İSTANBUL KÜLTÜR ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

REHBER HİZMET SUNUCULARI VE UYGULAMALARI

YÜKSEK LİSANS TEZİ

Fuat ALTUN

Anabilim Dalı: Bilgisayar Mühendisliği

Programı: Bilgisayar Mühendisliği Yüksek Lisans Programı

Tez Danışmanı: Yard.Doç.Dr. Kemal Yüksek

HAZİRAN 2004
İstanbul

İSTANBUL KÜLTÜR ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

REHBER HİZMET SUNUCULARI VE UYGULAMALARI

YÜKSEK LİSANS TEZİ
Fuat ALTUN
0109050001

Anabilim Dalı: Bilgisayar Mühendisliği

Programı: Bilgisayar Mühendisliği Yüksek Lisans Programı

Tez Danışmanı: Yard.Doç.Dr. Kemal Yüksek

HAZİRAN 2004
İstanbul

ÖNSÖZ

Rehber servisi teknolojileri hızla yaygınlaşmaktadır. Özellikle LDAP' ın (Ligthweight Directory Access Protocol) rehber servislerine erişim konusunda standart olarak kabul görmesinden sonra neredeyse tüm yazılım üretici firmalar uygulama yazılımlarına ve programlama dillerine, rehber servislerine erişim desteğini eklemişlerdir. Bu yaygınlaşmada LDAP' ın firma bağımsız bir protokol olmasının rolü büyüktür. Rehber servislerinin kullanımı ile gereksiz veri tekrarı, ortak veri kullanımı ve veri tutarsızlığı sorunlarına çözüm üretmek mümkün olacaktır.

Yrd.Doç.Dr. Kemal Yüksek (Tez Danışmanı), tez metninin düzeltilip, kolay anlaşılır getirilmesinde, tez çalışmasının yönlendirilmesinde önemli katkılarda bulunmuştur.
İstanbul Sanayi Odası, görev yapmakta olduğum kurum olarak maddi ve manevi, önemli katkılarda bulunmuştur.
Haktan Akın, görev yapmakta olduğum kurumda yöneticim olarak önemli desteği olmuştur.
Eğitim hayatım boyunca tüm ailemin unutulmaz katkıları olmuştur. Tez çalışmam süresince kendileriyle yeteri kadar ilgilenemediğim eşim ile kızlarımın sabır ve hoşgörüleri ayrıca önemli manevi destek olmuştur.
Ömer Baysan, arkadaşlığı ile önemli manevi desteği olmuştur. Bunun sürekli olmasını ümit ediyorum.

Burada adını hatırlayamadağım tezimle ilgili yardımda bulunan herkese teşekkür ederim.

Fuat ALTUN

Haziran-2004

İÇİNDEKİLER

KISALTMALAR

LDAP : Lightweight Directory Access protocol
DAP : Directory Access protocol
DUA : Directory User Agent
DSA : Directory System Agent
OSI : Open System Interconnection
DIB : Directory Information Base
SASL : Simple Authentication and Security Layer
ACL : Access Control Lists
LDIF : LDAP Data Interchange Format
NIS : Network Information System
NSS : Name Service Switch
PAM : Pluggable Authentication Modules
ITU-T : International Telecommunication Union -Telecommunication
MD5 : Message Digest 5
RPC : Remote Procedure Call
DIT : Directory Information Tree
SSH : Secure Shell
O : Organization
OU : Organizational Unit
CN : Common Name
DC : Domain Components

TABLO LİSTESİ

ŞEKİL LİSTESİ

ÖZET

Bilgisayar ağları ve Internet gibi donanımsal ve yazılım olarak dağıtık yapıların artmasıyla veri tekrarı, ortak veri kullanımı, veri tutarsızlığı gibi yeni kavramlar önem kazanmıştır. Bu çalışmada işletim sistemleri ve çok programlı ortamlarda, adı geçen kavramlara bağlı oluşan sorunlara çözüm olabilecek yapıların oluşturulması irdelenmiştir.

Özel olarak İşletim sistemleri bazında Linux (aynı zamanda UNIX) işletim sistemleri üzerindeki kullanıcı hesaplarının tek bir noktadan yönetilmesiyle veri tutarsızlığı ve gereksiz veri tekrarının nasıl önlenebileceği gösterilmiştir. Aynı zamanda mevcut sistemde her platformun kendi kullanıcı rehberinde bu bilgileri saklamak zorunda olduğu ve bunun bir çok sorunu beraberinde getirdiği vurgulanmıştır.

Çoklu programlar bazında ise, bir çok yazılım tarafından ortak kullanılması gereken örneğin Personele, Müşteriye ait bilgilerin standart bir veritabanı üzerinden erişilebilme zorunluluğu söz konusudur. Bu zorunluluktan dolayı her yazılım için veritabanları tasarlanmakta ve bir yazılım diğer yazılımın kullandığı verilere erişememektedir. Bu durumun, bilgilerin çeşitli kaynaklarda gereksiz yere tekrar edilmesine sebep olacağı açıktır. Böyle bir yapının yönetilebilirliği ise son derece zordur.

Yukarıda bahsedilen iki sorunun çözümü de rehber hizmet sunucularının (directory service servers) kullanımı ile sağlanabilir. Bu alanda en gelişmiş ve standart haline gelmekte olan rehber hizmet sunucusu teknolojisi LDAP' (Lightweight Directory Access Protocol). tır. LDAP bu anlamda bir çözüm olmakla beraber yazılım geliştiricilerin, yazılımlarına mutlaka bu desteği kazandırarak, bu yazılımların LDAP rehber sunucularına erişebilir olmalarını sağlamaları gerekmektedir.

Bu tez çalışmasında, yukarıda bahsedilen sorunların önerilen rehber hizmet sunucusu yapısı ile nasıl giderilebileceği kapsamında genel anlamda rehber hizmet sunucuları ve bu rehber hizmet sunucularına erişim yöntemleri incelenecektir. Bu bağlamda X.500, NIS, NIS+ ve özel olarak LDAP yapıları üzerinde durulacaktır. LDAP' ın bu sistemlere göre avantajı işlenecektir. LDAP rehber servis sunucuları ile İlişkisel Veritabanları (Relational Databases) arasında karşılaştırma yapılacaktır.

DIRECTORY SERVICE SERVERS AND APPLICATIONS

SUMMARY
With the increase in software and hardware such as Computer Networks and Internet, some new concepts like data redundancy, shared data use, and data inconsistency have gained importance.In this study, the formation of the structures that would be a solution to the problems related to these concepts is studied.

In particular,in the operating system Linux (also UNIX), by managing the user accounts from a center, it was shown how to prevent the data inconsistency and unnecessary data redundancy. At the same time, it was emphasized that these information must be stored in every platform‘s own user guide and that this causes many problems.

In multi-programs, for instance, there is a necessity of the accessibility of information about customers by a standart database, Personel which is needed to be used collectively by many software. Because of this compulsory, for each software a database is constructed and one software cannot access the other software’s data. It is clear that this situation will lead to unnecessary repitition of data in various sources. The management of such a structure is extremely difficult.

The solution to these two problems mentioned above can be supplied by using directory services. In this field, the most developed and currently being a standart directory service is LDAP(Lightweight Directory Access Protocol). From this point of view, LDAP is a good solution and software developers should provide their softwares with access to LDAP directory services by definitey supplying this supporter to their softwares.

In this thesis, directory services in the scope of how the problems mentioned above can be overcome by recommended directory service structure and the methods of accessing these directory services will be studied. It is going to be studied X.500, NIS, NIS+ and in particular LADP. The advantage of LADP for these systems will be held. LADP directory services and Relational Databases will be compared

1. GİRİŞ

Günümüzde hiçbir uygulama tek bir bilgisayar üstünde çalışmak üzere tasarlanmamaktadır. Bu uygulamaların sadece yerel bilgisayar ağlarında (local area network) kullanılacağını düşünerek bir tasarım yapmak yine kullanımda birçok aksaklığa sebep olacaktır. Internet artık hayatımızın vazgeçilmez bir parçası haline gelmiş durumdadır.Bu durumun farkında olan yazılım geliştirici firmalar işletim sistemlerini (operating systems) ve uygulama yazılımlarını Internet desteği ile birlikte kullanıcılara sunmaya başlamıştır. Hızla artan sayıda kullanıcı bu dev bilgi kümesinden yararlanabilmek için bu sanal dünyaya giriş yapmaktadır. Bu kullanıcıların artmasıyla birlikte gerek işletim sisteminin gerekse uygulama yazılımlarının kullanıcılarının yönetimi zorlaşmakta hatta çok büyük ağlarda neredeyse imkansız hale gelmektedir.

Bilgisayar sistemlerinin kullanıldığı orta ölçekli bir kuruluşta dahi Windows, Linux, Unix gibi işletim sistemlerine Web, Ftp, Telnet, Ssh[1] gibi uygulama programlarına Oracle, Mysql, PostgreSql, Db2, MS Sql Server gibi Veritabanı Yönetim programlarına ve çoğu ticari amaçla geliştirilmiş (muhasebe, insan kaynakları, üretim planlama vb.) yazılımlara rastlanmaktadır. Bahsedilen sistemlerin çoğu birlikte kullanılmaktadır. Bu işletim sistemleri ve yazılımların hepsi kendilerine özgü kullanıcı rehberleri kullanmaktadır. Yazılımların oluşturduğu sistem bir bütün olarak düşünüldüğünde, bu sisteme yeni bir kullanıcı eklenmek veya bu kullanıcı bilgisinin bir bölümünü değiştirilmek istendiğinde, her rehbere ayrı ayrı erişmek ve yine her rehbere ayrı ayrı ekleme veya değiştirme işlemini yapmak gerekir. Aynı problem sistemden bir kullanıcının silinmesi gerektiği zaman da söz konusu olacaktır. Bu durum yüzlerce, binlerce kişilik ağlarda büyük sorunlara sebep olabilir. Kullanıcıların erişim yetkileri konusunda tutarsızlıklar oluşacaktır. Örneğin sistemden tamamen kaldırılmış bir kullanıcının e-posta hesabı sistemde kalmaya devam edecektir. Sadece yönetimsel değil aynı zamanda güvenlik konusunda da

[1] SSH (Secure Shell), özellikle Telnet yerine tercih edilen uzaktan erişim protokoldür. Veriyi şifreleyerek gönderip aldığı için daha güvenlidir.

problemler söz konusudur. Bir kullanıcı sistemden silindiğinde aynı işlemlerin diğer rehber sistemlerinde de yapılması gerekmektedir. Fakat tüm rehber sistemlerinden silinmiş bir kullanıcı hesabının Veritabanı Yönetim Sisteminin rehber sisteminden silinmesi unutulursa, bu ciddi bir güvenlik açığı oluşturur. Bir kullanıcıya ait parolanın değiştirilmesi de mutlaka teker teker tüm rehber sistemleri üzerinde yapılmalıdır. [2]

Sadece kullanıcı hesaplarının değil bir çok uygulama yazılımı tarafından ihtiyaç duyulan Personele, Müşterilere vb. ait bilgilerinde her uygulamanın kendine özel yapısı içinde tutulması, bu bilgilerin her yerde tekrarlanmasını zorunlu hale getirecektir. Kaynaklar bu durumda verimsiz kullanılmış olacaktır. Aynı zamanda hem yönetilebilirlik hem de güvenlik konusunda açıklar meydana gelecektir. İletişim amaçlı saklanan kullanıcı bilgilerine başka bir uygulamadan veya aynı uygulamayla bile olsa başka bir noktadan erişilmesi söz konusu olmayacaktır.

LDAP (Lightweight Directory Access Protocol), rehber servis sunucularına erişmeyi ve buradaki veriler üzerinde işlem yapmayı sağlayan bir protokoldür.LDAP protokolü ile doğrudan olarak erişilen, X.500 sunucularına ihtiyaç duymayan ve TCP/IP üzerinde çalışan rehber sunucularına LDAP rehber sunucuları (LDAP directory server) adı verilmektedir. Michigan Üniversitesindeki bir grup tarafından geliştirilmiştir. ITU-T (International Telecommunication Union - Telecommunication) tarafından geliştirilmiş olan X.500 rehber standartı temel alınarak geliştirilmiştir. LDAP tamamıyla TCP/IP protokolü üzerinde çalışmaktadır. X.500 ise sisteme daha fazla yük getiren OSI[3] protokolünü kullanmaktadır. Bazı rehberler belirli tip verileri saklamak üzere özelleşmişlerdir. Örneğin DNS sistemi internetteki host ve IP adreslerini tutmak üzere özelleşmiştir. Finger sistemi ise Internet üzerindeki kullanıcılara ait bilgileri tutmaktadır. Oysa X.500 ve LDAP rehber sunucuları (directory servers) neredeyse her türlü bilgiyi saklayabilmek üzere tasarlanmışlardır.

---

[2] Kurumların mevcut güvenlik politikaları güvenilir olsada , anılan yapılar hata riskini yükseltmektedirler.

[3] OSI protokolü sistem kaynaklarını ciddi oranda kullanmakta, özellikle hafıza ve işlemciyi yoğun kullanmaktadır.

LDAP ile tüm rehber bilgilerini (kullanıcı hesapları, bilgisayar adları, Ethernet kart adresleri, adres defteri vb.) tek bir merkezde toplayıp onları buradan yönetmek mümkündür. Böyle bir yaklaşım büyük bilgisayar ağlarında sistem yönetimini büyük oranda kolaylaştırmaktadır. LDAP üzerinde nerdeyse her türlü bilgi tutulabilir.[4] Bu konuda herhangi bir kısıtlama getirilmemiştir. Bu yüzden LDAP üzerinde sadece kullanıcı hesaplarının veya adres defteri bilgilerinin tutulabileceğini düşünmemek gerekir. Aynı zamanda LDAP üzerinde tutulan bir varlığın bilgilerine zamanla eklemeler yapılabilir. Bu eklemeler sistemin yapısında değişiklikler gerektirmez. Bir personelin özlük bilgileri yanında ailesi ile ilgili ayrıntı bilgi tutulmak istenebilir. Bu talebin yerine getirilmesi sistemdeki eski yapıda hiçbir değişikliği gerektirmez. Oysa böyle bir işlem ilişkisel bir veritabanında (relational database) yapılmak istenseydi, sistem üzerinde değişikler yapılması gerekirdi. LDAP' ın sağladığı bu esneklik sistemin kolay genişleyebilir olmasını sağlamaktadır.

LDAP rehber sunucularına LDAP protokolü ile erişilmektedir. Fakat LDAP rehber sunucuları üzerinde tasarım yapmak ve sunucuları sisteme entegre etmek yeterli olmamaktadır. LDAP protokolünün hem işletim sistemi hem de uygulama yazılımları tarafından desteklenmesi gerekmektedir. Yazılım üreten firmaların mutlaka LDAP uyumlu yazılımlar üretmeleri gerekmektedir. Günümüzde bu sorun işletim sistemi olarak neredeyse çözülme noktasına gelmiştir. Microsoft firması Windows 2000 ürün ailesinden itibaren LDAP' ı işletim sisteminin içine gömmüştür. Microsoft bu teknolojiye Active Directory adını vermiştir. Linux sistemler için şimdilik böyle bir durum söz konusu değildir. Fakat bununla beraber bir çok popüler Linux dağıtımıyla beraber ücretsiz olan LDAP sunucu ve istemci yazılımları gelmektedir.

Yukarıda belirtilen sorunların çözümü LDAP ile sağlanabilmektedir. Bu tez çalışması kapsamında yukarıda bahsedilen iki sorunun çözümü olarak sistem gerçekleştirimleri yapılmıştır. Bunun dışında X.500 sistemi ve LDAP sistemi karşılaştırılmıştır. Kimlik doğrulama amacıyla kullanım için PAM (Pluggable Authentication Module) yazılım kütüphanesine değinilmiştir. Hangi rehber servisine hangi sırayla başvurulacağını belirlemek için NSS (Name Service Switch) sistemi

[4] LDAP sunucular üzerinde her türlü bilgi tutulabilmesine karşılık, örneğin DNS gibi rehber yapılarında IP numarası, Host ismi gibi kısıtlı bilgiler tutulabilmektedir.

incelenmiştir. Linux işletim sisteminin LDAP ile bütünleşik olarak çalışabilmesi çalışmalar yapılmıştır. Aynı zamanda LDAP üzerinde Microsoft firmasına ait adres defterleri bilgilerinin tutulabilmesi için şemalar üzerinde düzenlemeler yapılmıştır. Bunun dışında LDAP rehberi üzerindeki bilgi ağacının tasarımından bahsedilmiş ve buradaki yaklaşımlar örneklerle gösterilmiştir.

2. REHBER SERVİSLERİ VE TEMEL KAVRAMLAR

İşletim sistemleri ve uygulama yazılım sistemlerinin işletimlerini yönlendiren temel tanım bilgilerinin oluşturduğu bütüne rehber (directory) denir. Örneğin bir işletim sisteminde, işletimin her aşamasında değişik nedenlerle başvurulan kullanıcı bilgileri, ağ erişim bilgileri, yazıcı bilgileri gibi tanım bilgileri rehber bilgileridir. Bir e-posta okuma programı için, kullanıcının sık sık haberleştiği kişilerin adres bilgilerinin tutulduğu adres defteri de rehber bilgisi olarak düşünülür.

