Contents

Active Directory (AD) and Network Address Translation (NAT)

This guide includes the following sections:

- Overview
- Introducing the Technologies
- Network Communication Requirements
- NAT Scenarios
- Summary

Overview

When companies merge or one company acquires another company, one of the goals is to allow employees and applications to share data and communicate between the two companies. There may be many distributed applications that need to be enabled across the trust, but often Active Directory is the first common set of services that is being looked at closely as part of a merger as it provides core security services as users access resources in remote forests.

The common recommendations around running a network with Internet Protocol Version 4 (IPv4) addressing would often lead to IP address conflicts. The preferred approach is to always resolve the IP address conflicts. It is strongly suggested that re-addressing be your first choice in this situation before considering Network Address Translation (NAT) as a temporary solution. This allows for running with standard IP routing and standard operations for network management and name resolution. However, there might be situations where re-addressing the IP networks may not be a possible immediate solution. For example, changes to static IP addresses might impact hard dependencies for applications or user access and communications.

The intent of this document is to present some of the considerations you should take into account and highlight some of the problems and challenges of implementing NAT in an Active Directory infrastructure for the scenario described above. It does not attempt to answer all questions, as most answers will depend on the design of the deployment and the products used.

Active Directory provides services such as authentication and authorization to existing identities in the organization. These are considered infrastructure services that do not exist for their own purpose but for the purpose of supporting users and applications using these identities. Applications, and the protocols they use, also need to work across the NATs you implement. The work you invest in connecting the network and enabling Active Directory can be of no value if a required application cannot work across NAT. In addition, there could be considerations for using other networking services such as IPsec with NAT. For more information, see Gathering Information about Your Current Network Infrastructure (http://go.microsoft.com/fwlink/?LinkId=180566) in the Windows Firewall with Advanced Security Design Guide.

Important

Make sure you assess the risks of enabling applications and services across a NAT before proceeding with an implementation.

Microsoft understands that communication between domain controllers, and between clients and domain controllers, over a NAT is an issue that customers often run into in merger or acquisition scenarios. Although no evidence indicates that NAT cross-forest configuration would inherently break communication between domain controllers or between clients and domain controllers, this scenario is not tested by product groups for Active Directory or other technologies that utilize Active Directory such as Kerberos or DFS. Thus, the Active Directory Product Group *generally* does not recommend customers to configure domain controllers to communicate over a NAT. If a customer decides to configure domain controllers to communicate over a NAT, *the customer should perform extensive testing*.

We understand that for some customers re-addressing their computers is a time-consuming process and the urgent business requirements to enable domain controllers to communicate over a NAT. Therefore, Microsoft will consider issues that our enterprise customers face in NAT scenarios on a case-by-case basis to determine whether a software update is possible or appropriate. The likelihood that a software update is implemented for an untested configuration (such as a NAT deployment) is lower than for fully tested scenarios.

Introducing the technologies

To set the foundation for the discussion and to establish the terminology used in this topic, this section covers the role of the technologies in the context of this scenario.

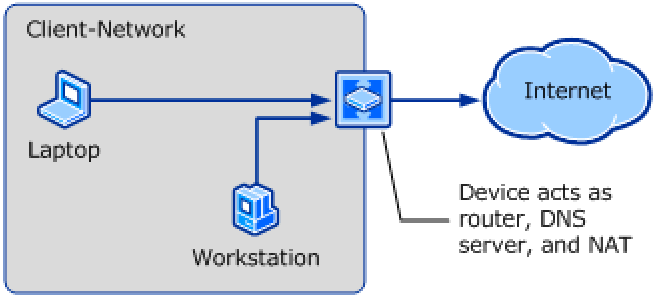
Network Address Translation (NAT)

Network address translation (NAT) is an Internet Engineering Task Force (IETF) standard used to allow multiple computers on a private network (using private address ranges such as 10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16) to share a single, globally routable IPv4 address. NATs are often deployed because public IPv4 addresses are becoming scarce. NAT can be an immediate but temporary solution to the IPv4 address exhaustion problem that will eventually be rendered unnecessary with IPv6 deployment.

The main purpose of NAT is to interconnect networks to allow applications, users and computers (clients) in one network to access resources in other networks. The NAT hides the network configuration of these clients from the other networks, so an independent addressing scheme can be used.

In general terms a NAT implementation defines a private side where clients are located and a public side with resources these clients connect to. A NAT router, which can also be a name resolution server (such as a DNS server) interconnects the private side (for example, the internal network) with the public side (for example, the Internet).

The following figure shows classical use of a NAT to connect clients to a big network.



When services that are running on the private side of the NAT need to be accessed from the public side, the NAT proxies the requests sent to one of its public IP addresses to a server on the private side. This feature is often called reverse publishing. For reverse publishing, the port stays the same most of the time, but each server would have a dedicated public IP address. In this case the clients on the public side of the NAT use public (published) IP addresses corresponding to servers in the private side. The NAT router uses the public IP addresses and TCP/UDP ports and forwards the network packets to the corresponding internal server.

Depending on the number of services you want to publish, the NAT setup and maintenance process can become more and more complex. For example, publishing a large number of IP addresses and names of services and maintaining the addresses becomes more complex.

Active Directory

Active Directory is a set of services that work together to provide directory lookup, authentication and authorization, security and other policies to the users and computers that are members of AD domains and forests.

These services are implemented as standard network servers, mostly Winsock-based servers and clients and sometimes using higher-level interfaces like Remote Procedure Call (RPC) or Server Message Block (SMB) communication. The implemented protocols are based on Internet Protocol (IP): Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) primarily (Internet Control Message Protocol (ICMP) is used for some diagnostic operations).

Active Directory typically uses Domain Name System (DNS) to resolve names of domains and servers. NETBIOS-based name resolution (WINS) can also be used for this purpose. For the case of a NAT implementation, a DNS server or a WINS server containing information about the services published can be deployed and made available to clients accessing these services.

