Best Practices in LDAP Security

September 2011

Dr Andrew Findlay

Skills 1st Ltd

Synopsis

LDAP servers are part of the critical infrastructure of most large organisations. They hold personal data subject to legal protection, and often act as the authoritative source of authentication and authorisation for multiple applications.

This paper divides LDAP security into three major requirements: *availability*, *integrity*, and *confidentiality*. Appropriate controls are proposed for each topic, noting the interactions and compromises that are required. Most of the controls are technical, relating to design and administration issues that affect all LDAP server products. The trade-off between technical and organisational controls is discussed, with reference to common human-factors issues.

1 Requirements

What is security? This rather over-used word now covers a multitude of things, ranging from nightclub bouncers to spies, and from junk bonds to "the state of feeling safe". In the context of information systems we need a tighter definition, and this is provided by the 'ISO27k' series of standards. Section 2.19 of ISO/IEC 27000:2009 [ISO27000] defines information security as "preservation of *confidentiality*, *integrity* and *availability* of information". It also notes that other properties, such as authenticity, accountability, non-repudiation, and reliability are relevant.

Following up the definitions of those terms, we find:

Confidentiality is the property that information is not made available or disclosed to unauthorized individuals, entities, or processes.

Integrity is the property of protecting the accuracy and completeness of assets, where *asset* is given the very broad definition "anything that has value to the organization" – a synonym for 'data' in the context of this paper.

Availability is the property of "being accessible and usable upon demand by an authorized entity". This is a property that is often neglected in favour of the others, but is in fact at the core of information security: if we are not protecting the availability of the service then there is little point in having it at all.

2 Controls

In the language of ISO/IEC 27000, *control* is a "means of managing risk, including policies, procedures, guidelines, practices or organizational structures, which can be administrative, technical, management, or legal in nature". The core standard, ISO/IEC 27001, is not prescriptive in this area: it refers to ISO/IEC 27002 which provides a list of example controls for organisations to choose from. The choice is to be guided by analysis of the risks, and controls may be drawn from other sources as appropriate. Following this model, the remainder of this paper presents some LDAP-related controls for consideration.

3 Technical Controls

3.1 Account management

Where LDAP entries represent accounts used by people or applications, it is important to have effective management processes in place. For personal accounts this should include automated updates from human-resources systems and other databases to provision new accounts and to disable old ones promptly. It is normally best to avoid deleting accounts, as that leads to a greater risk of re-using identifiers and also to problems in interpreting audit logs.

3.2 Authentication

LDAP servers generally support two different authentication methods: "simple bind" [RFC4513] and SASL [RFC4422]. If simple bind is in use then TLS should also be used, to prevent exposure of passwords on the network. As LDAP is often used to validate passwords for other services this is likely to be a very common situation. RFC4513 says that servers SHOULD disallow the use of passwords when TLS is not in use. Very few server products have this as their default setting but it should be seriously considered.

More secure mechanisms based on SASL should be used if possible. SASL EXTERNAL along with client-side certificates and TLS provides the most comprehensive protection, but does require the creation and management of an X.509 certificate for each user. Kerberos [RFC4120] carried by the SASL GSSAPI mechanism is a good choice in many environments.

Mechanisms such as DIGEST-MD5 avoid exposing the password on the network, but require the server to store every password in clear. This may be appropriate in some environments, but is not generally recommended.

In cases where one person acts with the delegated authority of another, it is common to find passwords being shared. This is against most organisations' policies. LDAP supports delegation without password sharing: for complete sessions using the SASL *authc/authz* concept, and for individual operations using the LDAP Proxied Authorization Control [RFC4370]. These mechanisms should be used in preference to password sharing.

Proxy authentication can also be used where a process such as a web app is acting on behalf of many users, but caution is advised: if the application is given the ability to act for any user then the consequences of a successful exploit may be unacceptable. Wherever possible, LDAP operations performed by an application should be done using the credentials of the user that triggered them.

3.3 Password Policy

Most LDAP systems store and validate passwords – indeed for many it is their primary function. Following the X.500 'get back exactly what you put in' principle, servers normally default to storing passwords in clear text or in a form that can be converted back to clear text. In most cases this is not necessary (but see DIGEST-MD5 in section 3.2 above). Wherever possible, passwords should be stored using a non-reversible cryptographic hash including a significant amount of salt. This provides the best possible protection against the recovery of passwords from stolen disks or backup tapes. SSHA-1 is the best commonly-implemented hash at present, but server administrators should consider moving to better hashes such as the SSHA-2 series when they become available. Note that passwords protected using AES and other symmetric algorithms are likely to be recoverable from stolen media with very little effort as the encryption keys are almost certain to be present on the same media.