2.1. Linux İşletim Sistemi Yerel Tanım Dosyaları

Linux işletim sisteminin ilk olarak kullanılmaya başlandığı yıllarda, bilgisayar sistemleri arasında bugünkü anlamda bir iletişim altyapısı gelişmemişti. Her bilgisayar kendisine ilişkin temel tanım bilgilerini, bugünkü ifadeyle çeşitli rehberlerini, yerel diskinde, tanım dosyaları olarak saklamaktaydı. Bilgisayarlar arası iletişimin çok yaygın olmadığı ve kurumsal ağlardaki bilgisayar sayısının bugünküne göre çok sınırlı olduğu bu zamanlarda, yerel tanım dosyaları kendilerinden beklenenleri yeterli bir şekilde yerine getirmekteydi. Bugün hala bu yerel tanım dosyaları kullanılmaya devam etmektedir. Bu dosyaların önemlilerinden bazıları ve işlevleri aşağıda verilmiştir:

• **/etc/passwd**: Linux kullanıcılarına ilişkin kimi bilgilerin saklandığı dosyadır. Bu bilgiler sırayla, kullanıcı adı, crypt fonksiyonu ile şifrelenmiş halde saklanan kullanıcı şifresi (password), kullanıcı numarası (user ID), grup numarası (group ID), kullanıcının gerçek hayattaki adı(gecos), başlangıç dizini yolu (home directory path) ve kullanıcı sisteme girerken çalıştırılan kabuk program (shell) bilgileridir.
Linux işletim sisteminde kullanıcı şifreleri, güvenlik nedenlerinden dolayı açık metin olarak saklanmaz. Kullanıcı şifreleri, crypt fonksiyonu ile şifrelenerek saklanırlar. Bu fonksiyon tek yönlü çalışan bir hash algoritmasına göre sonuç üretir. Günümüzün popüler Linux dağıtımlarında bu hash algoritması genellikle MD5' tir. Yani

şifrelenmiş metinden, açık metine dönüşüm yapılamaz. Şifre kontrolü, kullanıcının girdiği açık metin şifrenin, crypt fonksiyonu ile hash değerinin elde edilerek sistemde saklanan hash koduyla tekrar karşılaştırılması şeklinde olur.

• **/etc/group** : Linux kullanıcı gruplarına ilişkin kimi bilgilerin saklandığı dosyadır. Bu bilgiler sırayla, grup adı (group name), grup numarası (group ID) ve grup üyelerinin kullanıcı adlarının listesidir.

• **/etc/shadow** : Linux kullanıcı şifre bilgilerinin saklandığı bu dosyadır, kullanıcı şifresi, kullanıcı şifresinin en son değişme tarihi, şifrenin değiştirilmeden en az kaç gün kullanılması gerektiği, şifrenin değiştirilmeden en fazla kaç gün kullanılabileceği, şifrenin son kullanma tarihinin bitiminden kaç gün önce kullanıcının uyarılacağı, şifrenin son kullanma tarihinin bitiminden kaç gün sonra hesabın geçersiz olacağı, kullanıcı hesabının son kullanma tarihi, ve seçimli bir gösterge (flag) alanı saklanmaktadır.

Linux sistemlerinde, /etc/passswd dosyasının kullanıcıya ait şifreyi sakladığını ifade etmiştik. Bu dosyanın, kullanıcı tanıma işlemi dışında diğer sistem yazılımları tarafından da kullanılması, bu dosyanın bütün kullanıcılara okuma erişimi açık durumda bulunmasını gerektiriyordu. Bu yüzden, şifreli olarak saklansalar da, kullanıcı şifreleri açık bir tehdit altındaydı. İlk zamanlarda, bilgisayar sistemleri çok fazla gelişmediği için, bu şifreleri çözmek uzun zaman alıyordu. Fakat daha sonra sistemler hızlandıkça şifrelerin çözülmesi kolaylaştığı için, ek bir güvenlik getirmek amacıyla /etc/shadow dosyası kullanılmaya başlandı ve kullanıcı şifreleri bu dosyaya aktarıldı.[5] Bu dosyaya sıradan bir Linux kullanıcısının hiçbir erişim hakkı yoktur, sadece ayrıcalıklı kullanıcı olan root kullanıcısı erişebilir

• **/etc/hosts** : Sistemde, IP numarası ve İnternet adresi arasındaki eşlemenin yapıldığı bir sistem dosyasıdır. Bu dosya, DNS sisteminin kullanılmasıyla büyük ölçüde işlevini kaybetmiştir. Fakat, sistemin açılışı sırasında, henüz DNS hizmeti başlatılmadan bazı adreslerin çözümlenmesi gerekiyorsa, bu dosyaya ilgili adreslerin bilgileri girilir.

[5] Günümüzde tüm popüler Linux dağıtımları /etc/shadow dosyasını kullanmaktadır. Eğer istenirse bu dosya devre dışı bırakılabilir.

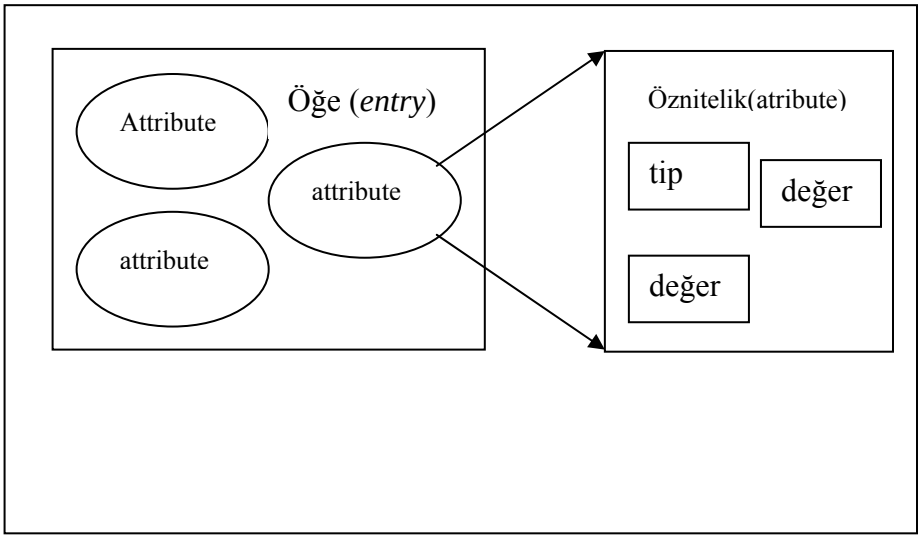
• **/etc/printcab** : Sistemde tanımlı bilgisayar yazıcılarına ilişkin bilgilerin saklandığı sistem dosyasıdır.

Bu sistem dosyaları, Linux'te bulunanların sadece bir kesimidir. Bu dosyalar dışında, sistem üzerinde çalışan hemen her hizmet programının, ayrı bir ayar ya da tanım dosyasına ihtiyacı vardır.

2.2. Öğe (Entry), Öznitelik (Attribute), Tür (Type), Değer (Value)

Rehber Bilgi Tabanı, herbiri (kullanıcı, cihaz gibi) varlıklarla ilgili bilgiyi saklayan öğelerden (entry) oluşur. Hakkında bilgi saklanmak istenen her bir varlık için, rehberde en az bir öğe bulunmalıdır. Her öğe, birden fazla özniteliğe (attribute) sahip olabilir. Bu özniteliklerden her biri, günlük hayattaki varlıkla ilgili bilgi tanımlarına karşılık gelir. Özniteliklerin bir kısmı zorunlu, bir kısmı seçimli olabilir. Her özniteliğin bir türü vardır. Bir özniteliğin türü, nesne öğesinin temsil ettiği varlığa veya daha doğru bir deyişle nesne öğesinin üyesi olduğu nesne sınıfına göre belirlenir [08]. Her özniteliğin, birden fazla değeri olabilir. Bu durum Şekil 2.1' de açıkça görülmektedir. Her değer, varlıkla ilgili, üzerinde işlem yapılan bilgiyi temsil eder.[01] Bir kurumda çalışanlar, rehberde temsil edilmesi gereken nesnelerdir. Telefon numarası bilgisi, "çalışan" için bir özniteliktir. Bu özniteliğin türü, karakter dizisidir. Her çalışan için bir ya da daha çok telefon numarası, bu öznitelik için değerlerdir.

Rehber Bilgi Tabanı içinde sadece nesne öğeleri tutulmaz. Nesneye alternatif adlar vermek amacıyla takma ad öğeleri de (alias entry) kullanılır [02]. Bu öğeler, kendi başlarına herhangi bir varlığa ilişkin bilgi içermeyip, sadece kullanıcıların bilgiye alternatif bir yolla erişmesini sağlamak amacıyla kullanılır.



Şekil 2.1 Öğe, Öznitelik, Tür ve Değer

2.3. Nesne Sınıfları (Object Classes)

Rehber Bilgi Tabanında saklanan herbir nesne öğesi, bir ya da daha çok sayıda nesne sınıfının bir örneğidir. (instance). Nesne sınıfları, bir sınıf hiyerarşisi içinde tanımlanırlar. Her nesne sınıfı, sınıf hiyerarşisi içinde bir nesne sınıfının alt sınıfıdır. Nesne sınıfları, nesne öğeleri için zorunlu ve seçimli öznitelikleri tanımlar. Bir nesne öğesi, zorunlu öznitelikler için mutlaka bir değere sahip olmalıdır. Nesne sınıflarının bir kısmı ITU-T (International Telecommunication Union-Telecommunication) tarafından tanımlanmıştır[6] [01]. ITU-T tarafından tanımlanan nesne sınıflarının bir kısmı ve kullanım amaçları şöyledir:

• top: Nesne hiyeararşisinin en üstünde yeralan sınıftır. Hiyerarşinin başlangıcını temsil eder ve her sınıf bu nesnenin bir olgusu olmak zorundadır.

• Country: Bir ülkeyi temsil eder.

• Organization: Bir kurumu temsil eder.

• OrganizationalUnit: Kurumun bir altbölümünü ya da şubeyi temsil eder.

• Person: Kişileri temsil eder.

• OrganizationalPerson: Bir kurumla ilişkilendirilen kişileri temsil eder.

[6] Rehber servislerini ilişkisel veritabanlarından ayıran özelliklerden biride rehber servislerinin önceden tanımlanmış olan nesne sınıflarına ve standartlaştırılmış şemalara sahip olmasıdır.

Rehber Bilgi Tabanında bulunabilecek nesne sınıfları, sadece ITU-T tarafından tanımlananlarla sınırlı değildir. Sınıf hiyerarşisinde, miras alma özelliği kullanılarak, yeni sınıf tanımları üretilebilir. ITU-T'nin tanımladığı nesne sınıflarının dışında, her üretici firma uluslarası standartlara göre kendi nesne sınıflarını tanımlayabilmektedir. Önemli olan bu sınıfların bir standart olarak kabul edilmesi ve geniş bir kullanım kitlesine sahip olmasıdır.

2.4. Rehber Bilgi Ağacı (DIT- Directory Information Tree) ve Rehber Şeması (Directory Schema)

Rehber Bilgi Tabanında nelerin bulunduğunun yanında, bunların nasıl bir düzende saklandığı da önemlidir. Bir rehber içindeki bilgiler, Rehber Bilgi Ağacı denen bir ağaç yapısına göre düzenlenir. Ağacın köküne yaklaştıkça ülke, kurum gibi daha genel varlıklar; ağacın dallarına gidildikçe kişiler, kurum çalışanları gibi daha özel varlıklar hakkında bilgiyi saklayan nesne öğeleri bulunur.
Bu ağaca nesne öğelerinin yerleşmesi belli kurallara göre olur. Her nesne öğesi, ağaç üzerindeki her konumda bulunamaz. Rehber Bilgi Ağacını düzenli, kolay erişilebilir bir halde tutabilmek için, Rehber Şeması denilen bir kurallar kümesine ihtiyaç vardır. Bu kurallar sayesinde Rehber Bilgi Ağacı, saklanan bilgi sürekli artsa bile, uzun süre işlevselliğini kaybetmeden bilgiyi düzenli saklama görevini yerine getirir.

Örnek bir Rehber Bilgi Ağacı üzerinde ülke, kurum, altbölüm, kişi gibi varlıkların nasıl yerleştiği Şekil 4 ve Şekil 5 'de gösterilmiştir. Şekilde, ITU-T kısaltmalarına uygun olarak, ülke için "C" (country), kurum için "O" (organization), altbölüm için "OU" (organizationalUnit), kişi için "CN" (commonName) ve etki alanı için "DC" (domain component) kısaltmaları kullanılmıştır.

2.5. Göreli Seçici Adlar (Relative Distinguished Names)

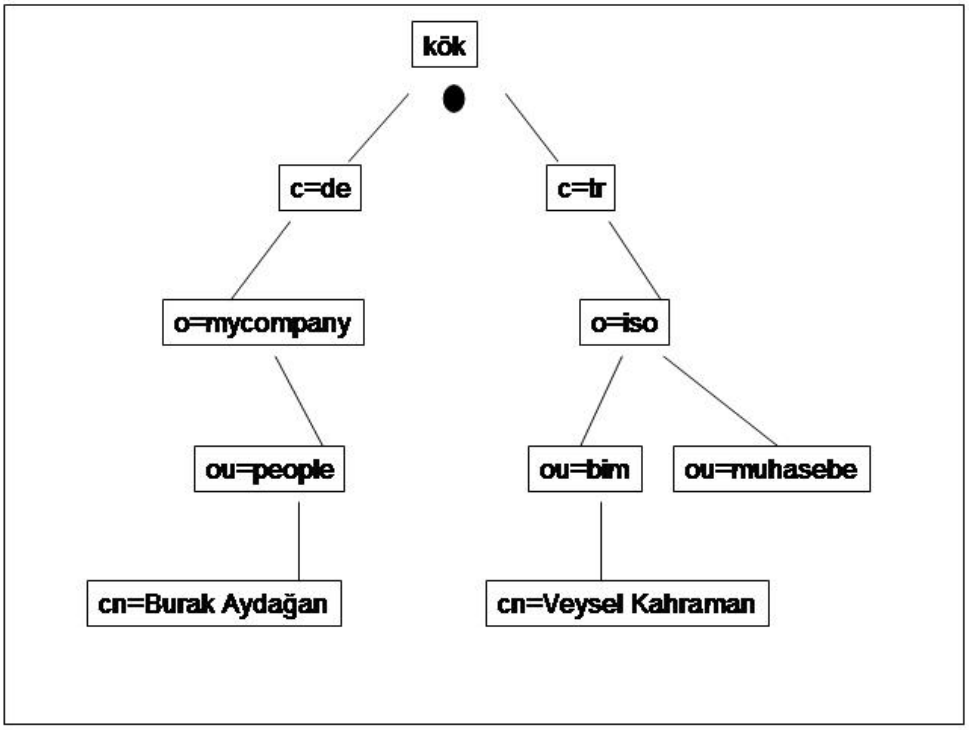
Ağaç üzerindeki her öğe, öğeye ilişkin öznitelik ve değer ikililerinden oluşan bir göreli seçici ada sahiptir. Örneğin Şekil 4'te, "İstanbul Sanayi Odası" adlı kuruma ilişkin göreli seçici ad, {O=iso} olarak verilmiştir. Başka bir tanımlama yapmak

istersek;, Göreli Seçici Ad(Relative Distinguished Names), Seçici Ad (Distinguished Names) içinde bulunan her bir bileşene verilen addır.

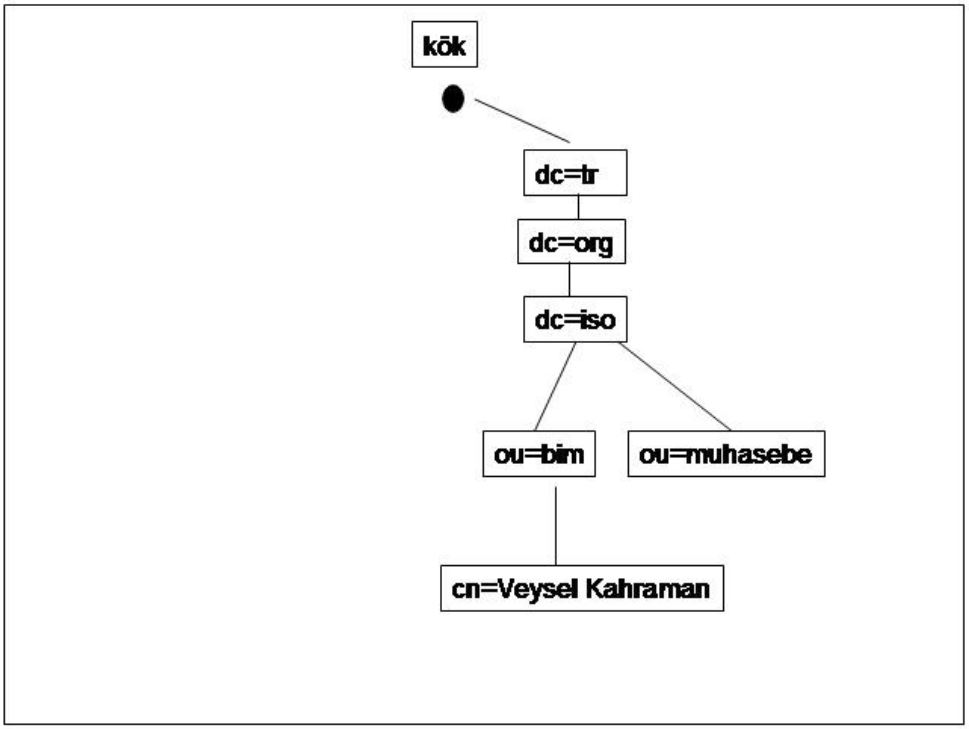
2.6. Seçici Adlar (Distinguished Names)

Ağaç üzerinde her öğenin, onu diğer öğelerden ayıran ve öğenin ağaç üzerindeki konumunu belirleyen bir seçici adı vardır. Bu seçici ad, göreli seçici adlar dizisinden oluşur. Bu dizi, ağacın en dibindeki kök öğesinden başlayıp ilgili öğeye doğru "gidilirken, üzerinden geçilen bütün öğelerin göreli seçici adlarının uç uca eklenmesiyle oluşur. Örneğin, Şekil 4'de "mycompany" adlı firmaya erişmek için kullanılan seçici ad, {O=mycompany, C=de} olarak verilmiştir. Ağaçta daha aşağılara inildikçe, bu seçici adlar ve erişim süresi uzar. Örneğin, "Burak Aydoğan" adlı kullanıcıya ilişkin seçici ad, {CN=Burak Aydoğan, OU=people, O=mycompany, C=de} olarak verilmiştir.

