Network Working Group
Request for Comments: 3088
Category: Experimental

K. Zeilenga
OpenLDAP Foundation

April 2001

OpenLDAP Root Service
An experimental LDAP referral service

Status of this Memo

Copyright Notice

Abstract

The OpenLDAP Project is operating an experimental LDAP (Lightweight Directory Access Protocol) referral service known as the "OpenLDAP Root Service". The automated system generates referrals based upon service location information published in DNS SRV RRs (Domain Name System location of services resource records). This document describes this service.

1. Background

LDAP [RFC2251] directories use a hierarchical naming scheme inherited from X.500 [X500]. Traditionally, X.500 deployments have used a geo-political naming scheme (e.g., CN=Jane Doe,OU=Engineering,O=Example,ST=CA,C=US). However, registration infrastructure and location services in many portions of the naming hierarchical are inadequate or nonexistent.

The construction of a global directory requires a robust registration infrastructure and location service. Use of Internet domain-based naming [RFC2247] (e.g., UID=jdoe,DC=eng,DC=example,DC=net) allows LDAP directory services to leverage the existing DNS [RFC1034] registration infrastructure and DNS SRV [RFC2782] resource records can be used to locate services [LOCATE].

1.1. The Glue

Most existing LDAP implementations do not support location of directory services using DNS SRV resource records. However, most servers support generation of referrals to "superior" server(s). This service provides a "root" LDAP service which servers may use as their superior referral service.

Client may also use the service directly to locate services associated with an arbitrary Distinguished Name [RFC2253] within the domain based hierarchy.

Notice:
The mechanisms used by service are experimental. The descriptions provided by this document are not definitive. Definitive mechanisms shall be published in a Standard Track document(s).

2. Generating Referrals based upon DNS SRV RRs

This service returns referrals generated from DNS SRV resource records [RFC2782].

2.1. DN to Domain Name Mapping

The service maps a DN [RFC2253] to a fully qualified domain name using the following algorithm:

```
    domain = null;
    foreach RDN left-to-right        // [1]

    {
        if not multi-valued RDN and
            RDN.type == domainComponent
        {
            if ( domain == null || domain == "." )
            {   // start
                domain = "";
            }
            else
            {   // append separator
                domain .= ".";
            }

            if ( RDN.value == "."  )
            {   // root
                domain = ".";
            }
            else
```

```
            {   // append domainComponent
                domain .= RDN.value;
            }
            continue;
        }
        domain = null;
    }
```

Examples:

Distinguished Name	Domain	
DC=example,DC=net	example.net	
UID=jdoe,DC=example,DC=net	example.net	
DC=.	.	[2]
DC=example,DC=net,DC=.	.	[3]
DC=example,DC=.,DC=net	net	[4]
DC=example.net	example.net	[5]
CN=Jane Doe,O=example,C=US	null	
UID=jdoe,DC=example,C=US	null	
DC=example,O=example,DC=net	net	
DC=example+O=example,DC=net	net	
DC=example,C=US+DC=net	null	

Notes:

0) A later incarnation will use a Standard Track mechanism.

1) A later incarnation of this service may use a right-to-left algorithm.

2) RFC 2247 does not state how one can map the domain representing the root of the domain tree to a DN. We suggest the root of the domain tree be mapped to "DC=." and that this be reversable.

3) RFC 2247 states that domain "example.net" should be mapped to the DN "DC=example,DC=net", not to "DC=example,DC=net,DC=.". As it is not our intent to introduce or support an alternative domain to DN mapping, the algorithm ignores domainComponents to the left of "DC=.".

4) RFC 2247 states that domain "example.net" should be mapped to the DN "DC=example,DC=net", not to "DC=example,DC=.,DC=net". As it is not our intent to introduce or support an alternative domain to DN mapping, the algorithm ignores domainComponents to the left of "DC=." and "DC=." itself if further domainComponents are found to the right.

5) RFC 2247 states that value of an DC attribute type is a domain component. It should not contain multiple domain components. A later incarnation of this service may map this domain to null or be coded to return invalid DN error.

If the domain is null or ".", the service aborts further processing and returns noSuchObject. Later incarnation of this service may abort processing if the resulting domain is a top-level domain.

