版本信息

日期	版本	描述	作者
2004-04-16	v1.0	最初版本	LDAPChina.com

Lightweight Directory Access Protocol (v3): Extension for Transport Layer Security

轻型目录访问协议（v3）传输层安全扩展

本备忘录的状态（Status of this Memo）

本文档定义了一个用于 Internet 通讯的 Internet 标准跟踪协议，为了发展的需要讨论和建议。对于这个协议的状况和地位请参照 Internet 官方协议标准（"Internet Official Protocol Standards"（STD 1））的当前版。这个备忘录的传播是不受限制的。

版权提示（Copyright Notice）

目　录

摘要（Abstract）

本文档为 LDAP 定义了"Start Transport Layer Security (TLS) Operation"（参考文档[LDAPv3]和[TLS]）（Start TLS 操作）。该操作提供了在 LDAP 关联中 TLS 的建立，并有该操作根据 LDAP 扩展请求定义。

1、本文档使用的约定（Conventions Used in this Document）

本文档中的关键字"**MUST**"，"**MUST NOT**"，"**REQUIRED**"，"**SHALL**"，"**SHALL NOT**"，"**SHOULD**"，"**SHOULD NOT**"，"**RECOMMENDED**"，"**MAY**"和"**OPTIONAL**"的含意与参考文档[ReqsKeywords]中描述的相同。

2、Start TLS 请求（The Start TLS Request）

本节描述 Start TLS 扩展请求和扩展响应：如何形成请求，如何形成响应，和列举不同的客户端**必须**（MUST）时刻准备好去处理的结果码。

本节接下将描述 Start TLS 操作的全部过程。

2.1、请求 TLS 建立（Requesting TLS Establishment）

客户端可以通过传输一个包含为 Start TLS 指定 OID 的 ExtendedRequest（参考文档[LDAPv3]）的 LDAP PDU（译者注：协议数据单元：Protocol Data Unit）来执行一个 Start TLS 操作：

1.3.6.1.4.1.1466.20037

LDAP 的 ExtendedRequest 有如下定义：

```
ExtendedRequest ::= [APPLICATION 23] SEQUENCE {
        requestName             [0] LDAPOID,
        requestValue            [1] OCTET STRING OPTIONAL }
```

形成 Start TLS 扩展请求时，将 requestName 域设置为上述的 OID 字符串。不填写 RequestValue 域。客户端**绝不**（MUST NOT）能在发送该请求后，再在该连接上发送任何 PDU，直到该客户端接收到了一个 Start TLS 扩展响应。

当一个 Start TLS 扩展请求被处理完毕时，服务器**必须**（MUST）返回一个包含 Start TLS 扩展响应的 LDAP PDU。LDAP 的 ExtendedResponse 定义如下：

```
ExtendedResponse ::= [APPLICATION 24] SEQUENCE {
        COMPONENTS OF LDAPResult,
        responseName     [10] LDAPOID OPTIONAL,
        response         [11] OCTET STRING OPTIONAL }
```

Start TLS 扩展响应**必须**（MUST）包含一个 responseName 域，该域**必须**（MUST）被设置为与 Start TLS 扩展请求中存在的 responseName 域相同的字符串值。不填写 response 域。服务器**必须**（MUST）将 resultCode 域或者设置为 success（成功）或者设置为将在 2.3 描述的其它值。

2.2、"Success"响应（"Success" Response）

如果 ExtendedResponse 包含一个 success（成功）的 resultCode，则说明服务器期望并能够协商 TLS。详细内容请参考第 3 节。

2.3、非"success"响应（Response other than "success"）

如果 ExtendedResponse 包含一个非 success 的 resultCode，则说明服务器并不期限望或无法协商 TLS。

如果 Start TLS 扩展请求没有成功，则 resultCode 将是下列之一：

operationsError (operations sequencing incorrect; e.g. TLS already established)

protocolError (TLS not supported or incorrect PDU structure)

referral (this server doesn't do TLS, try this one)

unavailable (e.g. some major problem with TLS, or server is shutting down)

如果客户端违反任何 Start TLS 扩展操作的先后次序的需要（在第 3 节描述），服务器**必须**（MUST）返回 operationsError。

如果服务器不支持 TLS（或者由于设计原因，或者由于当前设置的原因），那么它**必须**（MUST）将 resultCode 设置为 protocolError（参考文档[LDAPv3]的 4.1.1 节），或者设置为 referral。如果该服务器返回的 resultCode 为 referral，则它**必须**（MUST）在 LDAP 结果中包括一个实际的 referral 值。如果服务器不支持 TLS，客户端当前的会话是不受影响的。客户端**可以**（MAY）处理任何 LDAP 操作，或者它**可以**（MAY）关闭该连接。