When implementing NAT, you may want to consider modifying the Active Directory subnet and site topology, and the replication topology to meet your specific requirements. For a general discussion about Active Directory topology design, see Designing the Site Topology for Windows Server 2008 AD DS (http://go.microsoft.com/fwlink/?LinkId=179694).

Active Directory subnet and site topology

The following are some considerations you may have in mind in respect to the Active Directory subnet and site topology in a NAT environment.

For the typical case in which clients in the private side of the NAT access public resources:

1. Clients will connect to public resources using one unique IP public address, facilitated by the NAT. This means that site association for clients in the private side will need to be done using this IP address.
2. Clients can use name resolution servers on the public side of the NAT to discover public domain controllers.
3. If you want to designate a specific set of public domain controllers to serve clients on the private side of the NAT, you can create an Active Directory site for these domain controllers and create an Active Directory subnet with the public IP address of the NAT router. The clients will then be mapped to this subnet and site.

If clients in the public side need to access services in the private side:

1. Corresponding public IP addresses for these private resources need to be configured and published.
2. Clients in the public side will need to be able to resolve the names of resources hosting private services. This can be accomplished by allowing public clients to access a DNS or WINS name resolution server that resolves resource names to their corresponding published IP addresses.
3. For domain controller discovery in particular, the appropriate records need to be registered in the name resolution server. For DNS this means that the server should contain the appropriate SRV records for the discoverable domain controllers. This is accomplished by maintaining computer A records with the domain controller names pointing to the published domain controller IP addresses.
4. If you want to designate a specific set of domain controllers to serve clients in the other side of the NAT, you can create an Active Directory site for these domain controllers and create Active Directory subnets assigned to this site based on the public IP addresses of the clients. In this case, location of domain controllers will primarily be made using site SRV records. If no Active Directory subnets are created or there are clients IP addresses not mapped to an Active Directory subnet, domain controller discovery will use site-less SRV records. In this case, you can limit the number of domain controllers serving the clients by registering SRV records for only those domain controllers you want. For more information, see article 306602 in the Microsoft Knowledge Base (http://go.microsoft.com/fwlink/?LinkId=183352).

Active Directory replication topology

If you have domain controllers of a forest on both sides of the NAT, you will need to consider providing for Active Directory replication requirements. The key considerations are mainly related to DNS name resolution for both sides of the NAT, for every NAT you need to traverse. This paper discusses this problem in detail in section NAT Scenarios.

Network communication requirements

The Active Directory Network Operating System (NOS) implements several protocols, with services offered by domain controllers and clients that use them. Active Directory also makes use of protocols supported by other services that are hosted either on domain controllers or on separate servers. Some protocols are solely driven by clients or intermediate servers while others implement forwarding requests between domain controllers of the same or between domain controllers of different domains and forests.

For the purpose of this discussion we assume that many of the protocols used by the operating system are only needed within the local forest. We will go over the protocols that need special treatment or that are commonly seen as problematic.

In environments involving multiple organizations that merge their networks, the traffic is often filtered by ports. You may need to develop new rules for the inter-organization filtering. For a list of services in Windows and the port requirements they have, see article 832017 in the Microsoft Knowledge Base (http://go.microsoft.com/fwlink/?LinkId=179695).

Microsoft currently does not have holistic documentation about the machines that need to be contacted for a certain service or task. Some services document this as part of technical reference documentation, but customers need to figure out what it means to them when they connect the networks of two organizations. This topic provides links for a few crucial protocols.

Kerberos

This is the preferred authentication protocol of Active Directory. The requests are driven by clients and intermediate servers that implement a delegation scheme to other back-end services. The classical example is an application running on a web server that connects to a back-end database by using the user identity.

Clients locate Key Distribution Centers (KDCs) using service records, and they use the DC Locator to identify which domain controller (DC) to use.

A Kerberos transaction typically starts with a request to a KDC of the domain the user is defined in. Given the server identification, the KDC either hands out the ticket to the service, or returns a ticket to another KDC in a domain closer to the domain of the resource. These referral tickets are following the trust path within the forest and also across forests if there is a forest trust.

The client or intermediate server has the requirement that it needs to be able to locate a domain controller of every domain along the trust path between the user domain and the domain of the resource. Once the client has acquired the ticket it needs, it will send the ticket to the server along with application protocol data during a bind, session setup, connect or logon transaction.

In some documentation of Kerberos found on the Internet, the impression is raised that the client IP address must be included in ticket requests and it must be checked on the KDC against the source IP address seen in the network packet. However, this check is optional according to the Kerberos RFC 1510, although a number of other Kerberos implementations make this check by default. For more information, see Kerberos Processes and Interactions (http://go.microsoft.com/fwlink/?LinkId=179701).

Windows Kerberos does not include or check the client IP address by default. It has settings that allow this to be done for non-Windows Kerberos trusts, but this is also disabled by default. For more information, see article 318071 (http://go.microsoft.com/fwlink/?LinkId=179702) in the Microsoft Knowledge Base.

If you have enabled the IP address check in your configuration and you expect clients from behind a NAT to connect to your domain controller, you need to disable this check.

For more information about how Kerberos is used in trust referral processing, see the section titled Kerberos V5 Referral Processing (http://go.microsoft.com/fwlink/?LinkId=179714).

NTLM

NTLM was the main authentication protocol used until Windows 2000 introduced Kerberos. It is still used in many scenarios and is the main authentication protocol on external trusts between domains. NTLM can also be used in forest trust scenarios. Some applications still use or prefer NTLM.

