Global Information Assurance Certification Paper

Jeff Oakley

12/14/2003

Securing Directory Servers

TABLE OF CONTENTS

Abstract:

The intent of this paper is to discuss ways to secure directory servers. This is done by discussing security related issues that impact directory servers. Instances have been provided to show how security has been applied to protect directory servers.

One particular area of focus is on the directory products them selves. This is partially due to the lack of security in the LDAP protocol. LDAP is the protocol which is used to query the directory for information. LDAP was never really designed with security in mind. This paper points out that fact and shows how vendors are left to design security in their own products. A comparison is drawn between two products to cite the differences on how two companies chose to implement ACI security in their products.

Components to Consider for Directory Servers

Listed below are some items to consider to keep a directory server secure.

- Physical Security
- Authentication
- Use of Access Controls on the directory (ACI)
- Binding
- Operating System Security
- Available Directory Products

Physical Security

One of the most obvious ways to secure a directory sever is to limit physical access to it. Securing it behind locked doors prevents tampering, theft or hacking from the local console. This can be taken a step further by placing the server in a locked cabinet and in a protective environment for added protection. Every project that I work on requires the servers to be installed in such an environment. This still may not be enough protection for directory servers when you consider the type of environment and add accountability and trust into the equation. Take for instance, a directory server that hosts the public or private key certificates for your organization or your customers. This type of environment may demand a need to implement even more security mechanisms. Things like bio readers, electronic card readers, multiple factor authentication, physical keys, or more. This could be driven by requirements from a group like the financial industry or governmental organizations.

Such was the case on a project I worked on. It required the directory server, which was the certificate authority, be secured in a locked room. It takes two people to gain entrance to this room as designed. A bio reader and an electronic key card are placed far enough away from each other so that it's impossible for one person to gain entrance by them self. After the two people coordinate their keys, they are able to enter the locked vault. The other certificate servers are located in a separate room that requires entry via an

electronic key card and has a guard at the door 24 hours a day. Once inside the room the servers are located behind a metal fence that requires a key to open a physical pad lock. This key must be checked out by a technical analyst who is doing the work from the business area who is responsible for the integrity of the server. The server is further protected by a locked a cabinet.

Even with all these physical measures to control access to the server it would be worthless if the local console of the server isn't locked down with a user id and a password. Authentication is yet another step in securing the directory server.

Authentication

Authentication is the ability to prove you are who you say you are. It's defined in the computing dictionary as "The verification of the identity of a person or process. In a communication system, authentication verifies that messages really come from their stated source, like the signature on a (paper) letter."[1]

Almost all resources, with very few exceptions, are protected using user id's and passwords. Directory servers should always require users to pass their credentials before accessing any information on the directory. However, most directories when shipped have anonymous access turned on by default allowing anyone to access information in the directory. Anonymous access does not care about user id's or passwords. An LDAP query can be made to a server that has anonymous access turned on without authenticating the user to the server first. The only thing that may prevent the data from being accessed in a situation like this is an ACI rule, which is explained later. In some cases anonymous access is left turned on intentionally. A good example of this is for a directory server serving up white page information for a company. Most of the information in the white pages directory is common well known information that can be gained from other public sources. Most of the time anonymous access should be checked for and turned off if it is found enabled.

Creating complex password rules will keep the directory server even more secure, with regards to authentication. With the speed of today computers a standard eight digit password can be cracked with brute force methods within a matter of seconds, minutes, hours or days. The time it takes to do this depends on the following factors.[2]

- Number of ASCII characters used
- Displayable and Non Displayable
- Upper case or lower case
- Symbols
- Computer speed
- Computer crack program

Another method used to try and crack passwords is the dictionary method. Password crack programs go through known words in the dictionary. These dictionaries can be in any language and be different sizes. Users who use common words in the dictionary can have their passwords cracked very easily.

The company I work for implemented a strong authentication password policy for all of our servers and workstations. The password rule is listed below:

- Minimum length 12 characters
- Minimum of at least one upper and lower case character
- No words that can be found in any dictionary of any language
- No common information about you or your family
- One special character must be used such as a tilde or underscore

Authentication can be very sensitive depending on the environment that your directory server is in. If the server is located in the business to enterprise (B2E) environment there may be some level of trust since the users are your own employees. The security model the enterprise uses will still help guide how the server should be locked down.