Rehber bilgi ağacını tasarlarken Klasik yöntem (şekil 2.2) veya Internet adlandırma yöntemini (şekil 2.3) kullanabiliriz.



Şekil 2.2 Rehber Bilgi Ağacı (Klasik yöntem)



Şekil 2.3 Rehber Bilgi Ağacı (Internet adlandırma yöntemi)

2.7. Rehber Etki Alanı (Directory Domain) Kök Öğesi (Root Entry) ve Sonek (Suffix)

Rehberda tutulan bilginin güncel kalabilmesi için, ağaç üzerindeki dalların yönetiminin ağaç üzerinde bulunan kurumlara dağıtılması gerekir. Böylece üst otoritenin, alt dallardaki bilgiyi yönetgerek kalmaz. Kendi altındaki dalları yönetme yetkisi olan bölgelerin her birine Rehber Alanı denir. Şekil 4'de, {O=iso} ile tanımlanan "İstanbul Sanayi Odası", kendi altındaki bütün dallarla birlikte bir rehber alanı oluşturur.

Rehber sunucuda saklanan Rehber Bilgi Ağacının en üstünde bulunan öğeye kök öğesi denir. Bu öğeye ilişkin seçici ad, ilgili rehber sunucuda saklanan her öğe için sonektir. Şekil 4'de, İstanbul Sanayi Odasının'nın rehber alanıyla ilgili bilgilerin, kuruma ilişkin rehber sunucuda tutulduğu varsayılırsa, sunucunun rehber ağacının en üstünde bulunan {O=iso} öğesi, bu sunucu için kök öğesidir. Bu kök öğesinin seçici adı {O=iso, C=tr}, bu sunucuda tutulan her öğe için sonektir.

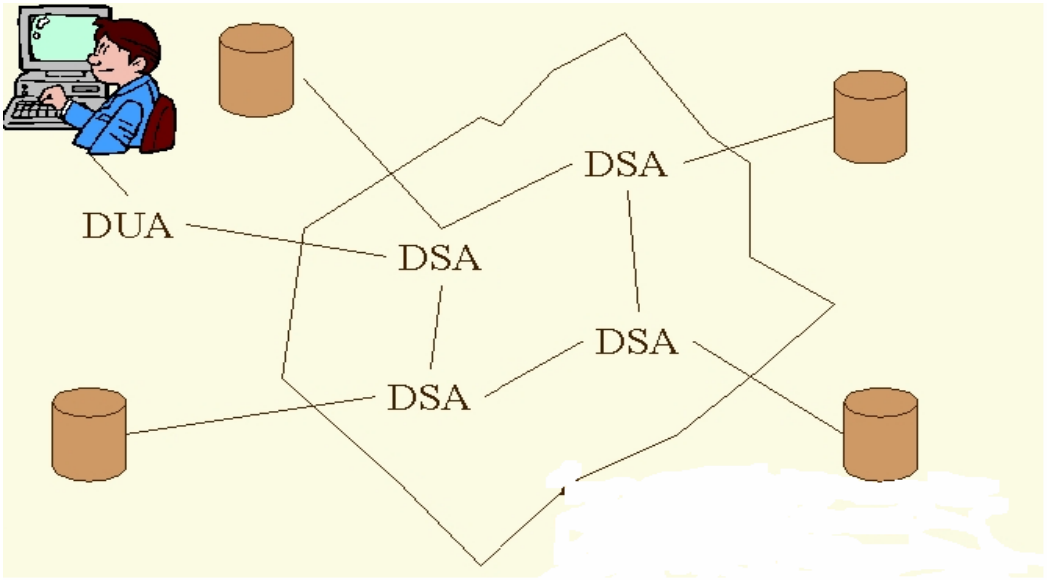
2.8. Rehber Yöneticisi (Directory Manager)

Rehber sunucuda saklanan bilgi üzerinde her türlü günleme ve bakım işlerini yapabilmek amacıyla, ağaç üzerindeki bütün öğelere sınırsız erişim ve günleme yetkisi bulunan kullanıcıya Rehber Yöneticisi denir. Rehber yöneticisi bir rehber ile ilgili her türlü işlemi yapmaya yetkilidir. Rehber yöneticisi özel bir öğedir ve bazen rehber bilgi ağacında saklanmayabilir. Örnek vermek gerekirse açık kod (open source) bir LDAP sunucu yazılımı olan OpenLDAP için Rehber yöneticisi slapd.conf dosyasında tanımlanmaktadır.[7]

2.9. Rehber Kullanıcı Temsilcisi (DUA- Directory User Agent)

Kullanıcılar, rehbere erişirken Rehber Kullanıcı Temsilcisi (DUA) adında bir yapıyı kullanırlar. DUA ile rehber arasında ilk bağlantı kurulduğunda, rehber sunucunun ve DUA'nın yetenekleri konusunda karşılıklı anlaşma sağlanır. Çünkü her rehber, tanımlı her işlemi gerçekleştiremeyebilir, her DUA gelen her yanıtı anlamayabilir. DUA, rehberde bilginin nasıl ve nerede tutulduğuyla ilgili ayrıntıları bilmez, sadece rehbere erişim noktası hakkında bilgi sahibidir. DUA ve rehber yapısı arasındaki ilişki Şekil 2.4'te gösterilmiştir.

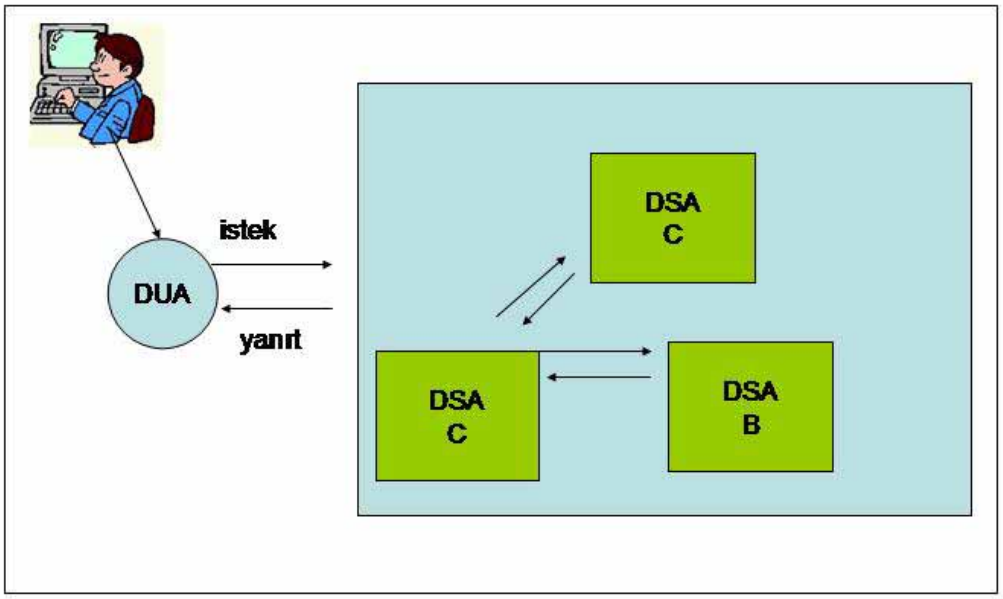
[7] Rehber yöneticisi, OpenLDAP tanım dosyası içinde "rootdn" anahtar kelimesi ile tanımlanmaktadır.



Şekil 2.4 Rehber Sistemine Erişim

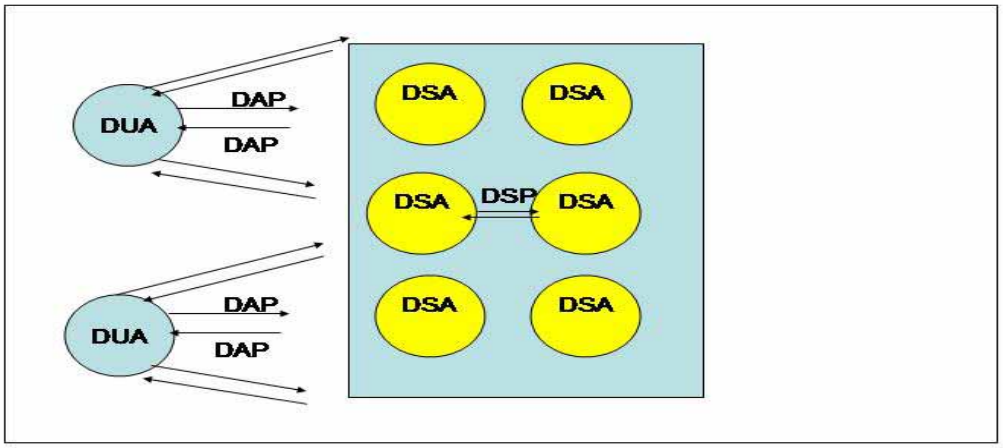
2.10. Rehber Sistemi Temsilcisi (DSA- Directory System Agent) ve Rehber Sistemi Protokolü(DSP- Directory System Protocol)

Rehber Sistemi Temsilcisi, DUA ve diğer DSA'lara Rehber Bilgi Tabanında saklanan bilgiye erişim imkanı sağlayan sunucu görevleridir. DSA ve DSA arası etkileşim, genelde bir referansı takip ederek istenen bilgiye ulaşma amacıyla kullanılır. DUA'dan gelen istemin DSA tarafından alınıp, referanslar takip edilerek istenen sonuca ulaşılması bir yöntemdir. Bu durum Şekil 7 'de gösterilmiştir. Şekilde 7'de, DUA, DSA A'dan bir istemde bulunur. DSA A, bu istemin yanıtı kendinde bulunmadığı için, kendi Rehber Bilgi Tabanından DSA B'ye olan referansı takip eder. DSA B, bu istemin yanıtının kendinde olmadığını anlayıp, istenen bilginin DSA C'de olabileceğini bir referansla DSA A'ya bildirir. Bu durum şekil 2.5' te açıkça görülmektedir.



Şekil 2.5 DUA-DSA Etkileşimi

Dağıtık yapıdaki DSA' lar kendi aralarında DSP (Directory Sistem Protocol) kullanarak iletişim kurmaktadırlar. Bu yapıyı aşağıda yer alan Şekil 2.6 da görülmektedir. Aynı zamanda DUA' lar DSA' lara erişim için DAP yapısını kullanmaktadır.



Şekil 2.6 DSA Etkileşimi

2.11. Erişim Kontrol Listeleri (ACL-Access Control Lists)

Rehberde saklanan her bilgiye, kullanıcıların istediği haklarla erişebilmesi sakıncalıdır. Bu probleme çözüm olarak, erişim kontrol listeleri kullanılır. Kullanıcıların her türlü bilgiye erişimi kısıtlanır. Erişim Kontrol Listelerini kullanarak, rehberin tamamına, rehberin bir kısmına, rehber ağacındaki bazı öğelere, rehberdeki bir öğeye veya arama kriterine göre seçilebilen bütün öğelere ve bu öğelerin özniteliklerine erişim hakları verilebilir. Ayrıca hakların hangi tür kullanıcılara verileceği de belirlenebilir.

3. REHBER SERVİSLERİNİN GELİŞİMİ

Hizmete dönük çeşitli yazılımların ihtiyaç duydukları tanım bilgilerinin birbirinden çok farklı, bağımsız bilgiler olabilmesi ve farklı ortamlarda saklanması yönetim ve bakım maliyetleri açısından büyük sorunlara yol açabilmektedir. Bu yapı içinde, kimi durumlarda aynı bilginin birden fazla kopyası bulunabilmekte yada farklı yerlerde saklanan bilgiler birbiriyle ilişkili olabilmektedir. Birbirleriyle ilişkili bilgiler birlikte güncelleme gerektirir. Birbirlerinden bağımsız bilgilere ilişkin güncellemeler de sorundur. Zira bunların güncellenmesi büyük bir olasılıkla farklı yerlerde yapılacağından sistem yöneticisi açısından fazladan öğrenme ve uygulama zamanı gerektirecektir. Bu da aslında kaynakların verimsiz kullanılmasıdır. Uygulamaya özgü rehber hizmetleri, yönetim maliyetleri yönünden olduğu kadar veri yapıları yönünden de olumsuzluklar içerir. Bir uygulamaya yönelik olarak tasarlanan rehber yapısı, her bilgiyi tutmaya ve her türlü bilgi üzerinden arama yapmaya izin vermez. Örneğin, bir Linux kullanıcısının, telefon, faks, posta adresi gibi bilgileri, sistem dosyalar içinde saklanamaz. Linux'te bunu yapabilmek için, uluslararası kuruluşlar tarafından yayınlanmış bir standart da yoktur. Bu sorunları aşabilmek için, uygulamadan bağımsız bir rehber sistemine ihtiyaç vardır.

Yukarda belirtilen sorunlar, CCITT (Consultative Committee for International Telegraphy and Telephony) tarafından dikkate alınarak, 1988 yılında uygulamadan ve ortamdan bağımsız X.500 Rehber Standartının ilk sürümü geliştirilmiştir. [09]
CCITT daha sonraki yıllarda adını ITU-T (International Telecommunication Union - Telecommunication) olarak değiştirdiği için X.500 ve ilgili standartlar ITU-T standardı olarak anılmaya başlanmıştır.

X.500 rehberlerine erişim için DAP (Diretory Access Protocol) kullanılmaktaydı. X.500 rehberleri ve DAP, yedi katmanlı bir model olan OSI (Open Sytems Interconnect) protokolünü kullanmaktaydılar. Oysa OSI protokolü sisteme çok fazla yük getiren bir yapı sağlamaktaydı. DAP' ın yapısı çok karmaşıktı. DAP çok kapsamlı özelliklere sahip bir protokoldü fakat bu özelikler çoğu uygulama için gereksizdi. [03]

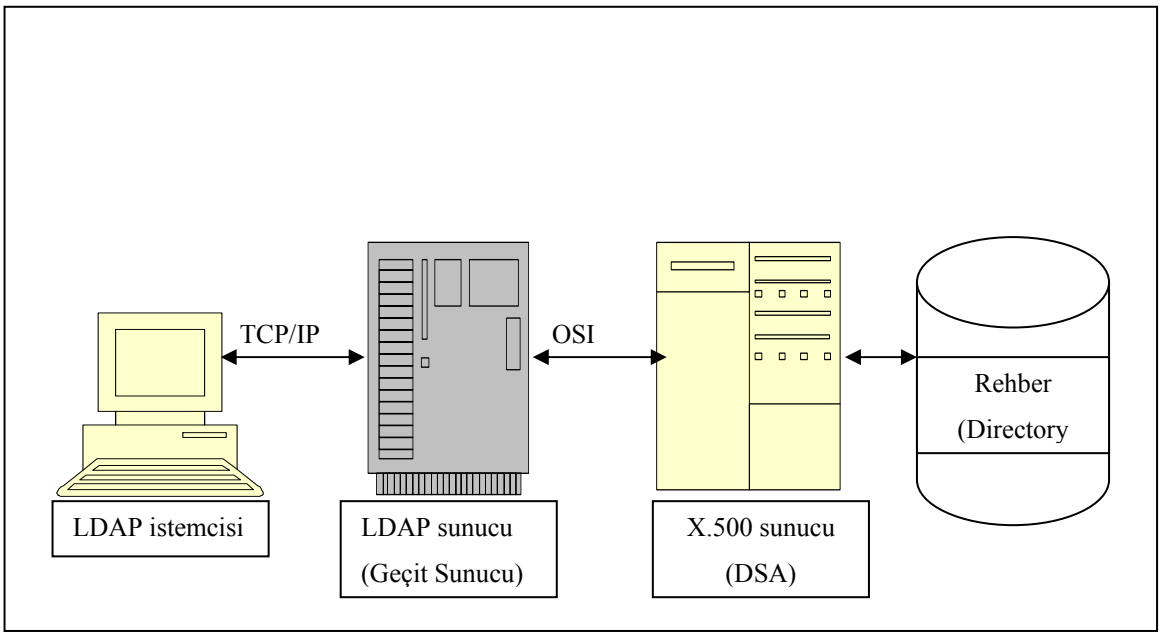
Bu olumsuzlukları dolayı Michigan Üniversitesindeki bir grup tarafından 1992 yılında ilk LDAP gerçekleştirimi hazırlanmıştır.[10] LDAP, X.500 rehberlerinin ve buna erişimi sağlayan DAP protokolünün aksine 4 katmanlı olan TCP/IP protokolü üzerinde çalışmaktadır. Böylece OSI protokolünün olumsuzluklarından (bilgisayarın hafıza ve işlemcisini yoğun kullanması, ağ üzerinde yüksek oranda trafik oluşturması) sistem etkilenmemiş olmaktadır.
LDAP'ın geliştirilmesinden sonra X.500 ve DAP kullanımı zamanla azalmış ve yerini LDAP' a bırakmıştır.