Many organisations have policies about the choice of passwords, the frequency of changing them, and what happens if there are repeated authentication failures. These can be implemented in LDAP systems [BEHERA], though care is strongly advised when using any features that can lock out an account. There are several problems, some related to the distributed nature of LDAP services which can make it impossible to maintain a truly global view of authentication failures. For the same reason, it is often difficult to detect locked-out accounts and to unlock them by administrative action.

Further, as LDAP is often used as a common authentication service behind several user-facing services, there is a serious risk of unintentional denial of service. An example of this is where a user changes their password in LDAP but forgets to (or is unable to) change the stored password in their e-mail client. Many mail clients will simply retry if they fail to download messages for any reason, and will rapidly trigger an account lockout affecting many other services. It is important to remember that *availability* is also part of security!

3.4 Access control

Access control is an important contributor to information security, but it is not standardised in LDAP. Each server product has its own access-control system, and the capability of these systems varies. Access-control lists affect both *integrity* and *confidentiality*, and their design can be an intricate process. The subject is too large to be treated in detail in this paper, but it is worth considering the requirements capture process described in section 7.1 of *Writing Access-Control Policies for LDAP* [FIN2009]:

- What are the subjects (users) and how can they be grouped into classes?
- What are the objects that we must control access to? Don't forget the nonleaf objects that make up the structure of the DIT.
- What is the security posture of the organisation – open to the world or tightly closed?
- How will entries be created and managed? If the directory will be the master source of data, who will be administering it?
- What will the directory be used for? What access is required for each application to work?

Once the basic policy has been set and the overall shape of the DIT has been determined, it is useful to work through all the relevant use-cases. For each one it should be possible to point to a specific entry in the DIT and ask questions like *Should user A be able to modify the telephone number in this entry? Should an anonymous user be able to read the surname?* The answers to these questions provide validation of the policy definition, and also become specific items in the test suite (see section 3.11 below).

Section 5 of [FIN2009] provides a list of design principles for access-control lists. Some of the more important ones are summarised here:

- ACLs are programs - they should be handled by programmers, not by data administrators.
- Place ACLs on the smallest possible number of entries.
- Write the tests *first*, as this helps to clarify exactly what the requirements are.
- Don't write individual account IDs into ACLs: give permissions to groups and allow administrators to control membership of the groups.

Where entries can be added to the directory by end-users or by data administrators, it may be appropriate to use DIT Structure and DIT Content controls to restrict the type of entries that can be added. This is because LDAP entries often grant the power to do particular things and this often works even if the entries are in the 'wrong' place, while the access-control lists may not provide adequate control over such entries.

Be aware that the details of access-control vary so much between server products that some policies cannot be completely implemented with some servers. If detailed access-control is important in a new project then this should be taken into account when choosing server software.

3.5 DIT Design

The shape of the DIT and the attributes chosen to form distinguished names can both have an effect on security. This is because it is not possible to give any sort of access to information in an entry without also disclosing the full DN of that entry. Taking an example from the original X.500 standard [X501], a person might have the LDAP DN:

CN=Smith,OU=Sales+L=Ipswitch,O=Telecom,C=UK

It would not be possible to give access to this person's *mail* attribute without also exposing the name and location of the department where they work. Similarly, it would not be possible to use the entry for authentication (needing only search access to *uid* and authenticate access to the password) without also disclosing the user's surname and place of work.

A more subtle problem stems from the inability of many LDAP servers to hide the *existence* of an entry whose DN has been guessed. An attacker can use the DN as the base of a search operation, and will often get a different error code for existing and non-existing entries even though the access-control lists apparently protect the entry completely.

Many problems can be avoided by collecting all entries describing people into a single container (e.g. OU=People,O=Telecom,C=UK) and by introducing an otherwise meaningless *uniqueIdentifier* attribute for use in the RDN.

3.6 Replication

Replication is an essential technique contributing to *availability*. Having multiple servers with identical data allows the *service* to continue even if one of the *servers* fails. Placing servers in multiple locations increases the range of threats that can be mitigated.

Internet-facing LDAP services should protect against distributed denial-of-service (DDoS) attacks, This is hard to achieve in practice due to the very large botnets that some attackers can mobilise, but one technique worth considering is to locate servers on disparate networks around the world and arrange that each server can only be reached from a defined range of source addresses. Even servers on internal corporate networks can be subject to DDoS attacks – either from malware or by accidental mis-configuration of desktop systems.

Replication can contribute directly to *confidentiality*, by providing public-facing replicas containing only a non-confidential subset of data from the main servers.

Providing high availability for updates is much harder than providing it for read-only operations. This is because there is a risk of irreconcilable changes being applied to two master servers while they are unable to communicate. In many cases it is better to provide a read-only service while recovering a failed master rather than risk the *integrity* of the data.

