Global Information Assurance Certification Paper

GIAC Advanced Incident Handling And Hacker Exploits

GCIH Practical Assignment

Version 2.1

Option One – Exploit in Action

Denial of Service Vulnerabilities In
Windows SMB Implementation

Submitted by Roger Millen

Abstract....3
1. Introduction....4
2. The Exploit....4
2.1 Name....4
2.2 Systems Affected....5
2.3 Brief Description....6
2.4 Variants....6
2.5 References....7
3. The Attack....8
3.1 Description & diagram of Network....8
3.2 Protocol Description....9
3.3 How the Exploit Works....12
3.3.1 Buffer Overflow....12
3.3.2 The SMB Buffer Overflow....14
3.3.3 Manually Running the Exploit....15
3.4 Description of the Attack....15
3.5 Signature of the Attack....16
3.6 How to Protect Against It....17
4. The Incident Handling Process....17
4.1 Preparation....17
4.1.1 The PC's....18
4.1.2 The Servers....19
4.1.3 The Infrastructure....20
4.1.4 Incident Handling....20
4.2 Identification....20
4.3 Containment....23
4.4 Eradication....23
4.5 Recovery....23
4.6 Lessons Learned....24
References....26

Abstract

While choosing a subject for this practical I researched several areas, read some of the completed practical's written by other candidates, and searched the web for an interesting subject. I wanted to add some value to my practical by including information about the type of environment, a University environment, in which I have worked. This environment is very different than a sanitized business environment because of its independence and diversity. Students, faculty and staff all have differing goals that make deployment of a secure environment difficult.

I choose this vulnerability because it was recently discovered, had a tool for exploiting it readily available, and had the potential to cause major damage in the hands of an aggressive attacker. Students are in tune with current exploit activities and quickly utilize these tools as toys to harass other students. The incident is typical of the type of thing we would see in a student lab, to a point where it is not taken as seriously as it should. This kind of environment is something businesses should be aware of, since upon graduation, these same students are being placed in those businesses with more than the applicable background that it originally hired them for.

This vulnerability is a fairly simple one, but contains a lot of the components of much more complex exploits. When the Microsoft bulletin was issued, it was only a matter of days before a tool was available and unprotected systems could be attacked. While writing the practical, the related Cisco vulnerabilities were announced only a few days before finishing a first draft. How much damage can be done with such a simple tool in an unprotected, unpatched environment? It is imperative that systems people keep their systems patched and have a system in place to stay current. Others, not necessary good guys, certainly are.

1. Introduction

The incident described in this paper is not an actual event. It is written for purposes of illustrating the exploit. The events as described are intended to describe some of the issues that face support and security personnel in a real world environment. The network is an actual configuration located in a single college in a University environment. Because of the diverse nature of this type of environment, security in one unit can be much different than the one described here. Financial and personnel resources are extremely strained in some units. This leads to leaving entire subnets unprotected which in turn makes them extremely vulnerable to being used as zombie systems.

Although many central IT organizations in a University will argue that they do not have control over individual units or the resources to monitor every unit, it is their responsibility to educate them by delivering resources and developing relationships to ensure the entire University is as secure as possible.

EDUCAUSE has taken a good first step in uniting Universities by creating a task force. http://www.educause.edu/asp/doclib/abstract.asp?ID=NET0027

"National leaders of higher education have endorsed a five-part Framework for Action for cyber security and are organizing a series of four NSF-sponsored workshops that will involve the entire community in the development of a more coherent national strategy"

Awareness and education is the key to creating secure educational institutions. My description of the incident is how I believe support personnel would react today to this kind of exploit. Hopefully, as more of us dedicate ourselves to take advantage of our security education to strengthen the effectiveness of these proposals, securing environments as well as handling of incidents will become standard procedure.

2. The Exploit

2.1 Name

"Microsoft Windows SMB Denial of Service Vulnerability" is the exploit to be examined.

Common Vulnerabilities and Exposures (CVE) has assigned this exploit as candidate number CAN-2002-0724. CAN-2002-0724 (under review)

"Buffer overflow in SMB (Server Message Block) protocol in Microsoft Windows NT, Windows 2000, and Windows XP allows attackers to cause a denial of service (crash) via a SMB_COM_TRANSACTION packet with a request for the (1) NetShareEnum, (2) NetServerEnum2, or (3) NetServerEnum3, aka "Unchecked Buffer in Network Share Provider Can Lead to Denial of Service".

The CERT Coordination Center has issued the following vulnerability notes:

1) Vulnerability Note VU#311619 http://www.kb.cert.org/vuls/id/311619

Microsoft Windows Server Message Block (SMB) fails to properly handle SMB_COM_TRANSACTION packets requesting NetServerEnum3 transaction.

2) Vulnerability Note VU#342243 http://www.kb.cert.org/vuls/id/342243

Microsoft Windows Server Message Block (SMB) fails to properly handle SMB_COM_TRANSACTION packets requesting NetShareEnum transaction.

3) Vulnerability Note VU#250635 http://www.kb.cert.org/vuls/id/250635

Microsoft Windows Server Message Block (SMB) fails to properly handle SMB_COM_TRANSACTION packets requesting NetServerEnum2 transaction.

2.2 System Effected

The following Microsoft Windows operating systems are affected by this exploit:

- Microsoft Windows NT 4.0 Workstation
- Microsoft Windows NT 4.0 Server
- Microsoft Windows NT 4.0 Server, Terminal Server Edition
- Microsoft Windows 2000 Professional
- Microsoft Windows 2000 Server
- Microsoft Windows 2000 Advanced Server
- Microsoft Windows XP Professional

Note: All Service Pack levels are affected.

This exploit also affects multiple Cisco products that use one of the vulnerable Windows operating systems.

Cisco Access Control Server (ACS) Any version
Cisco BBSM Any version
Cisco Broadband Troubleshooter (CBT) Any version
Cisco CallManager 3.0.x
Cisco CallManager 3.1.x
Cisco CallManager 3.2.x
Cisco Collaboration Server (CCS) Any version
Cisco DOCSIS CPE Configurator Any version
Cisco Dynamic Content Adapter (DCA) Any version
Cisco E-mail Manager (CEM) Any version
Cisco ICS 7750 1.x
Cisco ICS 7750 2.x

Cisco Intelligent Contact Manager (ICM) Any version
Cisco Media Blender (CMB) Any version
Cisco Network Registrar (CNR) Any version
Cisco Secure Policy Manager (CSPM) Any version
Cisco TrailHead Any version
Cisco Transport Manager (CTM) Any version
Cisco Unity Any version
Cisco User Registration Tool Any version
Cisco Works 2000
Cisco uOne Enterprise Edition Any version

2.3 Brief Description

This exploit is a result of an unchecked buffer in the SMB request function allowing a malformed request packet to cause a system crash resulting in a denial of service. A malformed packet could be sent by either a legitimate user or by anonymous access. It may be possible for an attacker to execute code through this exploit, but it is not known to be an issue at this time.

