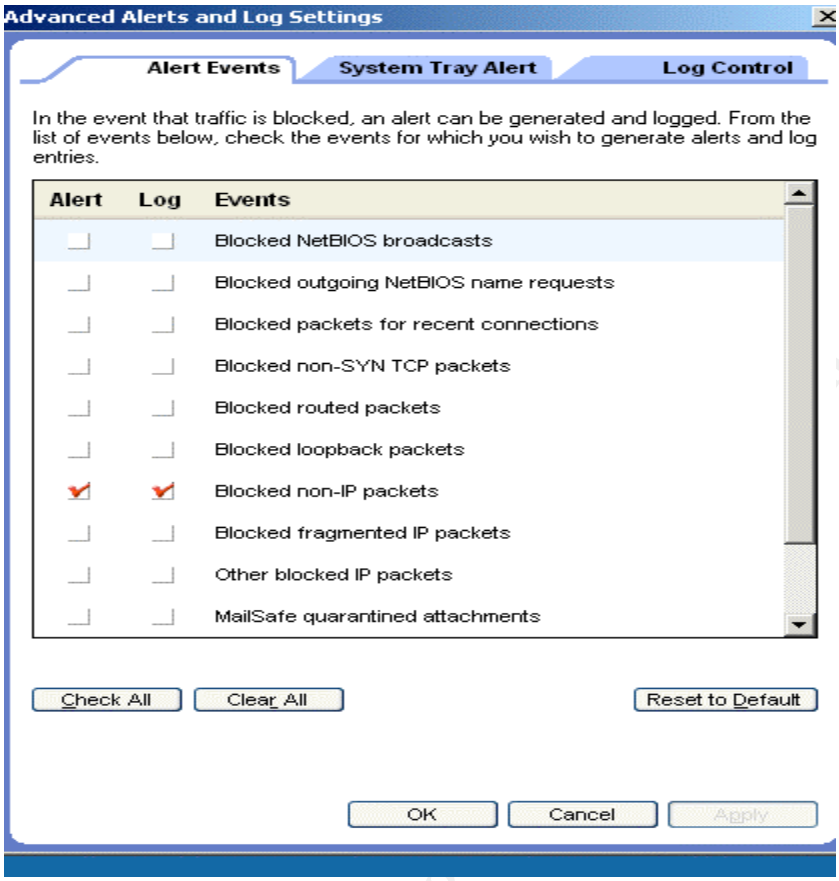


4. VPN: Though extremely difficult to deploy and slow a network could require that all data going over it be encrypted end to end.
5. One time passwords: If one time passwords or other two-factor authentication mechanisms were used (secure token etc.) it would be less likely that a person sniffing the network could reuse the same authentication information to enter the payroll system.
6. Desktop level firewalls: Desktop level firewall if programmed correctly, could be set to ignore ARP (or any request) from unknown sources. This would prevent ARP cache poisoning from occurring. (Note: This protection is not on by default. The screen shots below will show you how to prevent this attack using Zone Alarm Pro.)

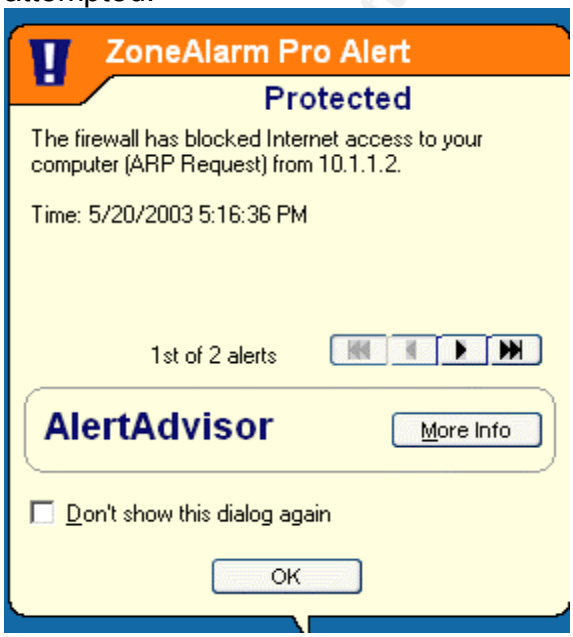
The configuration to turn on ARP Alerts is in the advanced tab of Zone Alarm: (It is unclear why this would not be on by default)