If high availability for updates is essential, there are some techniques that will reduce the risks of using multiple master servers:

- Do not permit entries to be renamed
- Try to ensure that changes to any given entry are always made on the same master server.

Be aware that the replication protocols are designed to ensure that user data within any given server will eventually match that on the master server, but that there is no guarantee on how long this might take. Any LDAP client that accesses multiple servers (perhaps as a result of using a load-balancer) may see inconsistencies affecting the *integrity* of its view of the data.

3.7 Network

Network firewalls are a well established security component. Appropriate firewall rules should be in place to protect all LDAP servers.

Be aware that some applications make long-lived LDAP connections which may be idle for substantial lengths of time (e.g. out of office hours). Firewalls that track TCP sessions can have a serious impact on the availability of these applications as such devices often silently drop idle connections after an hour or two. The problem is exacerbated by the typical firewall behaviour of refusing to send Port Unreachable or Reset packets when new traffic arrives on a dropped connection. If you have to use a firewall of this type, try very hard to make it close the TCP sessions properly and to send Port Unreachable packets when appropriate. Failing that, the only safe option is to configure clients and/or servers to close idle connections before the firewall does.

3.8 SSL and TLS

Most data carried by LDAP is likely to be sensitive, so sessions should be encrypted as a matter of course.

LDAP server products are required to support Transport Layer Security (TLS) if they support authentication, so this should be universally available. Most also support the older SSL encryption using a separate TCP port, though this usage has never been defined in a standard. Some organisations use SSL in the mistaken belief that port 636 is in some way more secure than port 389. This should be resisted: SSL has been deprecated for several years and there are known attacks against it that will not be fixed.

The correct and standard approach is to start LDAP without encryption and then negotiate the TLS security layer. If necessary, the server can be configured to refuse all operations other than 'Start TLS' until TLS is in place. It would still be wise to permit at least the root DSE to be read without TLS protection, as many LDAP clients need to read that to detect the server's ability to do TLS at all.

One important function of TLS is to provide proof to the client that it has connected to the correct server and that there is no man-in-the-middle attack in progress. To achieve this protection it is vital for all client systems to have trustworthy copies of the appropriate X.509 signer (CA) certificate, and for them to implement the correct validation checks during TLS setup.

Once TLS is in place on the connection, the client should re-read the root DSE and any other information that it plans to rely on. Servers may give different answers on secure connections, and in any case it is unwise to trust any information received over an unprotected link.

3.9 Server configuration

To protect service availability, servers should apply limits on the size of search results. Large result sets can consume significant amounts of memory, and can take a long time to transfer to the client. The appropriate value for the limit depends on the application: a server that is just supporting authentication for other applications might reasonably set the limit as low as two entries, but one supporting a browsable 'white pages' service might have to allow results of 100 entries or more.

The setup of the host operating system has a bearing on security. Normal good practice should be followed. Consideration should be given to using a dedicated machine or virtual machine to host LDAP server instances. If using virtual machines then be aware that some databases interact badly with some virtualisation technologies.

The standard LDAP TCP port is within the 'System Ports' range. On most Unix-like servers such ports can only be bound by the *root* user, so LDAP server processes are normally started by *root*. It is not desirable to run network-facing services under this all-powerful username, so a dedicated account should be provided for the server to switch to as soon as the port has been bound. Better still would be to start the server as a non-root user and either bind to a non-privileged port, or make use of the POSIX CAP_NET_BIND_SERVICE capability to permit binding to the standard port.

LDAP servers need a persistent data store. Some products use a networked relational database, others have an embedded database using local files. In either case the security of the database must be carefully managed to avoid an attacker bypassing the LDAP server and stealing data directly.

Some databases offer on-disk encryption. This may be a useful contribution to security, but bear in mind that the server must have access to the encryption key so in many cases an attacker who steals a copy of the database will also get the key to decrypt it from the same place.

All LDAP servers and databases store at least part of their configuration (and often all of their data) in local files. These must be properly protected by file-system permissions so that the contents cannot be read or written by any account other than the one that the server runs under. Some servers encrypt parts of their configuration, but as with encrypted databases it is likely that an attacker could steal the encryption keys as easily as the files themselves.

3.10 OS issues

LDAP servers can be very efficient, with one machine serving many thousands of clients. This leads to the risk of hitting file-descriptor limits and TCP connection limits in the operating system. Any such limits must be set to suitable values to preserve the availability and performance of the service. Servers that could be subject to DDoS attack may need these limits set to extreme values, with commensurate amounts of memory available in support.

Add-on security services such as Apparmor and SELinux are sometimes used to harden network-facing processes. This can be a useful backstop in case of coding errors in the LDAP server. It is important to review the configuration of these services, particularly if the LDAP server configuration being used is different from the supplier's default setup.