A Server Message Block (SMB) is the protocol that several developers use for access to shared files, printing functions, serial ports, and named pipes. SMB rides on top of NetBIOS session services in the higher application and presentation layers of the OSI model. SAMBA utilizes this protocol for access to Windows files on UNIX systems. The open source version is called the Common Internet File System (CIFS) and is being explored for use in other internet applications such as FTP and HTTP. A more in depth look into SMB will be included in Part 2 of this paper.

A buffer overflow is a condition that occurs when a defined array area is unchecked in an application for length and allows the buffer to overrun. This condition can be exploited to cause a return in the stack to either execute improper code or hang the system. This is not a specific operating system vulnerability, although operating systems can have unchecked code included in them, but rather an application vulnerability caused by improper programming practices. More technical explanations can be found later in this paper, and links to various resources are in the references section at the end of this paper.

2.4 Variants

I found no variants of this particular exploit, but the SMB protocol has been exploited in a number of different fashions. SMBRelay is a tool that has been used for several purposes such as man-in-the-middle attacks and can be setup as a trojan as a rouge SMB server used to collect passwords. There was also a NT exploit where a DIR../ from a SAMB client would cause the system to crash.

There are numerous buffer overflow exploits for many operating systems and applications. Any device that contains an operating system or application can be vulnerable to this type of exploit. Routers, switches, DNS Servers, Web Servers, palm devices, pocket PC's, and wireless devices can all be potential targets.

2.5 References

Microsoft Security Bulletin MS02-045 "Unchecked Buffer in Network Share Provider Can Lead to Denial of Service (Q326830)"

Microsoft Security Bulletin MS02-45

Microsoft knowledgebase article Q32680 "MS02-045: Unchecked Buffer in Network Share Provider May Lead to Denial-of-Service"

Microsoft Knowledgebase Article Q326830

These vulnerabilities were discovered and researched by Alberto Solino and Hernan Ochoa of the Security Consulting Services team at CORE SECURITY TECHNOLOGIES.

CORELABS – Advisories

Cisco Security Advisory 2002 September 18 16:00 (UTC -0400), "Cisco Security Advisory: Microsoft Windows SMB Denial of Service Vulnerabilities in Cisco Products - MS02-045"

Cisco Security Advisory

NTBugTraq vulnerability #5556 "Microsoft Network Share Provider SMB Request Buffer Overflow Vulnerability"

Microsoft Network Share Provider SMB Request Buffer Overflow

Proof of Concept tool and source code.

Proof of Concept

See section 1.1 for CERT advisories and related links.

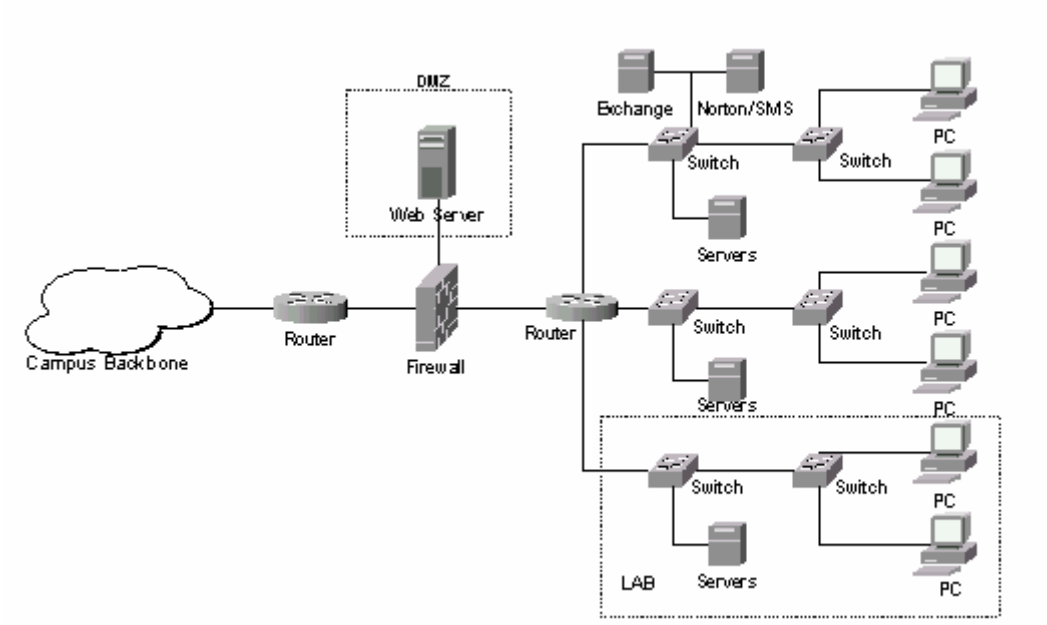
3. The Attack

3.1 Description and Diagram of Network

This network is located in one college unit on a University campus. The computer room is located in an environmentally controlled room with access restricted by smart card authentication, and video monitoring is installed. The illustration is a subset of the actual environment but includes all major components.

Router -	Cisco 3600 Series
Firewalls -	Checkpoint FW-1 V4.1
Switch -	Cisco 2948's
Servers-	Dell Power Edge 2600's and 4600's
Desktops-	Dell Optiplex's

Figure 3.1



The outside router is managed by central campus services and has anti-spoofing ACL entries. The internal router is configured to only route internal traffic, allows internal traffic to access the outside, and restricts access to any of the internal subnets from the student lab network. The environment contains 3 subnets, one of which is a student lab. There are approximately 500 PC's and laptops and all of them run the Windows 2000 operating system with service pack 3 installed.