Once this is done, Zone Alarm should be configured to log “blocked non-ip packets”: (again in the advanced tab)



If configured correctly you will receive an alert when ARP cache poisoning is attempted:



7. Modification to the ARP protocol: Though unlikely, the ARP protocol could be modified to only accept answers to an ARP request after they have been asked.
8. ARPWatch: ARPWatch or similar tool listens to the network (via the monitoring port of a switch) and maintains a table of MAC to IP translations. If ARPWatch detects a change (most likely indicating a reconfigured NIC or ARP poisoning attack) it sends an e-mail to the administrator to research.

Example ARPWatch Email:
Date: Wed, 16 Apr 2003 14:26:15 -0400
From: Arpwatch <arpwatch@localhost.localdomain>
To: root@localhost.localdomain
Subject: flip flop

hostname: <unknown>
ip address: 192.168.1.1
ethernet address: 0:4:5a:2d:d2:6b
ethernet vendor: <unknown>
old ethernet address: 0:10:a4:b0:44:74
old ethernet vendor: Xircom RealPort 10/100 PC Card
timestamp: Wednesday, April 16, 2003 14:26:07 -0400
previous timestamp: Wednesday, April 16, 2003 14:26:06 -0400
delta: 1 second

All IP based systems running on a LAN should be protected with some or all of the above methods to prevent against these types of sniffing attacks.

The Incident Handling Process

Preparation

There are a number of steps that a company should do to prepare for the unexpected like a computer breech. In this case, Company M, only had some of these policies in place.

The first step in preparation is Policy. I am referring to internal computer policies / warning banners and response strategies. Company M did have an internal computer policy in place, though this policy could have been more stringent. This policy expressly prohibited the use of sniffers and prohibited outside computers from being plugged directly into the Company's network, both of which were violated in this incident. The company also had a logon warning banner, but it is my opinion that this banner was insufficient. The logon banner did protect the company's right to trace and read data on its systems as seen in the following excerpt:

> *Users should not have any expectation of privacy with respect to information or communications stored on company systems, and such information and communications may be monitored, retained, used or deleted by the company at its discretion.*

The company's policy however did not expressly prohibit unauthorized users from accessing a system that they were not supposed to. The policy also did not expressly state that any data can and would be turned over to law enforcement as needed.

The response strategy also failed in some ways. The first step in a response strategy is the notification process. Company M did have a formal notification process in place. This policy stated that if a security breech was detected, the following people were to be notified: the CIO, the VP of HR, the VP of Public Relations, Lead Council, the Director of Networks and the Manager of Network Operations. The 24/7 NOC was responsible from making the appropriate calls as needed. Which can be seen from the notification list, the company did have a strong cross-department team in place.

One of the most difficult decisions during an incident is whether or not to alert law enforcement of a potential security breech. The internal policy stated that law enforcement would only be alerted if the CIO, VP of Public Relations and Lead Council all agreed that it was necessary.

A number of communications methods were in place, if needed, during an incident. In addition to standard electronic communication methods there was an out-of band email method (ie. hotmail accounts with PGP.) As well there were cell phone and pager capabilities. A war room was available in one of the

company's buildings that featured all necessary equipment including hardware, tools and electronic communication equipment normally found in a "jump bag." Company M did not feel that a "jump bag" was needed, as the war room was within walking distance to all of the office buildings. In addition, a bridge conference system was in place that all users could dial into, world wide.