In NTLM, the clients send user credentials to the server as part of the protocol data stream (for example, a Web server). The server will then check to see if the account passed is local to its Security Accounts Database (SAM), if not, it hands off the request to the domain controller of its primary domain, which it has the secure channel with. The domain controller also checks whether the account is local to its account database, if not, it will pass the credentials to a domain controller in the appropriate trusted domain through its secure channel.

Note that with transitive trusts, if the trusted domain is not a directly trusted, the domain controller will forward the credentials to a domain controller in the next domain of the trust path. The next domain controller in the authentication chain is found using the DC Locator.

So in NTLM, the communication requirement is that all domain controllers need to find domain controllers in the next hop along the trust path, they need to be able to find a domain controller for all directly trusted domains. You have to consider each hop following the NTLM trust path as a new connection, and know that it also may need to traverse a NAT.

The communication is facilitated using RPC calls of the Netlogon service, which shares its server TCP port with the other Active Directory RPC servers, or is using Named Pipes over SMB.

For more information about how NTLM is used in trust referral processing, see the section titled NTLM Referral Processing (http://go.microsoft.com/fwlink/?LinkId=179714).

Lightweight Directory Access Protocol (LDAP)

Applications are often querying the AD LDAP interface to get information. They do not always use the computer or user domain LDAP, but also the port 389 (SSL/TLS on 636) LDAP services of remote domains. Finding out about this can be difficult, as often applications do not document this or allow users to change the query location.

Applications can also use the Global Catalog (GC) service on port 3268 (SSL/TLS on 3269) of the local or remote forest. In Windows, the Object Picker can do this to find security principals to grant access to or to add to groups in a local resource or group.

The LDAP clients in the Windows operating system locate standard LDAP or GCs using service records, and they use the DC Locator to identify which domain controller to use. User applications that use LDAP may use a different way of identifying domain controllers, some may have them in a configuration store or even hard-coded into the application.

When you have identified the business requirement that an application needs to be used across the NAT, you also need to ensure the clients are able to resolve the target domain and server names and to initiate the LDAP or LDAP-GC session.

For its own purposes, a domain client only uses LDAP queries with the domains the computer and user logging on to the computer is member of for Group Policy. By definition, the Group Policy can only come from the same forest that the security principal is defined in.

For more information about implementation details on LDAP in Windows, see How Active Directory Searches Works (http://go.microsoft.com/fwlink/?LinkID=179715).

Server Message Block (SMB)

Strictly speaking, SMB (also referred to as Common Internet File System (CIFS)) is a resource server protocol and is beyond the scope of this guide. However, SMB is used by components of the Windows NOS.

When the user is logging on from a different forest across the NAT, it may also pull settings from domain controllers of that forest, including accessing other user environment items like roaming user profiles or home directories. For SYSVOL access, the client is using Distributed File System (DFS) to identify the server to talk to. DFS makes use of the Active Directory site topology in order to provide the user with the closest referral.

Note that there is a Group Policy setting named **Allow Cross-Forest User Policy and Roaming User Profiles** to prevent policy and roaming profiles to be applied across forests. The setting is located at:

Computer Configuration | Policies | Administrative Templates | System | Group Policy

Configure the setting as **Disabled** if you want to prevent this kind of access across the trust.

The Redirector may introduce a protocol dependency on ICMP, because in a number of scenarios where it resolves server names in DNS, it will ping the server to verify the address. For more information, see article 832017 in the Microsoft Knowledge Base (http://go.microsoft.com/fwlink/?LinkId=179695).

DNS

DNS is a central service to Active Directory, but it also helps to deliver the service when the DNS zones are stored in and replicated by Active Directory. In the more complex NAT scenarios, the accuracy and health of the DNS infrastructure is crucial for a NAT configuration to succeed.

Some NAT vendors offer translating the DNS server response payload coming from the private side of a NAT. It swaps all IP addresses of servers that are reverse-published to the public addresses. This is usually described as DNS traffic editing. This feature can help you avoid a managing a copy of DNS zones in the public network in many scenarios. Of course, this requires that the DNS traffic is not secured by IPSEC or DNS itself, so a payload signature still computes. It also requires that the DNS zone is stored only on the private side of the NAT, which is only the case in the simpler NAT deployments.

When you need to reverse-publish services, you need to determine which records are in the scope for that. For example, you might want to run the domain controllers you publish without any site-based records, or do the opposite and define a “NAT site” where you put all these domain controllers. For more information about managing the records registered by domain controllers using the DnsAvoidRegisterRecords registry setting, see article 267855 (http://support.microsoft.com/kb/267855) in the Microsoft Knowledge Base.

For more information about the communication architecture of DNS, see DNS Processes and Interactions (http://go.microsoft.com/fwlink/?LinkId=179717).

RPC

Active Directory uses various RPC interfaces (Netlogon, Local Security Authority/Policy management, Security Account Manager, and Active Directory Replication has one API called by clients). All these RPC interfaces by default use the same dynamic TCP server port.

You can specify the port to be used in the registry. For more information, see article 224196 (http://go.microsoft.com/fwlink/?LinkId=179718) in the Microsoft Knowledge Base.

Basically, RPC is yet another TCP-based set of protocols where the server is identified using the DC Locator process most of the time.

The RPC protocol itself does not introduce any new issues in NAT deployments. You may encounter issues though, with a given application’s RPC provider if they are encapsulating a client or server IP address in the RPC payload. If you are introducing any application that utilizes RPC in such an environment, testing should be done to verify proper functionality across the NAT connection.

We have reviewed the RPC interfaces used by Active Directory and there are no NAT dependencies. We did not specifically test all RPC interfaces offered by Windows, other Microsoft, or third-party products. The following list provides Active Directory RPC protocol specifications:

- NetLogon (http://go.microsoft.com/fwlink/?LinkId=179719)
- Local Security AuthorityLocal Security Authority (http://go.microsoft.com/fwlink/?LinkId=179721)
- Security Account Manager (http://go.microsoft.com/fwlink/?LinkId=179722)
- Active Directory Replication (http://go.microsoft.com/fwlink/?LinkId=179723)

Application Protocols

You need to make sure that the protocols of any application you want to use across NAT are capable of doing so. Even if the protocol is compatible with NATs, the software vendor may place and document restrictions on the placement of the application servers that may mean additional configuration or facilities to be installed.