4. X.500 REHBER STANDARTI VE LDAP

LDAP aslında rehber servislerine erişmek için tasarlanmış bir protokoldür. Bir rehber servisi (directory service) değildir. X.500 rehber standartı tanımlandığı zaman, X.500 rehber sunucuyla kullanıcı arasında iletişimi sağlamak amacıyla DAP (Directory Access Protocol) adlı protokol tanımlanmıştı. Bu protokol, bir OSI protokolü olarak tasarlanmıştır. Ancak, OSI protokolleri çok fazla sistem kaynağı(bilgisayar hafızası ve işlemci gücü) tüketimine neden olduğundan, DAP kullanımı yaygınlaşamamıştır.

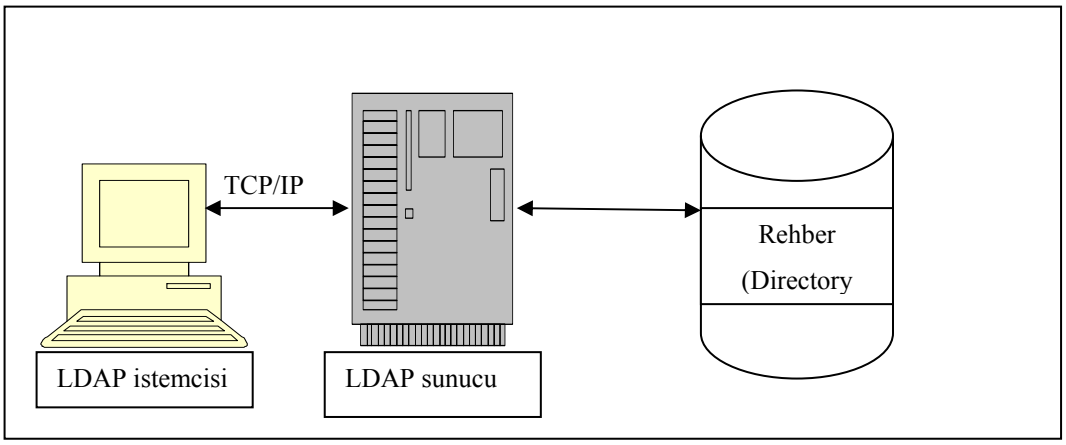
Aynı yıllarda, dört katmanlı olan TCP/IP (Transmission Control Protocol/ Internet Protocol) protokolünün uluslararası bir standart olması, bu konuda çalışanları TCP/IP tabanlı bir rehber erişim protokolü geliştirmeye yönlendirmiştir. Bu çalışmalar sonucunda, adından da anlaşılacağı gibi DAP protokolüne göre daha az kaynağa ihtiyaç duyan LDAP (Lightweight Directory Access Protocol) adlı protokolün ilk sürümü ortaya çıkmıştır [04]. Daha sonraki yıllarda bu geliştirme çabaları devam etmiş ve ikinci sürüm için standartlar yayınlanmıştır [05]. Gelişen teknolojinin yeni gereksinimleri ortaya çıkarmasının bir sonucu olarak, LDAP üçüncü sürümüne ilişkin standartlar da yayınlanmıştır [06]. Üçüncü sürüm, referanslar, güvenlik, uluslarası kullanılabilirlik ve genişleyebilirlik konularında ikinci sürümü geliştirmiştir.

Yukarıda da bahsettiğimiz gibi LDAP, rehber ile kullanıcı arasında iletişimi tanımlayan bir protokoldür. LDAP, istemcinin rehbere erişmesi için gerekli mesaj formatını ve işlem türlerini tanımlar, ancak rehberin yapısı hakkında bir standart tanımlamaz. Rehberin yapısı X.500 standardıyla belirlenir. LDAP, X500 standardıyla belirlenmiş bir rehbere erişip işlem yapmak için, daha önce ve rehberin yapısını gizleyen bir X.500 sunucuya (Directory System Agent-DSA) ihtiyaç duyar. Fakat X.500 sunucular yedi katmanlı OSI protokolünü taban aldığı için, TCP/IP protokolünü taban alan LDAP protokolü ile işlem yapmak mümkün değildir. Bunu mümkün kılmak için, LDAP sunucu olarak adlandırdığımız “geçit sunucuya” ihtiyaç vardır. Bu sunucu bir taraftan LDAP istemci ile TCP/IP protokolünü kullanarak haberleşirken, diğer taraftan X.500 sunucu ile OSI protokolünü kullanarak haberleşir LDAP ikinci sürüm (LDAP V.2) bu şekilde X.500 sunucusu için geçit olarak çalışmaktadır. Şekil 3.1’ de bu çalışma modeli görülmektdir.



Şekil 3.1 LDAP Sunucunun X.500 Sunucu için Geçit Olduğu Mimari

LDAP protokolünün ve TCP/IP'nin kullanımının yaygınlaşması ve OSI protokolünü kullanan ortamların azlığı, geliştiricilerin sadece TCP/IP ortamında çalışan mimariler üretmesine neden olmuştur. X.500 sunucunun aradan kaldırılıp sadece bir LDAP sunucu kullanılması, LDAP istemci açısından bir değişikliğe yol açmazken, TCP/IP protokolünün yaygınlaşması nedeniyle LDAP uygulamalarının daha geniş bir kullanım alanı bulmasına olanak sağlamıştır. Bir X.500 sunucuya ihtiyaç duymadan çalışabilen bu tür LDAP sunucular için stand-alone deyimi kullanılmaktadır LDAP üçüncü sürüm (LDAP V.3) bu şekilde çalışmaktadır. LDAP rehber sunucuları olarak adlandırılan bu yapı X.500 rehber sistemlerinin tüm özelliklerini içermez.(Şekil 3.2).



Şekil 3.2 Stand-Alone LDAP Sunucu Mimarisi

Yukarıdaki gösterilen LDAP sunucu mimarisine örnek olarak OpenLDAP Directory Server, NDIS Directory Server (Novell), Oracle Internet Directory (Oracle-OID) ürünleri gösterilebilir. Bu ürünlerin hepsi LDAP Rehber Sunucuları (LDAP directory servers) olarak isimlendirilmektedir.

LDAP'ın X.500 ve DAP'a göre avantajları aşağıda listelenmiştir.[03]

- Birincisi LDAP direkt olarak TCP protokolü üzerinde çalışmaktadır. DAP ise daha maliyetli olan OSI protokolü üzerine kurulmuştur. LDAP, OSI protokolünün oturum ve sunum katmanlarını (session and presentation layer) gerçeklemez. TCP/IP protokolü üzerinde uygulama (application layer) seviyesinde çalışır. LDAP, varsayılan (default) olarak 389 numaralı portu kullanır. LDAPS (SSL desteği + LDAP) 636 numaralı portu kullanmaktadır.

- İkincisi LDAP X.500 fonksiyonlarını basitleştirmiştir. DAP' ın read ve list fonksiyonları yerine search fonksiyonunu getirmiştir. Anlaşılması zor olan yapıları bırakmıştır.

- Üçüncüsü X.500 basit veri elemanları için bile yapısallığı yüksek olan ve kompleks yapılar kullanmaktadır. Oysa LDAP veri tipleri daha basit bir yapı olan string türündendir.

- Son olarak X.500 tek bir sunucu görüntüsü sergiler, LDAP ise dağıtık bir yapıya sahiptir.

Bir çok uygulama için LDAP'ın performansı yeterlidir. Bununla birlikte LDAP ve DAP protokolleri arasından dört alanda karşılaştırma yapılmıştır.[03]

- Sorgulama Süresi (Tablo 4.1)
- Sorgulama Büyüklüğü (Tablo 4.2)
- PDU kodlama (encoding) hızı (Tablo 4.3 ve Tablo 4.4)
- İstemci uygulamasının büyüklüğü ve karmaşıklığı

Tablo 4.1 Sorgulama Süresi Karşılaştırması

SORGULAMA	**DAP (ms)**	**LDAP (ms)**
Kimlik Doğrulamasız bağlantı	30	68
Kimlik Doğrulamalı Bağlantı	34	56
Basit Arama (1 entry)	32	41
Basit Arama (50 entry)	293	353

Tablo 4.2 Sorgulama Büyüklüğü Karşılaştırması

SORGULAMA	**DAP (ms)**	**LDAP (ms)**
Kimlik Doğrulamasız bağlantı	30	68
Kimlik Doğrulamalı Bağlantı	34	56
Basit Arama isteği	32	41
Basit Arama sonucu	293	353

Tablo 4.3 Kod Çözme (Decoding) Karşılaştırması

PDU Kompleksliği	**DAP**	**LDAP**
Basit	550	110
Orta	7925	714
Karmaşık	38393	2702

Tablo 4.4 Kodlama (Encoding) Karşılaştırması

PDU Kompleksliği	**DAP**	**LDAP**
Basit	24	6
Orta	1084	324
Karmaşık	2656	989

Tablo 4.5 İstemci Gerçekleştirimi Karşılaştırması

Metrik	**DAP**	**LDAP**
Toplam boyut	1484568	334552
Text	958564	221184
Veri	385024	73728
BSS	141080	38640
“;” sayısı	46746	1989
İf sayısı	9369	568

Karşılaştırma tablolarının sonuçlarından görüleceği gibi sorgulama süresi dışındaki kriterlerde LDAP daha avantajlı durumdadır.

5. İLİŞKİSEL VERİTABANLARI

LDAP rehber sunucusu (directory server) yerine getirdiği görev bakımından bir ilişkisel veritabanına (relational database) benzetilebilir. Bununla beraber aralarında pek çok noktada farklılıklar bulunmaktadır.

LDAP sunucular arama (search) işlemleri için optimize edilmişlerdir. Bu konuda çok hızlı sonuç döndürebilmektedirler. İlişkisel veritabanları ise hem arama hem veri üzerinde değişiklik yapma konularında yeteneklilerdir.

LDAP sunucular üzerinde genellikle çok az değişen, statik veriler tutulur. İlişkisel veritabanlarında ise hem statik hemde dinamik bilgiler tutulur.

LDAP sunucular hiyerarşik bir veritabanı yapısı sergilemektedir. Veriler bir ağaç yapısı şeklinde depolanmaktadır. Veri elemanları genellikle birbirinden bağımsızdır. Oysa ilişkisel veritabanlarında veri elemanları arasında karmaşık ve çoklu ilişkiler olabilmektedir.

LDAP sunucularında işlem (transaction) desteği bulunmamaktadır. İlişkisel veritabanlarının neredeyse hepsi bu desteği vermektedirler.
LDAP sunucu üzerinde geçmişe yönelik bilgi saklanması tercih edilmez. Çünkü bu veri miktarını arttıracaktır. Oysa ilişkisel veritablarında geçmişe yönelik bilgiler tutulmaktadır.

LDAP sunuculardа genişleyebilme özelliğine sahip, sabit çekirdek şemalar (core schema) bulunmaktadır. Bu şemalar çoğunlukla standarttır. Oysa ilişkisel veritabanlarındaki tüm şemalar kullanıcı tarafından tanımlanmaktadır. Bu da standart bir yapının oluşmasını engellemektedir.

LDAP sunucularda bilgi bütünlüğü (referential integrity) neredeyse yoktur. Oysa İlişkisel veritabanlarının temel kavramalarından biri bilgi bütünlüğüdür.

LDAP sunucuların genellikle doğası gereği dağıtık bir yapısı vardır. İlişkisel veritabanları ise genellikle merkeziyetçi bir yapı sergilerler.

LDAP sunucularda görüntü, ilişki, yabancı anahtar (view, join,foreign key) gibi karmaşık bilgi modelleri yoktur.

LDAP istemcileri, her türlü LDAP sunucuya sorunsuz erişebilir. İlişkisel veritabanı istemcileri ise sadece belirli tipteki veritabanı sunucularına erişebilirler.

LDAP sunuculardaki verileri ilişkisel veritabanları yerine düz dosyalarda (text files) tutmak isteyebiliriz. Bu durumda hem yönetilebilirlik söz konusu olmayacaktır, hem dosyalarda sadece veri bulunacaktır hem de veriye aynı anda erişmek problem olacaktır.

6. LDAP UYUMLU (LDAP-ENABLED) YAZILIMLAR

LDAP protokolünü kullanan bir istemci uygulaması geliştirmek hemen hemen günümüzdeki her programlama diliyle mümkündür. LDAP uyumlu uygulamalar geliştirmek için her programlama dili bir veya daha fazla yazılım kütüphanesi sağlamaktadır.

LDAP sunuculara erişim tüm programlama dilleriyle mümkün olsa da script dillerle LDAP uyumlu uygulama geliştirmek diğer dillere göre çok daha kolaydır. Performans olarak derlenen programlama dillerine göre çok az hız farkları vardır. Ama yine geliştirilecek LDAP uyumlu (LDAP enabled) yazılım için performans çok önemli ise C/C++ gibi derlenen diller tercih edilmelidir.

Script dillerle uygulama geliştirmenin bir çok avantajı bulunmaktadır. Script dillerle uygulama geliştirme maliyeti (overhead) derlenen (compile) dillerle uygulama geliştirme maliyetiyle karşılaştırıldığında script dillerin daha uygun bir çözüm olduğu görülür. Derlenen programlama dillerinin kaynak kod satır sayıları çok yüksek olmaktadır. Aynı zamanda derlenen programlama dillerinde bir çok kaynağın yönetimi programcının sorumluluğundadır. Örneğin PHP nin LDAP fonksiyonları arka planda C SDK sını kullanır. Fakat PHP ile LDAP erişimi doğrudan C SDK sı ile erişime göre çok daha kolaydır. PHP, Perl ve Python en popüler CGI scirpt dillerindendir. İçlerinde gelen bir çok modül sayesinde bir çok teknolojiye destek verirler. Özellikle bu destek veritabanı erişimi konusunda göze çarpar.

Günümüzün en popüler üç script dili ile LDAP uyumlu yazılımların geliştirilmesi aşağıda incelenmiştir.[07]

6.1 PERL içinde LDAP protokolü kullanımı:

Perl için 3 farklı LDAP gerçekleştiriminden söz etmek mümkündür

- Net::LDAPapi : 1998 den beri geliştirilmemektedir. Geliştiricisi tarafından artık desteklenmemektedir.
- PerLDAP : Mozilla projesi tarafından sağlanmaktadır. C SDK' sını kullanır. Bazı modülleri V3' ü desteklemektedir.
- Perl-LDAP : LDAP için doğal Perl gerçeklekleştirimidir. C SDK'sına ihtiyaç duymaz. Diğer sistemlere uyumlu hale getirilmesi en kolay olan yapıdır.

Aşağıda Şekil 6.1' de Perl-LDAP kütüphanesi kullanılarak yazılmış örnek bir arama programı yer almaktadır.

```
#!/usr/bin/perl

use NET::LDAP;

$ldap=Net::LDAP->new('ldap');
#$ldap değişkeni  bağlantıyı tutmaktadır(connection handler),
```

```
#ldap ise LDAP sunucunun adıdır.

$ldap->bind(dn=>'cn=admin,o=yoyodyne', password=>'plaintext');
$result=$ldap->search(basedn=>'o=yoyodyne',
        scope=>'one',filter=>'cn=admin',
        attrs=>['surname','mail','groupMembership']);
$result->code && warn $result->error;
foreach $entry ($result->all_entries) {
        @surname=$entry->get('surname');
        @mail=$entry->get('mail');
        @groupMembership=$entry->get('groupMembership');
        $dn=$entry->dn;
        print ("Info for $dn:\n");
        print ("Surname: ");
        foreach $surnamevalue (@surname) {print "\"$surnamevalue\" ");}
        print ("\n");
        print ("email: ");
        foreach $mailvalue (@mail) {print "\"$mailvalue\" ");}
        print ("\n");
        foreach $groupmembershipvalue (@groupmembership) {print
"\"$groupmembershipvalue\" ");}
        print ("\n\n");
        }
$ldap->unbind;
```

Şekil 6.1 LDAP işlemi için örnek PERL kaynak kodu

6.2 Python içinde LDAP protokolü kullanımı:

Ptyhon hem CGI hemde komut satırı programları geliştirmek için kullanılan popüler bir dildir. Guido Van Rossum tarafından geliştirilmiştir. Python program dizaynını okunabilir olmaya zorlar. Dikkat edilirse python kaynak kodunun okunurluğu göze çarpacaktır.