The response strategy did not take into account any notification of business partners. Though not directly applicable in this incident, there were no formal policies on how to deal with external partners when an incident occurs. Company M did have a number of business relationships, and provided various types of network access for these partners (frame relay, VPN etc.) thus this change in policy was strongly suggested.

One issue that the response team did not consider was access to the systems. It was assumed that the Network Operations Manager could provide access to all systems. In this particular incident the manager did have the appropriate access; a later review showed that this access was not universal.

An agreed upon reporting process was set in the computer administrative policy. If an incident occurs, all parties involved were to call into the bridge every two and half hours for a status update. After twelve hours if the incident was still occurring, and deemed important enough, the CEO was to be added to the conference call twice per day.

Communication within the company is a large part of being properly prepared for an incident. For Company M, this was easily solved because of the size of the company (less then five hundred employees) and the reporting structure. The technical members of the Incident Handling Team all had a working relationship with all of the other members of the technology staff and the application development team.

Company M also lacked a true disaster recovery plan. The company was well aware of this issue (as the CIO asked for money from the board annually to develop such a plan) but was unwilling at this time to finance it. There were only off-site backups performed on a standard weekly rotation.

Identification

The incident detailed within was identified during a routine financial audit of the payroll accounts. At first this abnormality in the reconciliation was discovered by a junior level member of the accounting group. After rechecking his work, he passed his finding on to the Comptroller. The Comptroller verified his findings and contacted the VP of HR, Benny to determine if the changes in payroll were accurate or an "error" had occurred. Benny researched the incident by contacting the payroll clerk (who was the) as well as the manager, director and VP of the affected employees. After confirming that the raises were not

intentional, Benny contacted the payroll company. The payroll company had archived the weekly files received from Company M. These files were decrypted, and the payroll company identified that the rate change was being sent from Company M five weeks prior. It was at this point that the company engaged the Incident Handling team.

Upon hearing the details of the information, the event was immediately classified as a potential incident, and the entire team was engaged. Based on how the infrastructure was laid out, it was unlikely that any automated systems would have detected this event. Company M did not have any type of IDS in place (in particular one that would detect ARP Cache poisoning.) No one was specifically responsible for reviewing web server access logs and/or firewall logs to make sure that only the appropriate personnel was accessing the systems. In addition there were no devices in place to detect foreign machines being added to a network and/or network cards in promiscuous mode.

The attacker in this case was not extremely technical, and thus the other standard signs of an intrusion: new user accounts, log file manipulation, multiple unsuccessful logons or similar were not attempted. Even if the attacker had left such signs, it is unlikely that the company would have detected the intrusion without a policy and/or tools in place to assist.

Once the systems in question were identified, the team worked backwards to attempt to decipher what had happened based on the signatures left on the systems. The team used both perimeter (firewall) and system level (IIS) logs to perform the analysis. There was no host level protection on this server.

First the team took a look at the IIS log files, for the week during which the data change occurred.
This is a sample of the data that was reviewed:

> 2003-05-17 17:15:34 10.1.1.62 - 10.1.2.160 80 GET /PPPP/ - 401 Mozilla/4.0+(
> compatible;+MSIE+6.0;+Windows+NT+5.1;+.NET+CLR+1.0.3705)
> 2003-05-17 17:15:51 10.1.1.62 usera 10.1.2.160 80 GET /PPPP/ - 200 - Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1;+.NET+CLR+1.0.3705)

From these logs, the incident team was able to identify the userid used to gain access to the system (usera) in addition to the IP address of the system used to access it (10.1.1.62) and the time of the logon. After this data was located, the team retrieved the firewall log files for this same time in order to confirm that this particular user was indeed the attacker and to make sure an intruder had not manually altered the IIS log files. The following data was located:

"183887" "17May2003" "17:15:35" "VPN-1 & FireWall-1" "eth-s1p1c0" "MainFirewall" "Log" "Accept" "http" "10.1.1.62" "10.1.2.160" "tcp" "48" "1060" "" ""
"183898" "17May2003" "17:15:51" "VPN-1 & FireWall-1" "eth-s1p1c0" "MainFirewal" "Log" "Accept" "http" "10.1.162" "10.1.2.160" "tcp" "48" "1084" "" ""

From this analysis it was concluded that the attacker was 10.1.1.62, and that they were able to use usera's id to access the system. It was known that usera's system was not 10.1.1.62, and thus more evidence was pointing towards an intrusion. At this point the event was classified as an incident.