NAT Scenarios

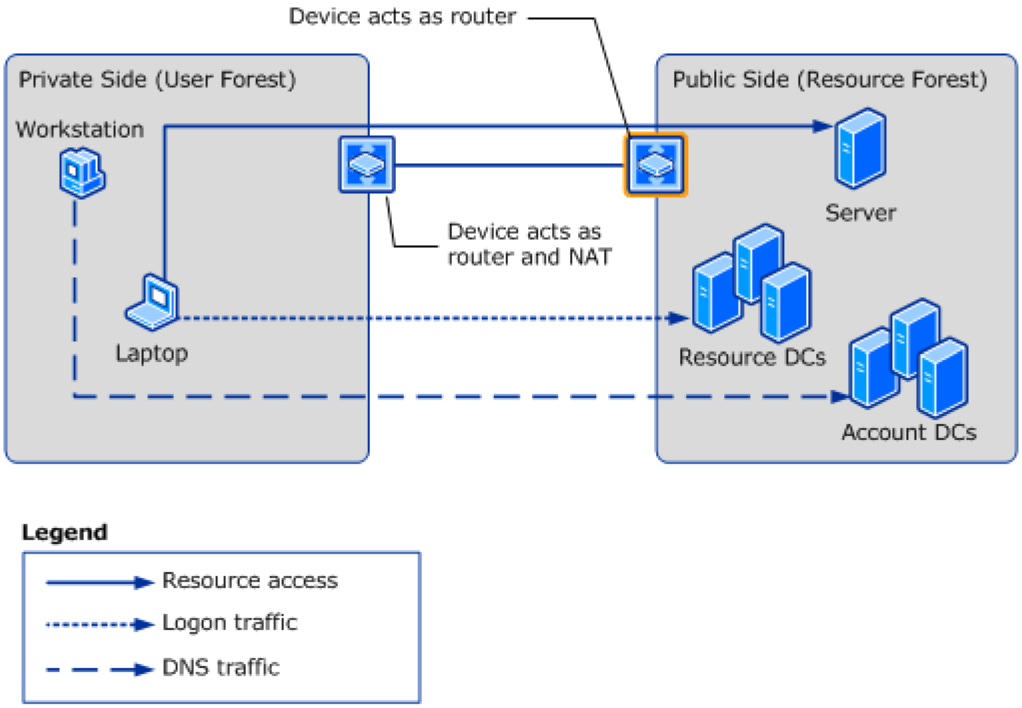
This section covers the following NAT scenarios:

- Clients behind NAT
- NAT between two Active Directory forests
- NAT between domain controllers of the same domain or forest

Clients behind NAT

The simplest scenario with NATs in the picture is when only clients are on the private side of a NAT, and all resources are on the public side of the NAT. This would be the case when you can move all servers of one company of the merger to the other company's network, and only leave clients and DHCP servers in the private network.

This scenario would also apply in a single company, when you have a complex network where you must use NAT to connect clients because they are unmanaged networks, such as independent agencies that nonetheless require access to your network.



There would be no resources published from the private side, and there are also no DNS servers in the private network. This scenario requires no manual configuration, as all server IP addresses used are public addresses, the NAT forwards the DNS queries to the actual DNS servers for the domains.

The scenario is very similar to the Internet access scenario described earlier.

NAT between two Active Directory forests

This section describes approaches you should consider when connecting two Active Directory forests using NAT. Be aware that the complexity of the deployment and operational procedures to maintain the infrastructure increases as more dependencies and service requirements between the two forests are added.

Two main approaches can be considered depending on the publishing requirements of services:

1. If domain controllers of only one forest need to be published to clients of the other forest, a regular NAT implementation approach can be followed. In this case, the private side of the NAT will contain the clients accessing domain controllers in the public side of the NAT. A one way forest trust between the two forests should exist for allowing access. The forest in the private side will be the trusted forest.

   Note

   For trust set up and secure channel maintenance you might need to reverse publish a domain controller (trusted domain controller) in the private side of the NAT, depending on where and how you create the trust. For more information about creating an incoming or outgoing trust, see Appendix: New Trust Wizard Pages (http://go.microsoft.com/fwlink/?LinkId=179724). For secure channel maintenance, the trusted domain controller needs to be accessible by the public domain controller. In this case, a reverse publishing of the trusted domain controller IP address will be needed.

2. If domain controllers in both forests need to be published for clients in the opposite forest, variations of the typical NAT scenario implementation might be followed. In this case, you may need to consider reverse publishing of services.

Domain controllers in one forest are published only (One NAT)

In this scenario, one forest is located on the private side of the NAT (private forest), and the other on the public side of the NAT. In this case, only resources in the public forest are accessible by users and computers in the private forest and not vice-versa. All domain controllers and servers of the forest on the public side of the NAT must use conflict-free IP addresses, because their IP addresses are not transformed as the network traffic traverses the NAT. Clients on the public side that do not need to be contacted by private clients do not need to have conflict-free addresses.