2.2. Locating LDAP services

The root service locates services associated with a given fully qualified domain name by querying the Domain Name System for LDAP SRV resource records. For the domain example.net, the service would do a issue a SRV query for the domain "_ldap._tcp.example.net". A successful query will return one or more resource records of the form:

```
_ldap._tcp.example.net. IN SRV 0 0 389 ldap.example.net.
```

If no LDAP SRV resource records are returned or any DNS error occurs, the service aborts further processing and returns noSuchObject. Later incarnations of this service will better handle transient errors.

2.3. Constructing an LDAP Referrals

For each DNS SRV resource record returned for the domain, a LDAP URL [RFC2255] is constructed. For the above resource record, the URL would be:

```
ldap://ldap.example.net:389/
```

These URLs are then returned in the referral. The URLs are currently returned in resolver order. That is, the server itself does not make use of priority or weight information in the SRV resource records. A later incarnation of this service may.

3. Protocol Operations

This section describes how the service performs basic LDAP operations. The service supports operations extended through certain controls as described in a later section.

3.1. Basic Operations

Basic (add, compare, delete, modify, rename, search) operations return a referral result if the target (or base) DN can be mapped to a set of LDAP URLs as described above. Otherwise a noSuchObject response or other appropriate response is returned.

3.2. Bind Operation

The service accepts "anonymous" bind specifying version 2 or version 3 of the protocol. All other bind requests will return a non-successful resultCode. In particular, clients which submit clear text credentials will be sent an unwillingToPerform resultCode with a cautionary text regarding providing passwords to strangers.

As this service is read-only, LDAPv3 authentication [RFC2829] is not supported.

3.3. Unbind Operations

Upon receipt of an unbind request, the server abandons all outstanding requests made by client and disconnects.

3.4. Extended Operations

The service currently does recognize any extended operation. Later incarnations of the service may support Start TLS [RFC2830] and other operations.

3.5. Update Operations

A later incarnation of this service may return unwillingToPerform for all update operations as this is an unauthenticated service.

4. Controls

The service supports the ManageDSAit control. Unsupported controls are serviced per RFC 2251.

4.1. ManageDSAit Control

The server recognizes and honors the ManageDSAit control [NAMEDREF] provided with operations.

If DNS location information is available for the base DN itself, the service will return unwillingToPerform for non-search operations. For search operations, an entry will be returned if within scope and matches the provided filter. For example:

```
c: searchRequest {
    base="DC=example,DC=net"
    scope=base
    filter=(objectClass=*)
    ManageDSAit
}

s: searchEntry {
    dn: DC=example,DC=net
    objectClass: referral
    objectClass: extensibleObject
    dc: example
    ref: ldap://ldap.example.net:389/
    associatedDomain: example.net
}
s: searchResult {
    success
}
```

If DNS location information is available for the DC portion of a subordinate entry, the service will return noSuchObject with the matchedDN set to the DC portion of the base for search and update operations.

```
c: searchRequest {
    base="CN=subordinate,DC=example,DC=net"
    scope=base
    filter=(objectClass=*)
    ManageDSAit
}

s: searchResult {
    noSuchObject
    matchedDN="DC=example,DC=net"
}
```

5. Using the Service

Servers may be configured to refer superior requests to <ldap://root.openldap.org:389>.

Though clients may use the service directly, this is not encouraged. Clients should use a local service and only use this service when referred to it.

The service supports LDAPv3 and LDAPv2+ [LDAPv2+] clients over TCP/IPv4. Future incarnations of this service may support TCP/IPv6 or other transport/internet protocols.

6. Lessons Learned

6.1. Scaling / Reliability

This service currently runs on a single host. This host and associated network resources are not yet exhausted. If they do become exhausted, we believe we can easily scale to meet the demand through common distributed load balancing technics. The service can also easily be duplicated locally.

6.2. Protocol interoperability

This service has able avoided known interoperability issues in supporting variants of LDAP.

6.2.1. LDAPv3

The server implements all features of LDAPv3 [RFC2251] necessary to provide the service.

6.2.2. LDAPv2

LDAPv2 [RFC1777] does not support the return of referrals and hence may not be referred to this service. Though a LDAPv2 client could connect and issue requests to this service, the client would treat any referral returned to it as an unknown error.