Yukarıda verilen örnek PERL kodunun yaptığı işi yapan Python kodu aşağıda Şekil 6.2' de verilmiştir.

```
#!/usr/bin/env python

import ldap

l=ldap.open('ldap')
l.simple_bind_s('cn=admin,o=yoyodyne','plaintext')
res = l.search_s('o=yoyodyne', ldap.SCOPE_ONELEVEL, "cn=admin",
["surname", "mail", "groupMembership"])
for entry in res:
        attrs=entry[1]
        surname=attrs["surname"]
        mail=attrs["mail"]
        groupmembership=attrs["groupMembership"]
        dn=entry[0]
        print "Info for %s:\n" % (dn,)
        for surnamevalue in surname:

        print "\"%s\" " % (surnamevalue,)
        print "\n"
        for mailvalue in mail:
                print "\"%s\" " % (mailvalue,)
        print "\n"
        for groupmembershipvalue in groupmembership:
                print "\"%s\" " % (groupmembershipvalue,)

        print "\n"

l.unbind()
```

Şekil 6.2 LDAP işlemi için örnek Python kaynak kodu

6.3 PHP içinde LDAP protokolü kullanımı:

PHP daha çok gömülü CGI programcıkları yazmak için kullanılan Perl, Basic ve ASP özelliklerini barındıran bir script dilidir. Komut satırı programlarında pek kullanılmaz. Yukarıda verilen örnek PERL kodunun yaptığı işi yapan PHP kodu aşağıda Şekil 6.3' de verilmiştir.

```
<?php

$ldap = ldap_connect("ldap");

ldap_bind($ldap, "cn=admin,o=yoyodyne", "plaintext");
$result = ldap_search($ldap, "o=yoyodyne", "cn=admin", array( "surname",
"mail" ,"groupMembership"));
foreach (ldap_get_values($ldap) as $entry) {
        $dn=$entry["dn"];
        $surname=$entry["surname"];
        $mail=$entry["mail"];
        $groupmembership=$entry["groupmembership"];
        print "Info for $dn\n";
        foreach ($surname as $surnamevalue) {
                print "\"$surnamevalue\" ";
                }
        print "\n";
        foreach ($mail as $mailvalue) {
                print "\"$mailvalue\" ";
                }
        print "\n";
        foreach ($groupmembership as $groupmembershipvalue) {
                print "\"$groupmembershipvalue\" ";
                }
```

```
        print "\n";
        }
ldap_unbind($ldap);
?>
```

Şekil 6.3 LDAP işlemi için örnek PHP kaynak kodu

7. LDAP İLE REHBER İŞLEMLERİ

Kullanıcıların rehber sistemleri üzerinde gerçekleştirebileceği işlemlerin ilk tanımı X.500 standart belgesinde yer almıştır. Fakat LDAP standartı ile ilgili belgelerde, X.500 standartında tanımlı işlemlerin bir kısmı birleştirilip tek bir işlem haline getirilmiş ve bazı yeni işlemler eklenmiştir. Bu yüzden, bu işlemleri LDAP standartları bağlamında anlatmak daha uygun olacaktır. LDAP işlemlerini, sorgulama (query), günleme (update) ve kullanıcı tanıma (authentication) işlemleri olmak üzere üç gruba ayırabiliriz. Sorgulama işlemleri arama, karşılaştırma ve iptal etme işlemleri olmak üzere üç tanedir. Günleme işlemleri ekleme, silme, günleme ve seçici ad değiştirme işlemleridir. Kullanıcı tanıma işlemleri rehbere bağlanma ve bağlantıyı kapatma işlemleridir.

7.1 Arama (Search)

Arama işlemi; Rehber Bilgi Ağacında tutulan bilginin tümü veya bir kısmı üzerinde,kullanıcının belirlediği bir arama kriterine göre arama yapmayı ve geri dönen sonuçları okuyabilmeyi ve listelemeyi sağlayan bir işlemdir. X.500 standart belgesinde okuma (read) ve listeleme (list) şeklinde iki işlem tanımlanmıştır. Okuma işlemi, bir öğeye ilişkin öznitelik değerlerinin okunabilmesini sağlayan bir işlemdir.Listeleme, Rehber Bilgi Ağacında bir öğenin hemen altında bulunan öğelerin listelenmesini sağlayan işlemdir.

LDAP standartında bu iki işlem arama işleminin içine gömülmüştür.

Arama işlemi genel veya özel olabilmesi nedeniyle en karmaşık ağaç işlemidir. Şekil 4'deki Rehber Bilgi Ağacı üzerinde yapılabilecek arama işlemlerine şöyle birkaç örnek verilebilir:

• Ağaç üzerindeki bütün kişilere ilişkin ad, soyad bilgilerini bulma.

•İstanbul Sanayi Odası Muhasebe Bölümündeki kişilere ilişkin ad, soyad, telefon numarası ve adres bilgilerini bulma.

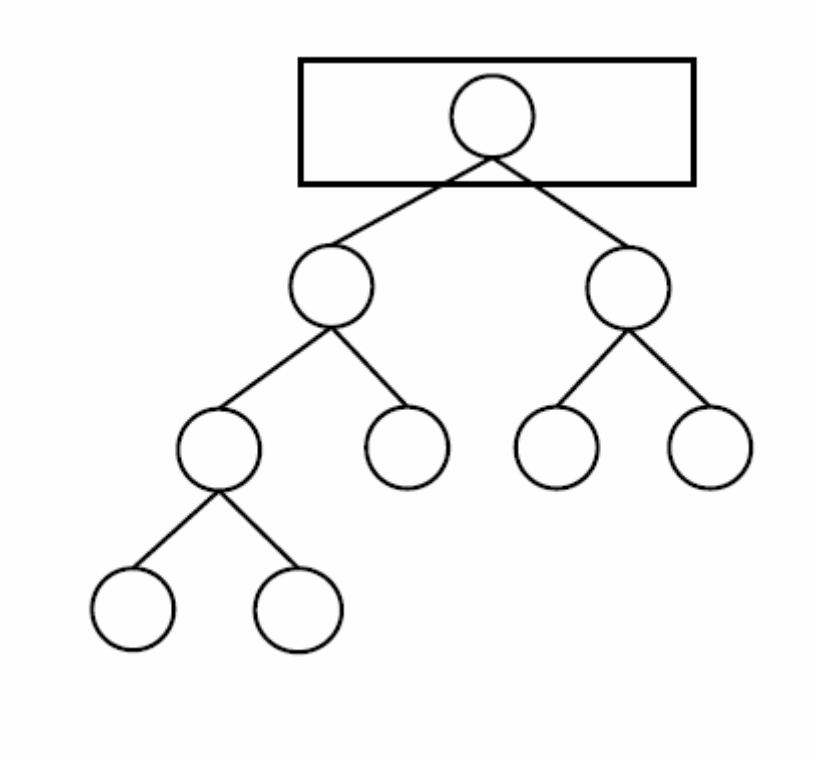
• İstanbul Sanayi Odasında, adında "Ahmet" geçen kişilerin ad, soyad ve telefon numarası bilgilerini bulma.

Arama işlemi için aşağıdaki parametrelere ihtiyaç vardır:

• **Arama Başlangıç Noktası (Search Base)**: Ağaç üzerinde aramanın başlayacağı öğenin seçici adı.

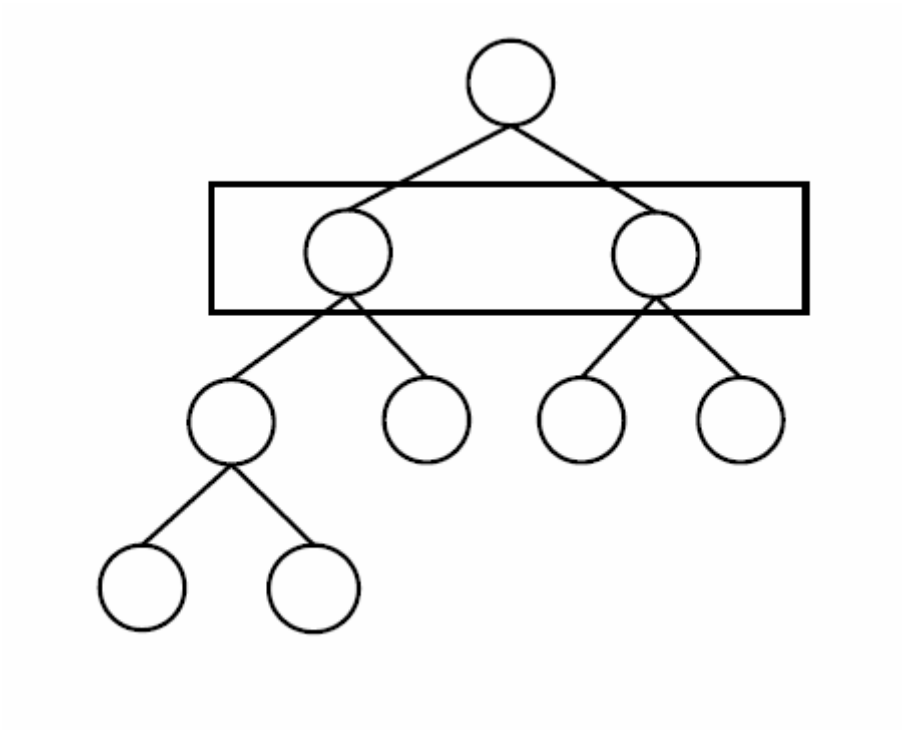
• **Arama Alanı (Scope)**: Aramanın ağaç içinde hangi derinliğe kadar yapılacağını ifade eder. Arama üç düzeyde olabilir:

a) baseObject düzeyi: Arama kriterinin yalnız arama başlangıç noktasındaki öğe üzerinde deneneceğini belirtir. Şekil 7.1' de bu durum gösterilmektedir.



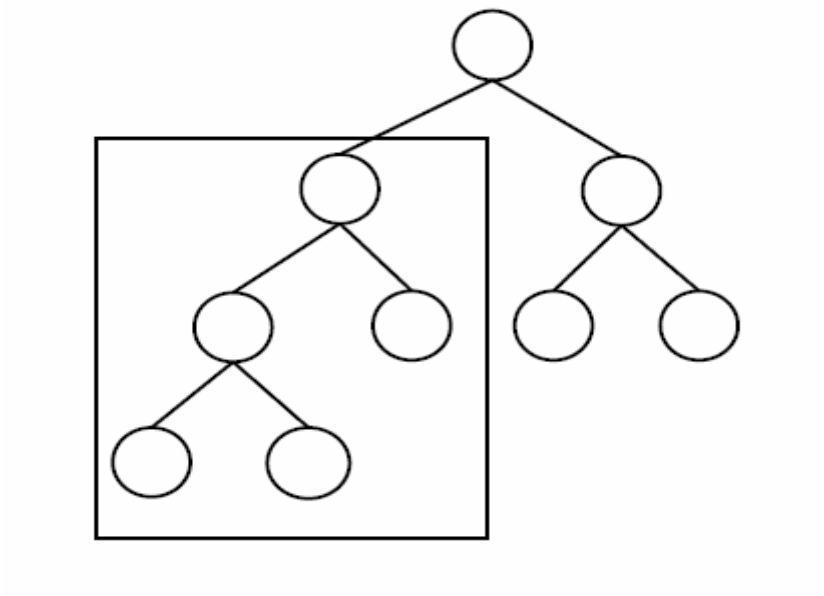
Şekil 7.1 Baseobject düzeyi

b) singleLevel düzeyi: Başlangıç noktasının yalnızca bir düzey altındaki öğeler üzerinde arama yapılacağını belirtir. Şekil 7.2' de bu durum gösterilmektedir.



Şekil 7.2 Singlelevel düzeyi

c) wholeSubtree düzeyi: Aramanın, başlangıç noktasından başlayarak bütün alt ağaç üzerinde yapılacağını belirtir. Şekil 7.3' de bu durum gösterilmektedir.



Şekil 7.3 Subtree düzeyi

• **Arama Kriteri (Search Filter)** : Arama sonunda geri döndürülecek öğelerin nasıl seçileceğini belirleyen kriterlerdir. Örneğin, adında "Mehmet" geçen kullanıcılar bir arama kriteridir.

• **Geri Döndürülecek Öznitelikler (Return Attributes)**: Arama sonucunda seçilen öğenin hangi özniteliğinin istemciye döndürüleceğini belirler. Bu parametre, arama sonucunda sadece istenen bilginin istemciye döndürülmesini sağlar. Örneğin, kişiler üzerinde yapılan bir aramada, kişilerin sadece ad ve soyadına ihtiyaç varsa, bu parametre sayesinde, arama sonucunda sadece bu bilgilerin istemciye döndürülmesi sağlanabilir. Böylece seçilen kişilerin bütün öznitelikleri döndürülmemiş olur.

7.2 Karşılaştırma (Compare)

Karşılaştırma işlemi, bir öğeye ilişkin bir özniteliğin kullanıcı tarafından verilen bir değerle karşılaştırmasını yapar. Karşılaştırma sonucunda değerler birbirine eşitse TRUE, değilse FALSE değeri döndürülür. Bu işlem, örneğin, kullanıcı tarafından girilen şifreleri kontrol etmek için kullanılabilir.

7.3 İptal Etme (Abandon)

Zaman uyumsuz rehber işlemlerinin , sonucu gelmeden önce iptal edilmesini sağlar.

7.4 Ekleme (Add)

Rehbere yeni bir öğe eklenmesini sağlar. Ekleme işlemi sonucunda, işlemin başarılı olup olmadığı bilgisi döndürülür.

7.5 Silme (Remove)

Rehberdeki bir öğenin silinmesini sağlar. Silme işlemi yalnızca ağacın en altındaki yaprak öğeleri üzerinde yapılabilir. Eğer bir öğenin altında alt öğeler varsa bu öğe

silinemez. Takma ad öğeleri silme işlemi sırasında takip edilmez. Silme işlemi sonucunda, işlemin başarılı olup olmadığı bilgisi döndürülür.

7.6 Günleme (Modify)

Öğelerin özniteliklerinin değerlerini günlemek için kullanılır. Günleme işlemi, bir özniteliğe yeni değer ekleme, özniteliğin bir değerini veya değerleri silme ve değiştirme şeklinde olabilir.

7.7 Seçici Ad Değiştirme (Modify Distinguished Names)

Öğelerin seçici adlarını değiştirmek veya ağaç üzerinde bir alt dalın yerini değiştirmek amacıyla kullanılır. Öğelerin seçici adları değiştirilirken, sadece göreli seçici ad kısmı değiştirilebilir. Bir alt dalın yeri, sadece aynı sunucu üzerinde değiştirilebilir, sunucular arası değişim olamaz.

7.8 Bağlanma (Bind)

LDAP oturum-tabanlı bir protokoldür. LDAP istemcinin sunucudan istemlerde bulunabilmesi için bir oturum açması gerekir. Bağlanma işlemi, kullanıcı tanıma işlemini gerçekleştirdikten sonra, LDAP istemci ile sunucu arasındaki oturumu başlatır. Kullanıcı tanıma işlemi için üç değişik yol tanımlanmıştır .

7.9 Bağlantıyı kapatma (Unbind)

LDAP istemci ile sunucu arasındaki oturumu kapatmak için kullanılır.

7.10 Rehber İşlemlerinden Geri Döndürülen Diğer Sonuçlar

Rehber işlemlerinden geri döndürülen bilgiler sadece öğelere ilişkin öznitelik bilgileri değildir. Aslında arama işlemi dışındaki işlemlerin hiçbiri, böyle bir sonuç

kümesi döndürmez. Rehber işlemleri sonucunda üç farklı sonuç daha dönebilir:• **İşlem Sonucu**: İşlemin başarılı şekilde tamamlanıp tamamlanmadığına ilişkin döndürülen değerdir.

• **Hata Kodları**: Bir işlemin doğru bir şekilde sonlandırılamadığı durumlarda, hatanın nereden kaynaklandığını bildirmek amacıyla istemciye döndürülen değerlerdir.

• **Referanslar**: Daha önceki kısımlarda anlatıldığı gibi, aranan bilginin rehber sunucuda bulunamadığı durumlarda, bilginin bulunabileceği sunuculara yönlendirmeyi sağlamak amacıyla kullanılan bir yoldur.

8. LDAP GÜVENLİK MİMARİSİ

LDAP protokolünde güvenlikten sözetmeden önce, güvenlik teriminin hangi kavramları içerdiğini anlamak gerekir. İletişim bağlamında güvenlik, genel olarak dört kavram esas alınarak incelenir:

• Kullanıcı Tanıma (Authentication) : Karşıdaki kişinin kim olduğundan veya olduğunu iddia ettiği kişi olup olmadığından emin olmak.

• Bütünlük (Integrity) : Alıcıya gönderilen iletinin, içeriği bozulmadan gönderildiği haliyle ulaşmasını sağlamak.

• Gizlilik (Confidentiality) : Alıcıya gönderilen iletideki bilginin gizliliğini koruyarak, sadece alıcı tarafından okunabilmesini sağlamak.

• Yetkilendirme (Authorization) : Bir istemde bulunan tarafın, bu istemde bulunmaya yetkili olup olmadığını denetlemek.

LDAP protokolünün güvenlik mimarisi, önceki kısımlarda kısaca değindiğimiz bağlanma (bind) işlemine dayanmaktadır. Dolayısıyla yukarıda anlattığımız kavramlar bağlanma işlemi bağlamında ele alınmaktadır. LDAP protokolünün

3.sürümünde, bu dört kavramdan yetkilendirme dışındakilere çözümler sunulmaktadır. Yetkilendirme işlemleri, X.500 Rehber Standartının bir parçası olan Erişim KontrolListeleri kullanılarak yapılır.

8.1 Kullanıcı Tanıma Olmadan Bağlanma (Anonymous)

Bu bağlanma yönteminde, kullanıcı kendini tanıtıcı herhangi bir bilgi vermeden, rehber sunucuya bağlanır. Kullanım amacı, herhangi bir kullanıcı tarafından erişilmesinde sakınca olmayan (public) bilgilere anonim erişimi sağlayabilmektir.
Genelde, bu yöntemle bağlanan kullanıcılara, sadece bilgileri okuma amaçlı izin verilir. Kuruma, kişilere, altbölümlere ilişkin telefon, adres, fax, e-posta bilgileri, anonim erişime açılabilecek bilgilere örnek olarak verilebilir.