If you have a directory server in the business to customer (B2C) environment it is imperative that you identify them correctly. The company I work for provides a B2C environment to our customers. Each user has personal information in this directory about them selves. If this information was disclosed to the wrong person authenticating into the directory it could cause severe litigation issues. One of the ways we solved this problem was to use a 3rd party vendor to help prove who the user says they are. Once authenticated, the user can access their personal information in the directory.

Choose a Security Model

There are many different security models that can be adapted to the environment in which you work. Listed below are some of the basic security models available. [3]

Basic Design Principles

Least privilege - fewest possible privileges for user.
Economy of mechanism - small, simple, straight forward.
Open design
Complete mediation - check every access
Permission based - default is denial of access.
Separation of privilege - no single super user.
Least common mechanism - avoid shared objects.

Easy to use

The method deployed in the organization that I work for is a hybrid. The least privileged, permission based, and separation of privilege models are all used. The reason for this type of implementation is to use the strengths of each of those models to help secure the organization. Least privileged only allows end users the minimum level of access necessary to do the job. The first thing any user on our network must do is authenticate. Everything is locked down unless you can authenticate which follows the permission based model. The other model we use is the separation of privilege. This is to keep a check and balance in our security environment.

The strength of using a hybrid model like this allows for a more secure environment. However there are potential monetary impacts due to the potential for higher operating costs to the organization. This is based on the amount of security work that is necessary to limit access to users. Again, this depends on each organizations implementation of the model they choose to deploy and weighing all the risks. When it comes right down to it, the one question that needs to be asked is how much risk your organization is willing to accept and design around it.

One thing that can not be over looked when considering the cost of deploying a more secure network is that of the reputation and trust of the organization to its customers. It may cost more to deploy a secure security model initially. However, the benefits far out weigh the costs when you think about the loss of customers and revenue due to a damaged reputation to the company.

The main points to note before choosing a directory product is to find out how each product implements security within the directory, what operating systems can it operate on, and will it meet the needs of the chosen security model.

Use Access Control Information (ACI's) to Secure Information

After you have authenticated to the directory server it is up to the authorization rules that have been applied to protect the data. This is done by using access control information (ACI). What is an ACI? Access control information allows the directory administrator to secure information in the directory.[4]

> An Access Control Instruction is a set of rules placed on the directory or a subset of the directory. The rules are evaluated by the server and either allow or deny permissions to a client request. Permissions are for instance read, write, search and compare.

Directories typically contain information on user objects like phone numbers, addresses, last and first name, and various other common attributes pertaining to the user. Directories can even store objects like servers, workstations, printers, certificates and other various devices. All of these objects may have their own unique set of attributes associated with them.

It is important to note that the implementation of ACI rules is left to the discretion of the directory vendor.[5]

> Access control policies are expressed in terms of access control factors. E.g., a request having ACFs i,j,k can perform operation Y on resource Z. The set of ACFs that a server makes available for such expressions is implementation-specific.

RFC 2820 Access Control Requirements for LDAP [6] confirms even more that there is no clear direction for ACL's in LDAP by stating that it is informational only and is not a standard. The security each vendor incorporates into their directory software is based on there own design ideas on how to secure data.

ACI's can be placed on any or every entry in a directory. It's not typical to place an ACI rule on every attribute unless it defaults that way from the vendor. One of the main purposes of a directory is to provide information to the person or program requesting it, usually with respect to information about people, very fast. If a directory has sensitive information about people in an attribute, such as a social security number, it can be secured by placing an ACI rule on it. This can protect it from being viewed or queried against.

Below is an example of the types of access control information that the SunOne product makes available based on their implementation of ACI's:[7]

- **Read –** this right allows information to be viewed
- **Search –** allows information to be viewed during an LDAP query of the directory
- **Compare –** this operation allows a query to the directory to compare a value with an object or attribute in the directory and return a positive or negative result on the operation instead of the actual data
- **Write –** allows modify, add and delete of the attribute values
- **Selfwrite –** this is for group operations only and allows users to add or remove themselves from groups in the directory
- **Add –** allows child entries to be added in the directory tree.
- **Delete –** this allows the objects to be deleted
- **Proxy –** allows an object to access directory information by using the rights of the an other object.

