

Browsing an NT Domain Through a VPN: Preparing Your Network

Version 1.1

NOTE: This document is obsolete as of CP2000. Please consult the Secure Domain Logon white-papers for configuration information.

Author: Ron Naken
Date: June 28, 1999
Purpose: To describe a network configuration that will allow VPN clients to browse and authenticate on a remote NT domain.

© 1999 CheckPoint Software Technologies LTD.

This document is written under the assumption that the firewall and SecuRemote client have already been installed and configured and that connectivity from the SecuRemote client to the remote network has already been confirmed using a tool such as PING.

Note: Throughout this document, the term “Windows 98” is used to describe both Windows 95 & Windows 98. The version of Windows NT covered in this document is 4.0.

We will start the discussion with a brief background in network browsing and follow with configuration. This document will focus on what must be done on your network to allow remote browsing and authentication on an NT domain. Other documents will be created to illustrate configuration of the Windows 98 and Windows NT clients.

For further reference, please see the following documents:

Browsing an NT Domain Through a VPN: The Windows 98 Client Using a NIC
Browsing an NT Domain Through a VPN: The Windows 98 Client Using Dial-Up
Browsing an NT Domain Through a VPN: The Windows NT Client Using a NIC
Browsing an NT Domain Through a VPN: The Windows NT Client Using Dial-Up

FAQ: NT Domain Authentication and Browsing Across “Hidden” Sites

Network Browsing: A Look Under The Network Neighbor’s Hood

One of the most challenging aspects of implementing Windows NT domains in the enterprise is understanding the underlying NetBIOS architecture and how it is effected by a WAN. Before we begin our configuration, we need to have a minimal understanding of Microsoft network browsing and terminology.

- Domain Master Browser – This machine is responsible for building and maintaining a list of resources for an entire domain. It compiles resource lists provided by Subnet Master Browsers, as well as resource lists from Master Browsers for other domains & workgroups. The Domain Master Browser is always the Primary Domain Controller (PDC) so there will be only one per domain.

Note: The term “resource” is being used here to describe a server with shared resources, as opposed to describing the actual share, itself.

- Subnet Master Browser – These machines are responsible for building and maintaining a list of resources for their own subnet. They feed their list to the Domain Master Browser and receive from the Domain Master Browser a list of resources for the entire network. There will be only one per subnet per domain.
- Backup Browser – These machines periodically receive resource lists from their Subnet Master Browser and act as a backup. There may be many Backup Browsers.

When a user first enters Network Neighborhood after startup, that user’s machine attempts to contact a Master Browser to locate a resource list to download. The process of locating this Master Browser is normally done through broadcasts. If the Master Browser cannot be located, the client cannot locate a list of resources for the domain, so

the user will see the following message in Network Neighborhood: "Unable to browse the network".

Another problem with domains on an internetwork is that name resolution for NetBIOS names is also done through broadcasts by default. This means that if we have a router sitting between a Microsoft client and a server, the client will not be able to locate that server without additional configuration. NetBIOS names are those names we assign to our network-enabled Microsoft machines during installation. These names are used by our NetBIOS-based services to reference machines, and are resolved to IP addresses when using NetBIOS over TCP/IP (NBT).

To solve the problem of locating Master Browsers and resolving NetBIOS names of hosts that are on a separate subnet from the client, we need to introduce WINS (Windows Internet Name Service) and LMHOSTS. WINS is a service similar to DNS (Domain Name Service) that can be used to maintain a dynamic database of NetBIOS names to IP address mappings – typically, the best solution in an enterprise for NetBIOS name registration and resolution. The LMHOSTS file, on the other hand, is a text file that can be configured with static entries. Let's take a quick look at WINS before we go into LMHOSTS. It will become very clear that WINS is a much easier solution to setup and maintain.

WINS is a standard service that can be installed on NT Server once TCP/IP has been installed. Once the WINS service is installed, all an administrator needs to do is configure the network servers and clients with their primary and secondary WINS (if there is a backup WINS) and the WINS service will take care of the rest. Hosts that are configured to use a WINS server are known as WINS clients. As WINS clients initialize during startup, they will register with the WINS server to provide the WINS with an IP address to NetBIOS name mapping for that client. When domain controllers initialize during startup, they will register their domain name with the WINS and register themselves as a domain controller for that domain. When a WINS client needs to resolve a NetBIOS name, or when a WINS client is looking for a domain controller or Master Browser, it will contact the WINS server to find its answer.

