

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskis

## On This Page

Introduction

Windows 2000 Component Overview

Description of the Windows 2000 Startup and Logon Process

User Logon

Conclusion

Appendix A: Test Environment

Appendix B: TCP/IP Ports Used in the Authentication Process

## Introduction

The client startup and logon process is the process the Microsoft Windows operating systems uses to validate a computer or User in the Windows networking environment. Developing an understanding of the client startup and user logon process is fundamental to understanding Windows 2000 networking. This white paper will provide the reader with detailed information on this process, including:

- How clients connect to the network with Windows 2000 Dynamic Host Configuration Protocol (DHCP), Automatic Private Internet Protocol (IP) addressing, and static addressing.
- How Windows 2000 clients use the Dynamic Domain Naming System (DDNS) support in Windows 2000 to locate domain controllers and other servers in the infrastructure needed during startup and logon. In addition we will show how Windows 2000 clients register their names in DDNS.
- How the Lightweight Directory Access Protocol (LDAP) is used during startup and logon to search the Microsoft Active Directory for required information.
- How the Kerberos security protocol is used for authentication.
- How MS Remote Procedure Calls (MSRPC) are used.
- How Server Message Block (SMB) is used to transfer group policy information and other data during the startup and logon process.

In addition to discussing the Windows 2000 core components used by the startup and logon process, the paper shows what happens and how much network traffic is generated during each part of the process. The discussion begins with an overview of the Windows 2000 components involved in the startup and logon process. We will then examine the Client Startup process and discuss the User logon process.

Throughout the discussion sample information from network monitor traces will be used to illustrate what is happening at that particular point. We have also made an effort to provide references

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskis

whenever possible to external sources of information where additional information can be found. The most common reference materials cited include:

- Internet Engineering Task Force (IETF) Requests for Comments (RFCs)
- Microsoft Windows 2000 Resource Kits
- Microsoft Support Knowledge Base articles
- Microsoft Notes from the Field books
- Various Web sites

Reading and understanding this white paper will allow systems architects and administrators to better engineer and support Windows 2000 networks. It should help network designers determine where to place key components to ensure reliable startup and logon in a Windows 2000 network. Support professionals will be able to use this paper to resolve problems by comparing the baseline information provided here to their environments.

## Audience

The target groups for this discussion are systems administrators and network architects who are planning, implementing, or managing Windows 2000 networks. It is expected that this group will have an understanding of the following topics:

- Microsoft Windows NT or 2000 networking concepts
- Basic knowledge of the TCP/IP protocol
- Some exposure to examining network traces

The Windows 2000 Resource Kit, Microsoft TechNet, and the *Notes from the Field* series offer more detailed discussions of core Windows 2000 services we will discuss as part of the client startup and logon process. It would be worthwhile to have access to these resources as supplementary resources while reading this paper.

## Windows 2000 Component Overview

In order to understand the Windows 2000 client startup and logon process, a discussion of the new or updated protocols and services that play a role in this process is needed. This section provides a brief overview of each of the following protocols and services involved:

- Dynamic Host Configuration Protocol (DHCP)
- Automatic Private IP Addressing

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskis

- Domain Naming System (DNS)
- Kerberos
- Lightweight Directory Access Protocol (LDAP)
- Server Message Block (SMB)
- Microsoft Remote Procedure Call (MSRPC)
- Time Service

More in-depth information on each protocol or service can be found using the references provided in each section.

## DHCP

The original objective of the Dynamic Host Configuration Protocol was to provide each DHCP client with a valid Transmission Control Protocol/Internet Protocol (TCP/IP) configuration.

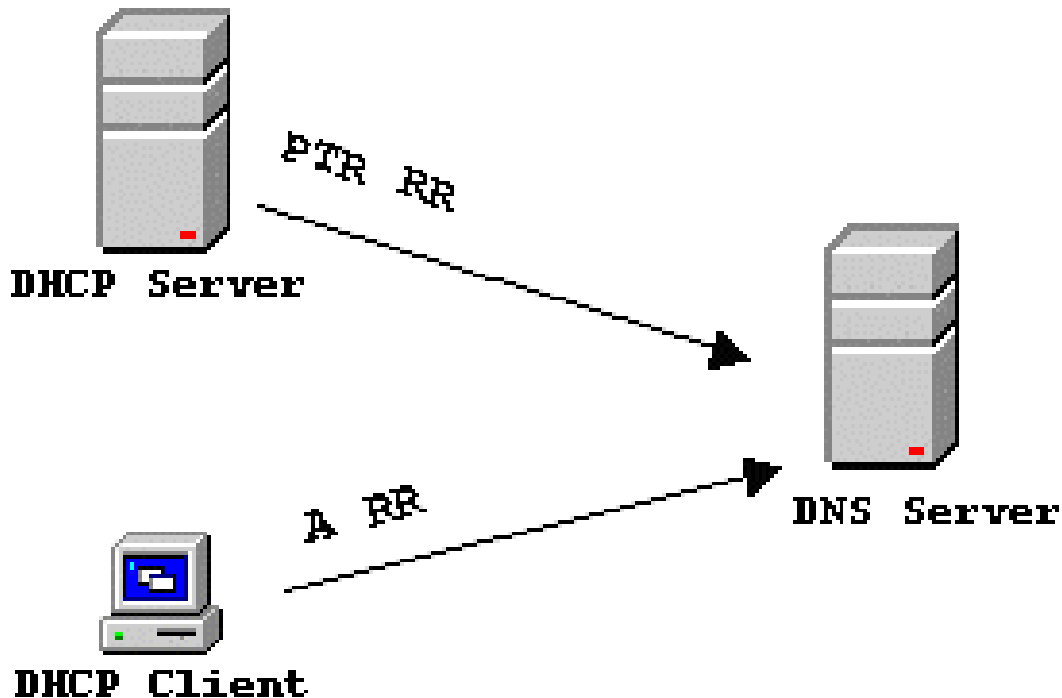
The process in general consists of eight messages:

- **DHCPDiscover.** A DHCP client uses this message in order to detect at least one DHCP server.
- **DHCPOffer.** Each DHCP server that receives the request from a client checks its scopes for a valid configuration set and offers this to the DHCP client.
- **DHCPRequest.** The DHCP client requests the first offer it receives from the DHCP server.
- **DHCPAcknowledge.** The selected DHCP server uses this message in order to confirm the lease with the DHCP client.
- **DHCPNack.** The DHCP server uses this message in order to inform a client that the requested TCP/IP configuration is invalid.
- **DHCPDecline.** The DHCP client uses this message in order to inform the server that an offered TCP/IP configuration is invalid.
- **DHCPRelease.** The DHCP client uses this message to inform the server that an assigned configuration is no longer in use by the client.
- **DHCPInform.** This is a new message defined in Request for Comments (RFC) 2131. If a client has already obtained an Internet Protocol (IP) address (for example, manual configuration), it may use this message to retrieve additional configuration parameters that are related to the IP address from a DHCP server.

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskis

This role of a DHCP server was extended with the availability of Dynamic DNS. In this case, the DHCP server can be used for the dynamic registration of the client's IP address and the hostname. In the default configuration, the DHCP server registers the IP address of the client with the DNS server. This is also known as Pointer Record (PTR RR).



For more information about DHCP, see:

- RFC 1541
- RFC 2131
- Dynamic Host Configuration Protocol, TCP/IP Core Networking Guide, Windows 2000 Server Resource Kit

## Automatic Private IP Addressing

Windows 2000 implements the Automatic Private IP Addressing (APIPA), which will provide an IP address to a DHCP client even if there is no DHCP server available. APIPA is designed for computers on single-subnet networks that do not include a DHCP server. APIPA automatically assigns an IP address from its reserved range, 169.254.0.01 through 169.254.255.254. What this means is that when a client fails to communicate with a local DHCP server at startup to renew its lease, it will use an APIPA assigned address until it can communicate with a DHCP server. This is different behavior from Windows NT 4.0 where the client would continue to lease a lease that had not expired even if the client could no longer contact the DHCP server.

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskis

Using the Registry Editor to create the following registry key can disable APIPA:

```
HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet
```

```
Services\Tcpip\Parameters\Interfaces\<adapter name>
```

Where <adapter name> is the name of the Dynamic Host Configuration Protocol (DHCP) configured adapter where you want to disable APIPA.

Add the following value to this key:

Value name: IPAutoconfigurationEnabled

Value type: REG\_DWORD

Value in hexadecimal: 0 (A value of 0 disables APIPA support on this adapter)

**Note:** If the IPAutoconfigurationEnabled entry is not present, a default value of 1 is assumed, which indicates that APIPA is enabled.

This change requires the computer to be restarted to take effect. It is documented in the articles 244268 and 244268 available at <http://search.support.microsoft.com/>

## DNS

The primary mechanism for service location and name resolution in Windows 2000 is the Domain Name System (DNS). Windows 2000 includes a DNS system that is tightly integrated with the operating system providing integration with the Active Directory and support for making Dynamic updates, but any BIND 8.2.2 compliant DNS can be used with Windows 2000. DNS support in Windows 2000 is intended as a standards-based replacement for the NetBIOS-based Windows Internet Naming Service (WINS), which was the previous service that provided dynamic support for Windows clients. Both services provide the ability for dynamic updating of system names into their databases but WINS is a flat-name space and does not scale as well as DNS. By moving to DNS, Windows 2000 not only conforms to Internet standards, it provides a hierarchical naming system that scales to meet the demands of large networks.

The Windows 2000 startup and logon process uses DNS to locate services like LDAP and Kerberos to retrieve the address of at least one controller and to register its hostname and IP address in DNS zone database.

The Windows 2000 DNS system and its requirements are covered in great detail in the Windows 2000 Resource Kit and the *Notes from the Field* book *Building Enterprise Active Directory Services*.

For more information, see:

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskis

- Windows 2000 Resource Kit: Name Resolution in Active Directory, Distributed Systems Guide
- Windows 2000 Resource Kit: Windows 2000 DNS, TCP/IP Core Networking Guide
- RFC: 1035, 1036

## Kerberos

Kerberos V5 is the default authentication protocol in Windows 2000. Kerberos originated as part of Project Athena at MIT in the late 1980s, and version 5 is described in the IETF RFC 1510.

Kerberos V5 is an authentication protocol. It allows mutual authentication between computers. In other words, it allows computers to identify each other. This is of course, the basis of all security systems. Unless a server is absolutely sure you are who you say you are, how can that server reliably control access to its resources? Once the server has positive identification of who you are, it can then make the determination about whether you are authorized to access the resource.

Kerberos, *per se*, does not authorize the user to access any resource, although the Microsoft implementation of Kerberos V5 does allow secure delivery of user credentials. (For the specification of the fields involved, see

<http://www.microsoft.com/technet/archive/interopmigration/mgmt/kerberos.mspix>.)

There are six primary Kerberos messages. The six messages are really three types of actions, each of which has a request from a client and a response from the Key Distribution Center (KDC). The first action occurs when the client types in a password. The Kerberos client on the workstation sends an "AS" request to the Authentication Service on the KDC asking the Authentication Service to return a ticket to validate the user is who they say they are. The Authentication Service verifies the client's credentials and sends back a Ticket Granting Ticket (TGT).

The second action is when the client requests access to a service or a resource by sending a TGS request to the Ticket Granting Service (TGS) on the KDC. The Ticket Granting Service returns an individual ticket for the requested service that the client can submit to whatever server holds the service or resource the clients wants.