Other directory vendors such as Critcial Path and their Injoin Directory Server (IDS) product have similar rights or types of permissions. Even though they appear to be similar they may perform operations in a slight or altogether different way. The ACI table, displayed below, was created by interpreting the descriptions of the permissions listed in the Critical Path Injoin Directory Server Administrator's Reference Guide. [8]

Permission	Attribute Description	Entry Description
Add	User can add the attribute and values	User can add an entry to the directory
Disclose on Error	If enabled allows the attributes value to be disclosed by an attribute or security error	Enables the name of the entry to be disclosed on error
Read	Returns value on a read or search operation	User has read access when the entry is used
Remove	Allows removal of the attribute or it's value	Allows removal of the entry
Browse	N/A	Attributes can be returned as entries in a list or search operation
Export	N/A	Allows user to move entries and subordinate entries from the current location to a new one in the directory
Import	N/A	Operates similar to the Export
Modify	N/A	Modify the DN of entry
Rename	N/A	Allows the entry RDN to be renamed and accounts for all of the subordinate entries as well
Return of DN	N/A	Allows the DN to be returned on error or normal conditions
Compare	Allows attributes to be used in a compare operation	N/A
Filter Match	Enables filters to be used on a search operation	N/A

If you compare the ACI's of both directory products you will note that they have operations that are named the same. However, they may operate differently from each other. The add ACI for SunOne only allows child entries to be added in the directory where the IDS add ACI allows the user to add an attribute or change it's value in addition to allowing the user to add an entry in the directory. This shows the contrast in functionality between the two products. It

also demonstrates that an add in one product means something different in another. It is essential that you understand the directory products implementation of the ACI rules in depth before applying them. This comparison of the two directory products implementation of ACI's support the statement that there is no hard and fast standard ACI rules. Each directory vendor deploys ACI's as they wish.

Precedence Rules for ACI's

It's not enough just to have permissions placed on the entries and attributes in the directory. Rules have to be established to determine the order in which they are processed. This is important because it's possible to have many ACI rules applied to a single entry or attribute. Again the rules may differ from one product to the next. It is dependent on how security is setup around the directory tree.

Let's say a deny rule is assigned to a user so that they can't read or view information in the salary attribute due to the sensitive nature of the attribute. Another right is granted to the same attribute to allow modify rights for a group which happens to contain the same user mentioned above. Obviously there is a conflict. What is the result if there are no rules in place? Precedence rules have been established to resolve these conflicts so that there is a predictable outcome. Again, different directory vendors implement them based on their own security design.

SunOne's product has a straight forward precedence scheme.[9] The request is processed by comparing the incoming request from the start of the targeted object in the directory all the way back up to the root. The reason for this is based on the inheritance rights from the top of the tree down unless there is an over riding right farther down in the directory tree. All of the deny rules are processed first. If there is an implicit deny set on the entry in the directory, the operation is stopped. If there are no deny rules matched, the allow processing starts checking from the targeted directory back to the root. If the allow tests have no match the request will still fail because there was no resultant match. This is due to the SunOne products hidden implicit deny at the root of the directory tree. This secures the directory immediately after the installation of the product. The request will be processed if all deny processing passes and an explicit match is found. If there is more than one match the first match found closest to the target will be used. There is also another possibility that a deny and an allow rule could be in conflict with each other. If this occurs the deny rule will always win. ACI processing can be implemented differently with products from other vendors.

Critical Path's implementation is more complex. This is based on their implementation of Security on the directory.[10] Their first precedence rule is to check against the ACI Attribute. Each attribute may have one or more values and each value is considered an item. Each of these items can have protected items which define the information in the entry that is to be protected with one or

more permission clauses. A permission clause has a list of the users who are allowed or denied access to the entry with the list of those permissions. The precedence rules are also contained in this ACI item. Every permission clause in an ACI item has it own precedence value. If there is no precedence value set it uses the default value. The ACI item also has an authorization level. It requires users to authenticate to this level in the directory or stronger to gain access to any of the permissions set in the ACI item. If this first rule does not satisfy the request a second and third check is made which are closely related. It deals more with specific identification of the user logged in versus a less specific. Finally, the last evaluation made determines that the ACI item that is closest to the target entry is the dominant rule.

This displays the contrast in the precedence rules from one vendor to another. The directory product's capabilities must be well known before trying to implement security on the directory.

Binding

ACI's also include bind information in them to help determine if an individual user or group can access the targeted data. For example, a user Friar binds to the directory with a distinguished name (DN) of cn=Friar Croli, ou=test, o=root, c=us . All subsequent LDAP requests or queries that come from the client where this user is authenticated to the directory will compare his DN to the DN stored in the ACI.