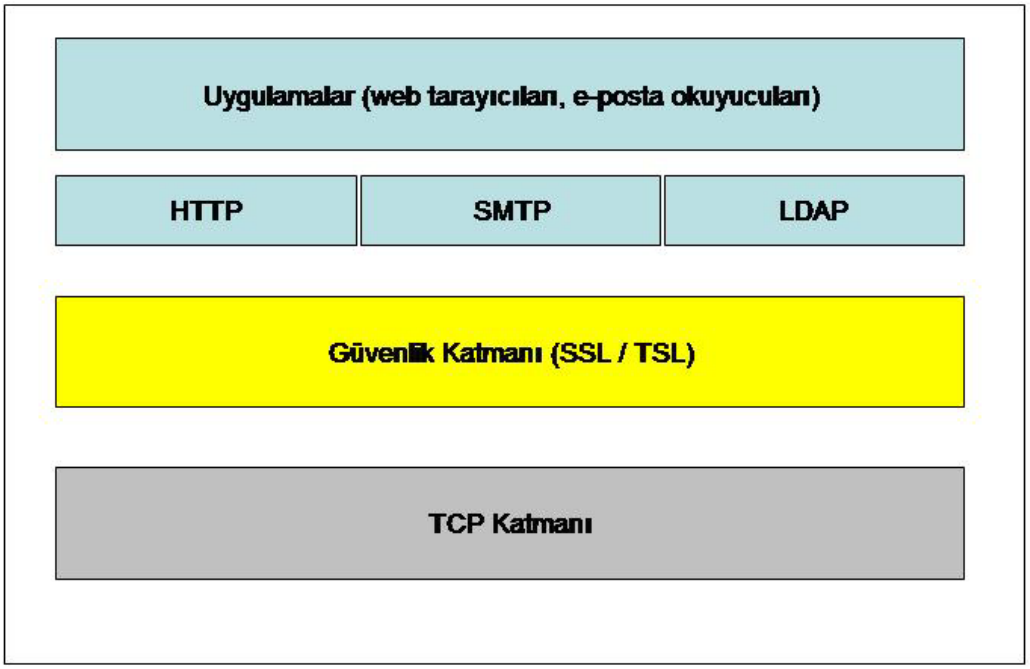
8.2 Basit Kullanıcı Tanıma ile Bağlanma (Simple Authentication)

Bu yöntemde kullanıcı, kimliğini tanıtıcı bir bilgi ve şifre bilgisi vererek rehbere bağlanır. Telnet, FTP gibi birçok İnternet uygulamasında kullanılan yönteme benzerdir. Fakat, Rehber Hizmetlerinde kimlik tanıtıcı bilgi olarak bir kullanıcı adı(login) yerine, Rehber Bilgi Ağacı üzerinde kullanıcıya ilişkin nesne öğesinin seçici adı girilir. Bu yöntemin kullanım amacı, bağlanan kişilere bir kimlik vermek ve böylece anonim olarak bağlanan kullanıcılara verilmek istenmeyen bazı yetkileri verebilmektir. Bu bağlanma yöntemi sayesinde, örneğin, kişilere ilişkin bilgileri günleme yetkisi, birkaç özel kullanıcıya veya kişilerin kendilerine verilebilir.

8.3 Basit Kullanıcı Tanıma ve Güvenlik Katmanı ile Bağlanma (SASL-Simple Authentication and Security Layer)

İlk iki bağlanma yönteminde, güvenlik açısından dikkate alınan tek kavram kullanıcı tanımaydı. Bu yöntemlerde, kullanıcı ile iletişimde açık metin (clear text) halindeileti gönderilir. Bu durum İnternet üzerinde kötü niyetli kişilerin saldırısına açık bir yaklaşımdır. SASL bağlanma yöntemi, bütünlük ve gizlilik sorunlarına çözümbulabilmek amacıyla, SSL (Secure Socket Layer) ve TLS (Transport Layer

Security).ara güvenlik katmanlarını kullanır. SSL ve TLS, bütünlüğü kontrolü yapabilmek için mesaj özeti yaratma algoritmalarını, gizliliği sağlamak için şifreleme algoritmalarını kullanmaktadır. SSL, Netscape firması tarafından geliştirilen bir ara güvenlik katmanıdır. Kullanıcı tanıma amacıyla açık anahtar şifreleme algoritması (Public Key Cryptosystem) ve sayısal sertifikaları (Digital Certificates); mesaj özeti üreterek bütünlük denetimi yapmak amacıyla SHA (Secure Hash Algorithm) ve MD5 algoritmalarını; mesaj içeriğini şifreleyerek gizliliği sağlamak amacıyla DES, 3DES gibi şifreleme algoritmalarını kullanır. TLS, SSL 3.0 sürümüne göre IETF (Internet Engineering Task Force) tarafından geliştirilmekte olan bir standarttır. SSL ve TLS'nin Internet protokolleri arasındaki yeri Şekil 8.1' de gösterilmiştir.



Şekil 8.1 SSL ve TSL' in bulunduğu katman

9. LDIF (LDAP Data Interchange Format)

Aktarma işlemini tek tek öğe bazında yapmak çok zahmetli ve zaman alıcı bir iştir ve genelde çoğu durum için uygun bir çözüm değildir. Büyük miktardaki verilerle işlem yapabilmek için, LDIF dosya formatı tanımlanmıştır. Bu dosya formatı, rehbere veri yüklemek (import) ve rehberden veri almak (export) için kullanılır. Çok basit, iyi tanımlanmış bir dosya formatı olduğu için, veri aktarımı sırasında birçok yazılım aracıyla birlikte kullanılabilir. Bu dosya formatı veri yükleme gibi toplu işlemler için kullanılabileceği gibi, rehber bilgisi üzerinde günleme, silme, ekleme gibi tekil işlemler için de kullanılabilir. Daha genel bir ifadeyle, Rehber Bilgi Tabanında yapılacak her türlü günleme için kullanılabilir.

LDIF dosya formatı, tutanaklar dizisinden oluşur. Bu dosyadaki her bir tutanak, bir öğeye ilişkin bilgileri veya varolan bir öğe üzerinde yapılacak günleme işlemlerini ifade eden ardışık satırlardan oluşur. Öğe bilgilerini içeren tutanaklar, rehbere yeni öğeler eklemek amacıyla kullanılır. Günleme işlemlerini içeren tutanaklar ise rehber ağacındaki öğeler üzerinde günleme, silme, ekleme yapmak amacıyla kullanılabilir.
Bir LDIF dosyasında, bu iki tür tutanak birden bulunamaz, yalnız bir tür bulunmalıdır. LDIF dosya formatı, Şekil 9.1' de gösterilmiştir.

```
[<id>]
dn: <distinguished name>
objectClass: <object class>
[objectClass: <object class>]
...
[<attribute type>[;language tag]:<attribute value>]
[<attribute type>[;language tag]:<attribute value>]
```

Şekil 9.1 LDIF Dosya Formatı

Bu dosya formatındaki deyimlerin tanımları şöyledir:

• id : Öğeye ilişkin seçimli öğe numarası.

• dn : Öğenin zorunlu seçici adı.

• objectClass : Öğenin nesne sınıfı. En az bir adet bulunması zorunludur.

• attribute type : Öğeye ilişkin öznitelik.

• attribute value : Öğenin bir özniteliğine ilişkin değer.

• language tag : Öznitelik değerinin ifade edilmesinde kullanılan dil.

Yukarıdaki tanımlara göre, bir öğeye ilişkin verilen bilgilerden sadece dn satırı ve en az bir adet objectClass satırı zorunludur. Fakat, eğer öğenin nesne sınıfınca belirlenmiş zorunlu öznitelikler varsa, bu özniteliklere ilişkin attribute type satırları da bulunmalıdır.

10. REHBERDE TUTULACAK VERİLER

Rehberde tutmayı planladığımız bilgilerin bazı özelliklere sahip olması rehberi daha amaca yönelik kullanmamızı sağlayacaktır. Bununla birlikte, istendiği takdirde rehber sistemlerinde her türlü verinin tutulabileceği unutulmamalıdır.

Rehberde saklanacak bilgiye sadece bir kişi erişecekse, bu bilginin rehber sisteminde saklanması anlamlı değildir. Rehber sistemleri ancak çok kullanıcı sistemlerde kullanılırsa amacına hizmet edebilir.

Rehber, daha çok okuma ağırlıklı bilgileri saklamak amacıyla kullanılmalıdır. Sık sık üzerinde değişiklik yapılan bilgiler, rehberde saklanmak için uygun değildir. Bu tür işlemler için ilişkisel veritabanları en uygun çözümdür.[8]

Rehberde büyük boyutlu ve yapısal olmayan bilgilerin saklanması doğru değildir. Örneğin bir kurumdaki ofis dokumanlarının saklanması için rehber sistemleri tavsiye edilmez. Bunun yerine bu tür dosyalar dokuman yönetimi üzerine özelleşmiş bir veri

[8] İlişkisel veritabanları bölümünde ayrıntılı olarak rehber sistemleri ve ilişkisel veritabanları araındaki farklardan bahsedilmiştir.

tabanında tutulmalı yada bu bilgilere Internet üzerinden de erişilecekse bir FTP (file transfer protocol) sunucusu üzerinde saklanmalıdır.

Rehber sunucuları üzerinde tutulabilecek bilgiye en güzel örnek e-posta uygulamalarıdır. İleriki kısımlarda bu işlemin nasıl yapılacağı ayrıntılı olarak anlatılacaktır.[9]

12. LINUX İŞLETİM SİSTEMİNDE KULLANICI HESABI YÖNETİM SEÇENEKLERİ

Önceki kesimlerde anlatıldığı gibi UNIX ilk geliştirildiği yıllarda kullanıcı hesabı yönetimi tamamen text dosyalar üzerinde ve kullanıcılar sadece o makine üzerinde tanımlı olacak şekilde yapılıyordu. Fakat kullanıcı sayısı artmaya başladıkça merkezi bir kullanıcı hesabı yönetim sistemi zorunlu hale gelmeye başlamıştır. Kullanıcı sayısı binlerle ifade edilmeye başlandığında ise yönetim imkansız hale gelmektedir.
Linux sistemler üzerinde kullanıcı hesaplarının merkezi bir noktadan yönetimi için Sun firması tarafından geliştirilen NIS (Network Information System) ve LDAP rehber servisi alternatifleri mevcuttur.

Fakat burada dikkat edilmesi gereken çok önemli bir nokta vardır. LDAP ile normal şartlar altında kullanıcılar için kimlik doğrulama (authentication) yapılamaz. Çünkü LDAP bir kimlik doğrulama mekanizması değil, bir rehber servisi veya başka bir deyişle hiyerarşik bir veritabanıdır. Bu yüzden LDAP sistemini kullanıcı hesabı yönetiminde kullanabilmemiz için NSS ve PAM sistemlerinin LDAP ile beraber kullanılması gerekmektedir.

NIS sistemi sadece UNIX/Linux sistemlerinde çalışmaktadır. Bu büyük bir dezavantajdır. Oysa LDAP neredeyse bütün sistemler tarafından desteklenmekte ve standart olarak kabul edilmektedir. Aynı zamanda LDAP rehber sunucularda tutulabilecek veri konusunda ise hiçbir sınır yoktur. NIS sisteminin arama kabiliyeti

[9] Bu işlem için tamamen açık kaynak koda sahip olan OpenLDAP yazılımı kullanılmıştır.

çok zayıftır. Erişim kontrol mekanizmasına (ACL) sahip değildir. Bu sebeplerden dolayı LDAP rehber servisleri NIS sistemlerine tercih edilmekte ve NIS sistemlerinin kullanımı gün geçtikçe azalmaktadır.

12.1. NIS (Network Information System)

Sun Microsystems tarafından geliştirilmiştir. Unix ve Linux işletim sistemleri üzerinde kullanılmaktadır.[10] NIS Network Information Service kelimelerinin baş harflerinden oluşmaktadır. Amacı bir ağ üzerindeki bütün makinalara, bütün ağ tarafından bilinmek kaydıyla bilgi akışı sağlamaktır. NIS tarafından dağıtılacak bilgiler, kullanıcı hesap isimleri, parolalar, kullanıcı dizinleri ve grup bilgileri olabilir.

NIS yardımıyla, NIS veri tabanında kayıtlı bir kullanıcı , ağ üzerinde NIS istemci programlarını çalıştıran herhangi bir makineden sisteme giriş yapabilir.NIS'in gelişmiş bir sürümü olan NIS+ , verileri korumak amacıyla şifreleme yöntemi kullanır. Fakat NIS sistemine göre çok daha karmaşık bir yapıya sahiptir.

NIS Servisi kullanıldığında, tüm sistemler tek bir veritabanından (MAPS) yönetilir. Değişiklikler tek bir noktadan yapılır ve otomatikman diğer sistemler de etkilenir. Bu sayede istemci makineler tek bir veritabanını ağ üzerinden kullanabilirler. NIS, Linux yerel tanım dosyalarında saklanan bilginin, NIS sunucu yazılımları ile diğer Linux sunuculara dağıtılmasını sağlayarak yönetim işlerini merkezileştirmeyi amaçlayan bir sistemdir.

NIS, yerel tanım dosyalarında bilgiyi dbm formatındaki dosyalarda saklar. Bu dosyalardaki bilgiler birincil sunucunun (master server) yerel tanım dosyalarından elde edilir. Sistem ile ilgili bilgiler günleneceği zaman, önce birincil sunucu üzerindeki yerel tanım dosyaları günlenir. Daha sonra değişen dosyaları bilgilerden yararlanılarak, dbm formatındaki dosyalar tekrar oluşturulur. Dbm formatındaki bu dosyalar, önceden belirlenen kriterlere göre arama yapmaya olanak sağlayacak

[10] Günümüzün popüler Linux dağıtımlarının çoğunda NIS, NIS+ ve LDAP kurulum sırasında seçenek olarak sunulmaktadır.

şekilde yaratılır. Örneğin, birincil sunucu üzerinde '/etc/passwd' dosyasından yararlanılarak, kullanıcı adına (login) ve kullanıcı numarasına (user ID) göre aramayı sağlayacak 'passwd.byname' ve 'passwd.byuid' adlarıyla iki dbm dosyası oluşturulur. Bu dbm dosyalarının kullanımı, telefon rehberi sarı sayfalarının kullanımına benzediğinden, NIS hizmetleri için yp (yellow pages) hizmetleri deyimi de kullanılır.

NIS hizmetlerinde tanım bilgilerinin birincil sunucu üzerinde toplanması, yönetim kolaylığı sağlamasına karşın, hatalar için tek kaynak noktasıdır (single point offailure). Başka bir deyişle, birincil sunucu kullanım dışı kaldığı zaman, kurum içinde bu sunucuya bağlı bütün hizmetler kullanım dışı kalır. Bu durumu önlemek için, kurum ağı üzerinde ikincil (yedek) sunucuların (slave servers) bulunması gerekmektedir. Bu ikincil sunucular, ilk çalıştıklarında tanım bilgilerini birincil sunucudan dbm veritabanı formatında alırlar. Daha sonra, birincil sunucu üzerindeki bilgilerde bir günleme yapılırsa, bu günlemeler birincil sunucu tarafından ikincil sunuculara gönderilir. Böylece kurumsal tanım bilgisi, güncel bir halde birden fazla sunucu üzerinde saklanmış olur. Eğer birincil sunucu kullanım dışı kalırsa, ikincil sunucular kullanıma girer NIS hizmetinin ilk kullanılmaya başladığı zamanlarda, ağ üzerindeki NIS hizmeti istemcileri, birincil sunucuyu broadcast IP paketleri göndererek bulabiliyorlardı. Bugün bu durum hala geçerli olmasına rağmen, yeni NIS sürümlerinde istemci üzerinde birincil sunucu adresini tanımlama olanağı da sağlanmıştır. Fakat her iki durumda da, NIS istemcilerinin, hangi NIS alanına (NIS domain) bağlanacağını bilmesi gerekir. NIS alanı, Internet adres tanımlarıyla ilgili olan DNS (Domain NameSystem) alanlarından farklıdır. NIS alanı, ortak sistem tanım bilgilerini paylaşan bir grup bilgisayardan oluşur. Bu gruplama, bir kurumun birbirinden bağımsız bölümlerine ilişkin tanım bilgilerinin, sadece ilgili bölümün sunucu ve istemcilerine dağıtılmasını sağlamak amacıyla kullanılır. Örneğin, bir üniversitede değişik bölümlere dağılmış çok sayıda Linux sunucusu bulunabilir. Bu sunucular, örneğin bölüm taban alınarak değişik NIS alanları içinde gruplandırılabilir. Bu yolla her bölüme ilişkin sistem yönetimi, diğer bölümlerden bağımsız, sadece o bölüme özgü tanım bilgilerine dayalı olarak gerçekleştirilebilir.

12.2. Active Directory

Active Directory Microsoft firmasına ait bir rehber sistemi olduğu için burada incelenmeyecektir. Tez kapsamında Linux ve UNIX işletim sistemleri incelenmektedir. Bununla birlikte Active Directory teknolojisi ile Microsoft firması bir rehber servisi (directory service) alt yapısını Windows 2000 ürün ailesinden itibaren işletim sistemi içine gömmeyi başarmıştır. Active Directory Windows 2003'ten itibaren LDAP V3 uyumlu hale getirilmiştir.