There are approximately twenty file servers running a mix of Windows 2000 and Windows NT. All subnets are a configured as part of a single domain with no trusted networks. Active Directory has not been implemented as of yet. Microsoft Exchange V5.5 server runs on Windows NT as well as other that are

running applications that vendors will not support under Windows 2000 so those systems are still running Windows NT. There are three modems connected on two servers that are running RAS for dial-in access to email. There is a Microsoft SMS server that is used for distributing applications and patches as well as being used for remote support. Print services and Norton anti-virus management server also run on the SMS server. Only necessary services are running on the servers and differ depending on infrastructure services they perform such as WINS, DHCP and relays, etc. The recommendations on page 33 in the SANS Securing Windows 2000 documents were followed and only necessary adjustments were made depending on the applications running. A more detailed description of server configurations is given in the incident handling section 4.1 on preparation.

The firewall is a Checkpoint V4.1 system running under Windows NT. The rules are configured as deny everything except for what is necessary to come in. The mail server is internal and not in the DMZ therefore the SMTP port is open to the mail server. Terminal services are also open for remote administration purposes. There is a DMZ that contains two web servers. One of the servers is a development server that mirrors the production server. VNC is used for access so that port is open to that server only. The switches are standard configuration without any VLANS.

3.2 Protocol Description

As described in section 2.3 the SMB protocol is utilized for accessing shared files, directories, and devices. SMB is a higher-level protocol that rides on top of other protocols. In the case of Windows it rides on top of NetBIOS, which in turn rides on top of TCP/IP. Figure 3.2 maps NetBIOS and SMB in the OSI reference model and is given to give the reader an idea of how SMB maps but NetBIOS doesn't map well to it.

> "The diagrams below attempt to show the components of the NetBIOS protocols and higher level protocols such as SMB in relation to the OSI Reference Model. Because the protocols were not developed specifically to comply with the OSI model any mapping is only approximate and intended as a guide. When protocols (such as NetBIOS) are encapsulated within other protocols (such as TCP/IP or IPX) it is particularly difficult to map these to a reference model, thus the diagrams below are intended to help show the relationships between the

protocols rather than provide a definitive mapping to the OSI model. [1]

Figure 3.2

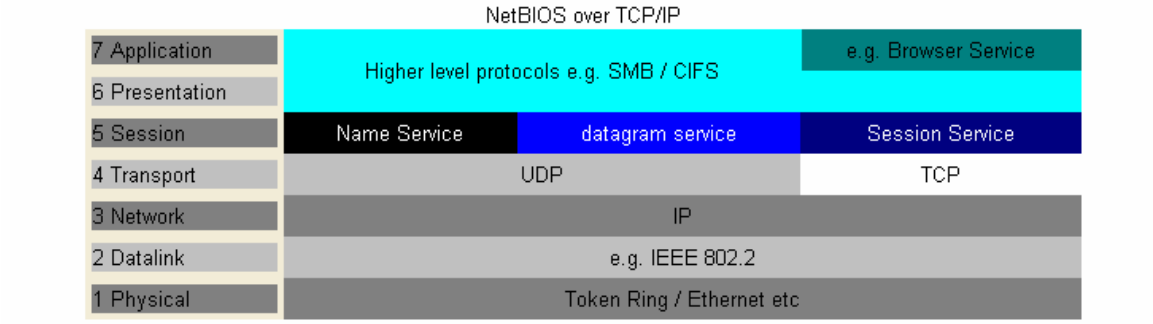
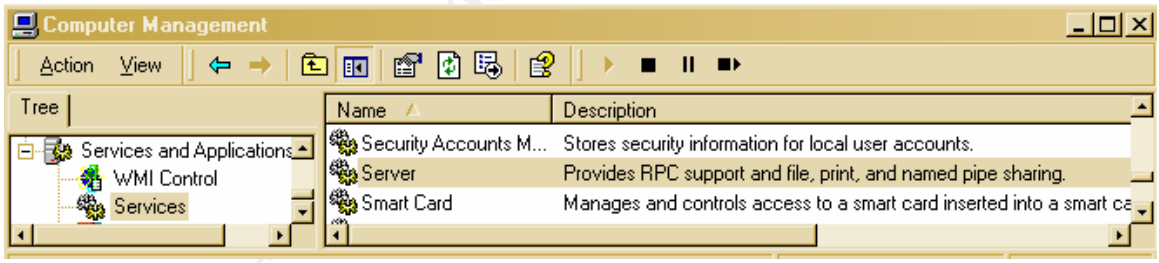


Figure 3.3 shows that in a Windows NT and 2000 environment SMB is part of the Server service. Since the computer browsing function also utilized SMB it is dependant on the Server service.

Figure 3.3



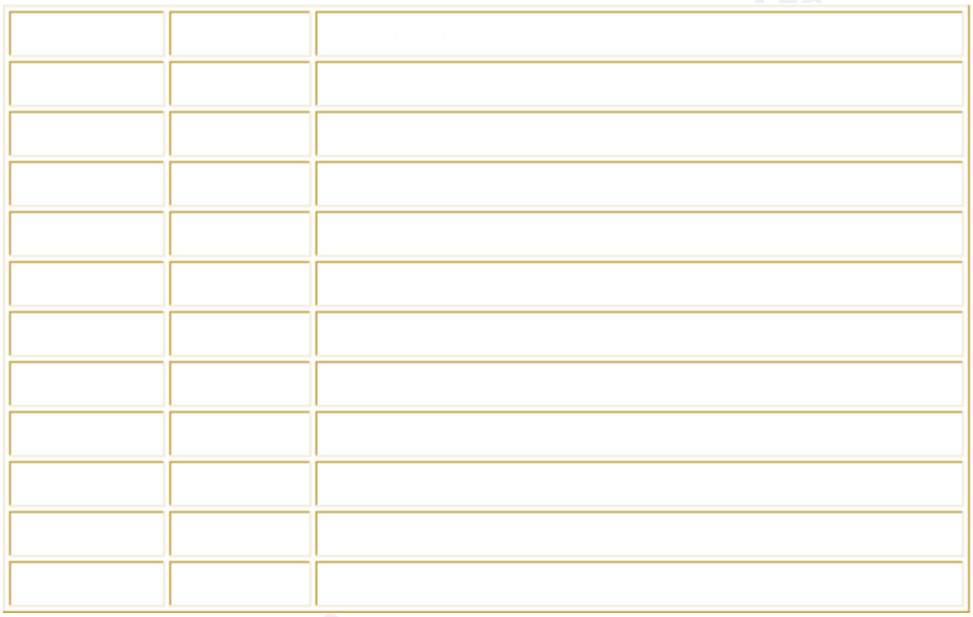
There are many variants and dialects of the SMB protocol. It was originally written by IBM in 1987 and has gone through several changes and additions over time. Microsoft has maintained backwards compatibility throughout their different iterations and several of the other variants are compatible as well.