Containment

The containment phase of the incident handling process is used to prevent the incident from getting worse. At the same time, this phase is used to acquire the necessary data (following procedure) to be used in legal proceedings if necessary. In this incident Company M did not handle the containment phase as it should have.

The first step the incident team did was to contact the payroll company. The team asked that the payroll company ignore all further data communications between their systems until further notice. Company M only made modifications to this system sporadically (approximately one or two times weekly) thus Company M defaulted back to verbal changes with the payroll company until the systems were secured.

The second step was to perform a backup of the victimized system. This procedure was done using the standard nightly backup rotation with BackupExec. This was a poor choice for backup, since it was not a bit by bit backup. Depending on the actual incident valuable data could have been lost without a bit by bit copy. WinDD, Ghost (though it must be configured correctly) or similar should have been used to perform a bit by bit backup. To configure Ghost correctly, you must run it from the command line with the –ID switches to capture all of the disk including the unpartitioned space (http://service1.symantec.com/SUPPORT/ghost.nsf/pfdocs/1998082612540625.)

The third step towards containment was the changing of usera's password. This too was a poor choice, as it would potentially tip off the intruder that they had been detected. In this particular incident, the intruder would have easily been able to regain the users password, even after the change was put in place. At the same time the company required that usera reset their password on all systems they had access to. It was my suggestion that this be expanded company wide, but the rest of the incident team declined to do so.

The fourth step was to analyze the data found on the backup copy of the system. Unfortunately, all of the data that could be analyzed had already been done in the identification phase.

The fifth and final step was to provide management with a report and a recommendation. Our recommendation was that the company should continue to verbally communicate with the payroll company (as opposed to using the compromised system) until the attackers could be identified and the system could be secured. In this case, management agreed with the team's recommendations, since there would be little business impact.

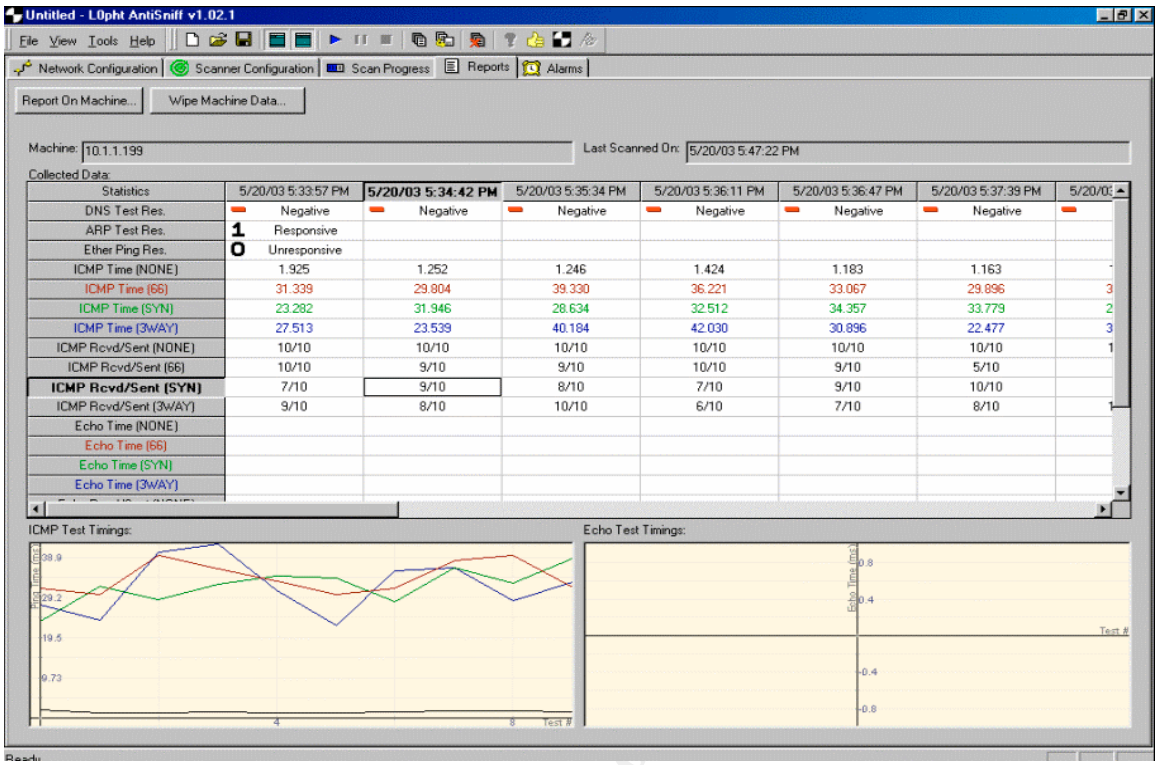
During this process only very few tools were used to contain the incident. A text editor was needed to review the IIS and firewall log files. If the incident team has properly used the backup procedures, at least three additional hard drives would have been used with WinDD. The team would have needed a system configured to analyze the WinDD backups. Finally the conference bridge was used to provide updates during the agreed upon time frames as per the policy.