12.3 LDAP ile Linux Sistemlerde Kimlik Doğrulaması:

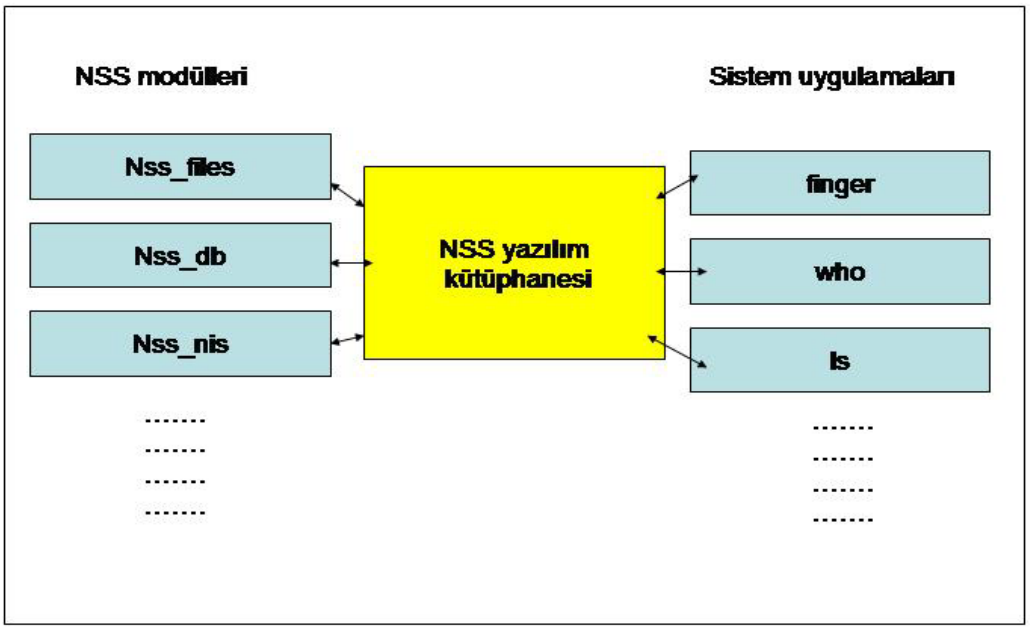
Yukarıda da bahsedildiği gibi LDAP kullanarak kullanıcı için kimlik doğrulaması yapılabilmesi NSS ve PAM yazılım kütüphaneleri ile mümkün olabilmektedir.

12.3.1 NSS (Name Service Switch) Sistemi

Önceki kesimlerde anlatıldığı gibi, UNIX'in ilk zamanlarında sistem tanım bilgileri, diğer deyişle rehber bilgileri, /etc rehberi altındaki yerel tanım dosylarında saklanmaktaydı. NIS, DNS gibi rehber hizmetlerinin kullanılmaya başlanmasıyla, bir sorgunun sonucunun birden çok hizmetten elde edilebilmesi söz konusu olmuştur. Bu durumda, hizmetlerin hangilerinin ve hangi sırayla çağırılarak arama yapılacağı sorun olmaya başlamıştır. Bu sorun ilk zamanlarda, UNIX'in önemli sistem işlevlerini bünyesinde barındıran, libc yazılım kütüphanesinin kaynak koduna müdahale edilerek aşılmaya çalışılmıştır. Fakat bu yolla, sadece sabit bir arama sırası tanımlanabiliyordu. Örneğin, bir kullanıcının bilgileri önce NIS'ten, bulunamazsa yerel dosyalardan aranacak şeklinde, sistem yöneticisinin isteğine göre değiştirilemeyen bir arama sırası takip ediliyordu. Bu yol, rehber bilgisine erişim için yeni hizmetler gerçekleştirmek isteyen geliştiriciler için de bir sorun olmaktaydı. Çünkü, doğrudan libc yazılım kütüphanesinin kaynak koduna müdahale edilmesi gerekiyordu. Bu durum, yazılım geliştiricilerin işini zorlaştırdığı gibi, değiştirilen libc kodunun dağıtılması ve sürüm denetiminin yapılması açısından da sorunlara yolaçabilir. Soruna çözüm bulmak için, NSS sistemi kullanılmaya başlanmıştır. NSS sistemi, ilk olarak Sun Microsystems tarafından geliştirilmiş, daha sonra glibc olarak bilinen GNU C yazılım kütüphanesine aktarılmıştır. Önceki yaklaşımın sorunlarına

çözüm olarak, hizmetler arasında arama sırası tanımlanmasına ve sisteme yeni hizmetlerin eklenmesine izin verilmektedir.

Şekil 12.1 de NSS sisteminin çalışma şekli görülmektedir.



Şekil 12.1 NSS Sisteminin Yapısı

NSS modülleri, NSS Yazılım Kütüphanesince tanımlanan ve Linux işletim sistemi için gerekli rehber bilgilerini simgeleyen bazı veritabanlarına erişim imkanı sağlar. Bu veritabanlarının adları ve kısa tanımları şöyledir :

• **aliases** : UNIX kullanıcılarının e-posta takma adları (e-mail aliases).

• **ethers** : Ağdaki bilgisayarların Ethernet kartı adresleri.

• **group** : UNIX kullanıcı grupları.

• **hosts** : Ağdaki bilgisayarların İnternet adresleri ve IP numaraları.

• **netgroup** : Ağ çapında özel yetki vermek istenen İnternet adresleri ve kullanıcı adları listesi.

• **network** : Alt ağ adları ve adresleri.

• **protocols** : Ağ protokollerinin bilgileri.

• **passwd** : UNIX kullanıcılarının bilgileri.

• **rpc** : Uzaktan Yordam Çağırma (Remote Procedure Call) ile ilgili bilgiler.

• **services** : Ağ hizmetlerinin adları ve port numaraları.

• **shadow** : Yetkisiz UNIX kullanıcılarından saklanmış şifre bilgileri.

Veritabanları, NSS Yazılım Kütüphanesi içinde tanımlandığı için, yeni bir veritabanı eklemek veya olanlarda değişiklik yapmak için glibc kaynak koduna müdahale etmek gereklidir.
NSS Yazılım Kütüphanesi, hangi veritabanları için hangi hizmetleri (diğer bir deyişle modülleri) kullanarak arama yapacağını /etc/nsswitch.conf ayar dosyasına göre belirler. Bu dosya, NSS sisteminin doğru çalışması açısından çok önemlidir.

Şekil 12.2'de bu dosya için bir örnek verilmiştir.

```
# /etc/nsswitch.conf
# Name Service Switch configuration file.
passwd: db files nis
shadow: files
group: db files nis
hosts: files nisplus nis dns
networks: nisplus [NOTFOUND=return] files
ethers: nisplus [NOTFOUND=return] db files
protocols: nisplus [NOTFOUND=return] db files
rpc: nisplus [NOTFOUND=return] db files
services: nisplus [NOTFOUND=return] db files
```

Şekil 12.2 Örnek bir /etc/nsswitch.conf dosyası

12.3.2 PAM (Pluggable Authentication Modules) Sistemi

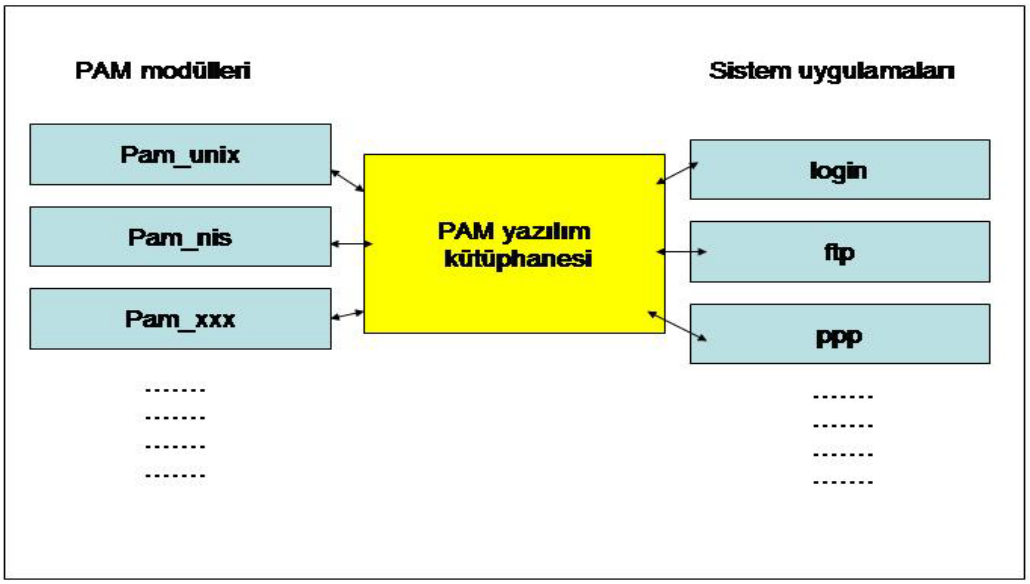
PAM, UNIX'te kullanıcı tanıma (authentication) işlemi için, değişik hizmetleri kullanma imkanı veren bir sistem kütüphanesidir. PAM sisteminin kullanım amacı NSS sistemi ile karıştırılmamalıdır. NSS, her türlü sistem uygulamasının ihtiyaç duyduğu kullanıcı, grup bilgileri, DNS adresleri gibi bilgilere erişim imkanı sağlayan bir sistemdir. Kullanıcı tanıma işlemi sırasında NSS sorgusu yapılsa da, bunun

işlemin sonucuna doğrudan bir etkisi bulunmaz. PAM, login, ftp, su gibi kullanıcı tanıma işlemi yapılması gereken uygulamalara hizmet etmek amacıyla tasarlanmıştır. Kullanıcı tanıma işlemine ihtiyaç duyan programların temel olarak yaptıkları iki önemli işlem vardır: Kullanıcı tanıma ve kullanıcının istediği hizmeti vermek. Eski UNIX sistemlerinde kullanıcı tanıma işlemi, genelde /etc/passwd dosyasındaki kullanıcı bilgilerine erişilerek yapılıyordu. Fakat, bu işlem sırasında kullanıcı tarafından girilen açık metin (clear text) şifrenin, bir şifreleme algoritması kullanılarak şifrelenmesi ve /etc/passwd dosyasındaki şifreyle karşılaştırılması uygulamanın kendisi tarafından yapılıyordu. Bu yöntem, uygulamanın geliştiricisi açısından fazladan bir programlama yüküne sebep olduğu gibi, yeniliklere de kapalı bir sistem oluşturur. Çünkü, /etc/passwd dosyası dışında, yeni kullanıcı tanıma yöntemlerini sisteme eklemek veya olanları güncellemek için yapılacak değişiklikler kullanıcı tanıma işlemi yapılan uygulamaların hepsine yansıtılmalıdır. PAM, kullanıcı tanıma işleminin ayrıntılarıyla ilgili sorunları çözerek, sistem uygulamalarını bu yükten kurtarır. Kullanıcı tanıma yöntemlerinde yapılacak değişiklikler, sadece PAM modülleri üzerinde yapıldığı için sistem uygulamaları etkilenmez. Böylece uygulama geliştiriciler, sadece kullanıcının istediği hizmeti doğru bir şekilde vermek üzerine yoğunlaşabilirler.

Kullanıcı tanıma işleminin PAM sistemi ile yapılması, güvenlik açısından da çok önemlidir. Uygulama geliştiricilerin tasarladıkları uygulamaya özgü kullanıcı tanıma yöntemleri, fazla denenmeden kullanıldıkları için güvenilir değillerdir. PAM sistemi,denenmiş ve güvenilir bir kullanıcı tanıma yöntemine olan ihtiyacı giderdiği gibi, güvenlik açısından öğrenilmesi gereken birçok ayrıntıdan uygulama geliştiricileri kurtarmaktadır.

PAM sisteminin yazılım mimarisi, NSS sistemininkine benzemektedir. Bu mimari temel olarak iki kesimden oluşur: PAM Yazılım Kütüphanesi, PAM modülleri. PAM Yazılım Kütüphanesi, sistem uygulamaları ile iletişim kuran kesimdir.[11,12,13] Bu kütüphane, sistem yazılımlarının ihtiyaç duyacağı işlevleri sunmaktadır. Bunun yanı sıra, PAM modüllerinin belleğe yüklenmesi ve modül işlevlerinin çağrılmasını da yapmaktadır.

Her bir PAM Modülü, kullanıcı tanıma için kullanılan bir hizmete karşılık gelir. Bu hizmetler, örneğin /etc/passwd dosyası, NIS gibi değişik kaynaklardan aldıkları bilgilerle kullanıcı tanıma işlemini gerçekleştirirler. Bu kaynakların neler olacağıyla ilgili bir sınırlama yoktur. Bir PAM modülü, göz retinası, parmak izi tarayarak veya ses kaydederek kullanıcı tanıma işlemini gerçekleştirebilir. Bunun için yapılması gereken, bir tarayıcı cihaz ile alınan kullanıcı bilgilerini, daha önce veritabanında saklanmış olan bilgiyle karşılaştıracak bir PAM modülü yazmaktır. PAM sisteminin yapısı Şekil 12.3' te gösterilmiştir.



Şekil 12.3 PAM Sisteminin Yapısı

PAM Yazılım Kütüphanesi, sistem uygulamaları için dört çeşit görevi yerine getirir :

• Kullanıcı Tanıma Yönetimi (Authentication Management) : Kullanıcı adı ve şifre gibi kullanıcıyı tanıtan bilgilerin alınması ve kullanıcının sistemde var olup olmadığının ve tanıtıcı bilgilerinin doğruluğunun denetlenmesidir.

• Kullanıcı Hesabı Yönetimi (Account Management) : Kullanıcı tanıma işlemi yapıldıktan sonra, kullanıcının istenilen hizmeti alma hakkının olup olmadığının denetlenmesidir.

• Oturum Yönetimi (Session Management) : Kullanıcı için hizmet oturumunun başlatılması ve sonlandırılmasından önce gerekli işlerin yapılmasıdır.

• Şifre Yönetimi (Password management) : Kullanıcı şifresinin değiştirilmesi sırasında gerekli işlerin yapılmasıdır.

Her PAM modülü, bu dört çeşit görevin hepsini veya bir kısmını gerçekleştirebilir. Örnek bir pam.conf dosyası içeriği Şekil 12.4' te görülmektedir.

```
login auth required pam_unix.so.1 debug
login auth optional pam_nis.so.1 try_first_pass
login account required pam_unix.so.1
login session required pam_unix.so.1
login password required pam_unix.so.1
rlogin auth sufficient pam_rhosts_auth.so.1
rlogin auth required pam_unix.so.1
other auth required pam_unix.so.1
other account required pam_unix.so.1
other session required pam_unix.so.1
other password required pam_unix.so.1
```

Şekil 12.4 Örnek Bir /etc/pam.conf Dosyası

13. REHBER SERVİSİ UYGULAMALARI

Bu bölüme kadar rehber servisleri ve ilgili teknolojilerinden ayrıntılı olarak bahsedildi. Bu bölümde ise iki adet rehber servisi uygulaması gerçekleştirilecektir.

Birinci uygulama Linux İşletim Sistemi üzerindeki kullanıcıların, LDAP rehber sunucuları kullanılarak tek bir noktadan kimlik doğrulamalarının (authentication) yapılmasını göstermektedir. Böylece kullanıcı bilgilerinin merkezi bir noktadan yönetimi mümkün olacaktır. Bilgi tekrarı, tutarsız verilerin oluşması ve bunlara bağlı olarak güvenlik risklerinin oluşması büyük ölçüde engellenecektir.

İkinci uygulama ise LDAP rehber sunucuları üzerindeki verilere, Microsoft Outlook, Microsoft Internet Explorer gibi LDAP uyumlu yazılımlardan erişilebilmesini sağlamaktır. OpenLDAP rehber sunucusunun üzerinde bulunan verilere, Microsoft firmasının LDAP uyumlu yazılımlarıyla doğrudan erişme imkanı yoktur. Bunun için bu uygulama gerçekleştirimi yapılırken,OpenLDAP içinde yer alan şemalara (schema) ek olarak şemalar tanımlanacak ve mevcut bazı şemalar üzerinde de değişiklikler yapılacaktır.

13.1. Uygulama 1:

Tek Bir Noktadan Kimlik Doğrulanması

Amaç:

Bu çalışmanın amacı Linux İşletim Sistemi üzerindeki kullanıcıların, LDAP rehber sunucuları kullanılarak tek bir noktadan kimlik doğrulamalarının (authentication) yapılmasıdır.

Gerçekleştirim:

Bu çalışmada rehber servisinin üzerine kurulacağı İşletim Sistemi olarak Mandrake Linux sürüm 9.2 kullanılmıştır. LDAP rehber sunucu (directory server) olarak açık

kaynak koda sahip olan (open source) OpenLDAP rehber sunucusu kullanılmıştır. OpenLDAP Mandrake Linux 9.2' ye ait CD ler içinde gelmektedir. Daha güncel sürümleri ücretsiz olarak http://www.openldap.org adresinden temin edilebilir.

Öncelikle Mandrake Linux 9.2' işletim sistemi üzerine aşağıdaki paketlerin kurulmuş olması gerekmektedir.

Tablo 13.1 Linux üzerinde olması gereken paketler

OpenLDAP sunucusunun düzgün çalışması için gereken paketler	
libldap2	
openldap	
openldap-clients	
openldap-migration	Sadece sunucu tarafında gerekli paketler. İstemci tarafında gereksiz.
openldap-servers	
nss_ldap	
pam_ldap	

Sunucu Tarafı Ayarları

Bundan sonra ilk adım olarak /etc/openldap/slapd.conf ayar dosyasını kullanarak aşağıda şekil 13.1' de gösterilen değişikliklerin bir metin editörü ile yapılması gerekmektedir.

```
database        ldbm
suffix          "dc=mylan,dc=net"
rootdn          "cn=root,dc=mylan,dc=net"
rootpw          {MD5}zYgLcm4KDb1CN/ENGdpG9A==
directory       /var/lib/ldap
index           objectClass,uid,uidNumber,gidNumber eq
index           cn,mail,surname,givenname       eq,subinitial
```

Şekil 13.1 /etc/openldap/slapd.conf dosyası

Yukarıda görülen {MD5}zYgLcm4KDb1CN/ENGdpG9A== aslında secret kelimesinin MD5 algoritması ile oluşturulan hash kodudur. Ve rootdn' in parolasıdır.