The SMB protocol operates in a client/server model. With this protocol each machine can be either a client or a server. The protocol itself has two parts. First there is the header information, which is a fixed size, and secondly there are the command strings, which can vary in size dramatically depending on the contents of the message.

[1] SMB in the OSI model

The following table (Figure 3.4) “shows the format of an SMB header. SMB commands are not required to use all the fields in the SMB header. For example, when a client first attempts to connect to a server, it does not yet have a tree identifier (TID) value - one is assigned after it successfully connects - so a null TID (0xFFFF) is placed in its header field. Other fields may be padded with zeros when not used.”[2]

Figure 3.4

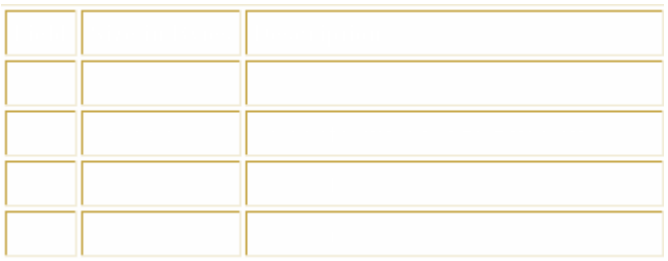


Following the header is the command or reply information. There are many available commands and codes. Some of the core commands are reading, writing, and deleting files and directories, starting and ending connections, verifying dialects, sending messages and broadcasts, etc. Command codes are successful request, bad password, and message sent and received codes.

The commands or reply are not of fixed length and vary depending on the transaction. Each command parameters and data will be different and not all of the fields need to be filled and are padded with zeros as they are in the header.

Figure 3.5

[2] From Using Samba



A SMB session is set up through the above protocol format in a request and response negotiation. First the client and server must establish a connection at he NetBIOS level. After the connection has been made, the client and server must decide on which dialect of SMB that they will communicate with and then the client is set up to make requests from the server.

As stated earlier, there are many dialects of the SMB protocol. A SMBnegprot command is sent from the client to the server with a listing of its supported dialects and the server responds with a listing of the dialects it supports or if it doesn't support any at all. If at all possible, an attacker would try to negotiate one of the earlier levels of SMB. Earlier dialects did not require a logon so access to information form these earlier supported dialects was readily available.

In more recent dialects, the next step would be the logon process. In the case of a Windows network share, each shared folder has its own set of possible users that is set up through the share permissions process. Default permissions on shares are full control for everyone, so it is important to be astute about changing the permissions when giving access to a shared folder. A logged on user will receive a user ID (UID) that is used in all subsequent request to the server.

After a user has logged on it makes a connection to a tree. The connection is an InterProcess Communications resource (IPC$) over which named pipes transactions are carried. The server supplies a tree connection ID (TID) that becomes part of the header and is also required to be sent by the client for identification. It is possible for a client to have more than one TID active at a time. Once connect to the tree, a client can issue commands to the server such as open file, read file, etc.

Referring to figure 3.4 the UID and TID are each 2-bye block within the SMB header.

SMB is also the protocol used for the browsing service. The Windows browsing service is what is used by "Network Neighborhood" to view available machines in a subnet. Since NetBIOS is a non-routable protocol the WINS service must be run in order for this information to be accumulated across subnets. The browser service accumulated NetBIOS names dynamically and makes that list available to systems on the network. It is possible then to do a "node status query" on the master browser an gather all available NetBIOS

names in one query. Then reverse NBT lookups cam be done to gather IP address information.

3.3 How the exploit works

3.3.1 Buffer Overflows

This exploit is a buffer overflow denial of service attack. A brief explanation of a buffer overflow is given in section 2.2. Links to some in depth technical papers concerning the subject are listed in the references section at the end of this paper.

Buffer overflows represent on of the most critical problems with applications on the Internet today and is one of the most common sources of security risk. The problem is largely a product of poor programming practices. Input information should be validated by the application code and if it is too large or is the incorrect data type the program should exit gracefully.

In order to understand how this exploit works, it will be necessary to get a little more information about how buffer overflow attacks work. A buffer is an allocated space in RAM that a program sets up to hold instructions and variables. Buffers can and are located adjacent to each other within the RAM. If the program does not check to ensure that only the maximum amount of information is either entered into a field or written to memory an overflow occurs writing information into areas allocated for different instructions or variables.

Two things can occur when this condition exists. First, important instructions for a program are overwritten causing the system to hang causing a denial of service. Second, the program returns to an overwritten portion of RAM that an attacker has placed executable code in, and then the program runs the malicious code instead of program code. This is potentially a disastrous situation as the executable code could accomplish a number of events including deleting or stealing files, run remote access trojan client programs, or completely destroy a system.

Buffer overflows can occur in a variety of ways. Below is a list of some of the most common types with a brief explanation of how they work.

Static buffer overruns

A static buffer overrun occurs when a buffer, which has been declared on the stack, is written to with more data than it was allocated to hold. The nonobvious versions of this error occur when unverified user input data is copied directly to a static variable, causing potential stack corruption.

Heap overruns

Heap overruns, like static buffer overruns, can lead to memory and stack corruption. Because heap overruns occur in heap memory rather than on the stack, some people consider them to be less able to cause serious problems;

nevertheless, heap overruns require real programming care and are just as able to allow system risks as static buffer overruns.

Array indexing errors

Array indexing errors also are a source of memory overruns. Careful bounds checking and index management will help prevent this type of memory overrun.

