**LDAP**

LDAP (Lightweight Directory Access Protocol) is a protocol used to access and manage directory information. A directory is a hierarchical data store that contains information about network resources such as users, groups, computers, printers, and other devices. LDAP provides some excellent functionality:

| **Functionality** | **Description** |
| --- | --- |
| Efficient | Efficient and fast queries and connections to directory services, thanks to its lean query language and non-normalised data storage. |
| Global naming model | Supports multiple independent directories with a global naming model that ensures unique entries. |
| Extensible and flexible | This helps to meet future and local requirements by allowing custom attributes and schemas. |
| Compatibility | It is compatible with many software products and platforms as it runs over TCP/IP and SSL directly, and it is platform-independent, suitable for use in heterogeneous environments with various operating systems. |
| Authentication | It provides authentication mechanisms that enable users to sign on once and access multiple resources on the server securely. |

However, it also suffers some significant issues:

| **Functionality** | **Description** |
| --- | --- |
| Compliance | Directory servers must be LDAP compliant for service to be deployed, which may limit the choice of vendors and products. |
| Complexity | Difficult to use and understand for many developers and administrators, who may not know how to configure LDAP clients correctly or use it securely. |
| Encryption | LDAP does not encrypt its traffic by default, which exposes sensitive data to potential eavesdropping and tampering. LDAPS (LDAP over SSL) or StartTLS must be used to enable encryption. |
| Injection | Vulnerable to LDAP injection attacks, where malicious users can manipulate LDAP queries and gain unauthorised access to data or resources. To prevent such attacks, input validation and output encoding must be implemented. |

LDAP is commonly used for providing a central location for accessing and managing directory services. Directory services are collections of information about the organisation, its users, and assets–like usernames and passwords. LDAP enables organisations to store, manage, and secure this information in a standardised way. Here are some common use cases:

| **Use Case** | **Description** |
| --- | --- |
| Authentication | LDAP can be used for central authentication, allowing users to have single login credentials across multiple applications and systems. This is one of the most common use cases for LDAP. |
| Authorisation | LDAP can manage permissions and access control for network resources such as folders or files on a network share. However, this may require additional configuration or integration with protocols like Kerberos. |
| Directory Services | LDAP provides a way to search, retrieve, and modify data stored in a directory, making it helpful for managing large numbers of users and devices in a corporate network. LDAP is based on the X.500 standard for directory services. |
| Synchronisation | LDAP can be used to keep data consistent across multiple systems by replicating changes made in one directory to another. |

There are two popular implementations of LDAP: OpenLDAP, an open-source software widely used and supported, and Microsoft Active Directory, a Windows-based implementation that seamlessly integrates with other Microsoft products and services.

Although LDAP and AD are related, they serve different purposes. LDAP is a protocol that specifies the method of accessing and modifying directory services, whereas AD is a directory service that stores and manages user and computer data. While LDAP can communicate with AD and other directory services, it is not a directory service itself. AD offers extra functionalities such as policy administration, single sign-on, and integration with various Microsoft products.

| **LDAP** | **Active Directory (AD)** |
| --- | --- |
| A protocol that defines how clients and servers communicate with each other to access and manipulate data stored in a directory service. | A directory server that uses LDAP as one of its protocols to provide authentication, authorisation, and other services for Windows-based networks. |
| An open and cross-platform protocol that can be used with different types of directory servers and applications. | Proprietary software that only works with Windows-based systems and requires additional components such as DNS (Domain Name System) and Kerberos for its functionality. |
| It has a flexible and extensible schema that allows custom attributes and object classes to be defined by administrators or developers. | It has a predefined schema that follows and extends the X.500 standard with additional object classes and attributes specific to Windows environments. Modifications should be made with caution and care. |
| Supports multiple authentication mechanisms such as simple bind, SASL, etc. | It supports Kerberos as its primary authentication mechanism but also supports NTLM (NT LAN Manager) and LDAP over SSL/TLS for backward compatibility. |

LDAP works by using a client-server architecture. A client sends an LDAP request to a server, which searches the directory service and returns a response to the client. LDAP is a protocol that is simpler and more efficient than X.500, on which it is based. It uses a client-server model, where clients send requests to servers using LDAP messages encoded in ASN.1 (Abstract Syntax Notation One) and transmitted over TCP/IP (Transmission Control Protocol/Internet Protocol). The servers process the requests and send back responses using the same format. LDAP supports various requests, such as bind, unbind, search, compare, add, delete, modify, etc.

LDAP requests are messages that clients send to servers to perform operations on data stored in a directory service. An LDAP request is comprised of several components:

1. Session connection: The client connects to the server via an LDAP port (usually 389 or 636).
2. Request type: The client specifies the operation it wants to perform, such as bind, search, etc.
3. Request parameters: The client provides additional information for the request, such as the distinguished name (DN) of the entry to be accessed or modified, the scope and filter of the search query, the attributes and values to be added or changed, etc.
4. Request ID: The client assigns a unique identifier for each request to match it with the corresponding response from the server.

Once the server receives the request, it processes it and sends back a response message that includes several components:

1. Response type: The server indicates the operation that was performed in response to the request.
2. Result code: The server indicates whether or not the operation was successful and why.
3. Matched DN: If applicable, the server returns the DN of the closest existing entry that matches the request.
4. Referral: The server returns a URL of another server that may have more information about the request, if applicable.
5. Response data: The server returns any additional data related to the response, such as the attributes and values of an entry that was searched or modified.

After receiving and processing the response, the client disconnects from the LDAP port.