6.2.3. LDAPv2+

LDAPv2+ [LDAPv2+] provides a number of extensions to LDAPv2, including referrals. LDAPv2+, like LDAPv3, does not require a bind operation before issuing of other operations. As the referral representation differ between LDAPv2+ and LDAPv3, the service returns LDAPv3 referrals in this case. However, as commonly deployed LDAPv2+ clients issue bind requests (for compatibility with LDAPv2 servers), this has not generated any interoperability issues (yet).

A future incarnation of this service may drop support for LDAPv2+ (and LDAPv2).

6.2.4. CLDAP

CLDAP [RFC1798] does not support the return of referrals and hence is not supported.

7. Security Considerations

This service provides information to "anonymous" clients. This information is derived from the public directories, namely the Domain Name System.

The use of authentication would require clients to disclose information to the service. This would be an unnecessary invasion of privacy.

The lack of encryption allows eavesdropping upon client requests and responses. A later incarnation of this service may support encryption (such as via Start TLS [RFC2830]).

Information integrity protection is not provided by the service. The service is subject to varies forms of DNS spoofing and attacks. LDAP session or operation integrity would provide false sense of security concerning the integrity of DNS information. A later incarnation of this service may support DNSSEC and provide integrity protection (via SASL, TLS, or IPSEC).

The service is subject to a variety of denial of service attacks. The service is capable of blocking access by a number of factors. This capability have yet to be used and likely would be ineffective in preventing sophisticated attacks. Later incarnations of this service will likely need better protection from such attacks.

8. Conclusions

DNS is good glue. By leveraging of the Domain Name System, global LDAP directories may be built without requiring a protocol specific registration infrastructures.

In addition, use of DNS service location allows global directories to be built "ad hoc". That is, anyone with a domain name can participate. There is no requirement that the superior domain participate.

9. Additional Information

Additional information about the OpenLDAP Project and the OpenLDAP Root Service can be found at <http://www.openldap.org/>.

10. Author's Address

Kurt Zeilenga
OpenLDAP Foundation

EMail: kurt@openldap.org

11. Acknowledgments

Internet hosting for this experiment is provided by the Internet Software Consortium <http://www.isc.org/>. Computing resources were provided by Net Boolean Incorporated <http://www.boolean.net/>. This experiment would not have been possible without the contributions of numerous volunteers of the open source community. Mechanisms described in this document are based upon those introduced in [RFC2247] and [LOCATE].

References

[RFC1034] Mockapetris, P., "Domain Names - Concepts and Facilities", STD 13, RFC 1034, November 1987.

[RFC1777] Yeong, W., Howes, T. and S. Kille, "Lightweight Directory Access Protocol", RFC 1777, March 1995.

[RFC1798] Young, A., "Connection-less Lightweight Directory Access Protocol", RFC 1798, June 1995.

[RFC2119] Bradner, S., "Key Words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2247] Kille, S., Wahl, M., Grimstad, A., Huber, R. and S. Sataluri, "Using Domains in LDAP/X.500 Distinguished Names", RFC 2247, January 1998.

[RFC2251] Wahl, M., Howes, T. and S. Kille, "Lightweight Directory Access Protocol (v3)", RFC 2251, December 1997.

[RFC2253] Wahl, M., Kille, S. and T. Howes, "Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names", RFC 2253, December 1997.

[RFC2255] Howes, T. and M. Smith, "The LDAP URL Format", RFC 2255, December 1997.

[RFC2782] Gulbrandsen, A., Vixie, P. and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.

[RFC2829] Wahl, M., Alvestrand, H., Hodges, J. and R. Morgan, "Authentication Methods for LDAP", RFC 2829, May 2000.

[RFC2830] Hodges, J., Morgan, R. and M. Wahl, "Lightweight Directory Access Protocol (v3): Extension for Transport Layer Security", RFC 2830, May 2000.

[LOCATE] IETF LDAPext WG, "Discovering LDAP Services with DNS", Work in Progress.

[LDAPv2+] University of Michigan LDAP Team, "Referrals within the LDAPv2 Protocol", August 1996.

[NAMEDREF] Zeilenga, K. (editor), "Named Subordinate References in LDAP Directories", Work in Progress.

[X500] ITU-T Rec. X.500, "The Directory: Overview of Concepts, Models and Service", 1993.

Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.