3.3.2 The SMB Buffer Overflow Attack

CERT Vulnerability Note #VU250635 includes the following description:

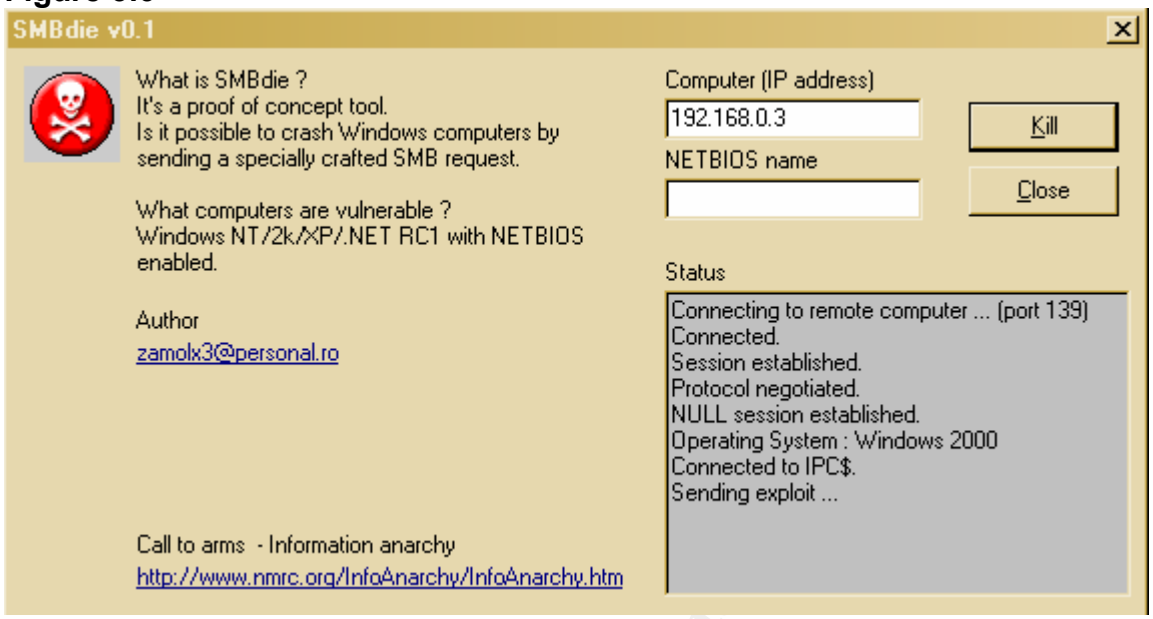
"SMB will crash if it receives a crafted SMB_COM_TRANSACTION packet requesting a NetServerEnum2 transaction. If either the 'Max Param Count' field or the 'Max Data Count' field of the field is set to zero (0), the destination SMB host will crash with a blue screen. This vulnerability can be exploited by both local and remote attackers. "

CERT Vulnerability Notes #VU311619 and #VU34243 state this same information except that they indicate "SMB_COM_TRANSACTION packet requesting *a NetServerEnum3* or *NetServerEnum* transaction" respectively. The other variation on these transactions is that a NetServerEnum transaction requires a user account in order for the exploit to work. NetServerEnum2 and NetServerEnum3 transactions only require anonymous access, which is the default mode for Windows installations.

There is a proof of concept tool called "SMBDie" that demonstrates this exploit. [3] Below (figure 3.6) is a screen shot of the tool. The status window shows the process the program takes. It is built from source code "smbnuke.c" available at the Russian web site http://www.security.nnov.ru/ and a search is done on SMB.

[3] Proof of Concept tool

Figure 3.6



To use this program all one needs to do is enter the IP address and NetBIOS name and click on kill. Getting the NetBIOS name is one of the necessary pieces of information that is needed by an attacker in order to perform this exploit. Internally, gathering NetBIOS names is a fairly simple process by either Once the IP address and NetBIOS name is collected, the SMB tool can be utilized. The process starts by connecting to port 139 on the machine that is to be exploited. It then establishes a NetBIOS session. Because of the various dialects of the SMB protocol it need to agree in which dialect it will communicate and negotiates a protocol. A null session is established through anonymous logon at IPC$ and the operating system is identified. It then sends the malformed packet.

The result is the attacked machine crashes and displays the dreaded blue screen with the following information:

*** STOP: 0x0000001E (0xC0000005, 0x804B818B, 0x00000001, 0x00760065)
KMODE_EXCEPTION_NOT_HANDLED
*** Address 804B818B base at 80400000, DateStamp 384d9b17 0 ntoskrnl.exe

The system will then dump its memory and then reboot if it is configured to do so.

3.3.3 Manually running the exploit

Building a tool to establish a NetBIOS session and sending a SMB request packet if one did not exist could be written using readily available code. For example, SMBlib is a free suite of programs which implement the SMB protocol, and can be used to create a SMB session. Other packet generation tools such as phNetProbe support SMB protocol and be used to generate a packet with the specific parm and data counts set to zero.

3.4 Description and Diagram of the attack

The diagram in Figure 3.1 illustrates the network configuration. Port 139 is blocked at the firewall so even this exploit can be executed remotely a NetBIOS session cannot be established from outside the firewall. Internally port 139 is left open for purposes logon, browsing, and file sharing.

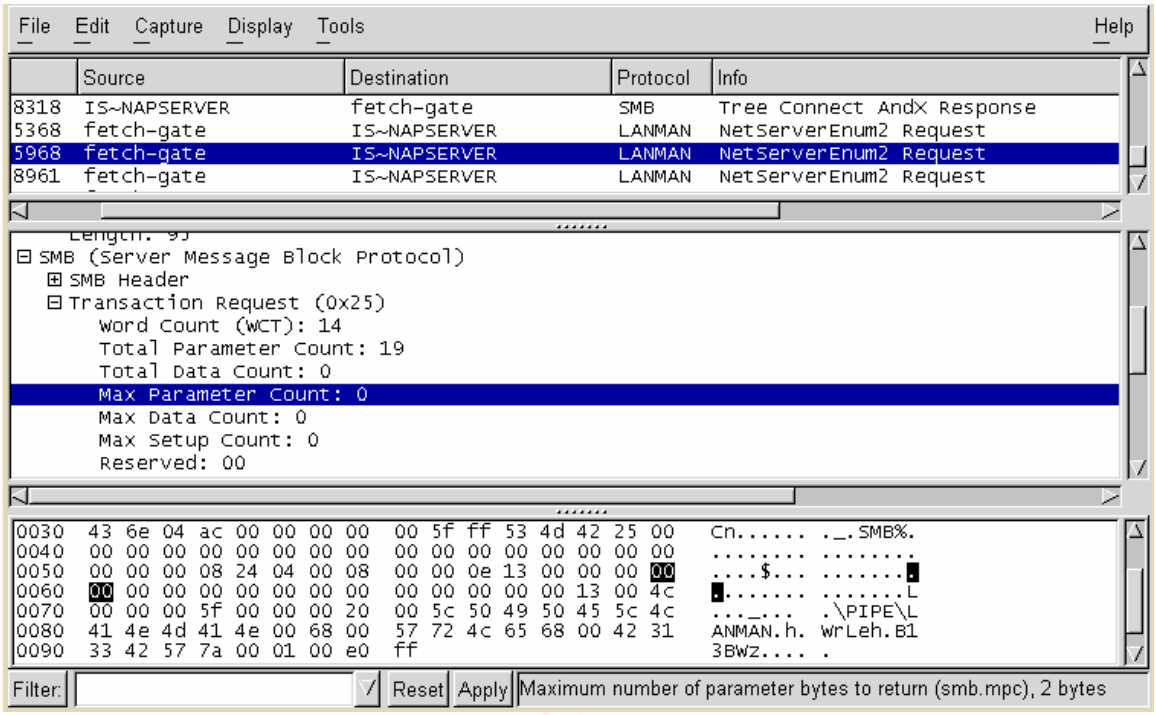
Internally, then any machine has the ability to launch this attack. The only information an attacker would need is the IP address and NetBIOS name of the machine they which to attack. Active IP connections can be found with a port scan of port 139 and then a NetBIOS name can be harvested as described in section 3.3.2. Utilizing the SMBDie tool it would be a simple process to crash the machine.

