

# Kerberos, DNS, LDAP, MSRPC

While Windows operating systems use a variety of protocols to communicate, Active Directory specifically requires [Lightweight Directory Access Protocol \(LDAP\)](#), Microsoft's version of [Kerberos](#), [DNS](#) for authentication and communication, and [MSRPC](#) which is the Microsoft implementation of [Remote Procedure Call \(RPC\)](#), an interprocess communication technique used for client-server model-based applications.

## Kerberos

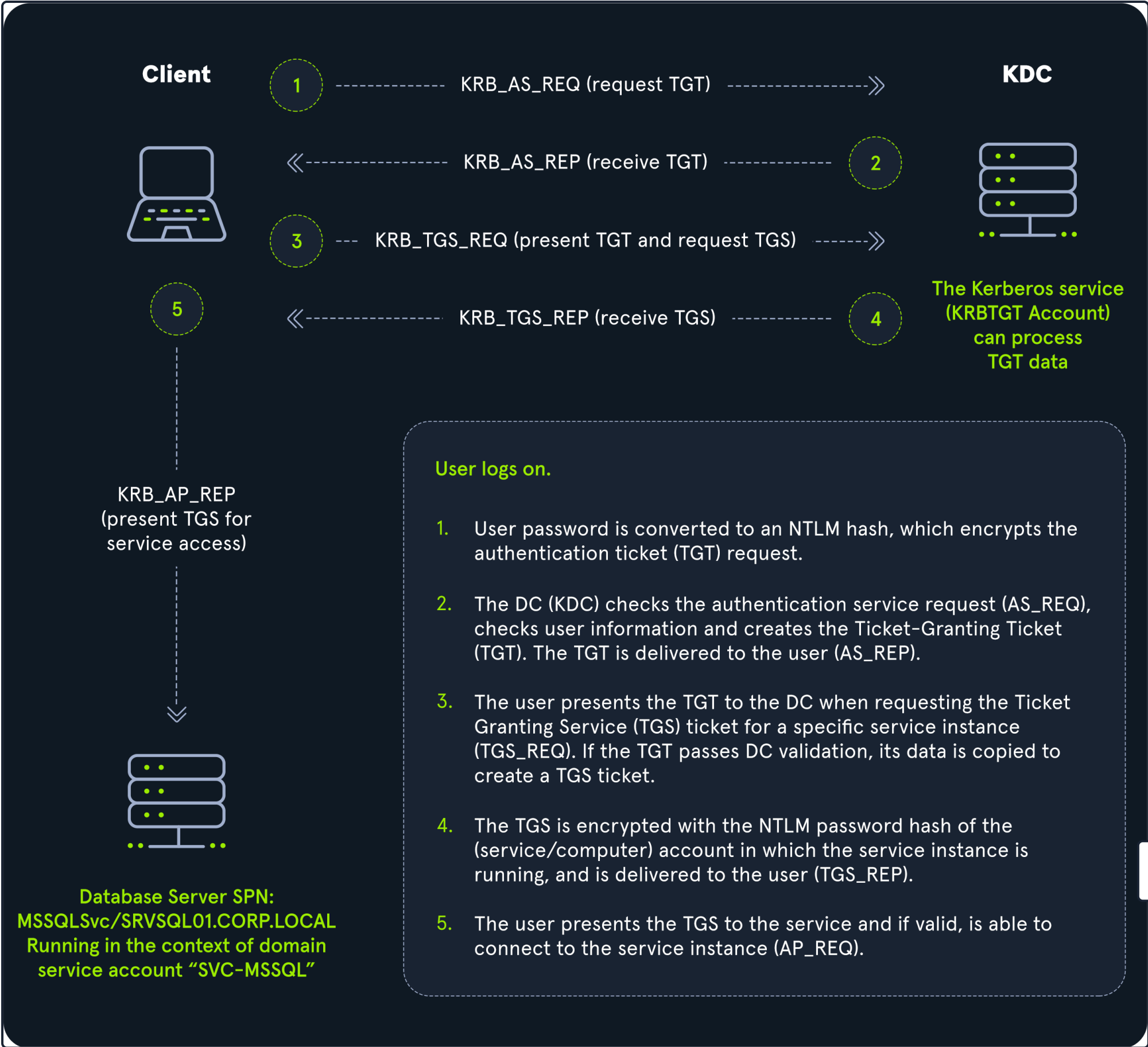
Kerberos has been the default authentication protocol for domain accounts since Windows 2000. Kerberos is an open standard and allows for interoperability with other systems using the same standard. When a user logs into their PC, Kerberos is used to authenticate them via mutual authentication, or both the user and the server verify their identity. Kerberos is a stateless authentication protocol based on tickets instead of transmitting user passwords over the network. As part of Active Directory Domain Services (AD DS), Domain Controllers have a Kerberos Key Distribution Center (KDC) that issues tickets. When a user initiates a login request to a system, the client they are using to authenticate requests a ticket from the KDC, encrypting the request with the user's password. If the KDC can decrypt the request (AS-REQ) using their password, it will create a Ticket Granting Ticket (TGT) and transmit it to the user. The user then presents its TGT to a Domain Controller to request a Ticket Granting Service (TGS) ticket, encrypted with the associated service's NTLM password hash. Finally, the client requests access to the required service by presenting the TGS to the application or service, which decrypts it with its password hash. If the entire process completes appropriately, the user will be permitted to access the requested service or application.