During the entire process all communication and notes were entered into a numbered workbook. This allowed for better evidence should the case go to court.

Eradication

From the previous phases the incident team was able to identify the source of the problem; unauthorized access using a valid username and password combination. In order to eradicate this vulnerability the team needed to identify the attacker as well as the methods used to perpetrate the system. It was believed that the attacker may have been associated with one of the employees whose pay had increased. It was also assumed that the attacker must have come from the IP address of 10.1.1.62. The incident team quickly identified this IP as the one belonging to Badh, who incidentally was one of the hourly workers whose pay had increased.

With HR present the incident team entered Badh's office during off hours. Upon entering the office a non-company M issued computer was seen attached to the network. From this evidence it was likely that Badh was the attacker. After this equipment was identified the incident team gained access to the switch to identify the IP address assigned to the foreign machine. At the time the legalities of accessing this machine were unknown. After identifying the IP, the incident team ran Antisniff against the NIC (picture below.) It came back in promiscuous mode, indicating that it maybe being used as a sniffer. This port was immediately disconnected from the network.



The fact that a non-company M issued machine was connected to the network, was enough cause to have Badh terminated. In addition the identification that the NIC associated with this machine was in promiscuous mode, was evidence to now require all users passwords in all systems to be changed.

When Badh returned to the office the next morning he was brought to HR. He was accused of breaking into the system to change pay rates. HR explained that if Badh was to work with the incident team to identify what he had done, Company M would not press charges. Badh explained all of the details on how he attacked the system, he also provided the incident team with a copy of his personal hard drive for further analysis. The team dissected his hard drive looking to see if other exploits were used, or other backdoors left connected. Unable to identify additional exploits the team considered the system eradicated, however the system was not yet considered secure.

The incident team also suggested to management that a security review and intrusion detection analysis be performed by a consultant. Management declined based on the associated costs.

Recovery

The first step in the recovery process was for the system owners to validate the data in the current system. The incident team hoped that the system would not need to be restored from tape, as they expected that the only change was the

data in some of the tables. The system administrators were able to verify that no changes to the OS or custom applications had occurred.

Since the data within the system was sensitive in nature (employee salaries) a subset of the incident handling team was able to work with the HR and payroll staff to sift through the data. Each record within the database was compared to the records recovered from the backup prior to the incident. Each changed record was flagged for HR review. HR reviewed each record to determine if the particular change was authorized. After reviewing each record the only unauthorized changes in the system were those for the hourly works which had already been identified. Each of these transactions were reversed.