Figure 3.7 is an Ethereal capture of the NetServerEnum2 request with the buffer overflow. The CERT vulnerability notes state:

"SMB may crash if it receives a crafted SMB_COM_TRANSACTION packet requesting a NetServerEnum2 transaction. If either the 'Max Param Count' field or 'Max Data Count' field of the packet is set to zero (0), the destination SMB host will crash with a blue screen. "

In the SMBdie implementation, both count fields are set to zero causing the system to crash. With these parameters set to zero, no space has been allocated in the stack for information causing the stack to be overrun and the system to crash.

Figure 3.7



3.5 Signature of the attack

On a Windows NT or 2000 machine there will be 2 events in the System event logs. The first event will be a “save dump” informational event that has the following information:

“The computer has rebooted from a bugcheck. The bugcheck was: 0x0000001e (0xc0000005, 0x804b818b, 0x00000001, 0x00760065). Microsoft Windows 2000 [v15.2195]. A dump was saved in: C:\WINNT\MEMORY.DMP.”

The second event will be and error event that says, “The previous system shutdown at <Time> on <Date> was unexpected.” During my test I found that this second entry was inconsistent and did not always get entered into the event logs.

Intrusion detection systems would detect scans to port 139, but since there are multiple exploits for this port it wouldn’t be assumed that it is related to this one. While a IDS rule could be written to look for the Max parm or data counts set to zero, I did not find one written and wouldn’t expect one to be. Firewalls would block or at least should block incoming traffic to this port as well as port 445 (NetBIOS sessions can utilize this port also) making remote exploitation nonexistent. Internally, the supplied patches should be applied. During my tests the session is refused after the Microsoft patch was installed.

3.6 How to protect against the exploit

Install the patches that Microsoft has released a patch for this exploit.[4] Cisco has also released a set of patches.[5] Other methods for protection from this exploit are:

1 – Disable anonymous access, e.g. Null Sessions.

This will not solve the problem from legitimate users.

2 – Block access to ports TCP/139 and TCP/445.

This will block access at the perimeter but if file and print sharing is necessary internally, it is not a viable option on those machines.

3 – Shutdown the LANMAN Server.

This prevents exploitation from any attacker but removes all file and print sharing functionality from the vulnerable server.

4. The Incident Handling Process

4.1 Preparation

The following sections describe details about how systems are configured and administered in order to maintain a secure environment. In this environment, the consideration given to maintaining and configuring systems is one of the areas that this IT department paid fairly close attention to. Indeed, it is one of the most critical areas for attention and also, in my experience, one of the most deficient. A system can only be prepared for known problems and the lag time between exploit and patch is sometimes unacceptable. As the numbers of viruses and other exploits rise exponentially, the idea of gaining a signature and pushing them out to protection software is at some point going to be infeasible. Two things need to happen in the future in order to defeat the proliferation of exploits. First, greater care needs to be taken in the development phase of applications and operating systems. Second, new methods for intercepting viruses, trojans, and other intrusions needs to be developed. Behavior based systems are in the works, but appear to be further down the road than what we would like before any meaningful implementations appear. For now, the time taken to prepare for the inevitable will be returned.

4.1.1 The PC's

All PC's and laptops are running Windows 2000 and are standardized in everyway possible. Only necessary services are running. File and print sharing is installed. Ghost images are made so all machine configurations are known at the

[4] Microsoft TechNet MS02-45 Patches
[5] Cisco Security Advisory - MS SMB DOS Vulnerabilities

time of installation. Patches and updates are pushed out to the machines using SMS after they have been evaluated. Generally, there is a waiting time before major patches or service packs are distributed to see if there are any problem reports. If there is an urgent hot fix that is an issue indigenous to the systems, it is applied immediately. Anti-virus definitions are distributed weekly using the Norton Anti-virus management server. Faculty and staff are asked to leave their systems on during a specified weeknight so that updates can occur without disruption of work. Many faculty and staff use a laptop docking system and are frequently out of the office or take their laptops home in the evening, so constant reminders are necessary to keep definitions up-to-date. Also, tracking patch and hot fix installation is difficult, so email is used to communicate to everyone about system management events that are going to occur. Computer and helpdesk staff will update machines individually by checking the installed list against inventory in the event there are people absent during SMS updates.

The college environment is sometimes difficult to lock down. It is basically an open environment that allows for creative license among the faculty who consider themselves independent contractors for the college, not employees. Systems are purchased from several different financial sources such as grants, personal money, donations, separate departmental budgets, as well as the college itself. Warning banners are not being displayed because while there is general agreement that the network can be monitored, there is no agreement of ownership.

There is a password rotation policy of 100 days and recommendations about choosing appropriate passwords is published in several areas and reinforced at every opportunity.

There has been no agreement concerning downloading of software. The computer staff continually warns users about the dangers of uncontrolled installation of software but no policy exists to prevent it. The computer staff's policy in the event of an issue occurring because of problematic software is to use the original ghost image of the machine and restore individual documents from backup. Backups are done through a central University system using a customized version of Tivoli to a central campus tape environment. The backups are performed incrementally each night. The problem of absenteeism of faculty and staff that have a laptop system exists for nightly backups just like there is with keeping virus definitions up-to-date, so continual reminders about backups are needed.

The student lab, consisting of approximately 100 machines, is set up with ghost partitions and every machine is ghosted every evening. All applications reside on the systems except for one client/server application and only one directory is set up for access on the server for class use. All students use one account and removable media is used during classes. Students utilize the campus email system and have no accounts on the internal system.