3.11 Testing

Every LDAP service should have a permanent test suite. This is particularly necessary where the access-control rules are complex. Tests should be written as part of the development process, and relevant parts of the test suite should be run frequently during development. When the service goes into production, the test suite should be kept up-to-date and run whenever any configuration changes are made. It may be necessary to split the tests into two sets so that as many as possible can be run routinely against the production environment.

Tests should cover at least:

- Access-control rules
- Authentication methods
- TLS
- Size limits
- Referential integrity (if the server is configured to enforce this)

4 Service management

4.1 Constant service

LDAP is often used by multiple user-facing services, so maintaining availability is extremely important. With careful planning it should be possible to do complete hardware and software upgrades without any break in service. This can be achieved using replica servers, and either client-side fallback or (preferably) LDAP-aware proxies and load-balancers.

Client software often deals poorly with server failures. A solution for this problem is to run an LDAP proxy service on each client system so that client software is always talking to a local (presumably highly available) service.

4.2 Development and Test environments

Normal good practice dictates that Development and Test environments should be completely isolated from the Production environment. A more difficult decision is what data to load in those environments: a copy of the production data gives the best emulation of the production environment, but the data is still sensitive and subject to strict legal protection.

A representative set of synthetic test data should be generated and used as the normal load on both Development and Test systems. The dataset must be large enough to give realistic results to client queries. Occasionally it may be necessary to load a copy of the Production data on the Test system, in which case care must be taken to protect it and to erase it completely when it is no longer needed.

5 Human factors

It is generally recognised that the legitimate users of an IT system are the most likely route for a successful attack on the security of that system. This should be countered by training and awareness campaigns, but system design has a part to play too. Any system that makes it too hard for people to do their job will eventually be subverted or bypassed by the users. This suggests that the technical controls should not be set too strictly.

Further, there has to be a trade-off between human-enforced and machine-enforced policies. The LDAP server cannot be aware of the *intention* behind a particular operation, so it can only enforce very simple rules based on the data that it holds. Business logic in other applications may have a wider view, but ultimately the decisions have to be made by people.

A model that has value in some organisations it to trust staff to make routine updates, but to notify each update to a 'directory editor' who can keep a broad watch on data quality.

6 Monitoring and Audit

One of the secondary security properties mentioned in [ISO27000] is *accountability*. This can be provided by logging and auditing.

Most LDAP servers have the ability to log changes (and usually other operations) in text format. This is very useful to developers and system managers, but the format is not always convenient for audit. Some servers have the ability to maintain a change-log or a generic operation-log in a database that can be searched using normal LDAP operations. This can be a good way to provide accountability, as it is usually easy to formulate queries such as 'On what occasions was the password changed on this account?'

Be aware that there is a cost to all logging, and for searchable logs it can be quite high because each event must be recorded and indexed in a database. This can make full accountability infeasible on busy services.

7 References

BEHERA *Password Policy for LDAP Directories*, Sermersheim, J., Poitou, L., Chu, H. Ed., draft-behera-ldap-password-policy-10.txt, August 2009

FIN2009 *Writing Access Control Policies for LDAP,* Findlay, A., UKUUG conference proceedings, spring 2009 http://www.skills-1st.co.uk/papers/ldap-acls-jan-2009/

ISO27000 ISO/IEC 27000:2009 *Information technology — Security techniques — Information security management systems - Overview and vocabulary,* ISO/IEC, Geneva, 2009

ISO27001 ISO/IEC 27001:2005 *Information technology — Security techniques — Information security management systems – Requirements*, ISO/IEC, Geneva, 2005

ISO27002 ISO/IEC 27002:2005 *Information technology — Security techniques — Code of practice for information security management*, ISO/IEC, Geneva, 2005

RFC4422 RFC4422: *Simple Authentication and Security Layer (SASL),* Melnikov, A., Ed., and Zeilenga, K., Ed., June 2006

RFC4510 RFC4510: *Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map,* Zeilenga, K., Ed., June 2006

RFC4513 RFC4513: *Lightweight Directory Access Protocol (LDAP): Authentication Methods and Security Mechanisms*, Harrison, R., Ed., June 2006

RFC4120 RFC4120: *The Kerberos Network Authentication Service (V5),* Neuman, C., Yu, T., Hartman, S., Raeburn, K., July 2005

X501 The Directory – Models, Recommendation X.501 / ISO9594-2, ISO/CCITT, Geneva 1988

Contact

Dr Andrew Findlay

Skills 1st Ltd
2 Cedar Chase
Taplow
Maidenhead
SL6 0EU

+44 1628 782565

andrew.findlay@skills-1st.co.uk

www.skills-1st.co.uk