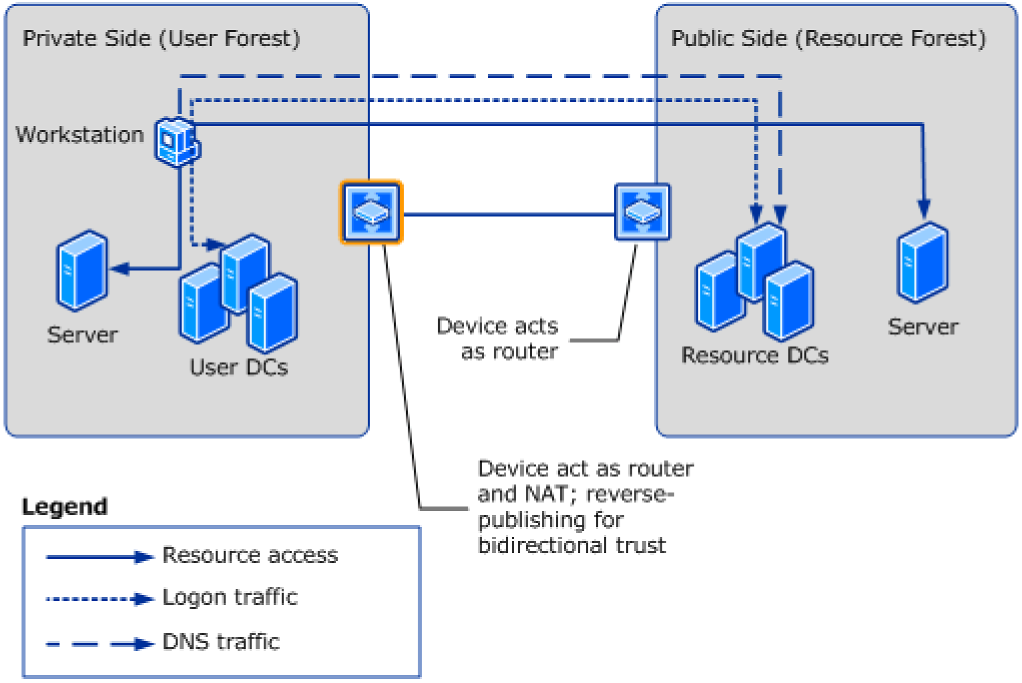
A forest trust, in which the private forest is the trusted forest, must exist in order to allow users and computers in the private forest to access resources in the public forest. If all communication is initiated from the private forest, you do not need to reverse-publish domain controllers in this forest.

You need reverse-publishing for some of the domain controller addresses on the private forest to enable NTLM authentication or allow for trust password management (see the Note in the previous section).

The clients in the private forest will discover domain controllers of the second forest using the name resolution servers in the public forest.

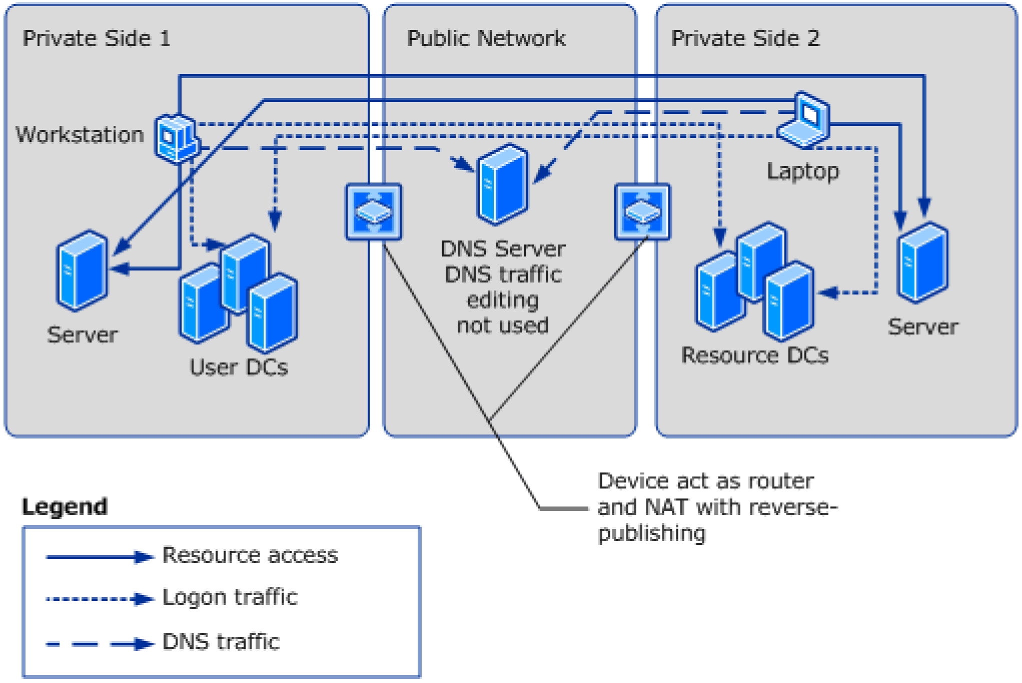
In this case, you also need a public DNS zone unless you have DNS traffic editing available.

The illustration shows the one-way trust case:


Private Side (User Forest)
Public Side (Resource Forest)
Workstation
Server
User DCs
Device acts as router
Resource DCs
Server
Device act as router and NAT; reverse-publishing for bidirectional trust
Legend
Resource access
Logon traffic
DNS traffic

Domain controllers in both forests are published (Two NATs)

If IP address conflicts cannot be resolved on both sides of the network and you have a two-way trust then both forests must run on the private side of a NAT. The configuration requires having a name resolution server on the public network hosting “public” DNS zones for domains on each forest.



In this configuration, the complexity of the configuration will increase vastly, as you require publishing the services of potentially many servers across the NAT. These include:

- Domain controllers of each end of the trust.
- Domain controllers of each domain where users should access resources across the trust.
- Domain controllers of domains of resources that should be accessed through the trust.
- Domain controllers of all domains in the trust path in both forests.
- The resource servers themselves.
- Distributed server applications that have servers on both sides of the NATs may require special attention.