Next the incident team researched ways of protecting against the sniffing attack used in this incident. The team identified a number of ways to more securely protect the web based systems. All web based systems within the organization were changed to digest authentication. In addition, each of the systems was issued an SSL certificate for an additional layer of security.

The incident team explained to management that additional security methods that should be put in place, but most were dismissed because of external and/or internal management costs. However ARPWatch was placed on the network to identify future ARP Poisoning attacks.

Once agreed upon, preventive measures were put in place. The incident team then reviewed the data one last time with the HR department. When this was data was verified it was left to the HR Department to put the system back into operation. The incident team did agree to monitor the logs of the server and the firewall to attempt to pinpoint any additional strange behavior for the following six months. In addition the incident team agreed to do a monthly data review of the system with HR.

Lessons Learned

Once the attacker was removed, the systems secured and put back into service a “lessons learned” meeting occurred within the organization. A “Final Report” was issued by the incident team. Covered within the report was a detailed description of the incident, how it was handled and what preventative actions were put in place for future incidents. In addition the document covered suggested future training (for both the incident team as well as the standard user community) and future technological changes and/or policies that should be put in to place.

The detailed description of the incident discussed all steps that were used by the attacker. The document began with an overview of the attempted SSL spoof technique. This portion of the document was eventually distributed to all network personnel within the organization. The document then continued with the

detailed information on how the attacker was able to gain access to the LAN, and ultimately acquire the usernames and passwords of the victim.

The report then discussed what policies and procedures have changed as a result of the incident:

1. All basic authentication website were changed to digest authentication with SSL
2. ARP Watch was installed to detect foreign systems being plugged into the network in addition to the flip-flop of MAC and IP addresses.
3. Anti-sniff was added to the network administrators' tool bag. These test were scheduled to be performed on an unannounced schedule in order to detect sniffers on the network.

The document also contained a number of suggestions that were sent to management:

1. Update the user policy to prevent unauthorized users from accessing systems.
2. Modification to the business partner alert strategy. While the incident team was working on the case it was obvious that there was no documented business partner contact strategy, should an incident involve a partner.
3. Unified system access. While working on the incident, the team also determined that in certain circumstance they may not have the correct authority to access all systems.
4. The creation of a log files analysis system. Such a system would be able to point out suspect usage patterns.
5. The introduction of an IDS and host based intrusion tools.
6. An internal study of employee behavior by HR to determine if these types of incidents are predictable, and what from an HR perspective could be done to prevent them in the future.

The final report also contained each of the steps and technologies that were used to ultimately catch the attacker. The report also featured a detailed list of events from the HR perspective, should the attacker cause any legal issues in the future. Finally a group was formed within the IS organization to prepare a security presentation for all of Company M's staff. Covered in this presentation was how security effects each of the employees, and how insecure the Internet may be.

References

Dhar, Sumit. “Web Watch.” March 5, 2002. URL: http://www.linuxjournal.com/article.php?sid=5869

Drury, Jason. “Sniffers: What are they and How to Protect From Them.” November 11, 2000 URL: http://www.sans.org/rr/switchednet/sniffers.php

Duffy, Richard. “Finding Dsniff on Your Network.” November 28, 2001 URL: http://www.sans.org/rr/paper.php?id=262

Radke, Andre. “HTTP Authentication Schemes.” November 13, 1999 URL: http://frontier.userland.com/stories/storyReader$2159

Sipes, Steven. “Intrusion Detection FAQ, Why your switched network isn't secure.” September 10, 2000. URL: http://www.sans.org/resources/idfaq/switched_network.php

Synamntec, “Switches: Alphabetical list of switches (Ghost.)” May 6, 2003. URL: http://service1.symantec.com/SUPPORT/ghost.nsf/pfdocs/1998082612540625