The third action is when the Kerberos client actually submits the service ticket to the server and requests access to the service or resource. These are the AP messages. In Microsoft's implementation, the access security identifiers (SIDs) are contained in the PAC that is part of the ticket sent to the server. This third action need not have a response by the server unless the client has specifically asked for mutual authentication. If the client has marked this exchange for mutual authentication, the server returns a message to the client that includes an authenticator timestamp.

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskis

For a typical domain logon, all three of these actions occur before the user is allowed access to the workstation.

For more information about Kerberos on Windows 2000, see

<http://www.microsoft.com/windows2000/techinfo/howitworks/security/kerberos.asp>.

## LDAP

Lightweight Directory Access Protocol is a Directory Access Protocol (DAP) designed to allow clients to query, create, update, and delete information stored in a directory. It was initially used as a front-end to X.500 directories, but can also be used with stand-alone directory servers. Windows 2000 supports both LDAP version 3 and version 2.

## LDAP Process

The general model adopted by LDAP is of clients performing protocol operations against servers. A client transmits a request describing the operation to be performed to a server. The server is then responsible for performing the necessary operations in the directory. Upon completion of the operations, the server returns a response containing any results or errors to the requesting client.

The LDAP information model is based on the entry, which contains information about some object (for example, a person). Entries are composed of attributes, which have a type and one or more values. Each attribute has a syntax that determines what values are allowed in the attribute and how those values behave during directory operations. Examples of attribute syntaxes are for IA5 (ASCII) strings, URLs, and public keys.

## LDAP Features

Windows 2000 supports the LDAPv3 protocol as defined in RFC 2251. Key aspects of the protocol are:

- All protocol elements of LDAPv2 (RFC 1777) are supported.
- The protocol is carried directly over TCP or other transport, bypassing much of the session/presentation overhead of X.500 DAP.
- Most protocol data elements can be encoded as ordinary strings (for example, distinguished names)
- Referrals to other servers may be returned (described in the next section).
- Simple Authentication and Security Layer (SASL) mechanisms may be used with LDAP to provide association security services.

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskis

- Attribute values and distinguished names have been internationalized through the use of the International Standards Organization (ISO) 10646 character set.
- The protocol can be extended to support new operations, and controls may be used to extend existing operations.

## LDAP Referral

An LDAP referral is a domain controller's way of indicating to a client application that it does not have a copy of a requested object (or, more precisely, that it does not hold the section of the directory tree where that object would be, if in fact it exists) and giving the client a location that is more likely to hold the object, which the client uses as the basis for a DNS search for a domain controller. Ideally, referrals always reference a domain controller that indeed holds the object. However, it is possible for the referred-to domain controller to generate yet another referral, although it usually does not take long to discover that the object does not exist and to inform the client. Active Directory returns referrals in accordance with RFC 2251.

## RootDSE

The rootDSE represents the top of the logical namespace and therefore the top of the LDAP search tree. The attributes of the rootDSE identify both the directory partitions (the domain, schema, and configuration directory partitions) that are specific to one domain controller and the forest root domain directory partition.

The rootDSE publishes information about the LDAP server, including what LDAP version it supports, supported SASL mechanisms, and supported controls, as well as the distinguished name for its subschema subentry.

Clients connect to the rootDSE when making LDAP at the start of an LDAP operation.

## LDAP over TCP

LDAP message PDUs (Protocol Data Units) are mapped directly onto the TCP byte stream. RFC 2251 recommends that server implementations provide a protocol listener on the assigned port, 389. The Active Directory uses port 389 as the default port for domain controller communications. In addition, the Active Directory supports port 636 for LDAP Secure Sockets Layer (SSL) communications. A Windows 2000 domain controller that is a Global Catalog (GC) server will listen on port 3268 for LDAP communications and port 3269 for LDAP SSL communications.

## LDAP During the Startup and Logon Process

LDAP is used extensively during the Windows 2000 startup and logon process. The client uses LDAP during the domain controller locator process to get the domain controller it will use. LDAP is also used to find the applicable group policy objects for the computer or user. Finally, LDAP is used to locate the appropriate certificates for the client during certificate auto enrollment.

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskas

For more information, see:

- Windows 2000 Resource Kit
- Windows 2000 Active Directory Technical Reference
- RFCs 1777, 1778, 1779, 1959, 1960, 1823
- RFCs 2251-2256

## SMB

The Server Message Block protocol is the resource-sharing protocol used by MS-Net, LAN Manager, and Windows Networking. In addition, there are SMB solutions for OS/2, Netware, VMS, and Unix from vendors such as AT&T, HP, SCO, and, via Samba, over 33 others. The SMB protocol is used in a client-server environment to access files, printers, mail slots, named pipes, and application programming interfaces (APIs). It was jointly developed by Microsoft, IBM, and Intel in the mid-1980s. As illustrated by the following chart, SMB will run over multiple network protocols:

OSI			TCP/IP		
Application	SMB				Application
Presentation					
Session	NetBIOS		NetBIOS	NetBIOS	
Transport	IPX <sup>1</sup>	NetBEUI	DECnet	TCP&UDP	TCP/UDP
Network				IP	IP
Link	802.2, 802.3,802.5	802.2 802.3,802.5	Ethernet V2	Ethernet V2	Ethernet or others
Physical					

In SMB communication, a client connects to the server by negotiating a dialect. Once the client has established a connection, it can then send commands (SMBs) to the server that allow the client to access resources.

The SMB commands can be generally categorized into four parts:

- Session Control
- File Commands
- Print Commands

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskis

- Message Commands

SMB security has evolved as the platforms that use it have evolved. The base SMB protocol model defines two levels of security:

- **Share level.** Protection is applied at the share level on a server. Each share can have a password, and a client needs only that password to access all files under that share. This was the first security model that SMB had and is the only security model available in the Core and CorePlus protocols. Windows for Workgroups vserver.exe implements share level security by default, as does Windows 95.
- **User Level.** Protection is applied to individual files in each share and is based on user access rights. Each user (client) must log in to the server and be authenticated by the server. When it is authenticated, the client is given a user ID, which it must present on all subsequent accesses to the server. This model has been available since LAN Manager 1.0.

The SMB protocol has gone through many revisions over time. The most current version of SMB implemented in Windows 2000 is the Common Internet File System (CIFS), which is a slight variant of the NT LM 0.12 version used previously. The next section goes into the details of this modern implementation.

## Windows 2000 support for SMB via the CIFS

The Common Internet File System is the standard way that computer users share files across corporate intranets and the Internet in a Windows network. The CIFS is an enhanced version of the SMB protocol. CIFS is an open, cross-platform implementation of SMB that is currently a draft Internet standard. CIFS was introduced in Service Pack 3 for Windows NT 4.0 and is the native file sharing protocol for Windows 2000. The CIFS is a variant of the NTLM 0.12 protocol.

## Windows 2000 SMB/CIFS Protocol Implementation

CIFS defines a series of commands used to pass information between networked computers. The redirector packages requests meant for remote computers in a CIFS structure. CIFS can be sent over a network to remote devices. The redirector also uses CIFS to make requests to the protocol stack of the local computer. CIFS messages can be broadly classified as follows:

- Connection establishment messages consist of commands that start and end a redirector connection to a shared resource at the server.
- Namespace and File Manipulation messages are used by the redirector to gain access to files at the server and to read and write them.

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskis

- Printer messages are used by the redirector to send data to a print queue at a server and to get status information about the print queue.
- Miscellaneous messages are used by the redirector to write to mail slots and named pipes.

In Windows 2000, CIFS supports distributed replicated virtual volumes (such as Distributed File System [DFS]), file and record locking, file change notification, read-ahead and write-behind operations. CIFS communications are established via standard SMB session and name resolution mechanisms.

## The SMB/CIFS Process in Windows 2000

When there is a request to open a shared file, the I/O calls the redirector, which in turn requests the redirector to choose the appropriate transport protocol. For NetBIOS requests, NetBIOS is encapsulated in the IP protocol and transported over the network to appropriate server. The request is passed up to the server, which sends data back to satisfy the request.

In Windows NT 4.0, Windows Internet Name Service (WINS), and Domain Name System (DNS), name resolution was accomplished by using TCP port 134. Extensions to CIFS and NetBT now allow connections directly over TCP/IP with the use of TCP port 445. Both means of resolution are still available in Windows 2000. It is possible to disable either or both of these services in the registry.

## SMB Utilization During the Startup and Logon Process

The Windows 2000 client startup and logon process uses SMB to load Group Policy objects applicable to that workstation or user. The basic SMB operation that is observed during the startup and logon process is SMB dialect negotiation. This is an exchange between the client and the server to determine which SMB dialect they will be able to use. SMB will also be used to make a DFS referral for the share that is being accessed. The client loading Group Policy objects will create the majority of SMB traffic during the startup and logon process.

For more information, see:

- Windows 2000 TCP/IP Protocols and Services Technical Reference

## MSRPC

The simplest way to describe an RPC is a request by one computer to use the processing resources on another. The RPC protocol permits one process to request the execution of instructions by another process located on another computer in a network.

The RPC process consists of:

- **Client application.** Requests the remote execution.

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskis

- **Client stub.** Translates calls into/from standard network representation (NDR) format.
- **Client RPC Runtime Library.** Converts NDR into network messages
- Network Transport: Handles the network communications.
- **Server RPC Runtime Library.** Converts NDR into network messages.
- **Server stub.** Translates calls into/from standard network representation (NDR) format.
- **Server application.** Executes the requested instructions.

The RPC procedures are uniquely identified by an interface number (UUID), an operation number (opnum), and a version number. The interface number identifies a group of related remote procedures. An example for an interface is net logon, which has the UUID 12345678-1234-ABCD-EF00-01234567CFFB.

An example for an RPC call is:

```
MSRPC: c/o RPC Bind: UUID 12345678-1234-ABCD-EF00-01234567CFFB call 0x1
assoc grp 0x0 xmit 0x16D0 recv 0x16D0
MSRPC: Version = 5 (0x5)
MSRPC: Version (Minor) = 0 (0x0)
MSRPC: Packet Type = Bind
+ MSRPC: Flags 1 = 3 (0x3)
MSRPC: Packed Data Representation
MSRPC: Fragment Length = 72 (0x48)
MSRPC: Authentication Length = 0 (0x0)
MSRPC: Call Identifier = 1 (0x1)
MSRPC: Max Trans Frag Size = 5840 (0x16D0)
MSRPC: Max Recv Frag Size = 5840 (0x16D0)
MSRPC: Assoc Group Identifier = 0 (0x0)
+ MSRPC: Presentation Context List
```

RPC are independent from the low-level transport protocol. Microsoft RPC (MSRPC) can be layered on top of several different transport protocols such as TCP/IP, Internetwork Packet Exchange/Sequenced Packet Exchange (IPX/SPX), or NetBIOS Enhanced User Interface (NetBEUI).

Most of the RPC interfaces use dynamic ports for the network communication. In this case it is necessary to involve a specific interface, called the End Point Mapper. The End Point Mapper is

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskaskas

always listening on port 135 for TCP/IP and the End Point Mapper's UUID is E1AF8308-5D1F-11C9-91A4-08002B14A0FA.

The client has to bind to an interface first before it can call its procedures. If the bind process was successful, it can send a request to the End Point Mapper, in which it includes the UUID of the target interface. The End Point Mapper sends back the port number the client can use for the communication.

The following table shows the communication sequence for this process.