The LMHOSTS file is another solution that can be used to provide NetBIOS to IP address mappings. It is located in the \WINDOWS directory on Windows 98, and under \WINNT\SYSTEM32\DRIVERS\ETC on Windows NT. LMHOSTS is a text file that is structured similar to a standard HOSTS file; however, there are directives we can use to specify what type of mapping we are creating. Here is an example:

LMHOSTS:

```
1.2.3.4      ARTHUR      #PRE #DOM:CAMELOT
1.2.3.5      LAUNCELOT  #PRE #DOM:CAMELOT
```

The above sample entries make use of two tags in the LMHOSTS file:

#PRE

Pre-loads the entry into the NetBIOS name cache when the name cache is initialized. This occurs during startup and can be forced at the command prompt:

nbtstat -R

#DOM:xxxxxx

This registers the entry as a domain controller for the domain specified.

So in our above examples, we have one entry that reads: “There is a host whose IP address is 1.2.3.4 and whose NetBIOS name is ARTHUR. This entry will be pre-loaded into the NetBIOS name cache. This host is also a domain controller for a domain named CAMELOT.”

The next entry reads: “There is a host whose IP address is 1.2.3.5 and whose NetBIOS name is LAUNCELOT. This entry will be pre-loaded into the NetBIOS name cache. This host is also a domain controller for a domain named CAMELOT.”

It is important to point out that entries in the LMHOSTS file which are #PRE will not be processed by a host during normal use of the LMHOSTS file unless they have already been loaded into the NetBIOS name cache. If you add #PRE entries to the LMHOSTS file, make sure you type “**nbtstat -R**” to reinitialize the name cache.

Another important point about LMHOSTS is that it is only read during name resolution. This means that as a user logs on and his host needs to locate a domain controller for authentication, or as a user opens Network Neighborhood to browse, the LMHOSTS file will not be read to get a list of servers to contact. The entries will need to be both #PRE and #DOM:xxx and be pre-loaded into the name cache for these services to work properly.

Though LMHOSTS is not an ideal solution since it is a text file that must be maintained on each host that needs to locate resources across an internetwork, it can be somewhat centrally located and managed. The #INCLUDE directive can be used to include a network based LMHOSTS file that is shared world-readable. And the #BEGIN_ALTERNATE and #END_ALTERNATE directives can be used to provide redundant copies. Here is an example:

```
LMHOSTS:  
1.2.3.4      ARTHUR      #PRE #DOM:CAMELOT  
1.2.3.5      LAUNCELOT  #PRE #DOM:CAMELOT  
#BEGIN_ALTERNATE  
#INCLUDE \\ARTHUR\PUBLIC\LMHOSTS  
#INCLUDE \\LAUNCELOT\PUBLIC\LMHOSTS  
#END_ALTERNATE
```

Any #INCLUDE tags that appear between #BEGIN_ALTERNATE and #END_ALTERNATE will be considered duplicate include files. This will allow us to keep redundant copies of one file. When the LMHOSTS file is read from top-to-bottom during name resolution or name cache initialization, if the first alternate #INCLUDE is found, the second will not be processed.

Note: It is important to point out that during name resolution, #PRE entries are skipped – they are assumed to have been preloaded into the NetBIOS name cache.

Configuring Your Network to Support Remote VPN Browsing & Authentication

Before we begin, we need to address the issue of network address translation (NAT) with NetBIOS, since it may cause complications. If your SecuRemote clients will be accessing a remote network that is hidden using NAT, you will need to enable encapsulation on the VPN to work around the NetBIOS NAT issue and/or to allow routing to internal non-routable addresses. This (“Tunnel Mode”) will be enabled by default if you are using IKE; however, if you will be using FWZ, you will need to enable “Encapsulate SecuRemote Connections” on the FWZ properties of your firewall object in the security policy.

Understand that this particular document only describes what is necessary from the company’s network to allow clients to remotely browse and authenticate on an NT domain. Other documents will follow that describe the client configuration for both Windows 98 and Windows NT as a SecuRemote client to remotely access an NT domain through a VPN.

What we will accomplish in the following steps is to setup a WINS server that will allow our remote clients to resolve our internal NetBIOS names from remote, as well as locate domain controllers and master browsers.

1. Configure the SecuRemote VPN.
2. Verify network connectivity from the SecuRemote client to the remote network through the VPN. Check the Firewall-1 Log Viewer to ensure the communications were decrypted/encrypted by the firewall.