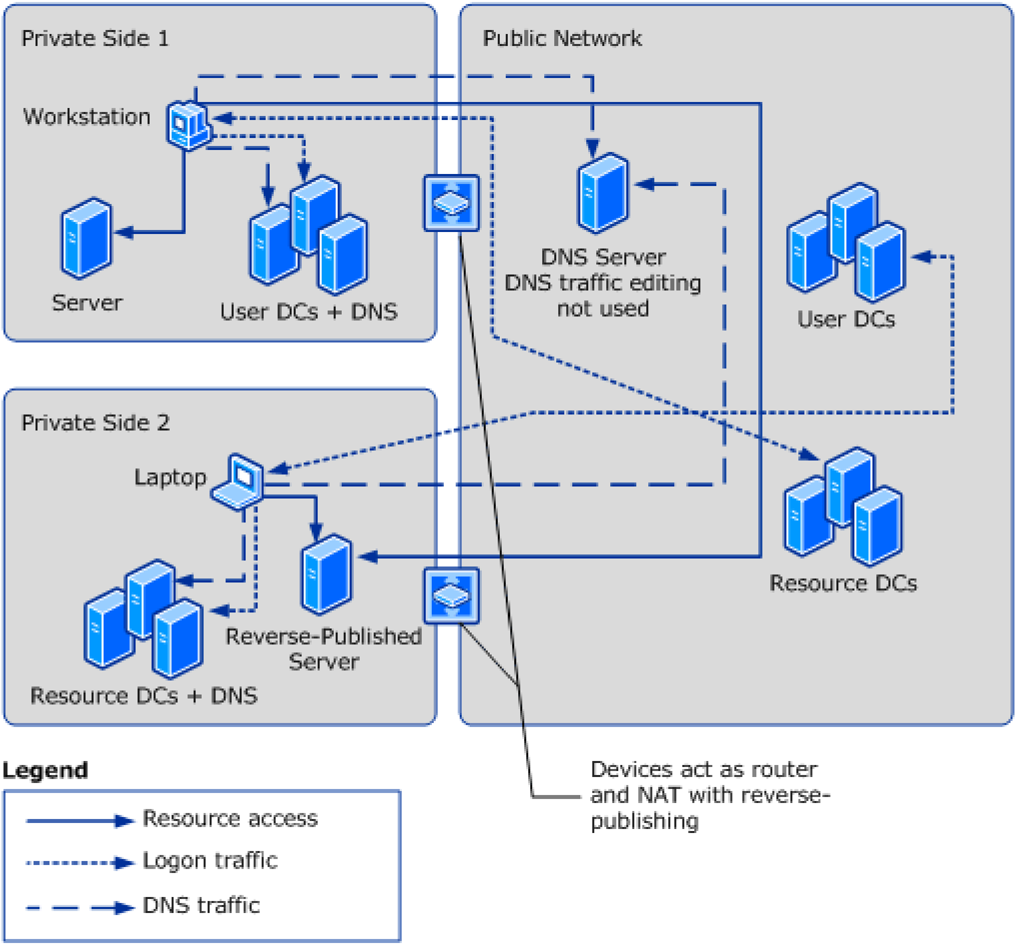
For each server you will have a public IP address managed by the NAT. This IP address needs to be published in DNS as well, but the server will not be able to manage this address in DNS. The name of the DNS zone on the public side of the NAT is the same as on the private side, so you would require a static DNS zone on the public network. Each DNS server on a private side would forward queries for the remote forest into a DNS server running on the public network (or reverse-published on the public network).

An alternative is a DNS server on the private network where the NAT maps private to public IP addresses as the DNS traffic traverses the NAT (DNS traffic editing).

Domain controllers of both forests are in the public network (Two NATs)

In this approach, domain controllers from both forests are placed in the public network (domain controllers seen as public from both private networks). The advantage of this approach is that there is a set of domain controllers that are reachable from all networks using known addresses.

On the other hand, a set of domain controllers from the private networks would still need to be reverse-published in order to allow for Active Directory and SYSVOL replication with domain controllers in the public network.



Another variation of the same scenario is when all domain controllers of a certain forest are on the public side of the NAT. This requires that you are able to concentrate the domain controllers into the public network. That variation is simpler from an Active Directory standpoint as the considerations will be reduced to client systems in the private networks. You may still need custom DNS zones and reverse publishing for servers on the private side of the NATs.

NAT between domain controllers of the same domain or forest

The next level of complexity is reached where domain controllers from the same forest are on both sides of the NATs, in more than one private network, or in private and public networks. In addition to managing the DNS name resolution for the different networks, you also may have to consider Naming Context presence in the various network segments and replication topology.

We will not consider trusts in this case, as the trusting or trusted domains or forests would just be another set of clients or servers you need to worry about. The considerations for using trusts are the same as in section NAT between two Active Directory forests.