Frame	Source	Destination	Protocol	Description
1	Client	Server	MSRPC	c/o RPC Bind: UUID E1AF8308-5D1F-11C9-91A4-08002B14A0FA I
2	Server	Client	MSRPC	c/o RPC Bind Ack: call 0x1 assoc grp 0xC85D xmit 0x16D0 recv
3	Client	Server	MSRPC	c/o RPC Request: call 0x1 opnum 0x3 context 0x0 hint 0x84
4	Server	Client	MSRPC	c/o RPC Response: call 0x1 context 0x0 hint 0x80 cancels 0x0

It is also possible to encapsulate MSRPCs into SMB. In this case, the client and server are using a handle to a previously opened file in order to exchange data.

The Windows 2000 startup and logon process uses the Netlogon and the Directory Replication Service (DRS) interface. The Netlogon interface is used to establish the secure communications channel between the client and a domain controller in a domain. The Directory Replication Service is primarily used for communication between Domain Controllers and Global Catalog servers. It does, however, provide an interface used during the logon process. The DRS provides a method to convert names into a format that is useable by LDAP.

For more information, see

- "Active Directory Client Network Traffic," *Notes from the Field Building Enterprise Active Directory Services*
- "Analyzing Exchange RPC Traffic Over TCP/IP [159298]" on TechNet

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskis

## Time Service

Windows 2000 includes the W32Time (Windows Time) time service that provides a mechanism for synchronizing system time between Windows 2000 clients in a domain. Time synchronization occurs during the computer startup process.

The following hierarchy is used by systems in the domain to perform time synchronization:

- All client desktops nominate as their in-bound time partner their authenticating domain controller.
- All member servers follow the same process as client desktops.
- All domain controllers in a domain nominate the primary domain controller (PDC) Operations Masters as their in-bound time partner.
- PDC Operations Masters follow the hierarchy of domains in the selection of their in-bound time partner.

**Note:** Operations Masters are described in Chapter 7 of the Distributed Systems Guide in the Windows 2000 Server Resource Kit.

For more information, see:

- 216734 for a description of the time sync process and how to set up an authoritative time source
- RFC: 1769, 2030

## Windows 2000 Domain Authentication Methods

Windows 2000 supports two different authentication methods for domain logons: Kerberos and NTLM. The use of Kerberos as the authentication protocol for Windows 2000 changes the default Windows authentication protocol from NTLM to a protocol that is based on Internet Standards.

The default authentication protocol in Windows 2000 is based on MIT's Kerberos version 5, outlined in RFC 1510. Using Kerberos for authentication fundamentally changes the way security information is exchanged between clients, servers, and domain controllers in the Windows network.

With Kerberos, the client requests what is called a session ticket from the KDC. The session ticket contains information that the server can use to validate the client has the necessary authentication for the server. The client then sends these tickets to authenticate with the resource it is trying to use. Kerberos only provides authentication information from the client to the resource it is trying to access. Kerberos does not provide authorization to access resources; it simply supplies the authorization to the system.

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskis

To provide legacy support, Windows 2000 continues to provide support for NTLM authentication. Windows 2000 will use NTLM for authentication when not in a Windows 2000 domain, as a member of a NT 4 domain or workgroup, or when there is a need to access older applications that require NTLM for authentication.

This discussion will focus on the Kerberos authentication protocol and how it is used during the startup and logon process. The startup and logon process using the NTLM protocol is unchanged from Windows NT 4.0 and is covered in detail in *Notes from the Field* series book *Building Enterprise Active Directory Services*.

For more information, see:

- Windows 2000 Resource Kit: Authentication; Distributed Systems Guide

## Description of the Windows 2000 Startup and Logon Process

When you think of logon in the Windows environment, you typically think of the process of a user pressing ctrl-alt-delete and entering his or her username and password (credentials) to gain access to a system. These credentials will give users access either to resources on the computer they are working on or if the system is part of a network, these credentials give access to resources on the network such as applications, files, or printers that the user has been authorized to access.

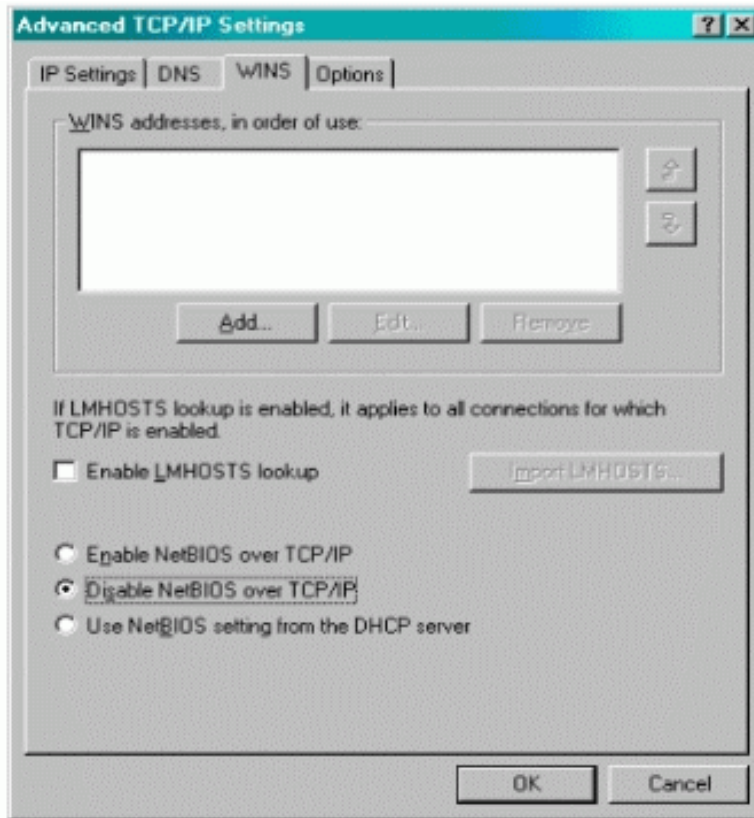
The process that allows users to access resources does not start when the user logs on to the system, but begins well before that when the system is started. In a Windows 2000 domain environment, the computer needs to establish itself as a valid member of a domain before users will be able to logon to that system and access other resources on the network.

The objective of the following sections is to describe the steps that are involved in the Windows 2000 startup and logon process. The NetBIOS over TCP/IP functionality is disabled on all computers to set the focus on network traffic that occurs in an environment that has only Windows 2000 computers.

However, it is possible to configure each computer to be compatible with Windows NT and Windows 95 and Windows 98 clients. A detailed description of the startup and logon traffic that is associated with Windows NTbased clients can be found in the "Active Directory Client Network Traffic" chapter of the *Notes from the Field* series book *Building Enterprise Active Directory Services*. The following graphic illustrates the configurations.

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskas

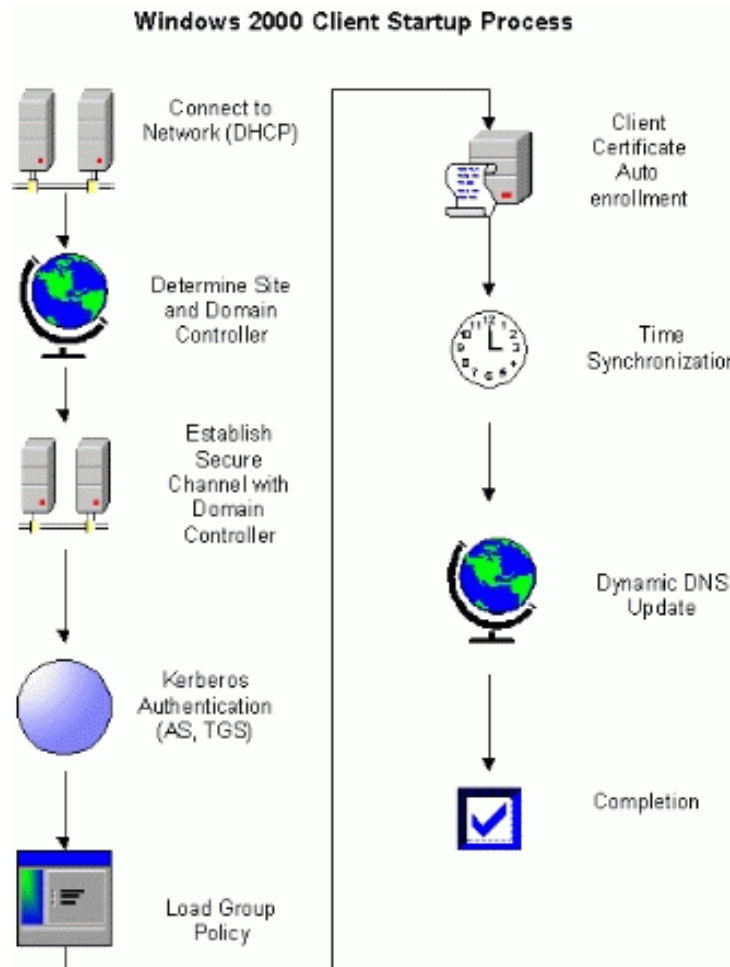


## The Computer Startup Process

A computer that is a member of a Windows 2000 domain goes through a startup process that connects it to the domain it is a member of. This startup process allows services on the computer to interact or be interacted with on the network, and more importantly, it is required in order for users to interactively log on. This process flow shows the computer startup process.

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskas



## Connecting to a Network

The computer startup process begins when the computer connects to the network and loads the TCP/IP protocol. TCP/IP can be configured to use either static or dynamic configuration. Dynamic configuration means using the DHCP, which is a well-documented technology that is a core component of the Windows Server operating system. Static addressing means that TCP/IP configuration information has been manually configured on the computer. Typically, static addresses are used for resources that do not change very often, such as routers or servers. In the examples in this paper, the only systems that use static addresses are the servers.

The DHCP process generates the following frames on the network when the client connects to the network. The sequence of "Discover," "Offer," "Request," and "ACK" in the first four frames is the DHCP process in action. These four frames generate 1,342 bytes (about 342 bytes per DHCP frame) of network traffic, but this will vary depending upon the number of DHCP options specified. The Reverse ARPs (RARP) in frames 5 through 8 are performed by the client to ensure the address is not in use by another computer on the network. Each RARP frame creates about 60 bytes of network traffic or around 300 bytes for the address check sequence.

## Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskas

Frame	Source	Destination	Protocol	Description
1	Client	*BROADCAST	DHCP	Discover
2	Server	*BROADCAST	DHCP	Offer
3	Client	*BROADCAST	DHCP	Request
4	Server	*BROADCAST	DHCP	ACK
5	Client	*BROADCAST	ARP_RARP	ARP: Request, Target IP: 10.0.0.100
6	Client	*BROADCAST	ARP_RARP	ARP: Request, Target IP: 10.0.0.100
7	Client	*BROADCAST	ARP_RARP	ARP: Request, Target IP: 10.0.0.100
8	Server	*BROADCAST	ARP_RARP	ARP: Reply, Target IP: 10.0.0.100

It is important to note that if a client already possesses a lease, then it will simply renew the lease with the DHCP server when it restarts. The renewal includes only the Request and Acknowledgement packets shown in the first two frames below. The client still performs the RARP process to ensure its address is not in use.