Kerberos authentication effectively decouples users' credentials from their requests to consumable resources, ensuring that their password isn't transmitted over the network (i.e., accessing an internal SharePoint intranet site). The Kerberos Key Distribution Centre (KDC) does not record previous transactions. Instead, the Kerberos Ticket Granting Service ticket (TGS) relies on a valid Ticket Granting Ticket (TGT). It assumes that if the user has a valid TGT, they must have proven their identity. The following diagram walks through this process at a high level.



## Kerberos Authentication Process

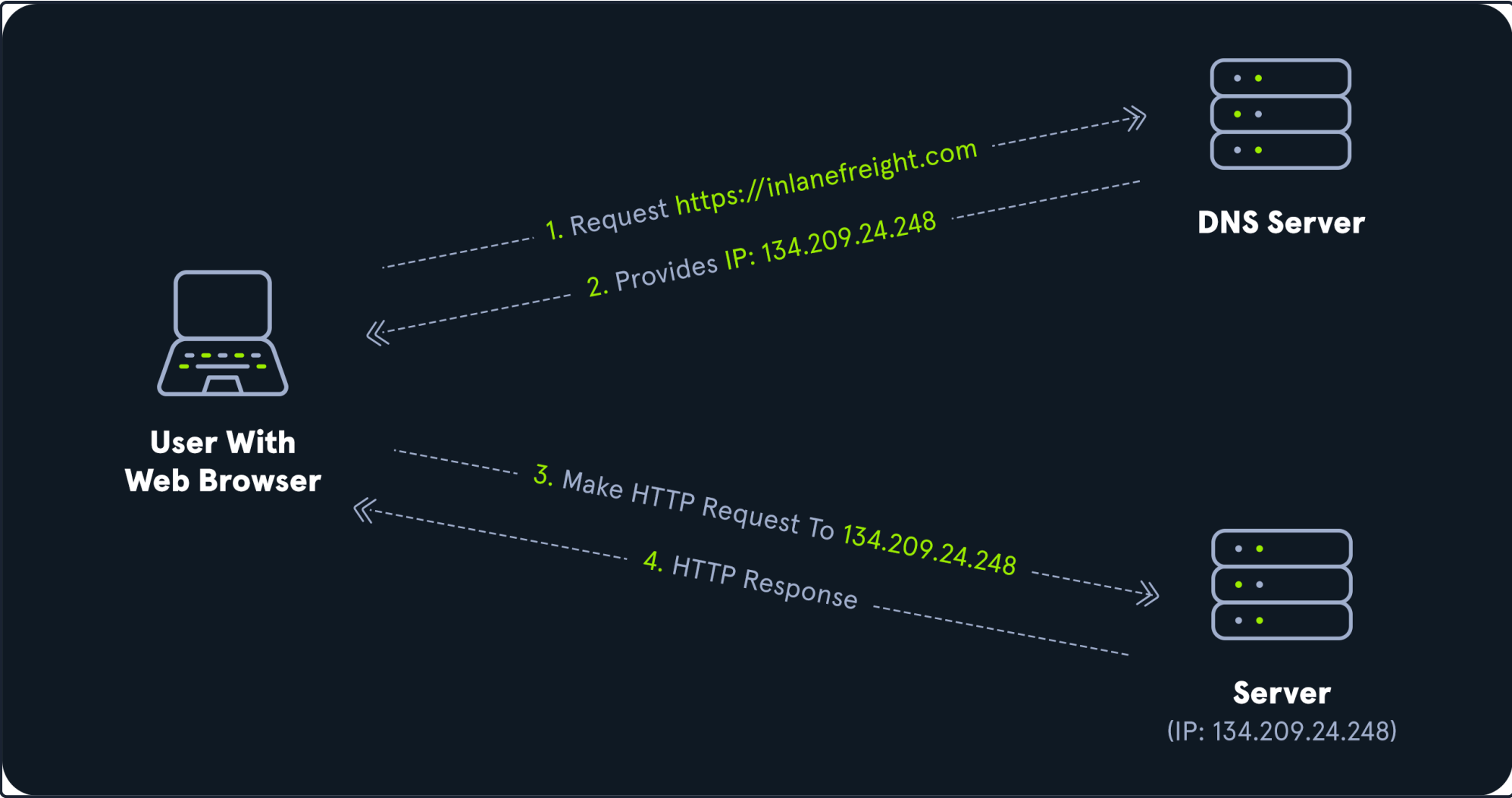
1. When a user logs in, their password is used to encrypt a timestamp, which is sent to the Key Distribution Center (KDC) to verify the integrity of the authentication by decrypting it. The KDC then issues a Ticket-Granting Ticket (TGT), encrypting it with the secret key of the krbtgt account. This TGT is used to request service tickets for accessing network resources, allowing authentication without repeatedly transmitting the user's credentials. This process decouples the user's credentials from requests to resources.
2. The KDC service on the DC checks the authentication service request (AS-REQ), verifies the user information, and creates a Ticket Granting Ticket (TGT), which is delivered to the user.
3. The user presents the TGT to the DC, requesting a Ticket Granting Service (TGS) ticket for a specific service. This is the TGS-REQ. If the TGT is successfully validated, its data is copied to create a TGS ticket.
4. The TGS is encrypted with the NTLM password hash of the service or computer account in whose context the service instance is running and is delivered to the user in the TGS\_REP.
5. The user presents the TGS to the service, and if it is valid, the user is permitted to connect to the resource (AP\_REQ).