Students frequently deliver to faculty, and bring to the lab, disks that contain viruses. Virus activity on student home machines is an ongoing battle

that is fought with free virus software, education, and staff support. Keeping this environment isolated from everything is immensely important to maintain network security.

4.1.2 The Servers

The servers are located in a controlled environment utilizing smart card access and video monitoring. Patches and updates are installed after review for necessity and testing has occurred. Incremental backups are performed nightly and full backups once a week. Each server has been reviewed for necessary services and anti-virus scans are run regularly. The Exchange server has an anti-virus scanning package installed. The administrator account has a name other than administrator and the default account is set up to inform the network administrator through email if anyone tries to login on through that account. The SANS step-by-step guide was used during each server setup and appropriate recommendations were used during configuration. Passwords are changed every 100 days and utilize a strong password policy. Three logon attempts are allowed before locking an account. Event logs are reviewed each morning and a third party event log system is in place to notify the network administrator of defined events.

A separate server was purchased for purposes of testing system restorations. Ghost images are made of each server at the time of installation and a log is kept of any system or configuration changes that are made. There is also a small test network that contains at least one of each type of system in the event of any catastrophic event. Any server can be set up from scratch in a minimal amount of time.

Remote Access Services (RAS) is running for accessing email from home or for those traveling. Three modems are connected with only one local and one 800 number needed which will search for an unused modem. Email is the only accessible system or application through the modems. Web email is also available and given the numbers of IIS vulnerabilities is also a constant concern.

Vendors are a constant issue because patch levels and operating systems are kept at lower standards to accommodate their applications. Support for the applications are dropped by the vendor if patches are installed without their approval. Remote support also becomes an issue, as some vendors will require a modem be placed directly on the server or insist on telnet access to the system. These machines are a constant security concern and plans are underway to institute a VPN access system and policy.

4.1.3 Infrastructure

Switches are configured with strong passwords. They are located in controlled closets accessible only by approved personnel utilizing smart card locks. All wiring is in conduit running through ceilings and walls.

Access to the network is controlled in the student lab to only their subnet. The file server is isolated to that subnet, and the subnet itself is denied access to the other subnets by ACL's in the router.

4.1.4 Incident Handling

Since this is a college unit within the University, it utilizes the legal resources of the larger organization. Any contact with law enforcement or media is conducted through them. Communication policies were established between the unit and the central organizations legal, human resource, and public affairs departments. Internally, the incident handling team consists of the network administrator, two other IT staff members, a human resource representative, the administrative and finance manager, and a departmental manager. The team has the authority to take possession of any software or equipment necessary and to monitor all network activity.

A process for dealing with incidents has been developed and checklists were made. A ""jump bag" was put together and is kept in the server room. The process was designed to identify an incident, contain, eradicate the attack, and recover the system in the quickest manner possible while still maintaining efficiency and effectiveness in keeping records and evidence.

Reporting of incidents is encouraged and is reinforced during meetings, training, and publications. All support occurs through the help desk and each call is entered into database. Help desk personnel are trained to report unusual activity to the network administrator.

4.2 Identification

A call is received by the help desk from a faculty member stated that his system has crashed. Since the definition of "crashed" by all faculty and staff is different, the support person asked for further account of what exactly happened. The faculty member responded saying that he was writing an email to an associate and his machine suddenly had nothing but a blue screen with a bunch of letters and numbers on it. Then the computer rebooted and seems fine. The support person, whose other line began ringing, responded with "Well, don't worry about it. Windows does that sometimes."

Two days later the faculty member called again. "It happened again and this time I lost several pages of a document I was working on."

The support person answering the phone was not the same one that answered previously. "What happened again?" he asked.

"I was working on an important document when my machine just went to a blue screen and then rebooted."

"OK, I'll be there in a few minutes," he said.

The support person did a quick search in the problem database for entries about the faculty member. The last entry was from a couple of weeks ago when he had complained that his machine was running slow. They had found that he had downloaded a shareware application that would track his workout schedule and diet information. When the program was removed, performance was restored to normal. There was no mention of the blue screen incident that occurred a few days ealier.

The technician went to the faculty member's office, sat down at the machine and found that he had downloaded a different fitness application and several games were installed just recently. "You know our policy about shareware software and if it causes problems on your machine. All these problems started after you downloaded these programs so we'll need to take your machine downstairs and install your original image and restore your files."

The technician then checked to be sure proper backups were available and copied any files that had been worked on that day.

"I have work that I need to get done. How long is this going to take?" asked the faculty member.

"A couple of hours. I'll get it back to you as soon as I can," was the response.

The machine was ghosted and files were restored. The machine was delivered to the faculty member's office and checked to see that everything was in order. A warning about the evils of downloading untested applications ensued and the faculty member was back in business.

Monday afternoon and the help desk phone rang again. It happened again. The technician went to the machine and it was still in the same condition as when he left it. No other software had been loaded. This time the technician decided to take a look at the event logs and found the following informational log at the time the reboot occurred.

"The computer has rebooted from a bugcheck. The bugcheck was: 0x0000001e (0xc0000005, 0x804b818b, 0x00000001, 0x00760065). Microsoft Windows 2000 [v15.2195]. A dump was saved in: C:\WINNT\MEMORY.DMP."

The technician left to do research on the log entry. He loaded his copy of the technet CD and did a few searches. The general theme of the information in the technet articles seemed to relate to either a hardware issue or a driver issue. Since the drivers on all the machines were the same and no problems had occurred, it was most likely a hardware problem. The support department keeps loaner PC's for these situations and the technician setup the computer with everything the faculty member needed and swapped the system.

Wednesday afternoon came and the computer crashed again. The support technicians and network administrator had a meeting to discuss the problem and try to brainstorm a solution. Options were limited since new hardware and standard software that were known to be good were used and the problem continued to occur. Could it be coming from the network? Looking at the

support database information, it was noticed that the problem occurs at almost the same time every other day. Is there some kind of connection? They know that the computer had all the most current service packs, up to service pack 3. Also some of the other security patches had been applied as well. The next step was to take a look at the most recent security bulletins to see if there could be any connection.