Frame	Source	Destination	Protocol	Description
1	Client	*BROADCAST	DHCP	Request
2	Server	*BROADCAST	DHCP	ACK
3	Client	*BROADCAST	ARP_RARP	ARP: Request, Target IP: 10.0.0.100
4	Client	*BROADCAST	ARP_RARP	ARP: Request, Target IP: 10.0.0.100
5	Client	*BROADCAST	ARP_RARP	ARP: Request, Target IP: 10.0.0.100
6	Server	*BROADCAST	ARP_RARP	ARP: Reply, Target IP: 10.0.0.100

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskas

## Domain Controller Detection

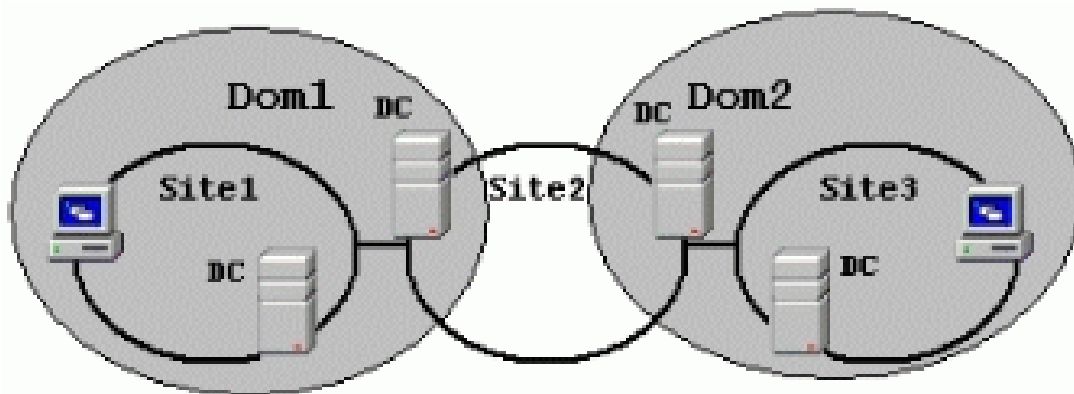
A client computer needs to get the latest configuration status during each startup phase. Therefore, it has to locate at least one controller in its domain.

In a Windows 2000 domain, each controller is also an LDAP server. In order to retrieve a list of available controllers, the client can query the DNS for SRV resource records with the name *\_ldap.\_tcp.dc.\_msdcs.DnsDomainName*.

The following frames show an example of this.

Frame	Source	Destination	Protocol	Description
1	Client	Server	DNS	0x1:Std Qry for _ldap._tcp.dc._msdcs.main.local. of type Srv Loc on class INET addr.
2	Server	Client	DNS	0x1:Std Qry Resp. for _ldap._tcp.dc._msdcs.main.local. of type Srv Loc on class INET addr

The Windows 2000 domain is an administrative boundary, which is independent from the structure of a given network. The computer in a given environment can be grouped into sites. A site in Windows 2000 is defined as a set of IP subnets connected by fast, reliable connectivity. As a rule of thumb, networks with LAN speed or better are considered fast networks for the purposes of defining a site. A domain can span multiple sites and multiple domains can cover a site.





# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskas

If it is possible for the DNS server to locate the requested information, it sends back a list of all known domain controllers in the site.

DNS: Answer section: \_ldap.\_tcp.Site2.\_sites.dc.\_msdcs.dcclab.local. of type Srv Loc on class INET addr.(2 records present)

DNS: Resource Record: dcclab22.dcclab.local. of type Host Addr on class INET addr.

DNS: Resource Record: dcclab21.dcclab.local. of type Host Addr on class INET addr.

The client randomly picks up one controller for the additional communication process, and it does not distinguish between local or remote subnets because it considers each member of its site as a computer that is reasonably close to the client.

As already mentioned, it is possible to have an influence on the controller selection in form of the site concept. After retrieving a domain controller, the client tries to determine whether the controller is the closest one in form of LDAP queries.

Frame	Source	Destination	Protocol	Description
1	Client	Server	LDAP	ProtocolOp: SearchRequest (3)
2	Server	Client	LDAP	ProtocolOp: SearchResponse (4)
3	Client	Sever	LDAP	ProtocolOp: SearchRequest (3)
4	Server	Client	LDAP	ProtocolOp: SearchResponse (4)

In the query, the client requires a match for attributes such as its:

- DNS domain name
- Host name
- Domain globally unique identifier (GUID)
- Domain security identifier (SID)

If the controller does have exactly this information in its Active Directory database, it passes back information about itself such as:

- DomainControllerName
- DomainControllerAddress

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskas

- DomainControllerAddressType
- DomainGUID
- DomainName
- DNSForestName
- DCSiteName
- ClientSiteName

The most important information for the client is the site name. The hex dump of the response from the server will contain only one site name if the client is a member of the controller's site:

```
00000: 00 A0 C9 F1 A0 00 00 01 02 33 BF E7 08 00 45 00 . . . . .3..E.
00010: 00 D4 E9 90 00 00 80 11 00 00 0A 00 00 16 0A 00 ._.....
00020: 00 18 01 85 04 04 00 C0 B6 57 30 84 00 00 00 9C .....[para]W0...
00030: 02 01 02 64 84 00 00 00 93 04 00 30 84 00 00 00 ...d..."..0...
00040: 8B 30 84 00 00 00 85 04 08 6E 65 74 6C 6F 67 6F 0.....netlogo
00050: 6E 31 84 00 00 00 75 04 73 17 00 00 00 FD 01 00 n1...u.s.....
00060: 00 48 44 82 88 4E 79 85 47 A8 CA 16 1D 55 23 B2 .HDNyG..U#
00070: E0 06 64 63 63 6C 61 62 05 6C 6F 63 61 6C 00 C0 .dcclab.local.
00080: 18 08 64 63 63 6C 61 62 32 32 C0 18 06 44 43 43 ..dcclab22..DCC
00090: 4C 41 42 00 08 44 43 43 4C 41 42 32 32 00 09 44 LAB..DCCLAB22..D
000A0: 43 43 4C 41 42 32 34 24 00 17 44 65 66 61 75 6C CCLAB24$.Default
000B0: 74 2D 46 69 72 73 74 2D 53 69 74 65 2D 4E 61 6D t-First-Site-Nam
000C0: 65 00 C0 50 05 00 00 00 FF FF FF FF 30 84 00 00 e.P....0..
000D0: 00 10 02 01 02 65 84 00 00 00 07 0A 01 00 04 00 .....e.....
000E0: 04 00 ..
```

If the client is communicating with a controller that is not in the client's site, the controller will also pass back the name of the client's proper site:

```
00000: 00 20 78 E0 AA 2B 00 20 78 01 80 69 08 00 45 00 . x+. x.i..E.
00010: 00 C9 FD A8 00 00 7F 11 28 64 0A 00 00 16 0B 00 ....(d.....
00020: 00 02 01 85 04 03 00 B5 C8 55 30 84 00 00 00 91 .....U0...'
00030: 02 01 01 64 84 00 00 00 88 04 00 30 84 00 00 00 ...d.....0...
00040: 80 30 84 00 00 00 7A 04 08 6E 65 74 6C 6F 67 6F 0...z..netlogo
00050: 6E 31 84 00 00 00 6A 04 68 17 00 00 00 7D 01 00 n1...j.h....}..
00060: 00 48 44 82 88 4E 79 85 47 A8 CA 16 1D 55 23 B2 .HDNyG..U#
00070: E0 06 64 63 63 6C 61 62 05 6C 6F 63 61 6C 00 C0 .dcclab.local.
```

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskas

```

00080:  18 08 64 63 63 6C 61 62 32 32 C0 18 06 44 43 43  ..dcclab22..DCC
00090:  4C 41 42 00 08 44 43 43 4C 41 42 32 32 00 0B 44  LAB..DCCLAB22..D
000A0:  43 43 52 4F 55 54 45 52 32 24 00 05 53 69 74 65  CCROUTER2$..Site
000B0:  32 00 05 53 69 74 65 31 00 05 00 00 00 FF FF FF  2..Site1.....
000C0:  FF 30 84 00 00 00 10 02 01 01 65 84 00 00 00 07  0.....e....
000D0:  0A 01 00 04 00 04 00  .....
```

In this case, the client sends another query to the DNS server asking for the list of controllers in this side. The following table shows an example of this. The client is looking for a domain controller in Site2 and switches to Site1 after the LDAP.

Frame	Source	Destination	Protocol	Description
1	Client	Server	DNS	Ox1:Std Qry for _ldap._tcp.Site2._sites.dc._msdcs.dcc lab.local.
2	Server	Client	DNS	Ox1:Std Qry Resp. for _ldap._tcp.Site2._sites.dc._msdcs.dcc lab.local
3	Client	Server	LDAP	ProtocolOp: SearchRequest (3)
4	Server	Client	LDAP	ProtocolOp: SearchResponse (4)
5	Client	Server	LDAP	ProtocolOp: SearchRequest (3)
6	Server	Client	LDAP	ProtocolOp: SearchResponse (4)
7	Client	Server	DNS	DNS Ox2:Std Qry for _ldap._tcp.Site1._sites.dc._msdcs.dcc lab.local.
8	Client	Server	DNS	Ox2:Std Qry Resp. for _ldap._tcp.Site1._sites.dc._msdcs.dcc lab.local

It is not necessary to have a domain controller in each site. Each domain controller checks all sites in a forest and the replication cost. A domain controller registers itself in any site that doesn't have a domain controller for its domain and for which its site has the lowest-cost connection. This process is also known as automatic site coverage. What this means is that clients will use the next domain controller that it has lowest cost to get to.

The default location process of the closest domain controller consists of 10 network packets and creates around 2,000 bytes of traffic.

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskas

Frame	Source	Destination	Protocol	Description
1	Client	Server	ARP_RARP	ICMP Echo: From 10.00.00.24 To 10.00.00.22
2	Server	Client	ARP_RARP	ICMP Echo Reply: To 10.00.00.24 From 10.00.00.22 10.0.0.22 10.0.0.24
3	Client	Server	DNS	0x1: Std Qry for _ldap._tcp.Default-First-Site-Name._sites.dc._msdcs.dcclab.local.
4	Client	Server	DNS	0x2: Std Qry for _ldap._tcp.Default-First-Site-Name._sites.dc._msdcs.dcclab.local.
5	Server	Client	DNS	0x1: Std Qry Resp. for _ldap._tcp.Default-First-Site-Name._sites.dc._msdcs.dcclab.
6	Server	Client	DNS	0x2: Std Qry Resp. for _ldap._tcp.Default-First-Site-Name._sites.dc._msdcs.dcclab.
7	Client	Server	LDAP	ProtocolOp: SearchRequest (3)
8	Server	Client	LDAP	ProtocolOp: SearchResponse (4)
9	Client	Server	LDAP	ProtocolOp: SearchRequest (3)
10	Server	Client	LDAP	ProtocolOp: SearchResponse (4)
11	Client	Server	ARP_RARP	ICMP Echo: From 10.00.00.24 To 10.00.00.22
12	Client	Server	ARP_RARP	ICMP Echo Reply: To 10.00.00.24 From 10.00.00.22 10.0.0.22 10.0.0.24

More details about the domain locator process can be found in the chapter "Name Resolution in Active Directory" of the Windows 2000 Resource Kit.

## Establishing a Secure Channel with the Domain Controller

When the client determines its site and domain controller, it can then create a secure channel with that domain controller. A secure channel is a connection between a domain member and a domain controller established to retrieve domain specific information, to update computer-specific information in the Active Directory, such as the computer password, and to validate the domain membership.

The process starts with a negotiation of the SMB dialect both parties will use. The SMB protocol has undergone many revisions and extensions since its release in 1984. This means the client and the server may not necessarily be using the same SMB dialect. In this case, both sides are Windows

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskis

2000, which uses the dialect NTLM 0.12. This dialect allows exchanges to use Unicode. Prior to this, exchanges were made in ASCII. The benefit of Unicode strings is they can include file names, resource names, and user names.