如果服务器支持 TLS，但由于某种原因不能建立一个 TLS 连接（例如：证书服务器没

有响应，LDAP 服务器不能联系到 TLS 实现，或者该 LDAP 服务器正处于关闭处理中），它**必须**（MUST）返回不可用（unavailable）。客户端**可以**（MAY）重试 Start TLS 操作，或者客户端**可以**（MAY）处理任何其它 LDAP 操作，或者它**可以**（MAY）关闭连接。

3、Start TLS 操作的先后顺序（Sequencing of the Start TLS Operation）

本节描述客户端和服务器为建立 TLS 所**必须**（MUST）遵守的全部处理过程。这些过程考虑到了与 LDAP 相关的全部安全的各种不同方面，包括合成安全级（resultant security level）的发现和客户端授权身份析判定。

应该注意的是在 LDAP 关联上建立 TLS，对客户端授权的详细影响将在第 5 节详述。

3.1、在 LDAP 关联上请求 Start TLS（Requesting to Start TLS on an LDAP Association）

在建立了 LDAP 关联之后，客户端**可以**（MAY）在任何时间发送 Start TLS 扩展请求。然而当出现下列情况时，客户端**绝不**（MUST NOT）能发送 Start TLS 扩展请求。

1、 如果在当前连接上已建立了 TLS；

2、 在多级（multi-stage）SASL 协商期间；

3、 如果在该连接上存在任何未完成的 LDAP 操作。

违返任何这些要求的结果将导致返回 operationsError 的 resultCode，这些内容在 2.3 节已描述过。

当客户端发送 Start TLS 请求时，它**可以**（MAY）已经执行了绑定操作，也可以并有绑定。

如果客户端在发送任何其它请求之前，未建立一个 TLS 连接，并且服务器需要该客户端在执行某个特定请求前，建立一个 TLS 连接，那么该服务器**必须**（MUST）拒绝那个请求，并返回结果为 confidentialityRequired 或者 strongAuthRequired。随后，客户端**可以**（MAY）发送一个 Start TLS 扩展请求，或者它**可以**（MAY）选择关闭该连接。

3.2、启动 TLS（Starting TLS）

如果服务器期望并能够协商 TLS，则它将返回一个 resultCode 为 success（成功）扩展

响应；如果不能，则它将返回其它 resultCode，这在前面章节中已介绍过。

在成功的情况下，客户端在该连接上停止传输 LDAP 请求，并且**必须**（MUST）或者开始一个 TLS 协商，或者关闭该连接。客户端将直接在底层传输连接上使用 TLS 记录协议（TLS Record Protocol）发送 PDU，以初始化 TLS 协商（参考文档[TLS]）。

3.3、TLS 版本协商（TLS Version Negotiation）

被使用的 TLS 或 SSL 的版本协商是 TLS 握手协议（Handshake Protocol）的一部分（参考文档[TLS]）。请参考该文档以了解更多细节。

3.4、合成安全层发现（Discovery of Resultant Security Level）

一个 LDAP 关联上的 TLS 连接建立之后，双方**必须**（MUST）单独决定是否继续基于已达到的私有层（privacy level）。确定 TLS 连接的私有层是依赖实现的，并且通过与每一方各自的本地 TLS 实现来完成。

如果客户端或服务器决定认证层或私有对于它继续处理层并不足够高，它**应该**（SHOULD）在 TLS 协商完成后，立刻关闭 TLS 连接（参见下面 4.1 和 5.2 节）。

客户端**可以**（MAY）尝试再次 Start TLS，或者**可以**（MAY）发送一个解绑定（unbind）请求，或者发送任何其它 LDAP 请求。

3.5、客户端授权身份的判定（Assertion of Client's Authorization Identity）

在接收到一个表示成功的 Start TLS 扩展响应时，客户端**可以**（MAY）声明一个特定的授权身份，该授权身份在决定客户端授权状态时被应用。客户端使用一个指定 SASL"EXTERNAL"机制（参考文档[SASL]）的 LDAP 绑定请求来完成此内容，参见下面 5.1.2 节。

3.5、服务器身份检查（Server Identity Check）

客户端为了防止人工介入（man-in-the-middle）攻击，**必须**（MUST）检查存在于服务器证书消息中的服务器身份是否与服务器的主机名相符。

根据下列规则执行检查：

- 客户端**必须**（MUST）使用打开 LDAP 连接的服务器主机名作为值与服务器证书中表达的服务器名进行比较。客户端**绝不能**（MUST NOT）使用服务器的规范 DNS 名或任何其它名称的原始形式。
- 如果证书中存在类型 dNSName 的 subjectAltName 扩展，它**应该**（SHOULD）被用于服务器身份的源。
- 检查是区分大小写的。
- 允许"*"通配符，如果存在，它仅应用于最左边的名称组件中。