Domain controllers in private networks only

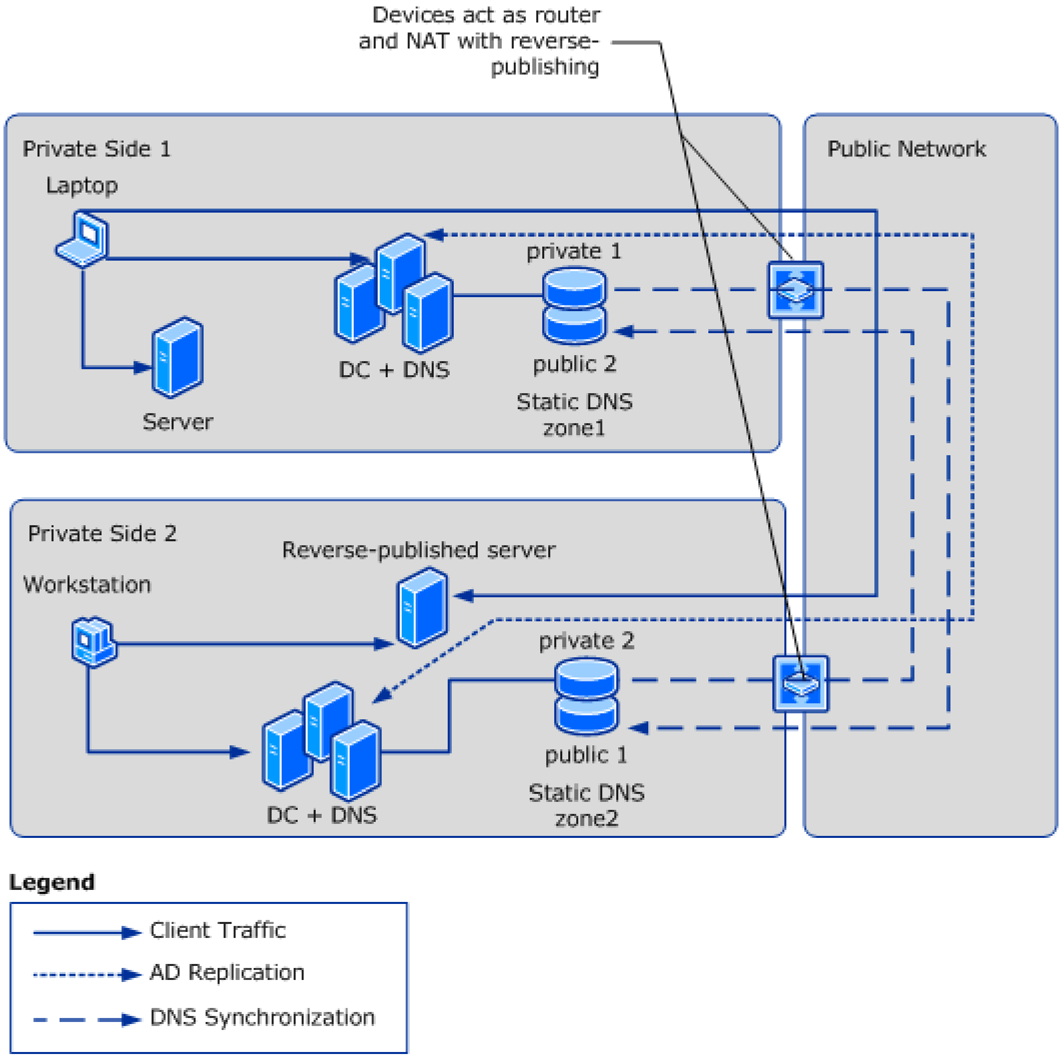
In this case, you could implement a Virtual Private Network (VPN) across the NATs and use transparent IP routing. However, this requires that you have routing domains that allow you to route transparently. This would require that all domain controllers have conflict-free addresses. Please remember that Active Directory may not be the only application you need to get working across the NAT. Examples can be Exchange or ISA Server deployment.

The clients will have duplicate IP addresses on either private side (otherwise there is no reason to use NAT). You need to decide whether you want to map these duplicate IP addresses in Active Directory Sites and Services at all and rely on site-less names, or if you want to tolerate parts of the clients to talk to remote domain controllers.

If a VPN cannot be implemented, you can follow a similar approach of having static DNS zones (as you require with NAT between forests). However, in this case, Active Directory-integrated DNS cannot be used as conflicting private addresses would be across all domain controllers in the domain. One approach you can consider is to follow a “split-brain” DNS solution where you manage parts of the infrastructure manually. This means that in a particular network you will have DNS servers with SRV records of domain controllers in that network only. For domain controllers in the other networks you will need to manually maintain CNAME and host A records for their published IP addresses in order to allow Active Directory and SYSVOL replication.

Note

The meaning of the arrows in the following diagrams is different from previous diagrams. “Client traffic” that has been represented by three different arrow types in previous diagrams are now using solid lines and the other arrow types stand for Active Directory replication and DNS synchronization. This hides some of the complexity of this deployment.



Domain controllers in both the public and private networks

In this scenario, domain controllers of the same forest are in private networks as well as in the public network. You can consider "mapping" the public domain controller records to the DNS servers in the private networks. These records would be treated like private domain controller records with conflict-free addresses that can bypass the NAT.

In the following diagram, below you see three parts of the DNS "brain" in three parts of the network. When you have enough domain controllers in the public network to satisfy all replication requirements, you can skip synchronizing the records of the reverse-published domain controllers into the DNS "brain" of the other private networks. However, they need to be present on the public network so the public domain controllers can replicate.