The Kerberos protocol uses port 88 (both TCP and UDP). When enumerating an Active Directory environment, we can often locate Domain Controllers by performing port scans looking for open port 88 using a tool such as Nmap.

## DNS

Active Directory Domain Services (AD DS) uses DNS to allow clients (workstations, servers, and other systems that communicate with the domain) to locate Domain Controllers and for Domain Controllers that host the directory service to communicate amongst themselves. DNS is used to resolve hostnames to IP addresses and is broadly used across internal networks and the internet. Private internal networks use Active Directory DNS namespaces to facilitate communications between servers, clients, and peers. AD maintains a database of services running on the network in the form of service records (SRV). These service records allow clients in an AD environment to locate services that they need, such as a file server, printer, or Domain Controller. Dynamic DNS is used to make changes in the DNS database automatically should a system's IP address change. Making these entries manually would be very time-consuming and leave room for error. If the DNS database does not have the correct IP address for a host, clients will not be able to locate and communicate with it on the network. When a client joins the network, it locates the Domain Controller by sending a query to the DNS service, retrieving an SRV record from the DNS database, and transmitting the Domain Controller's hostname to the client. The client then uses this hostname to obtain the IP address of the Domain Controller. DNS uses TCP and UDP port 53. UDP port 53 is the default, but it falls back to TCP when no longer able to communicate and DNS messages are larger than 512 bytes.



Forward DNS Lookup

Let's look at an example. We can perform a `nslookup` for the domain name and retrieve all Domain Controllers' IP addresses in a domain.

Kerberos, DNS, LDAP, MSRPC

```
PS C:\htb> nslookup INLANEFREIGHT.LOCAL

Server:      172.16.6.5
Address:     172.16.6.5

Name:        INLANEFREIGHT.LOCAL
Address:     172.16.6.5
```

Reverse DNS Lookup

If we would like to obtain the DNS name of a single host using the IP address, we can do this as follows:

Kerberos, DNS, LDAP, MSRPC

```
PS C:\htb> nslookup 172.16.6.5

Server:      172.16.6.5
Address:     172.16.6.5

Name:        ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL
Address:     172.16.6.5
```

Finding IP Address of a Host

If we would like to find the IP address of a single host, we can do this in reverse. We can do this with or without specifying the FQDN.