The client proceeds by connecting to the netlogon interface of the target. The End Port Mapper must be involved in this process in order to connect the client with the correct port on the server. Finally, the client sends three calls (NetrServerReqChallenge, NetrServerAuthenticate3, NetrLogonGetdomainInfo) to the interface. This process produces approximately 4,600 bytes of traffic.

Frame	Source	Destination	Protocol	Description
1	Client	Server	SMB	C negotiate, Dialect = NT LM 0.12
2	Server	Client	SMB	R negotiate, Dialect # = 5
3	Client	Server	MSRPC	c/o RPC Bind: UUID E1AF8308-5D1F-11C9-91A
4	Server	Server	MSRPC	c/o RPC Bind Ack: call 0x1 assoc grp 0xD52C
5	Client	Server	MSRPC	c/o RPC Request: call 0x1 opnum 0x3 contex
6	Server	Client	MSRPC	c/o RPC Response: call 0x1 context
7	Client	Server	MSRPC	c/o RPC Bind: UUID 12345678-1234-ABCD-EF0
8	Server	Client	MSRPC	c/o RPC Bind Ack: call 0x1 assoc grp 0x1C04B
9	Client	Server	R_LOGON	RPC Client call logon: NetrServerReqChallenge(..)
10	Server	Client	R_LOGON	RPC Server response logon: NetrServerReqChallenge()
11	Client	Server	R_LOGON	Error: Bad Opcode (Function does not exist)
12	Server	Server	R_LOGON	Error: Bad Opcode (Function does not exist)
13	Client	Client	MSRPC	c/o RPC Bind: UUID 12345678-1234-ABCD-EF0
14	Server	Client	MSRPC	c/o RPC Bind Ack: call 0x3 assoc grp 0x1C04B
15	Client	Client	R_LOGON	Error: Bad Opcode (Function does not

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskis

				exist)
16	Server	Client	R_LOGON	Error: Bad Opcode (Function does not exist)

**Note:** The current version of Netmon cannot resolve the calls NetrServerAuthenticate3 and NetrLogonGetdomainInfo correctly. In the previous table, these calls are shown as errors with a Bad Opcode.

**Important:** When an environment is mixed after an upgrade of a Windows NT 4 domain to Windows 2000, it is important to be aware of the following situation. When the only available domain controller for a Windows 2000 client to authenticate with is a Windows NT 4.0 backup domain controller, it will be unable to establish a secure channel. This is by design to increase security. Windows 2000 clients know what type of domain they belong to and will not downgrade their authentication method when setting a secure channel. The following trace sequence shows this scenario. It appears to be identical to the Windows 2000 client in a Windows NT 4.0 domain. The main appears after this because the client was unable to setup a secure channel, thus domain authentication fails.

## Kerberos Authentication and Session Creation

After the secure channel has been established, the client will retrieve all necessary tickets to establish an IPC\$ session with the controller. Because all Windows 2000 domain controllers are Kerberos Key Distribution Center (KDC), the client tries to detect the closest KDC in the same way it has already done it for the LDAP services.

The first step the client has to perform is the authentication in form of an AS ticket exchange. If this is successfully finished, it requests tickets for the controller (computer name\$) and the Kerberos service (krbtgt) that is running on the controller. The packet exchange produces approximately 8 kilobytes (KB). The actual size in a given environment depends on the number of Global and Universal groups that the client is a member of. Each additional group will add about 28 bytes to the total.

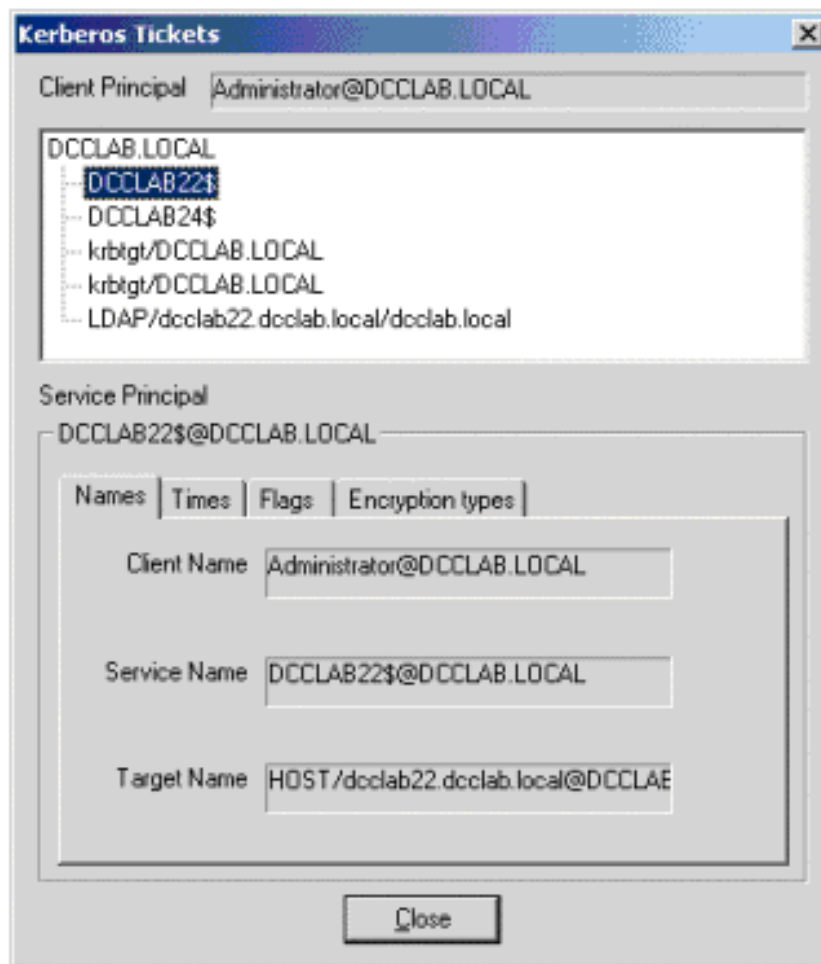
Frame	Source	Destination	Protocol	Description
1	Client	Server	SMB	DNS 0x3: Std Qry for _kerberos._tcp.Default-First-Site-Name._sites.dc._msdcs.DCCLAB.LOCAL. of type Srv Loc on class INET
2	Server	Client	SMB	DNS 0x3: Std Qry Resp. for _kerberos._tcp.Default-First-Site-Name._sites.dc._msdcs.DCCLAB.LOCAL. of type Srv Loc on class

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskas

3	Client	Server	LDAP	LDAP ProtocolOp: SearchRequest (3)
4	Server	Server	LDAP	LDAP ProtocolOp: SearchResponse (4)
5	Client	Server	Kerberos	Kerberos KRB_AS_REQ ( <b>request for TGT</b> )
6	Server	Client	Kerberos	Kerberos KRB_AS_REP
7	Client	Server	Kerberos	Kerberos KRB_TGS_REQ ( <b>request for DC\$</b> )
8	Server	Client	Kerberos	Kerberos KRB_TGS_REP
9	Client	Server	R_LOGON	Kerberos KRB_TGS_REQ ( <b>request for Kerberos Service</b> )
10	Server	Client	R_LOGON	Kerberos KRB_TGS_REP

All the tickets a client has obtained can be viewed with the Kerbray utility from the Microsoft Resource Kit.



# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskis

Finally, the client can connect to the IPC\$ share of the controller, which produces around 3,600 bytes of traffic.

Frame	Source	Destination	Protocol	Description
1	Client	Server	SMB	SMB C session setup & X
2	Server	Client	SMB	SMB R session setup & X
3	Client	Server	SMB	SMB C tree connect & X, Share = \\DCCLAB22.DCCLAB.LOCAL\IPC\$
4	Server	Client	SMB	SMB R tree connect & X, Type = IPC\$

## DFS Referral

The client then makes a Distributed File System (DFS) referral. A DFS referral is the DFS client-server protocol to get DFS-specific information that exists on the server to the client. It occurs whenever necessary. The general referral process is started when the client sends an SMB packet, indicating it is a DFS referral. The server passes this request to the DFS driver to complete the request. Subsequently, any access to network shares could result in a DFS referral request, if the client does not already have information about that share. Windows 2000 is a DFS version 5.0 client. This version allows caching of referrals to a DFS root or link for a (administrator configurable) specific length of time.

When the client starts up, a number of DFS referral requests are made from the client to one of the domain controllers within the client computer's domain that responds to the request. This process is required so that the client will be ready to handle any requests to domain-based DFS shares.

The first two requests serve to initialize the DFS client. The first is used to learn the names of all trusted Windows 2000 domains that could be accessed by the client. The second is used to obtain a list of domain controller in the domain order by local site first. The names returned contain both the Netbios name and DNS names of the domains.

The third request is made to obtain the actual server path to connect to a sysvol share.

The DFS referral creates a minimum of 394 bytes of traffic. The actual amount of traffic generated will depend on the number of trusted domains and DCs in the local domain returned in the reply.

By default the DFS referral to learn domain configuration is repeated every 15 minutes.

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskaskas

Frame	Source	Destination	Protocol	Description
1	Client	Server	SMB	C transact2 NT Get DFS Referral
2	Server	Client	SMB	R transact2 NT Get DFS Referral (response to frame 105)

The client then pings and makes an LDAP request to get the domain controller again.

Frame	Source	Destination	Protocol	Description
1	Client	Server	ICMP	Echo: From 10.00.00.100 To 10.00.00.22
2	Server	Client	ICMP	Echo Reply: To 10.00.00.100 From 10.00.00.22
3	Client	Server	UDP	Src Port: Unknown, (1041); Dst Port: Unknown (389); Length = 209 (0xD1)
4	Server	Client	UDP	Src Port: Unknown, (389); Dst Port: Unknown (1041); Length = 188 (0xBC)

## Name Translation

Each object in the Active Directory has a name. There are different formats for names available, such as the user principal names, distinguished names, and the earlier "domain\user" names from Windows NT. It is not necessary for a name to be a string. In general, everything that uniquely identifies an object can be considered as a name. Depending on the service that needs a name as parameter, it might be necessary to convert a given name from one format into another.

This is the objective of an API called DsCrackNames, which is used to map names from one format to another. Details about this call can be obtained from the Microsoft Developers Network (MSDN). Before a client can call this function, it has to bind a handle to the directory service with DSBind and if the operation is done it has to unbind from the directory with DSUnbind.

The following frames show the network traffic that comes along with the translation process. The entire process produces approximately 6,600 bytes of traffic.