Note: In the example below, we verified our connection using telnet. We can confirm the VPN is working by the decrypt action for user “joe” which was the username we entered when SecuRemote prompted us to authenticate for the VPN.

..	Date	Time	Ori..	Type	Action	Service	Source	Destinati..	Proto.	Rule	S_Port	User
0	29Oct...	15:50:23	firew	control	ctl							
1	29Oct...	15:50:28	firew	log	decrypt	telnet-23	joe-pc	mail-server	tcp	1	1153	joe

3. Make sure your SecuRemote “Client Encrypt” rule allows the “NBT” group of services or “Any” for access to the remote servers and domain controllers. This is shown below.

Note: The NBT service group will include NetBIOS name resolution, file sessions, and domain authentication.

No.	Source	Destination	Service	Action	Track	Install On	Time
1	Sales@Any	Encrypt_Domain	NBT	Client Encrypt		Gateways	business_hc

4. Create a WINS server (or use LMHOSTS). See “Installing WINS on NT Server” below. If you are using NAT or non-routable addresses to hide your internal network, you will need to add a static NAT entry for your WINS server to give it a publicly accessible address.

Note: Remote VPN clients will need to access this WINS server as their NetBIOS based services are initializing. If they cannot contact their WINS server at that time, they will either not be able to browse the network, or they will experience delays in populating their browse list depending on their configuration. With the configurations we will discuss, NetBIOS based services will initialize before the VPN is established. This is why at least one WINS needs to be publicly accessible.

5. Create a rule on the firewall to allow remote clients to contact the WINS without requiring a VPN. The rule should allow only the “nbname” service to ensure only NetBIOS name resolution requests can be sent to the WINS. The rule will look as follows:

No.	Source	Destination	Service	Action	Traffic
1	Any	WINS	nbname	accept	

6. Configure your servers and domain controllers to be clients to the WINS server. See “Configuring a WINS client on NT” below.

Note: You may wish to configure only the servers your remote clients will access as WINS clients. You must configure a domain controller as a WINS client so remote hosts will be able to locate them for authentication and browsing. It is a good idea to include the PDC, since account management (i.e. adding a machine account to the domain) is always done on the PDC and later replicated to the BDC’s. Failure to include the PDC would prevent remote hosts from joining the domain, among other things.

7. Configure the SecuRemote clients as shown in the other documents referenced above in the section “For Further Reference...”

Installing WINS on NT Server

Installing a WINS server on NT is deceptively simple. Keep in mind that the WINS service on NT will not know of itself, so you might consider setting the WINS server up as a WINS client to itself after you get it installed (see below). This document will not discuss replication between WINS servers.

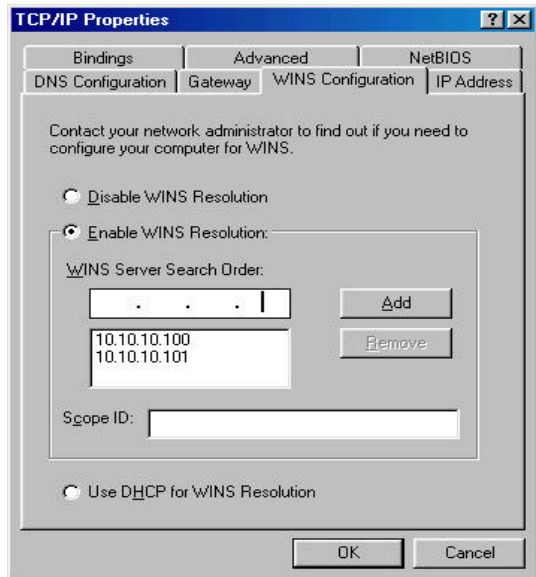
1. Go to Control Panel->Network. Select the “services” tab and add a service. Select WINS and click OK.
2. Once the WINS service is installed, you will need to reboot the machine.

Configuring a WINS Client on NT

1. Go to Control Panel->Network. Select the “Protocols” tab and edit properties of “TCP/IP Protocol”.
2. Select the “WINS Address” tab and enter the IP address of the WINS server in the “Primary WINS Server” field.
3. Click OK.

Configuring a WINS Client on Windows 98

- 1) Go to Control Panel->Network. Select properties of the "TCP/IP" that is bound to the network adapter card you will be using. Dial-up Networking is covered in another document.
- 2) Select the "WINS Configuration" tab and enter the IP address of the WINS server in the "WINS Server Search Order" field. See below.



- 3) Click OK.
- 4) Remember to restart the machine after configuring it as a WINS client.