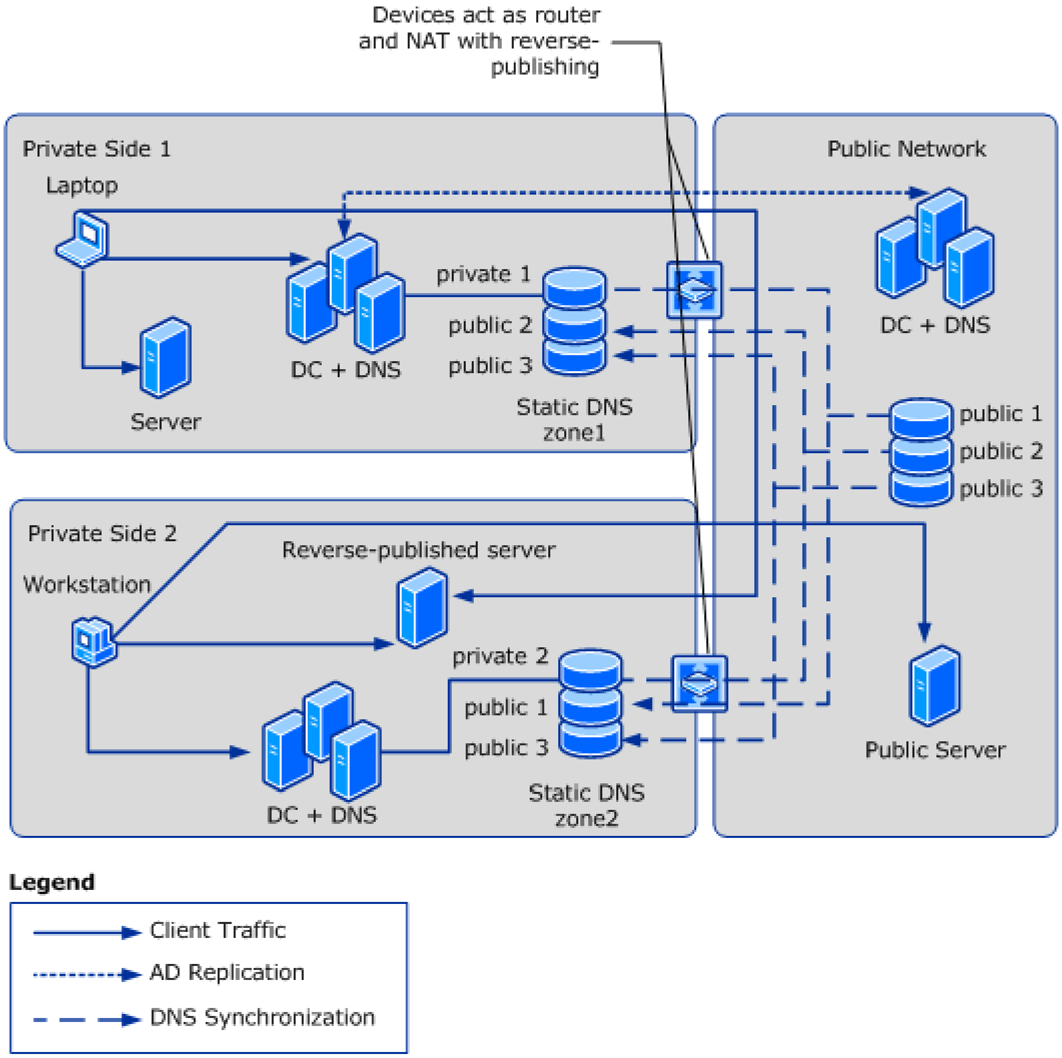


Table of Scenarios

The table summarizes the scenarios discussed in this paper. It should help you determine the DNS requirements. The column **DNS Zones for Forest** has hints on synchronization requirements of the zones you create. These do not refer to zone transfers; the contents of the zones will differ as the public zone cannot contain private addresses.

Placement of DCs of the forest	Trust Direction	Connectivity Required for NTLM or Kerberos	DNS Zones for Forest	Will DNS traffic editing help?	Reverse Publishing of DCs
All on private side of NAT	Trusted	NTLM	One zone private, one	Yes	Yes

Placement of DCs of the forest	**Trust Direction**	**Connectivity Required for NTLM or Kerberos**	**DNS Zones for Forest**	**Will DNS traffic editing help?**	**Reverse Publishing of DCs**
			zone public		
Some on one private network, some on public network	Trusted	NTLM	One zone private, one zone public *Public zone needs to be synchronized to each private zone*	Should not be needed	Maybe
All on public network	Trusted	NTLM	One zone public	No	No
DCs in public and more than one private network	Trusted	NTLM	One public zone, and one zone for each private network *Public zone needs to be synchronized to each private zone*	No, no single consistent DNS zone	Avoid
All on private side of NAT	Trusting	Kerberos	One zone private, one zone public	Yes	Yes
Some on one private network, some on public network	Trusting	Kerberos	One zone private, one zone public *Public zone needs to be synchronized to each private zone*	Only required for server and application records in private network	Maybe
All on public network	Trusting	Kerberos	One zone public	Only for server and application records in	No

Placement of DCs of the forest	Trust Direction	Connectivity Required for NTLM or Kerberos	DNS Zones for Forest	Will DNS traffic editing help?	Reverse Publishing of DCs
				private network	
DCs in public and more than one private network	Trusting	Kerberos	One public zone, and one zone for each private network *Public zone needs to be synchronized to each private zone*	No, no single consistent DNS zone	Avoid
All on private side of NAT	Bidirectional	Both	One zone private, one zone public	Yes	Yes
Some on one private network, some on public network	Bidirectional	Both	One zone private, one zone public *Public zone needs to be synchronized to each private zone*	Only required for server and application records in private network	Maybe
All on public network	Bidirectional	Both	One zone public	Only for server and application records in private network	No
DCs in public and more than one private network	Bidirectional	Both	One public zone, and one zone for each private network *Public zone needs to be synchronized to each private*	No, no single consistent DNS zone	Avoid

Placement of DCs of the forest	**Trust Direction**	**Connectivity Required for NTLM or Kerberos**	**DNS Zones for Forest**	**Will DNS traffic editing help?**	**Reverse Publishing of DCs**
			zone		

Summary

Introducing NAT in your environment adds complexity to your overall network configuration. It also adds complexity to applications on the network, and Active Directory is not an exception.

When you review the scenarios you will see that the larger the distribution of domain controllers in your forest within the network, the more complex the setup will be. The complexity of managing DNS will become a big problem, as you may need to manually manage a static DNS zone for each private network with domain controllers of the forest.

Running in a deployment with Active Directory trusts or Active Directory and SYSVOL replication over NATs, you will need to manage the elevated risk that comes with maintaining additional manual configuration. Even if your NATs can edit DNS frames, you still have to keep track of domain controller changes and their required configuration changes on the NATs. Amongst these are adding or removing domain controllers, changing the IP addresses of domain controllers, moving the PDC Emulator role (sometimes still used by applications), and so on.

You should consider building a similar list of changes that will affect your NAT configuration for any other service that is using the network across the NATs before you proceed with a NAT implementation in your environment.