例如：*.bar.com 将检查 a.bar.com，b.bar.com 等。但并不检查 bar.com。如果证书中存在给定类型的多个身份（例如，存在多个 dNSName 名），则这些值的任何一个都是可以接受的。

如果在上述检查中，主机名与证书中的 dNSName 身份无法匹配，则面向用户的客户端**应该**（SHOULD）或者通知用户（客户端**可以**（MAY）给予用户一个无论如何都继续的机会）或者终断连接并且声明那个服务器的身份是可疑的。自动化客户端**应该**（SHOULD）关闭连接，返回并且（或者）记录一个错误，该错误声明那个服务器的身份是可疑的。

通过了本节描述的服务器身份检查后，客户端**应该**（SHOULD）准备作进一步的检查来保证服务器被授权以提供客户端所期望的服务。客户端**可能**（MAY）需要使用本地策略信息（local policy information）。

3.7、服务器功能信息刷新（Refresh of Server Capabilities Information）

当 TLS 会话建立时，客户端**必须**（MUST）刷新所有已缓冲的服务器功能信息（例如：从服务器的根 DSE，参见参考文档[LDAPv3]的 3.4 节）。对于防止可以改变任何在 TLS 建立之前检索得到服务器功能信息的主动中间攻击（active-intermediary attacks）是必须的。服务器**可以**（MAY）在 TLS 建立之后提供不同的功能。

4、关闭 TLS 连接（Closing a TLS Connection）

4.1、温和的关闭（Graceful Closure）

或者客户端或者服务器都**可以**（MAY）在一个 LDAP 关联上通过发送一个 TLS 关闭警告来终止 TLS 连接。这样做将保持 LDAP 关联的完整。

在关闭 TLS 连接之前，客户端**必须**（MUST）或者等待所有已发出的 LDAP 操作完成，或者明确地放弃它们（参考文档[LDAPv3]）。

在关闭 TLS 连接的发起者发送完一个关闭警告之后，它**必须**（MUST）忽略任何 TLS 消息直到它接收到来自另一方的警告。它将终止发送 TLS 记录协议 PDU（TLS Record Protocol PDU。PDU：协议数据单元），并且在接收警告之后**可以**（MAY）发送和接收 LDAP PDU。

另一方如果它接收到一个关闭警告，它**必须**（MUST）立刻发送一个 TLS 关闭警告。它随后将终止发送 TLS 记录协议 PDU，并且**可以**（MAY）发送和接收 LDAP PDU。

4.2、强行关闭（Abrupt Closure）

或者客户端或者服务器都**可以**（MAY）通过删除底层 TCP 连接来强行关闭整个 LDAP 关联和任何建立在此关联上的 TLS 连接。在这种情况下，服务器**可以**（MAY）预先发送一个关闭连接通知（Notice of Disconnection，参考文档[LDAPv3]）给客户端。

5、TLS 对客户端授权身份的影响（Effects of TLS on a Client's Authorization Identity）

本节描述由于在一个 LDAP 关联上建立 TLS 带来的对客户端授权身分的影响。首先将描述缺省的影响，随后对包括错误情况的客户端授权身份判定进行简要讨论，最后描述关闭 TLS 连接的影响。

授权身份和相关概念在参考文档[AuthMeth]中定义。

5.1、建立 TLS 连接的影响（TLS Connection Establishment Effects）

5.1.1、缺省的影响（Default Effects）

当在 LDAP 关联上建立 TLS 连接时，包括匿名状态在内的任何先前已建立的认证和授权身份**必须**（MUST）保制保留。在这种情况下，甚至保留服务器通过 TLS 请求客户端认证的信息。例如：服务器在 TLS 协商期间请求客户端提供它的证书（参考文档[TLS]）。

5.1.2、客户端授权身份的判定（Client Assertion of Authorization Identity）

客户端**可以**（MAY）隐式地请求它的 LDAP 授权身份来源于它的已认证的 TLS 凭证或者它**可以**（MAY）显示地提供一个授权身份和判定它将与它的已认证的 TLS 凭证结合使用。前者被称为隐式判定，后者被称为显示判定。

5.1.2.1、隐式判定（Implicit Assertion）

隐式授权身份判定在 TLS 建立之后完成，TLS 通过调用一个使用**不**（SHALL NOT）包括可选的凭证八进制字符串（可以绑定请求的 SaslCredentials 序列中找到）的"EXTERNAL"机制名（参考文档[SASL]，[LDAPv3]）的 SASL 形式的绑定请求来建立。服务器将按照本地策略（local policy）从由 TLS 凭证提供的认证身份中得到客户端授权身份。如何实现此判定的底层机制每种具体的实现是不同的。