This is accomplished in SunOne's product by using keywords in the ACI. There are two keywords that can be used. They are userdn and groupdn. It depends on which keyword that is used in the bind rule of the ACI as to whether the user or group gains access. In the Critical Path IDS directory the bind information is part of the permission clause in the item.

ACI Security Summary

Setting up ACI's can become very confusing and messy after months and years of changes. Due to the nature of how ACI's can be set at any level within the directory the actual rights the user may have to information may be more than was originally intended or planned. The more complex the directory structure and the more ACI rules implemented can create for quite an administrative mess. Not only is it a nightmare to manage but the potential for security holes are now possible which may be difficult to identify. The way to counter this problem is by getting a tool to help look at all of your ACI rules that are in place and to help diagnose any vulnerable areas.

Choose a directory product that has good solid ACI security features in it. Use the least privileged model concept and deny all access from the root of the tree down. Only enable the attributes that you want displayed. Some products come with this as the default and others you physically have to configure this

way. This will ensure that only the data you want to be accessed through an LDAP request will happen.

During the ACI design phase use a flow charting tool to help draw out the security. You have to know how the rules are applied for each of the vendor's products. Chart the location of each ACI and the impact on rules above or below it. This can be a very tedious exercise. It's also possible to make mistakes using a manual flow charting program.

Keep the ACI attribute itself secured. If the ACI entry can be viewed it could give a hacker very important information about the users, groups and rights they have in the directory. This could gain them access further into the directory or other systems that use single signon.

The directory should never display user passwords in a readable format. The users password should be restricted from being queried against so that a hacker can't run a password crack program against a user.[11] This attribute should be locked down with an ACI and limited to the systems security personnel.

Certain attributes that contain sensitive information about an object should be protected with ACI's. A prime example would be a user object's social security number attribute or information about someone's salary.

Directories and there clients are written to communicate on TCP port 389. This is an industry standard. Changing the port number to a lesser known port number could help secure the directory even more. However, the drawbacks may outweigh the security benefit gained. The client has to be able to be configurable enough to change ports. Off the shelf software that queries against the directory also would have to be able to change port numbers too and this would have to be distributed to the clients universally across the enterprise.

ACI Tools

There are also tools that are available that can help to list all of the ACI's in your directory.

Internet2 Ldap Analyzer [12]

PROTOS project's LDAPv3 test suite

Lock Down the Operating System (OS)

Most of the focus so far has been discussing the security on the directory itself. However, if you don't lock down the OS your directory is just as vulnerable. Most directory platforms run on multiple operating systems. The most popular platforms are Unix, Linux and Windows. All of these operating systems have server hardening procedures that should be followed. Securing an

Operating System won't be discussed here since the focus of this paper is on securing the directory. However, it is mentioned here because the directory is not secure if the OS is not secure.

Directory Products

List of the directory products that are available:

- Solstice
- Microsoft Windows 2000 Active Directory
- Netscape
- SunOne
- Critical Path Injoin Directory Server
- Novell E@Directory
- IBM Secureway
- OpenLDAP

Summary

There are several vendor products to choose from. Depending on the size and complexity of your organization you may need to implement more than one. The reasons for doing this could be many. In the company I work for we have Windows 2000 AD. It's strength is authenticating users on the network and making resources available but not robust enough to server as our B2C directory servers. We also have Netscape, SunOne and Critical Paths directories in place as well. We are striving to find a directory product that we can migrate to so that it lessens the complexity and reduces the cost by maintaining one product instead of many.

So before choosing a product, do the homework first, find out what kind of security is available and verify it will meet all the demands of your organization. Secure it physically to prevent tampering. Lock down the operating system that the directory is on. Implement a secure ACI scheme to protect the data on the directory.

End Notes and References:

1 No Author. "hyperdictionary:computing dictionary:authentication." URL:http://www.hyperdictionary.com/search.aspx?Dict=&define=authentication. (28 Nov. 2003)

2 Shaffer, George. "Password Cracking Goals, Techniques and Relative Merits and Cracking Times of Different Techniques." GeodSoft:Good and Bad Passwords How-To. 2000-2004. URL: http://geodsoft.com/howto/password/cracking_passwords.htm#howlong (29 Nov 2003)

3 No Author. "Operating System Security" URL:http://216.239.37.104/search?q=cache:bWFNC-weSuEJ:www.andrew.cmu.edu/course/95-752/notes/OperatingSystem1.ppt+%22access+control+information%22+%2B+%22security+holes%22&hl=en&ie=UTF-8 (5 Dec. 2003)