Kerberos, DNS, LDAP, MSRPC

```
PS C:\htb> nslookup ACADEMY-EA-DC01

Server:      172.16.6.5
Address:     172.16.6.5

Name:        ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL
Address:     172.16.6.5
```

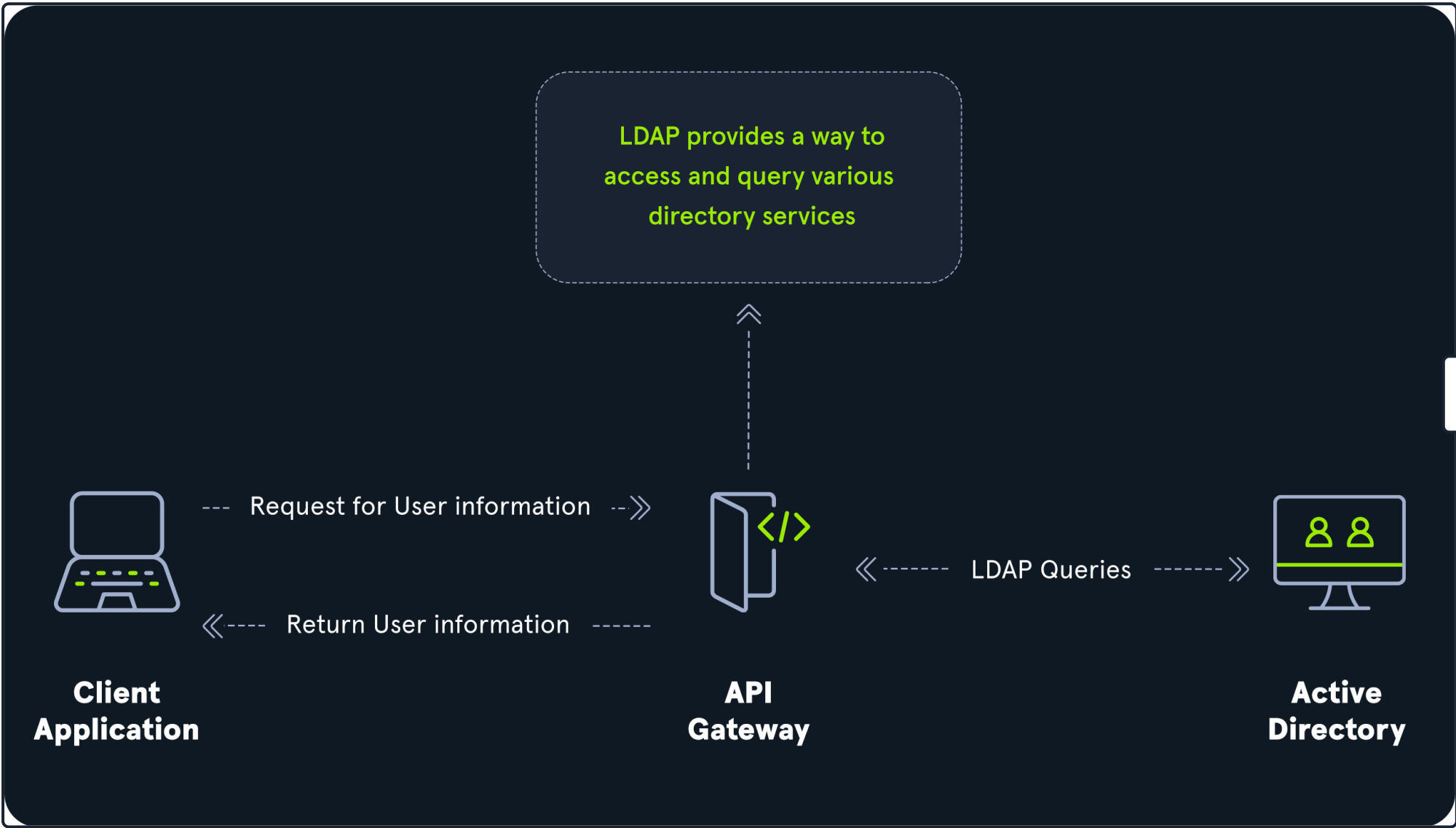
For deeper dives into DNS, check out the [DNS Enumeration Using Python](#) module and the DNS section of the [Information Gathering - Web Edition](#) module.

## LDAP

Active Directory supports [Lightweight Directory Access Protocol \(LDAP\)](#) for directory lookups. LDAP is an open-source and cross-platform protocol used for authentication against various directory services (such as AD). The latest LDAP specification is [Version 3](#), published as RFC 4511. A firm understanding of how LDAP works in an AD environment is crucial for attackers and defenders. LDAP uses port 389, and LDAP over SSL (LDAPS) communicates over port 636.

AD stores user account information and security information such as passwords and facilitates sharing this information with other devices on the network. LDAP is the language that applications use to communicate with other servers that provide directory services. In other words, LDAP is how systems in the network environment can "speak" to AD.

An LDAP session begins by first connecting to an LDAP server, also known as a Directory System Agent. The Domain Controller in AD actively listens for LDAP requests, such as security authentication requests.