5.1.2.2、显示判定（Explicit Assertion）

显示授权身份判定在 TLS 建立之后完成，TLS 通过调用一个使用**将**（SHALL）包括凭证八进制字符串的"EXTERNAL"机制名（参考文档[SASL]，[LDAPv3]）的 SASL 形式的绑定请求来建立。该字符串**必须**（MUST）被按照参考文档[AuthMeth]中第 9 节的描述构建。

5.1.2.3、错误情况（Error Conditions）

对于任何一种判定，服务器**必须**（MUST）核对客户端在它的 TLS 凭证中提供的认证身份是否被允许镜像至被判定的授权身份。如果客户端未被授权，该服务器**必须**（MUST）拒绝一个在绑定响应中带有 invalidCredentials 结果码绑定操作。

另外，对于任何一种判定形式，如果在做 SASL EXTERNAL 绑定请求时 TLS 会话未在客户端和服务器之间建立并且没有其它外部认证凭证资源（例如：IP 级安全（参考文档[IPSEC]））；或者如果在建立 TLS 会话过程中，服务器没有请求客户端认证凭证，那么 SASL EXTERNAL 绑定**必须**（MUST）失败，并返回结果码 inappropriateAuthentication。

在上述绑定操作失败后，该 LDAP 关联的任何客户端认证和授权状态全部丢失，所以在失败后，LDAP 关联处于匿名状态。虽然服务器可以（MAY）使用一个 TLS close_notify 消息来结束 TLS 连接，但此时 TLS 连接状态并未受到绑定失败的影响(as it MAY at any time)。

5.2、关闭 TLS 连接的影响（TLS Connection Closure Effects）

TLS 连接的关闭**必须**（MUST）导致 LDAP 关联变为匿名认证和授权状态，而不管通过 TLS 建立的状态，也不管在 TLS 连接建立之前的认证和授权状态。

6、安全考虑（Security Considerations）

LDAP 使用 TLS 协议的目标是保证连接的机密性和完整性，并且可选地提供认证。TLS 明确地提供了这些功能（在参考文档[TLS]中描述）。

所有通过 Start TLS 操作得到的安全全部通过 TLS 自身的使用而得到。Start TLS 操作本身不提供任何附加安全。

TLS 的使用不提供或不保证由 LDAP 目录服务器掌管的数据的机密性和（或）正确性（non-repudiation）。它也不保护数据不被服务器管理员所监视。一旦建立，TLS 仅提供并保证通过 LDAP 关联传输的操作和数据的机密性和完整性，并且仅当在客户端和服务器支持并协商后才提供这种保证。

由于 TLS 的使用提供的安全的级别直接依赖于正使用的 TLS 实现的质量和那个实现使用的风格。另外，一个主动中间攻击能够从根 DSE 的 supportedExtension 属性中删去 Start TLS 扩展操作。所以，一旦 TLS 被建立并且在开始 TLS 连接开始之前，两端（指客户端和服务器）**应该**（SHOULD）独立地确定和允诺达到的安全级别。例如：TLS 连接的安全级别可能被协商得低至明文水平。

客户端**应该**（SHOULD）或者警告用户何时已达到的安全级别没有提供机密性和（或）完整性保护，或者可配置去拒绝在未到达可接受安全级别上进行处理。

客户端和服务器实现者**应该**（SHOULD）应用一些没有被 TLS 实现所提供的方法去保证适当的凭证和其它凭证数据的保护。

服务器实现者**应该**（SHOULD）允许服务器管理员去选择是否并且何时需要连接机密性和（或）完整性，也允许选择是否并且何时需要通过 TLS 进行客户端认证。

7、感谢（Acknowledgements）

本文档作者们感谢 Tim Howes，Paul Hoffman，John Kristian，Shirish Rai，Jonathan Trostle，Harald Alvestrand 和 Marcus Leech 对本文档的贡献。

8、参考书目（References）

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

[IPSEC] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[LDAPv3] Wahl, M., Kille S. and T. Howes, "Lightweight Directory Access Protocol (v3)", RFC 2251, December 1997.

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

[SASL] Myers, J., "Simple Authentication and Security Layer (SASL)", RFC 2222, October 1997.

[TLS] Dierks, T. and C. Allen. "The TLS Protocol Version 1.0", RFC 2246, January 1999.

9、作者地址（Authors' Addresses）

Jeff Hodges
Oblix, Inc.
18922 Forge Drive
Cupertino, CA 95014
USA
Phone: +1-408-861-6656
EMail: JHodges@oblix.com

RL "Bob" Morgan
Computing and Communications
University of Washington
Seattle, WA
USA
Phone: +1-206-221-3307
EMail: rlmorgan@washington.edu

Mark Wahl
Sun Microsystems, Inc.
8911 Capital of Texas Hwy #4140
Austin TX 78759

USA
EMail: M.Wahl@innosoft.com

10、知识产权声明（Intellectual Property Rights Notices）

略

11、完整的版权声明（Full Copyright Statement）

略