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskas

Frame	Source	Destination	Protocol	Description
1	Client	Server	MSRPC	c/o RPC Bind: UUID E1AF8308-5D1F-11C9-91A4-08002B14A0FA call 0
2	Server	Client	MSRPC	c/o RPC Bind Ack: call 0x1 assoc grp 0xD52D xmit 0x16D0 recv 0x1
3	Client	Server	MSRPC	c/o RPC Request: call 0x1 opnum 0x3 context 0x0 hint 0x84
4	Server	Client	MSRPC	c/o RPC Response: call 0x1 context 0x0 hint 0x80 cancels 0x0
5	Client	Server	MSRPC	c/o RPC Bind: UUID E3514235-4B06-11D1-AB04-00C04FC2DCD2 call 0
6	Server	Client	MSRPC	c/o RPC Bind Ack: call 0x1 assoc grp 0x1C04C xmit 0x16D0 recv 0x
7	Client	Server	MSRPC	c/o RPC Alt-Cont: UUID E3514235-4B06-11D1-AB04-00C04FC2DCD2 call 0
8	Server	Client	MSRPC	c/o RPC Alt-Cont Rsp: call 0x1 assoc grp 0x1C04C xmit 0x16D0 recv 0x
9	Client	Server	MSRPC	c/o RPC Request: call 0x1 opnum 0x0 context 0x0 hint 0x38
10	Server	Client	MSRPC	c/o RPC Response: call 0x1 context 0x0 hint 0x3C cancels 0x0
11	Client	Server	MSRPC	c/o RPC Request: call 0x2 opnum 0xC context 0x0 hint 0x6E
12	Server	Client	MSRPC	c/o RPC Response: call 0x2 context 0x0 hint 0xB4 cancels 0x0
13	Client	Server	MSRPC	c/o RPC Request: call 0x3 opnum 0xC context 0x0 hint 0x6E
14	Server	Client	MSRPC	c/o RPC Response: call 0x3 context 0x0 hint 0xAC cancels 0x0
15	Client	Server	MSRPC	c/o RPC Request: call 0x4 opnum 0x1 context 0x0 hint 0x14
16	Server	Client	MSRPC	c/o RPC Response: call 0x4 context 0x0 hint 0x18 cancels 0x0

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskas

## LDAP RootDSE

The client then requests information from the LDAP RootDSE. The RootDSE is a standard attribute defined in the LDAP 3.0 specification. The RootDSE contains information about the directory server, including its capabilities and configuration. The search response will contain a standard set of information as defined in RFC 2251. One of the items returned with this response is the supported Simple Authentication and Security Layer (SASL) mechanism. In this case it returns GSS-SPNEGO.

Frame	Source	Destination	Protocol	Description
1	Client	Server	LDAP	ProtocolOp: SearchRequest (3)
2	Server	Client	LDAP	ProtocolOp: SearchResponse (4)
3	Client	Server	Kerberos	KRB_TGS_REQ
4	Server	Client	Kerberos	KRB_TGS_REP
5	Client	Server	LDAP	ProtocolOp: BindRequest (0)
6	Server	Client	LDAP	ProtocolOp: BindResponse (1)

## Load Group Policy

Next, the computer loads applicable Group Policy objects. The client then completes an RPC call to convert its name to a distinguished name and performs an LDAP lookup for policy information that applies to this particular computer and then loads that information using Server Message Block (SMB).

## Policy Search

The following frames show the client performing a binding operation to the LDAP directory. LDAP queries require the client to bind to the Directory Service before making a search for information. At this stage of the client logon process, the client is binding to the Active Directory to make a search for Group Policies that apply to the client. This sequence also shows the client making an LDAP request to determine what Group Policies apply. Each bind operation creates about 1,675 bytes of traffic. The policy search creates about 3,527 bytes of traffic in this case.

Frame	Source	Destination	Protocol	Description
1	Client	Server	LDAP	ProtocolOp: BindRequest (0)
2	Server	Client	LDAP	ProtocolOp: BindResponse (1)
3	Client	Server	LDAP	ProtocolOp: BindRequest (0)

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskas

4	Server	Client	LDAP	ProtocolOp: BindResponse (1)
5	Client	Server	TCP	.AP..., len: 173, seq: 978423034-978423207, ack: 3068556899, win: 17069, src: 1048 dst: 389
6	Server	Client	TCP	AP..., len: 294, seq: 3068556899-3068557193, ack: 978423207, win: 16081, src: 389 dst: 1048
7	Client	Server	TCP	....S., len: 0, seq: 978497639-978497639, ack: 0, win: 16384, src: 1050 dst: 389
8	Server	Client	TCP	.A..S., len: 0, seq: 3068641675-3068641675, ack: 978497640, win: 17520, src: 389 dst: 1050
9	Client	Server	TCP	.A...., len: 0, seq: 978497640-978497640, ack: 3068641676, win: 17520, src: 1050 dst: 389
10	Client	Server	LDAP	ProtocolOp: BindRequest (0)
11	Server	Client	LDAP	ProtocolOp: BindResponse (1)
12	Client	Server	TCP	.AP..., len: 129, seq: 978498933-978499062, ack: 3068642008, win: 17188, src: 1050 dst: 389
13	Server	Client	TCP	.AP..., len: 171, seq: 3068642008-3068642179, ack: 978499062, win: 16098, src: 389 dst: 1050
14	Client	Server	TCP	.AP..., len: 203, seq: 978499062-978499265, ack: 3068642179, win: 17017, src: 1050 dst: 389
15	Server	Client	TCP	.AP..., len: 201, seq: 3068642179-3068642380, ack: 978499265, win: 17520, src: 389 dst: 1050
16	Client	Server	TCP	.AP..., len: 467, seq: 978423207-978423674, ack: 3068557193, win: 16775, src: 1048 dst: 389
17	Server	Client	TCP	.AP..., len: 1273, seq: 3068557193-3068558466, ack: 978423674, win: 17520, src: 389 dst: 1048

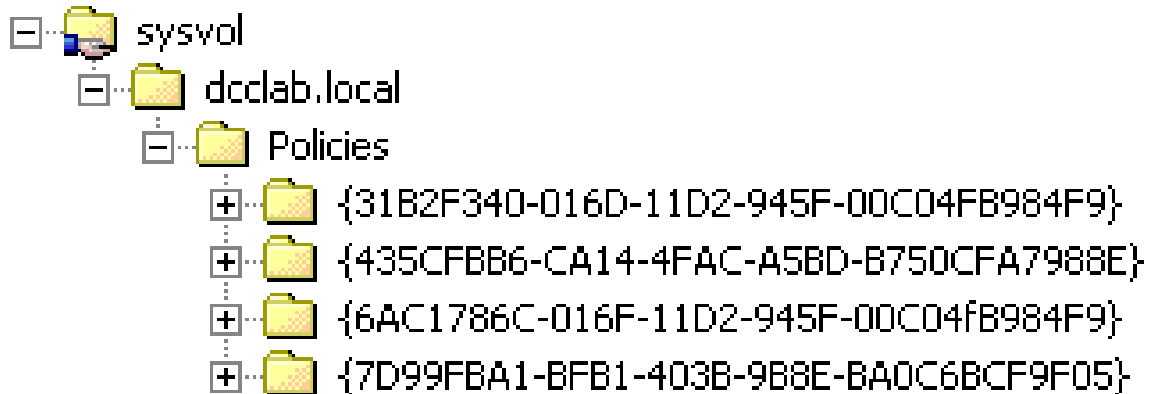
After the client determines what policies are applicable, it makes a second DFS referral. This will occur ahead of most attempts by a client in Windows 2000 to connect to a share point.

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskas

## Policy Load Using SMB

The part completes with the client connecting to the SYSVOL (a standard share point on a domain controller) on the domain controller and downloads its policies generating 1,018 bytes of traffic.



This is a very simple configuration, so this number will grow with more sophisticated Group Policy implementations.

Frame	Source	Destination	Protocol	Description
1	Client	Server	SMB	C tree connect & X, Share = \\DCCLAB22.MAIN.LOCAL\SYSVOL
2	Server	Client	SMB	R tree connect & X, Type = _
3	Client	Server	SMB	C NT create & X, File = \main.local\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\gpt.ini
4	Server	Client	SMB	R NT create & X, FID = 0x4000
5	Client	Server	SMB	C read & X, FID = 0x4000, Read 0x1a at 0x00000000
6	Server	Client	SMB	R read & X, Read 0x1a

## Client Certificate AutoEnrollment

Each time Group Policy objects are applied, the client completes the autoenrollment event. The autoenrollment event does the following:

- Checks the status of the computer's certificates, and if they are not OK, the client autoenrolls the client

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskis

- Downloads the enterprise's certification authority (CA) certificates from the Active Directory enterprise root store (= PKI trust anchors)
- Downloads certificates of CAs capable of issuing smart card certificates from Active Directory

The client makes a request to get the LDAP RootDSE (see frames above) information and then uses LDAP to complete autoenrollment.

Frame	Source	Destination	Protocol	Description
1	Client	Server	DNS	DNS 0x3: Std Qry for _ldap._tcp.Site2._sites.dc._msdcs
2	Server	Client	DNS	DNS 0x3: Std Qry Resp. Auth. NS is dcclab.local.
3	Client	Server	DNS	DNS 0x4: Std Qry for _ldap._tcp.dc._msdcs.dcclab22.dcc
4	Server	Client	DNS	DNS 0x4: Std Qry Resp. Auth. NS is dcclab.local. of
5	Client	Server	LDAP	ProtocolOp: BindRequest (0)
6	Server	Client	LDAP	ProtocolOp: BindResponse (1)
7	Client	Server	LDAP	ProtocolOp: BindRequest (0)
8	Server	Client	LDAP	ProtocolOp: BindResponse (1)
9	Client	Server	LDAP	ProtocolOp: SearchRequest (3)
10	Server	Client	LDAP	ProtocolOp: SearchResponse (simple) (5)
11	Client	Server	LDAP	ProtocolOp: UnbindRequest (2)
12	Client	Server	LDAP	ProtocolOp: BindRequest (0)
13	Server	Client	LDAP	ProtocolOp: BindResponse (1)
14	Client	Server	LDAP	ProtocolOp: SearchRequest (3)
15	Server	Client	LDAP	ProtocolOp: SearchResponse (simple) (5)
16	Client	Server	LDAP	ProtocolOp: UnbindRequest (2)
17	Client	Server	LDAP	ProtocolOp: UnbindRequest (2)

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskis

Closer examination of frame 3 in the previous frame reveals that this is a request for configuration information about the public key services in the domain. The following example provides a more detailed view of one of the frames.

```
LDAP: ProtocolOp: SearchRequest (3)
LDAP: ProtocolOp = SearchRequest
LDAP: Base Object = CN=Public Key
Services,CN=Services,CN=Configuration,DC=dcclab,DC
LDAP: Scope = Single Level
LDAP: Deref Aliases = Never Deref Aliases
LDAP: Size Limit = No Limit
LDAP: Time Limit = 0x00002710
LDAP: Attrs Only = 0 (0x0)
LDAP: Filter Type = Equality Match
LDAP: Attribute Type = cn
LDAP: Attribute Value = NTAAuthCertificates
LDAP: Attribute Value = cACertificate
```

## Time Synchronization

Next, the client updates its time with its authenticating domain controller. The following set of frames shows the time synchronization process. Notice that this occurs on port 123. This sequence creates 220 bytes of network traffic.

Frame	Source	Destination	Protocol	Description
1	Client	Server	NTP	Src Port: Unknown, (1051); Dst Port: Network Time Protocol (123); Length = 76 (0x4C)
2	Server	Client	NTP	Src Port: Network Time Protocol, (123); Dst Port: Unknown (1051); Length = 76 (0x4C)

## Dynamic Domain Name System Update

The last part of the startup process is for the client to perform its name update in the DNS database. The Windows 2000 dynamic DNS update is based on RFC 2136. This process is based on RFC 2136.

The client first determines whether the DNS server has the authority for the client's zone. This sequence creates 225 bytes of traffic.