The relationship between AD and LDAP can be compared to Apache and HTTP. The same way Apache is a web server that uses the HTTP protocol, Active Directory is a directory server that uses the LDAP protocol.

While uncommon, you may come across organization while performing an assessment that do not have AD but are using LDAP, meaning that they most likely use another type of LDAP server such as [OpenLDAP](#).

### AD LDAP Authentication

LDAP is set up to authenticate credentials against AD using a "BIND" operation to set the authentication state for an LDAP session. There are two types of LDAP authentication.

1. **Simple Authentication:** This includes anonymous authentication, unauthenticated authentication, and username/password authentication. Simple authentication means that a **username** and **password** create a BIND request to authenticate to the LDAP server.



2. **SASL Authentication:** [The Simple Authentication and Security Layer \(SASL\)](#) framework uses other authentication services, such as Kerberos, to bind to the LDAP server and then uses this authentication service (Kerberos in this example) to authenticate to LDAP. The LDAP server uses the LDAP protocol to send an LDAP message to the authorization service, which initiates a series of challenge/response messages resulting in either successful or unsuccessful authentication. SASL can provide additional security due to the separation of authentication methods from application protocols.


LDAP authentication messages are sent in cleartext by default so anyone can sniff out LDAP messages on the internal network. It is recommended to use TLS encryption or similar to safeguard this information in transit.

## MSRPC


As mentioned above, MSRPC is Microsoft's implementation of Remote Procedure Call (RPC), an interprocess communication technique used for client-server model-based applications. Windows systems use MSRPC to access systems in Active Directory using four key RPC interfaces.

Interface Name	Description
lsarpc	A set of RPC calls to the <a href="#">Local Security Authority (LSA)</a> system which manages the local security policy on a computer, controls the audit policy, and provides interactive authentication services. LSARPC is used to perform management on domain security policies.
netlogon	Netlogon is a Windows process used to authenticate users and other services in the domain environment. It is a service that continuously runs in the background.
samr	Remote SAM (samr) provides management functionality for the domain account database, storing information about users and groups. IT administrators use the protocol to manage users, groups, and computers by enabling admins to create, read, update, and delete information about security principles. Attackers (and pentesters) can use the samr protocol to perform reconnaissance about the internal domain using tools such as <a href="#">BloodHound</a> to visually map out the AD network and create "attack paths" to illustrate visually how administrative access or full domain compromise could be achieved. Organizations can <a href="#">protect</a> against this type of reconnaissance by changing a Windows registry key to only allow administrators to perform remote SAM queries since, by default, all authenticated domain users can make these queries to gather a considerable amount of information about the AD domain.
drsuapi	drsuapi is the Microsoft API that implements the Directory Replication Service (DRS) Remote Protocol which is used to perform replication-related tasks across Domain Controllers in a multi-DC environment. Attackers can utilize drsuapi to <a href="#">create a copy of the Active Directory domain database</a> (NTDS.dit) file to retrieve password hashes for all accounts in the domain, which can then be used to perform Pass-the-Hash attacks to access more systems or cracked offline using a tool such as Hashcat to obtain the cleartext password to log in to systems using remote management protocols such as Remote Desktop (RDP) and WinRM.

### Questions

 Cheat Sheet

Answer the question(s) below to complete this Section and earn cubes!

+ 1 

What networking port does Kerberos use?

88

 Submit

 Hint

+ 0 

What protocol is utilized to translate names into IP addresses? (acronym)

dns

 Submit

 Hint

+ 0 

What protocol does RFC 4511 specify? (acronym)

ldap

 Submit

 Hint

← Previous

Next →

 Mark Complete & Next

 Cheat Sheet


 Go to Questions

Table of Contents

Active Directory Overview

Why Active Directory?	✓
Active Directory Research Over the Years	✓

Active Directory Fundamentals

 Active Directory Structure	✓
 Active Directory Terminology	✓
 Active Directory Objects	✓
 Active Directory Functionality	✓

Active Directory Protocols

 Kerberos, DNS, LDAP, MSRPC	✓
 NTLM Authentication	✓



All About Users

 User and Machine Accounts	✓
 Active Directory Groups	✓
 Active Directory Rights and Privileges	✓

Digging in Deeper

 Security in Active Directory	✓
 Examining Group Policy	✓

Getting Our Hands Dirty

 AD Administration: Guided Lab Part I	✓
 AD Administration: Guided Lab Part II	✓

Module Summary

Wrapping It Up	✓
----------------	---

My Workstation

OFFLINE

▶ Start Instance

∞ / 1 spawns left