4 Garrels, Machtelt. "LDAP Operations HOWTO." 2003 URL:http://tille.soti.org/training/ldap/ldapbasics-x892.html (26 Nov. 2003)

5 Harrison, R. "draft-ietf-ldapbis-authmeth-08.txt." 26 October 2003 URL: http://www.ietf.org/internet-drafts/draft-ietf-ldapbis-authmeth-08.txt Appendix B2. Access Control Factors (1 Dec. 2003)

6 Network Working Group. "Request for Comments: 2820 / Access Control Requirements for LDAP". May 2000. URL:ftp://ftp.rfc-editor.org/in-notes/rfc2820.txt (12 Dec. 2003)

7 Garrels, Machtelt. "What are ACIs." LDAP Operations HOWTO:Chapter 4. 2003 URL:http://tille.soti.org/training/ldap/ldapbasics-x894.html#sect_04_02_01 (26 Nov. 2003)

8 No Author. "Chapter 1 Concepts and Attributes: Access Control Information (ACI)." Injoin Directory Server Administrators Reference Guide. February 2002: 3-4

9 Garrels, Machtelt. "What are ACI's." LDAP Operations HOWTO: Chapter 4 Access Control. 2003 URL:http://tille.soti.org/training/ldap/ldapbasics-x894.html (26 Nov 2003)

10 No Author. "Chapter 1 Concepts and Attributes: Access Control Information (ACI)." Injoin Directory Server Administrators Reference Guide. February 2002: 5-8

11 Shaffer, George. „Password Cracking Goals Techniques and Relative Merits and Cracking Times of Different Techniques." GeodSoft:Good and Bad Passwords How-To. 2000-2004. URL: http://geodsoft.com/howto/password/cracking_passwords.htm

12 Piket, Todd. "LDAP Analyzer version 2.0.0." URL: http://ldap.mtu.edu/internet2/analyzer/help/aci_analysis_info.shtml

Last Updated: July 28th, 2017

Upcoming Training

SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UT	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS New York City 2017	New York City, NY	Aug 14, 2017 - Aug 19, 2017	Live Event
Virginia Beach 2017 - SEC401: Security Essentials Bootcamp Style	Virginia Beach, VA	Aug 21, 2017 - Aug 26, 2017	vLive
SANS Chicago 2017	Chicago, IL	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
Community SANS Pasadena SEC401 @ NASA	Pasadena, CA	Aug 23, 2017 - Aug 30, 2017	Community SANS
Mentor Session - SEC401	Minneapolis, MN	Aug 29, 2017 - Oct 10, 2017	Mentor
SANS San Francisco Fall 2017	San Francisco, CA	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Tampa - Clearwater 2017	Clearwater, FL	Sep 05, 2017 - Sep 10, 2017	Live Event
Mentor Session - SEC401	Edmonton, AB	Sep 06, 2017 - Oct 18, 2017	Mentor
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
Mentor Session - SEC401	Ventura, CA	Sep 11, 2017 - Oct 12, 2017	Mentor
Community SANS Albany SEC401	Albany, NY	Sep 11, 2017 - Sep 16, 2017	Community SANS
Community SANS Dallas SEC401	Dallas, TX	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS Columbia SEC401	Columbia, MD	Sep 18, 2017 - Sep 23, 2017	Community SANS
SANS Copenhagen 2017	Copenhagen, Denmark	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Boise SEC401	Boise, ID	Sep 25, 2017 - Sep 30, 2017	Community SANS
Baltimore Fall 2017 - SEC401: Security Essentials Bootcamp Style	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
Community SANS New York SEC401	New York, NY	Sep 25, 2017 - Sep 30, 2017	Community SANS
Rocky Mountain Fall 2017	Denver, CO	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS DFIR Prague 2017	Prague, Czech Republic	Oct 02, 2017 - Oct 08, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Oct 02, 2017 - Oct 07, 2017	Community SANS
Community SANS Sacramento SEC401	Sacramento, CA	Oct 02, 2017 - Oct 07, 2017	Community SANS
Mentor Session - SEC401	Arlington, VA	Oct 04, 2017 - Nov 15, 2017	Mentor
SANS October Singapore 2017	Singapore, Singapore	Oct 09, 2017 - Oct 28, 2017	Live Event
Community SANS Indianapolis SEC401	Indianapolis, IN	Oct 09, 2017 - Oct 14, 2017	Community SANS