Wagner, Robert. “Address Resolution Protocol Spoofing and Main In The Middle Attacks.” September 27, 2001 URL: http://www.sans.org/rr/paper.php?id=474

Whalen, Sean. “An Introduction to Arp Spoofing.” Version 1. April, 2001 URL: http://chocobospore.org/projects/arpspoof/arpspoof.pdf

Ye, Eileen and Yuan, Yougu and Smith, Sean. “Web Spoofing Revisited: SSL and Beyond.” February 1, 2002. URL: http://www.cs.dartmouth.edu/~pkilab/papers/tr417.pdf

“How SSL Works” URL: http://developer.netscape.com/tech/security/ssl/howitworks.html

“Address Resolution Protocol (ARP.)” February 28, 2000. URL: http://www.microsoft.com/windows2000/en/datacenter/help/default.asp?url=/windows2000/en/datacenter/help/sag_tcpip_und_arp.htm

Tools:

Authentix: (Basic web based authentication mechanism) http://www.flicks.com/authentix_isp/

Base64 Decoder: http://www.opinionatedgeek.com/dotnet/tools/Base64Decode/Default.aspx

Cain and Able (sniffing, arp spoofing and more): http://www.oxid.it/cain.html

Dsniff (sniffing utility for Linux and more):
http://www.monkey.org/~dugsong/dsniff/.)

Demos:

WebSpoofing Demo: http://www.cs.dartmouth.edu/~pkilab/demos/spoofing/

Last Updated: May 23rd, 2017

Upcoming Training

SANS Madrid 2017	Madrid, Spain	May 29, 2017 - Jun 03, 2017	Live Event
SANS Atlanta 2017	Atlanta, GA	May 30, 2017 - Jun 04, 2017	Live Event
Community SANS Virginia Beach SEC504*	Virginia Beach, VA	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS San Francisco Summer 2017	San Francisco, CA	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS Rocky Mountain 2017	Denver, CO	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Charlotte 2017	Charlotte, NC	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Rocky Mountain 2017 - SEC504: Hacker Tools, Techniques, Exploits and Incident Handling	Denver, CO	Jun 12, 2017 - Jun 17, 2017	vLive
SANS Thailand 2017	Bangkok, Thailand	Jun 12, 2017 - Jun 30, 2017	Live Event
Mentor Session - SEC504	Reston, VA	Jun 13, 2017 - Aug 01, 2017	Mentor
SANS Minneapolis 2017	Minneapolis, MN	Jun 19, 2017 - Jun 24, 2017	Live Event
SANS Paris 2017	Paris, France	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Cyber Defence Canberra 2017	Canberra, Australia	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, Japan	Jul 05, 2017 - Jul 15, 2017	Live Event
SANS ICS & Energy-Houston 2017	Houston, TX	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Los Angeles - Long Beach 2017	Long Beach, CA	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, Singapore	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Seattle SEC504	Seattle, WA	Jul 10, 2017 - Jul 15, 2017	Community SANS
Community SANS Sacramento SEC504	Sacramento, CA	Jul 17, 2017 - Jul 22, 2017	Community SANS
Community SANS Ottawa SEC504	Ottawa, ON	Jul 17, 2017 - Jul 22, 2017	Community SANS
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
Community SANS Des Moines SEC504	Des Moines, IA	Jul 24, 2017 - Jul 29, 2017	Community SANS
Community SANS Annapolis SEC504	Annapolis, MD	Jul 24, 2017 - Jul 29, 2017	Community SANS
Community SANS Phoenix SEC504	Phoenix, AZ	Jul 24, 2017 - Jul 29, 2017	Community SANS
Security Awareness Summit & Training 2017	Nashville, TN	Jul 31, 2017 - Aug 09, 2017	Live Event
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
Community SANS Raleigh SEC504	Raleigh, NC	Aug 07, 2017 - Aug 12, 2017	Community SANS
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS New York City 2017	New York City, NY	Aug 14, 2017 - Aug 19, 2017	Live Event