The last security patches applied to the image or pushed out to systems was MS Bulletin MS02-024, concerning an authentication flaw that could lead to elevated privileges, and MS Bulletin MS02-31, a patch for Word and Excel macro issues. The network administrator had reviewed up to bulletin MS02-40, but hadn't had time to look at the most recent bulletins. After reviewing the bulletins, of which there were a few that needed to be applied, the only one that was related to the problem was bulletin MS02-45. It was decided to set up a sniffer to look at the traffic on that port and watch it Friday afternoon. The faculty member was going to be away that day so the support staff could monitor the system. Meanwhile information about the bulletin was looked at from resources such as CERT, CVE, Microsoft, and other related links.

Friday afternoon came and as predicted the system blue screened at approximately the same time. Sniffer traces at the time of the crash were looked at and a NetServerEnum2 request was made at that time. Looking at the trace, the packet had source IP address that was an internal address. The address was being used by one of the other faculty members in an office on the floor below.

The network administrator went down to the office and knocked but no one answered and the door was locked. He went down the hall to the faculty member's secretary who told him that the faculty member was out of town and had been for the last week or so, and would be back on Monday. When asked if anyone else could be using that office he was told that there was a graduate student that was doing some work for the faculty member, but she didn't know when he was or wasn't there. The network administrator decided to talk to the faculty member on Monday.

Monday arrived and the network administrator visited the faculty member. The graduate student had indeed been finishing some work for the faculty member and had access to his office. The student had also finished the work that needed to be done and had left for home that weekend, which happened to be in China. There was no evidence of the SMBdie tool of other tool that may have been used for the attack. The only thing that was noticed was that Norton anti-virus real time protection had been disabled.

4.3 Containment

In this incident, once identification of the exploit had been determined, the process of containment was limited. Comparison of sniffer traces between the SMBDie tool captured in a test environment and the ones that were captured during the attack, were essentially identical. It was noted that when the tool was

unzipped the Norton anti-virus would catch it and delete it, so the real-time protection needed to be disabled.

Conversations with the original faculty member reveled that he had had some altercations with the student in question earlier in the summer but wouldn't elaborate on the issue. The student had graduated at the end of the spring semester and had agreed to spend the summer working for the other faculty member through the summer session. The student also had morning seminars each Monday, Wednesday, and Friday, from which the assumption was made that he worked in the faculty member's office on those afternoons. The student had a job lined up and had to be back home around the beginning of the fall semester. The SMBDie tool was released on August 25th, which was around the time the first attacks took place. No legal action was taken.

4.4 Eradication

A ghost image was made of the machine the student was using and then was re-imaged with the original standard image to ensure that no other tools, trojans, or other malicious software were on the machine. The faculty member's machine that was attacked was given back his original machine that was also re-imaged and an image of the attacked machine was also made.

It was possible that the student had shared this tool with other students. Since the patch was capable of ensuring that this vulnerability could not be exploited, it wasn't an issue.

4.5 Recovery

The Microsoft patch for this, as well as ones that had not been applied yet, was pushed out to all machines. Discussions were held to determine if any of the other workaround recommendations were in order, in the event other exploits were developed that could take advantage of the configuration of the standard systems. Since file sharing was being used extensively, the workarounds were not an immediate option. Active directory implementation was being planned and with it the elimination of NetBIOS which would close the problem ports. A process of identifying those who were using file sharing and utilizing the servers for that purpose, was begun.

Several tests were run to ensure that the patch indeed was able to stop the exploit. On Windows 2000 systems, the patch replaces this file:

Date	Time	Version	Size	Path and File name
23-Jul-2002	20:28	5.0.2195.5971	92,432	%WINDIR%\System32\Xactsrv.dll

Because of file dependencies, this update requires Windows 2000 Service Pack 2 (SP2) or Service Pack 3. It also requires the system to be rebooted.

When running the SMBdie tool the SMB session is refused after applying the patch. Microsoft has fixed the code so proper bounds checking is done.

4.6 Lessons Learned

Several mistakes were made during the process of this attack. They began with the first call to the help desk. A blue screen machine should always be checked starting with the event logs. Many times the particular file that is causing the problem will be reveled. Entries in the support database should be entered for all calls. If these procedures were followed, the attacker would probably have been caught.

The second support person made the same mistake by assuming it was bad shareware that was the problem and not checking the logs. Following their own policy to re-image the machine before following this procedure, deleted valuable information about the attack. This could have been vital information if legal action had been taken. Procedures for re-imaging a machine in this type of event should include making an image of the machine first. The cost of a couple of CD's and the time it takes is minimal.

In the preparation section it was stated that help desk personnel were trained to report unusual activity to the network administrator. In this incident, that training did not go far enough. In the future, all blue screen occurrences should get reported. The question that should be asked is "When did this become an Incident?" The definition of and event, as stated in the SANS Incident Handling materials, is any observable occurrence in a system and/or network and an incident is an adverse event referring to harm or the intent of harm. With that definition in mind, this became an incident when the faculty member lost his work, and should have been treated as such from that point on.

The network administrator had waited too long before assessing and testing current security bulletins. A more efficient system of collecting, reviewing, testing, and applying patches and service packs is a common issue and needs to become a priority among support personnel.

The connection between help desk staff, network administrators and security personnel cannot be over emphasized. A two-way communication channel that contains trust, education, and information needs to be established. My observation in several environments is that there are often huge gaps in the communication between these entities and this one was no exception.

While actual monetary loss could be considered minimal by this attack, there was a lot of frustration and time lost on the part of the faculty member that was attacked, as well as the support personnel. Because of these very simple procedural errors, far greater damage could have occurred.

References

SMB Protocol References

What is SMB? – Richard Sharpe
http://samba.anu.edu.au/cifs/docs/what-is-smb.html

An Introduction to SMB/CIFS – From “Using Samba”, O’Reilly
http://www.oreilly.com/catalog/samba/chapter/book/ch03_03.html

CIFS: Common Insecurities Fail Scrutiny – Hobbit
http://www.insecure.org/stf/cifs.txt

NetBios, NetBEUI, NBF, SMB, CIFS Networking – Timothy Evans
http://ourworld.compuserve.com/homepages/timothydevans/contents.htm

Buffer Overflow Reverences

Smashing The Stack For Fun And Profit - Aleph One
http://www.phrack.org/phrack/49/P49-14

Win32 Buffer Overflows – Barnaby Jack
http://www.phrack.org/phrack/55/P55-15

The Tao of Buffer Overflow – Dildog
http://www.cultdeadcow.com/cDc_files/cDc-351/

Cerberus Information Security – White Papers
http://www.cerberus-infosec.co.uk/papers.shtml