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskas

Frame	Source	Destination	Protocol	Description
1	Client	Server	DNS	0x1: Std Qry for dcclab24.main.local. of type SOA on class INET addr.
2	Server	Client	DNS	0x1: Std Qry Resp. Auth. NS is main.local. of type SOA on class INET addr. : Name does not exist

Next, the client makes the dynamic update of its name in the DNS server. This creates about 1,800 bytes of traffic if the client has to update both (A RR and PTR RR).

Frame	Source	Destination	Protocol	Description
1	Client	Server	DNS	DNS 0x4: Dyn Upd PRE records to dcclab24.dcclab.local.
2	Server	Client	DNS	241 99.703125 00010233BFE7 INTEL F1A000 DNS 0x4: Dyn Upd Resp. PRE records to dcclab24.dcclab
3	Client	Server	DNS	DNS 0x5: Std Qry for 0.0.10.in-addr.arpa. of type SOA
4	Server	Client	DNS	0x5: Std Qry Resp. for 0.0.10.in-addr.arpa. of type SOA
5	Client	Server	DNS	0x6: Dyn Upd PRE/UPD records to 24.0.0.10.in-addr.arpa
6	Server	Client	DNS	0x6: Dyn Upd Resp. PRE/UPD records to 24.0.0.10.in-addr.arpa

The actual size of the packets for the dynamic update depends on many conditions. First of all, it depends on whether the client was already registered. Next, it depends in whether there is a conflict with an already registered entry. Another aspect for the client traffic is whether DHCP is in use. The default configuration in this scenario is that the client is updating its A RR, whereas the DHCP server is responsible for the PTR RR.

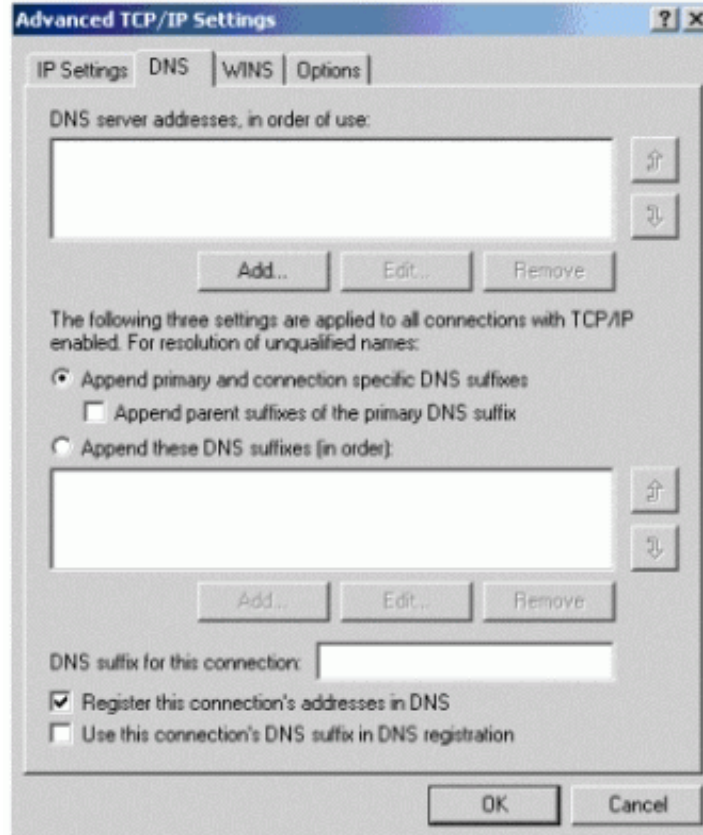
Last but not least, is the traffic also depending on the configuration of the DNS server and how the secure dynamic update behavior is configured.

**Note:** If DDNS is not being used in the environment, you should consider turning off the client's ability to make dynamic updates. Turning this off will save the network traffic generated by the unneeded

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskas

attempts to make the update by the client in an environment that does not support it. This feature is turned off in the advanced TCP/IP properties for the particular network connection. The location is shown in the following illustration:



## Completion

The computer startup sequence is completed with the client breaking down its open connections to the domain controller in a sequence similar to the one illustrated in the following table. This creates 473 bytes of traffic.

Frame	Source	Destination	Protocol	Description
1	Client	Server	SMB	C tree disconnect
2	Server	Client	SMB	R tree disconnect
3	Client	Server	SMB	C tree disconnect
4	Server	Client	SMB	R tree disconnect
5	Client	Server	SMB	C logoff & X
6	Server	Client	SMB	R logoff & X

# Windows 2000 Startup and Logon Traffic Analysis

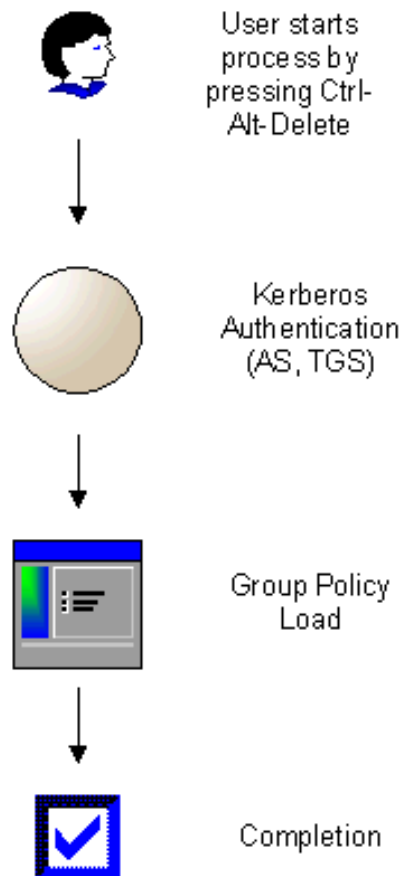
By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskas

This traffic sequence is useful to help determine the break between the computer startup and a user logon in a network trace. The system is now available for use. The system display will show the Ctrl-Alt-Delete screen.

## User Logon Overview

After the system startup is complete, the Ctrl-Alt-Delete screen appears on the console. A user will be able to make a domain logon from this system. Pressing Ctrl-Alt-Delete and entering a valid set of domain user credentials (user name and password) initiates the interactive client logon process that ends with the Windows NT shell being loaded and the user being able to interactively use the systems. It is important to note that the user logon process is essentially an abbreviated version of the computer startup process using a subset of the processes described previously. The following diagram shows this process.

### Windows 2000 Client Logon Process



# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskaskas

## Logon Flow

### User Identification

Windows 2000 provides three ways for a user to enter account information at logon. The first method is to use the Security Accounts Manager (SAM) account name and select the domain. This is the default method for logon. The second method is to use the fully qualified name, which would appear as `<user>@<domain-org-company-com>`. The third alternative is use the User Principle Name (UPN), which is described in the Windows 2000 Resource Kit as follows:

A user principal name (UPN) is an e-mail-like name that uniquely represents a user. A UPN consists of two parts, a user identification portion and a domain portion. The two parts are separated by an "@" symbol, to form `<user>@<DNS-domain-name>`, for example, `liz@noam.reskit.com`. Every user is automatically assigned a default UPN, where the `<user>` portion of the name is the same as the user's logon name, and the `<DNS-domain-name>` portion of the name is the DNS name of the Active Directory domain where the user account is located. When logging on using a UPN, users no longer have to choose a domain from a list on the logon dialog box.

You can set UPNs to arbitrary values. For example, even if Liz's account is in the `noam.reskit.com` domain, her UPN could be set to `liz@reskit.com`. When the user logs on, the user account to be validated is discovered by searching the global catalog for a user account with a matching UPN value. By making UPN values independent from domain names, administrators can move user accounts between domains, leaving UPN values unchanged and making interdomain moves more transparent to users."

UPN names are resolved to a user and domain by performing a domain controller lookup of the Global Catalog bnhhhhh (GC). UPN logon is only supported when the domain is in Native Mode.

### Kerberos Authentication for User Logon

The user logon operation generates a Kerberos Authentication request to get its session ticket. Then logon process then requests a session key for a service from the KDC via the Ticket Granting Service.

Frame	Source	Destination	Protocol	Description
1	Client	Server	Kerberos	KRB_AS_REQ
2	Server	Client	Kerberos	KRB_AS_REP
3	Client	Server	Kerberos	KRB_TGS_REQ
4	Server	Client	Kerberos	KRB_TGS_REP

The logon process requests the following tickets during the user logon process:

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskas

- Kerberos: Server Name = <clientname>\$
- Kerberos: Server Name = <dc name>\$
- Kerberos: Server Name = krbtgt.<dns domain name>
- Kerberos: Server Name = ldap.<dc name>.<dns domain name>

As mentioned earlier, the size of the Kerberos packets depends on the number of groups a user is a member of.

## Group Policy Load

The process of retrieving Group Policy information is the same as previously described in the computer startup section. The client uses the "DS API DSCrackNames" to perform a name translation and retrieves the information about the policies to load via LDAP.

```

000000A0 00 00 00 5D 04 5B 5B 4C 44 41 50 3A 2F 2F 43 4B  ...  .  .  .  LDAP://CN
000000B0 3D 7B 33 31 42 32 46 33 34 30 2D 30 31 36 44 2D  =| 31B2F340-016D-
000000C0 31 31 44 32 2D 39 34 35 46 2D 30 30 43 30 34 46  11D2-945F-00C04F
000000D0 42 39 38 34 46 39 7D 2C 43 4B 3D 50 6F 6C 69 69  B984F9} ,CN=Polie
000000E0 69 65 73 2C 49 4E 5D 53 79 79 74 65 6D 2C 44 43  ses ,CN=System,DC
000000F0 8D 6D 61 69 6E 2C 44 43 3D 6C 6F 6B 61 6C 38 30  =main,DC=local,0
00001000 5D 3D 84 00 00 00 3E 02 01 1D 73 84 00 00 00 36  lDd...>...s&...5
00001100 04 33 6C 64 61 70 3A 2F 2F 6D 61 69 6E 2E 6C 6F  . 3ldap://main.lo
00001200 63 61 6C 2F 43 4E 3D 43 6F 6B 66 69 67 75 72 61  cal/CN=Configura
00001300 74 69 68 6E 2C 44 43 3D 6D 61 69 6E 2C 44 43 3D  tion,DC=main,DC=
00001400 6C 6F 69 61 6C 30 84 00 00 00 10 02 01 1D 65 84  local0a.....sa

```

The client then establishes an SMB connection to the controller and downloads the necessary policy files.

Frame	Source	Destination	Protocol	Description
1	Client	Server	SMB	C negotiate, Dialect = NT LM 0.12
2	Server	Client	SMB	R negotiate, Dialect # = 5
3	Client	Server	SMB	C session setup & X
4	Server	Client	SMB	R session setup & X
5	Client	Server	SMB	C tree connect & X, Share = \\DCCLAB22.MAIN.LOCAL\SYSVOL
6	Server	Client	SMB	R tree connect & X, Type = _
7	Client	Server	SMB	C NT create & X, File = \main.local\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\gpt.ini

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskis

8	Server	Client	SMB	R NT create & X
9	Client	Server	SMB	C read & X, FID = 0x8005, Read 0x1a at 0x00000000
10	Server	Client	SMB	R read & X, Read 0x1a

This sequence shows the client connecting to the system volume and loading the user's policy. Information on the traffic generated by the loading of group policy information can be found in Chapter 5 of the Microsoft Press book *Building Enterprise Active Directory Services*, in the *Notes from the Field* series.

## Completion

The client then closes its connection to the domain controller. This happens on Port 445. It is represented in the trace like the following table.

Frame	Source	Destination	Protocol	Description
1	Client	Server	SMB	SMB C tree disconnect
2	Server	Client	SMB	SMB R tree disconnect
3	Client	Server	SMB	SMB C logoff & X
4	Server	Client	SMB	SMB R logoff & X

The user has now logged on. The following table shows a summary of the network traffic for a user logon.

Protocol	Frames	Bytes Claimed
SMB	16	3070
ICMP	4	114
UDP	12	96
NBT	17	1164
TCP	86	6019
MSRPC	16	3872
LDAP	4	3226

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskis

Kerberos	12	14229
----------	----	-------

**Note:** Please keep in mind that the traffic in the previous table represents a user who is just a member of the default groups and that no specific Group Policies were set.

## Conclusion

We have now provided a detailed examination of the Windows 2000 computer startup and user logon process. As stated in the introduction, understanding this process will assist with both infrastructure design and systems administration in Windows 2000 networks.

This document covers a great deal of material and should probably be read multiple times to fully understand and kept handy as a reference.

To help cement your understanding of the concepts described in the paper, I would suggest setting up a test environment and using network monitor to make some traces of the process. Use this document as a reference when examining the traces to understand what is happening at each point.

Questions and comments can be directed to [gmolnar@microsoft.com](mailto:gmolnar@microsoft.com)

## Appendix A: Test Environment

The following system configurations were used to validate the Windows 2000 startup and logon process.

Environment	Server	Client
Windows 2000	1 Windows 2000 domain controller in Domain Main.Local running DNS DHCP WINS	1 Windows 2000 Professional Desktop 1 Windows 2000 Professional Notebook
NT 4	1 NT 4 SP6a domain controller running DHCP WINS	1 Windows 2000 Professional Desktop
Mixed	1 upgraded Windows 2000 domain controller running DNS DHCP WINS 1 NT 4.0 SP6a BDC	1 Windows 2000 Professional Desktop
Forest	1 Windows 2000 Server Domain Controller in Domain Corp.Main.Local running: DNS DHCP	1 Windows 2000 Professional Desktop in Domain Corp.Main.Local 1 Windows 2000 Professional Desktop in Domain Field.Main.Local

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskas

2 Windows 2000 Servers configured as RRAS routers using a Null Modem Cable to simulate a slow link 1 Windows 2000 Domain Controller in Domain Field.Main.Local 1 Windows 2000 Domain Controller in Domain Field.Main.Local running: DNS (Secondary) DHCP
--

All computers (except where noted) were networked using a 10/100 Ethernet hub. The clients used various Ethernet cards. All clients were connected to the network at 100 MB.

All network traces were made using Network Monitor v5.00.646. A parser for Kerberos traffic was added to Network Monitor.

All network tests and configurations discussed in this document use the TCP/IP transport protocol. TCP/IP is the default network protocol for Windows 2000 and many of the services that logon and startup use require TCP/IP.

## Appendix B: TCP/IP Ports Used in the Authentication Process

The following table is a comprehensive list of ports used by Windows.

Port	TCP/UDP	Function Description
20	TCP	FTP
21	TCP	FTP
23	TCP	Telnet
25	TCP	IIS SMTP
31	TCP	Netmeeting
42	TCP	WINS Replication
52	TCP	Netmeeting
53	UDP	DNS Name Resolution SQL TCP lookup
53	TCP	DNS SQL TCP lookup
67	UDP	DHCP Lease (BOOTP)

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskas

68	UDP	DHCP Lease
80	TCP	IIS HTTP
88	UDP	Kerberos
88	TCP	Kerberos
110	TCP	POP3
119	TCP	NNTP
135	TCP	<b>Location Service</b> RPC RPC EP Mapper SQL RPC session mapper WINS Manager DHCP Manager MS DTC
137	UDP	<b>NetBIOS Name Service</b> SQL RPC Lookup Logon Sequence NT 4.0 Trusts NT 4.0 Secure Channel Pass Through Validation Browsing Printing SQL Named Pipes lookup
137	TCP	WINS Registration
138	UDP	NetBIOS Datagram Service Logon Sequence NT 4.0 Trusts NT 4.0 Directory Replication NT 4.0 Secure Channel Pass Through Validation NetLogon Browsing Printing
139	TCP	NetBIOS Session Service NBT SMB File Sharing Printing SQL Named Pipes session Logon Sequence NT 4.0 Trusts NT 4.0 Directory Replication NT 4.0 Secure Channel Pass Through Validation NT 4.0 Administration Tools (Server Manager, User Manager, Event Viewer, Registry Editor, Diagnostics, Performance Monitor, DNS Administration)
161	UDP	SNMP

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskas

162	UDP	SNMP Trap
215	TCP	Netmeeting
389	TCP	LDAP
443	TCP	HTTP SSL
445	TCP	SMB or CIFS
464	UDP	Kerberos kpasswd
500	UDP	IPSEC isakmp IKE
531	TCP	IRC
560	TCP	Content Replication Service Site Server
636		LDAP over SSL
731	TCP	Netmeeting
Dynamic	UDP	Netmeeting
888	TCP	Login and Environment Passing
Dynamic	TCP	Directory Replication
1109	TCP	POP with Kerberos
1433	TCP	SQL TCP session
1645	UDP	RADIUS Authentication
1646	UDP	RADIUS Accounting
1723	TCP	PPTP Control Channel (IP Protocol 47 GRE)
1755	TCP	Netshow
Dynamic	UDP	Netshow
1812	UDP	RADIUS Authentication
1813	UDP	RADIUS Accounting
1863	TCP	MSN Messenger
2053	TCP	Kerberos de-multiplexor
2105	TCP	Kerberos encrypted rlogin
3268		Global Catalog LDAP

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskas

3269		Global Catalog LDAP over SSL
3389	RDP	Terminal Services
8000	TCP	CyberCash (credit gateway)
8001	TCP	CyberCash (admin)
8002	TCP	CyberCash (coin gateway)
10140-10179	TCP	DCOM port range

For all the ports on Windows NT, look on your local computer:

`%windnt%/system32/drivers/etc/services`

The following table lists common ports.

<b>Authentication</b>	<b>Authentication services verify the identity of a user or device requesting access to a resource.</b>		
	SERVICE		TYPE
	AFS/Kerberos authentication service		TCP Port 7004 - afs3-kaserver
	AFS/Kerberos authentication service		UDP Port 7004 - afs3-kaserver
	Authentication Service		TCP Port 113 - ident
	Authentication Service		UDP Port 113 - ident
	Certificate Distribution Center		TCP Port 223 - cdc
	Certificate Distribution Center		UDP Port 223 - cdc
	Funk Software Inc.		TCP Port 1505 - funkproxy
	Funk Software Inc.		UDP Port 1505 - funkproxy

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskas

	Login Host Protocol (TACACS)	TCP Port 49 - bbn-login
	Login Host Protocol (TACACS)	UDP Port 49 - bbn-login
	TACACS-Database Service	TCP Port 65 - tacacs-ds
	TACACS-Database Service	UDP Port 65 - tacacs-ds

<b>Directory Service/Name Resolution Directory Services provide name resolution and lookup capabilities, allowing users or devices to locate resources on the network by human readable or well-known names.</b>		
	SERVICE	TYPE
	AppleTalk Name Binding	TCP Port 202 - at-nbp
	AppleTalk Name Binding	UDP Port 202 - at-nbp
	Directory Location Service	TCP Port 197 - dls
	Directory Location Service	UDP Port 197 - dls
	Directory Location Service Monitor	TCP Port 198 - dls-mon
	Directory Location Service Monitor	UDP Port 198 - dls-mon
	Lightweight Directory Access Protocol	TCP Port 389 - ldap
	Lightweight Directory Access Protocol	UDP Port 389 - ldap
	Microsoft-DS	TCP Port 445 - microsoft-ds
	Microsoft-DS	UDP Port 445 - microsoft-ds
	Microsoft's Windows Internet Name Service	TCP Port 1512 - wins

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskas

	Microsoft's Windows Internet Name Service	UDP Port 1512 - wins
	NETBIOS Name Service	TCP Port 137 - netbios-ns
	NETBIOS Name Service	UDP Port 137 - netbios-ns
	NIC Host Name Server	TCP Port 101 - hostnames
	NIC Host Name Server	UDP Port 101 - hostnames
	Prospero Directory Service non-priv	TCP Port 1525 - prospero-np
	Prospero Directory Service non-priv	UDP Port 1525 - prospero-np
	Domain Name Server	TCP Port 53 - domain
	Domain Name Server	UDP Port 53 - domain
	Host Name Server	TCP Port 42 - nameserver
	Host Name Server	UDP Port 42 - nameserver
	HOSTS2 Name Server	TCP Port 81 - hosts2-ns
	HOSTS2 Name Server	UDP Port 81 - hosts2-ns
	streettalk	TCP Port 566 - streettalk
	streettalk	UDP Port 566 - streettalk

Encryption		
	SERVICE	TYPE
	Kerberos	TCP Port 750 - kerberos-sec
	Kerberos	TCP Port 751 - kerberos_master
	Kerberos	TCP Port 88 - kerberos

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskas

	Kerberos	UDP Port 750 - kerberos-sec
	Kerberos	UDP Port 751 - kerberos_master
	Kerberos	UDP Port 88 - kerberos
	kerberos administration	TCP Port 749 - kerberos-adm
	kerberos administration	UDP Port 749 - kerberos-adm
	Kerberos Key Distribution Center	Windows NT Service - Kerberos Key Distribution Center
	kerberos-master	TCP Port 751 - kerberos-master
<p><b>Remote Access/VPN Remote Access &amp; VPN services allow users or devices to access remote networks as though they had local connections to that network. This is different from Remote Control Software where users actually assume control of a host on a remote network.</b></p>		
	SERVICE	TYPE
	any private dial out service	TCP Port 75 -
	any private dial out service	UDP Port 75 -
	Apple Remote Access Protocol	TCP Port 3454 - mira
	IPSEC driver	Windows NT Service - IPSEC driver
	pptp	TCP Port 1723 - PPTP
	Routing and Remote Access	Windows NT Service - Routing and Remote Access
	Shiva	TCP Port 1502 - shivadiscovery
	Shiva	UDP Port 1502 - shivadiscovery
	TIA/EIA/IS-99 modem server	TCP Port 380 - is99s
	TIA/EIA/IS-99 modem server	UDP Port 380 - is99s

# Windows 2000 Startup and Logon Traffic Analysis

By Greg Molnar, Keith Olinger, David Trulli, and Markus Vilcinskas

<p>Routing Routing protocols allow for the transmission of information between networks. TCP/IP is omitted from this list as it is assumed to be running on all hosts on the network. Protocols other than TCP/IP are important to note as they may indicate extranet support for different types of client operating systems and/or network configurations.</p>		
	SERVICE	TYPE
	AppleTalk Protocol	Windows NT Service - AppleTalk Protocol
	AppleTalk Routing Maintenance	TCP Port 201 - at-rtmp
	AppleTalk Routing Maintenance	UDP Port 201 - at-rtmp
	Appletalk Update-Based Routing Pro.	TCP Port 387 - aurp
	Appletalk Update-Based Routing Pro.	UDP Port 387 - aurp
	AppleTalk Zone Information	TCP Port 206 - at-zis
	AppleTalk Zone Information	UDP Port 206 - at-zis
	Border Gateway Protocol	TCP Port 179 - bgp
	Border Gateway Protocol	UDP Port 179 - bgp
	IPX	TCP Port 213 - ipx
	IPX	UDP Port 213 - ipx
	Local routing process (on site)	UDP Port 520 - router