Bu parola değiştirilmek istenirse aşağıdaki komut kullanılmalıdır.

```
[root@ldap]#slappasswd -h {MD5}
```

Bu işlemden sonra test amacıyla OpenLDAP sunucusunun başlatılması gerekmektedir.

```
[root@ldap]#service ldap start
```

OpenLDAP başladıktan sonra deneme amaçlı bir sorgu çalıştırılır.

```
[root@ldap]# ldapsearch -x -b '' -s base '(objectclass=*)' namingContexts
```

Aşağıda şekil 13.2' de yer alan satırlara benzer satırlar görüyorsak rehber sunucusu çalışyor demektir.

```
# filter: (objectclass=*)
# requesting: namingContexts
dn:
namingContexts: dc=mylan,dc=net
# search result
search: 2
result: 0 Success
# numResponses: 2
# numEntries: 1
```

Şekil 13.2 LDAP sorgu sonucu

OpenLDAP sunucumuzun Linux İşletim Sistemimiz her açıldığında otomatik çalışmasını istiyorsak aşağıdaki komutu kullanmamız gerekir.

```
[root@ldap]# chkconfig ldap on
```

Sonraki adım verinin transferi adımıdır. Bu aşamada yerel dosyalarda bulunan tanımlar rehber sunucusuna taşınır. Bunun için öncelikle /usr/share/openldap/migration dizinindeki migrate_common.ph dosyasında aşağıda şekil 13.3' te yer alan değişiklikler yapılır.

```
$DEFAULT_MAIL_DOMAIN = "mylan.net";
$DEFAULT_BASE = "dc=mylan,dc=net";
$DEFAULT_MAIL_HOST = "mail.mylan.net";
$EXTENDED_SCHEMA = 1;
```

Şekil 13.3 migrate_common.ph dosyası

Bundan sonra yine aynı dizin içinde aşağıdaki komut çalıştırılır.

```
[root@ldap]# ./migrate_all_online.sh
```

Karşımıza gelecek olan sorulara aşağıdaki gibi yanıtlar verilmesi gerekmektedir.

```
Enter the X.500 naming context you wish to import into: [dc=mylan,dc=net]
Enter the name of your LDAP server [ldap]: localhost
Enter the manager DN: [cn=manager,dc=mylan,dc=net]:
cn=root,dc=mylan,dc=net
Enter the credentials to bind with: secret
Do you wish to generate a DUAConfigProfile [yes|no]? No
```

Kontrol etme amacıyla aşağıdaki sorgu çalıştırılır ve benzer bir çıktı görülmesi beklenir. Burada ağımızda wrkstation adina sahip bir bilgisayar olduğu varsayılıyor. Biz bu ismi farklı verebiliriz.

```
[root@ldap]# ldapsearch –LL –H ldap://localhost -b"dc=mylan,dc=net" -x
"(cn=wrkstation)"
```

```
dn: cn=wrkstation.mylan.net,ou=Hosts,dc=mylan,dc=net
objectClass: top
objectClass: ipHost
objectClass: device
ipHostNumber: 10.0.10.23
cn: wrkstation.mylan.net
cn: wrkstation
```

Buraya kadar sistem beklenen sonuçları veriyorsa yerel dosyalardaki tanımlar OpenLDAP rehber sunucusuna doğru aktarılmış demektir.

Buraya kadar yapılan işlemler rehber sunucusu tarafında yapılması gereken işlemlerdi. Bundan sonrakiler istemci tarafında yapılacaktır. Biz istemci olarak sunucu bilgisayarımızla aynı bilgisayarı kullanacağımız için ayarları aynı bilgisayar üzerinde yapacağız.

İstemci Tarafı Ayarları

İlk olarak /etc/ldap.conf dosyasına aşağıda şekil 13.4' te yer alan satırlar girilir. Bu tanımlar ile istemci LDAP sunucu olarak hangi bilgisayara bağlanacağına karar verir. 127.0.0.1 adresi ile kendi bilgisayarımızı sunucu olarak gösterdik.

```
host 127.0.0.1
base dc=mylan,dc=net
rootbinddn cn=proxyuser,dc=mylan,dc=net
scope one
pam_filter objectclass=posixaccount
pam_login_attribute uid
pam_member_attribute gid
pam_template_login_attribute uid
pam_password md5
nss_base_passwd         ou=People,dc=mylan,dc=net?one
nss_base_shadow         ou=People,dc=mylan,dc=net?one
nss_base_group          ou=Group,dc=mylan,dc=net?one
```

```
nss_base_hosts                  ou=Hosts,dc=mylan,dc=net?one
```

Şekil 13.4 ldap.conf dosyası

/etc/nsswitch.conf dosyasına aşağıdA şekil 13.5' te yer alan satırlar girilir. Bu tanımlar ile istemcinin kimlik doğrulaması işlemi için önce yerel dosyalara sonra LDAP rehber sunucusuna bakması belirtilir.

```
passwd:     files ldap
shadow:     files ldap
group:      files ldap
hosts:      files ldap dns
```

Şekil 13.5 nssswitch.conf dosyası

Son olarak kimlik doğrulama işlemlerinin hangi kütüphanelerle (.so uzantılı dosyalar) yapılacağını /etc/pam.d dizini altındaki dosyalarda belirtiriz. /etc/pam.d/system-auth dosyasında aşağıda şekil 13.6' da yer alan değişiklikler yapılır.

```
#%PAM-1.0
auth        required      /lib/security/pam_env.so
auth        sufficient    /lib/security/pam_unix.so likeauth nullok
auth        sufficient    /lib/security/pam_ldap.so use_first_pass
auth        required      /lib/security/pam_deny.so
account     required      /lib/security/pam_unix.so
account     [default=bad success=ok user_unknown=ignore \
service_err=ignore system_err=ignore] /lib/security/pam_ldap.so
password    required      /lib/security/pam_cracklib.so retry=3
password    sufficient    /lib/security/pam_unix.so nullok use_authtok \
md5 shadow
password    sufficient    /lib/security/pam_ldap.so use_authtok
password    required      /lib/security/pam_deny.so
session     required      /lib/security/pam_mkhomedir.so skel=/etc/skel/ \
umask=0022
session     required      /lib/security/pam_limits.so
session     required      /lib/security/pam_unix.so
session     optional      /lib/security/pam_ldap.so
```

Şekil 13.6 system-auth dosyası

etc/pam.d/passwd dosyasında aşağıda şekil 13.7’ de yer alan değişiklikler yapılır.

```
#%PAM-1.0
auth       sufficient   /lib/security/pam_ldap.so
auth       required   /lib/security/pam_pwdb.so shadow nullok
account    sufficient   /lib/security/pam_ldap.so
account    required /lib/security/pam_pwdb.so
password   required            /lib/security/pam_cracklib.so retry=3 minlen=4  \
dcredit=0  ucredit=0
password   sufficient   /lib/security/pam_ldap.so use_authtok
password   required            /lib/security/pam_pwdb.so use_authtok nullok \
md5 shadow
```

Şekil 13.7 passwd dosyası

Uygulamaya ait sonuç:

Bu işlemlerden sonra istemci bilgisayarımız kimlik doğrulamasını OpenLDAP rehber sunucusu üzerinden yapacaktır. Kullanıcı hesapları için öncelikle istemcinin kendi üzerindeki yerel kullanıcı tanımı dosyalarına (passwd ve shadow) bakılacaktır. Aranılan kullanıcı hesabı burada yoksa LDAP rehber sunucuna bakılacaktır.

Biz burada istemci olarak ta Mandrake Linux 9.2 yi kullandık. İstemciyi hazır hale getirmek için gerekli tanım dosyaları üzerinde ayarlamalar yaptık. Tez kapsamında istemci sistem olarak Suse Linux 9.0 sistemi de kullanılmıştır. Suse Linux 9.0 işletim sisteminde LDAP istemci ayarları çok daha kolaydır. Bunun için bu işletim sisteminde grafik arabirimli sistem yönetim aracı olan YAST yazılımını kullanılarak kimlik doğrulaması için LDAP' ın kullanılması gerektiğini söylemek yeterlidir.

13.2 Uygulama 2:

Ldap Uyumlu Yazılımlarla Verilere Erişim

Amaç:

Bu çalışmanın amacı LDAP rehber sunucuları üzerindeki verilere, Microsoft Outlook, Internet Explorer gibi LDAP uyumlu yazılımlardan erişilebilmesini sağlamaktır.

Gerçekleştirim:

Bu çalışmada rehber servisinin üzerine kurulacağı İşletim Sistemi olarak Mandrake Linux sürüm 9.2 kullanılmıştır. LDAP rehber sunucu (directory server) olarak açık kaynak koda sahip olan (open source) OpenLDAP rehber sunucusu kullanılmıştır. OpenLDAP Mandrake Linux 9.2' ye ait CD ler içinde gelmektedir. Daha güncel sürümleri ücretsiz olarak http://www.openldap.org adresinden temin edilebilir.

OpenLDAP rehber sunucusunun üzerinde bulunan verilere, Microsoft firmasının LDAP uyumlu yazılımlarıyla doğrudan erişme imkanı yoktur. Bunun için OpenLDAP içinde yer alan şemalara (schema) ek olarak şemalar tanımlanmalı ve mevcut bazı şemalar üzerinde de değişiklikler yapılmalıdır.

İlk olarak /etc/openldap/schema/ dizini altında bir editör yardımı ile extension.schema adında bir düz yazı (text) dosyası oluşturup aşağıda şekil 14.1' de yer alan verileri girmemiz gerekir

Daha sonra bu dosya /etc/openldap/slapd.conf dosyası içinden referans verilmelidir. Burada yapılan değişikliklerle OpenLDAP, Microsoft firmasına ait ürünlerin ihtiyacı olan schema elemanlarına sahip olacaktır.

```
Attributetype ( 1.3.6.1.4.1.4203.666.100.121
        NAME ( 'rdn' )
        SUP name )
attributetype ( 1.3.6.1.4.1.4203.666.100.122
        NAME ( 'otherFacsimiletelephoneNumber' )
        SUP telephoneNumber )
attributetype ( 1.3.6.1.4.1.4203.666.100.123
        NAME ( 'IPPhone' )
        SUP telephoneNumber )
# This attribute handles MS/Outlook and Netscape Communicator
attributetype ( 1.3.6.1.4.1.4203.666.100.124
        NAME ( 'URL' 'homeUrl' )
        SUP name )
attributetype ( 1.3.6.1.4.1.4203.666.100.125
        NAME ( 'comment' )
        SUP name )
attributetype ( 1.3.6.1.4.1.4203.666.100.126
        NAME ( 'conferenceInformation' )
        SUP name )
attributetype ( 1.3.6.1.4.1.4203.666.100.127
        NAME ( 'reports' )
        SUP manager )
objectclass     ( 1.3.6.1.4.1.4203.666.100.1
    NAME 'officePerson'
        DESC 'Office employee or computer user'
    SUP inetOrgPerson
    STRUCTURAL
        MAY  ( c $
               rdn $
               otherFacsimiletelephoneNumber $
               IPPhone $
               URL $
               comment $
               reports $
               conferenceInformation )
)
```

Şekil 14.1 extension.schema dosyası

Tablo14.1 extension.schema dosyası

Ms/□utlook alan adları	Ldap öznitelikleri	İlgili şema dosyaları	Üyesi oldukları objectclass' lar
Name:	Cn (Common Name)	core.schema	objectPerson
Email Address:	Mail	core.schema	inetOrgPerson
Job Title:	Title	core.schema	organizationalPerson
Department:	Ou (Organizational Unit) "ou" tanımlı değilse "department" kullan.	Ou: core.schema department: extension.schema	ou: organizationalPerson department: officePerson
Office:	physicalDeliveryOfficeName	core.schema	organizationalPerson
Company Name:	o (Organization)	core.schema	inetOrgPerson
Business Web Page:	URL, homeURL	extension.schema	officePerson
First Name:	givenName	core.schema	inetOrgPerson
Middle Name:	İnitials	core.schema	inetOrgPerson
Last Name:	Sn (Surename)	core.schema	objectPerson
Notes:	Comment	extension.schema	officePerson
Netmeeting Server:	conferenceInformation	extension.schema	officePerson
Digital ID:	userCertificate	extension.schema	inetOrgPerson
Manager:	Manager	cosine.schema	inetOrgPerson
Reports:	Reports	extension.schema	officePerson
Business - Street Address:	postalAddress	core.schema	organizationalPerson
Business - City:	l (Locality)	core.schema	organizationalPerson
Business - State/Province:	St	core.schema	organizationalPerson
Business - Zip Code:	postalCode	core.schema	organizationalPerson
Business - Country/Region:	C	core.schema	officePerson
Home - Street Address:	homePostalAddress	cosine.schema	inetOrgPerson
Telephone:			
Business:	telephoneNumber	core.schema	organizationalPerson
Business Fax:	facsimileTelephoneNumber	core.schema	organizationalPerson
Home:	homePhone	cosine.schema	inetOrgPerson
Home Fax:	otherFacsimiletelephoneNumber	extension.schema	officePerson
Mobile:	Mobile	cosine.schema	inetOrgPerson
Pager:	Pager	cosine.schema	inetOrgPerson
IPPhone:	IPPhone	extension.schema	officePerson

Uygulamaya ait sonuç:

Bu işlemlerden sonra OpenLDAP rehber sunucusu üzerinde tutacağımız veriler Microsoft Outlook, Internet Explorer gibi LDAP uyumlu yazılımlardan erişilebilir ve sorgulanabilir olacaktır. Yukarda gösterilen şema üzerinde yapılacak değişikliklerle Netscape ve Mozilla gibi yazılımlarla da verilere erişim mümkün olacaktır.

14. SONUÇ

Bu çalışmada, rehber hizmet sunucuları ve bu yapılarla gerçekleştirilecek önemli uygulamalar incelenmiştir. Ayrıca mevcut rehber hizmet sunucuları çeşitli kriterlere göre karşılaştırılmış neticede LDAP rehber hizmet sunucu yapısının gerek rahat kullanım gerekse esneklik bakımından uygunluğu vurgulanmıştır. Rehber hizmet sunucularının erişim protokollerinin incelenmesi çalışmada yer verilen diğer bir husustur.

Linux sistemleri üzerinde kullanıcı kimlik doğrulaması ve tüm sistemler tarafından ortak kullanılan verilerin merkezileştirilmesi konusunda gerçekleştirim yapılmıştır. LDAP, hem X.500 sunucularına (DSA-Diretory System Agent) (LDAP geçit sunucu kullanılarak) hem de LDAP rehber servislerine erişim için destek sunmaktadır[14]. LDAP bir çok programlama dili tarafından desteklenmektedir. Büyük problemlerden biri yazılım üreten firmaların yazılımlarını LDAP uyumlu (LDAP enabled) hale getirmelerinde yaşanmaktadır. Ancak yazılımlar tümüyle LDAP uyumlu hale geldiklerinde rehber servislerden tam verim elde edilecektir.

Linux işletim sistemi üzerindeki sistem yazılımlarının hepsi kendine özgü kullanıcı rehberleri kullanmaktadır. Bu yazılımların oluşturduğu sistem bir bütün olarak düşünüldüğünde, bu sisteme yeni bir kullanıcı eklenmek veya bu kullanıcı bilgisinin bir bölümünü değiştirilmek istendiğinde, her rehbere ayrı ayrı erişmek ve yine her rehbere ayrı ayrı ekleme, silme veya değiştirme işlemini yapmak gerekir ki bu durum yüzlerce, binlerce kişilik ağlarda büyük sorunlara sebep olabilir. Sadece yönetimsel değil aynı zamanda güvenlik konusunda da problemler söz konusudur. Tez çalışması kapsamında bu merkezi olmayan rehberlerin bir LDAP rehber sunucusu üzerinde tutulması için gerekli çalışmalar yapılmış ve kullanıcı kimlik doğrulama işlemi merkezi yapıya kavuşturulmuştur. Kimlik doğrulama çalışmalarında LDAP istemci olarak Popüler bir Linux dağıtımı olan Suse Linux 9.0 kullanılmıştır. LDAP sunucu sistemi olarak ise yine popüler bir Linux dağıtımı olan Mandrake Linux 9.2

kullanılmıştır. Bu sistemin üzerinde tamamiyle açık kod (open source) olan OpenLDAP sunucu yazılımı kullanılmıştır.Kullanıcı hesaplarının tek bir merkezden yönetimi konusunda alternatif olarak Sun firması tarafından geliştirilen NIS yapısı karşımıza çıkmaktadır. Fakat LDAP karşısında NIS yapısının gün geçtikçe kullanım alanları daralmaktadır. Kullanıcı kimlik doğrulama yapısının LDAP ile kullanılabilmesi için PAM_LDAP ve NSS_LDAP yapıları incelenmiş gerçekleştirimde kullanılmıştır. Yeni kullanıcı tanıma yöntemlerinin diğer işletim sistemleri ve sistem uygulamaları ile birlikte kullanılması da mümkündür. LDAP sunucu standartı, bu yeni yöntemlerin ağ üzerindeki her türlü uygulama ve işletim sistemi tarafından kullanılabilecek şekilde yaygınlaşması için gerekli altyapıyı sunmaktadır.

Öncelikle Linux kullanıcı ve grup bilgilerinin LDAP rehberine aktarılması sağlanmıştır. Linux işletim sisteminin kullanıcı rehber hizmetlerinin şu anki tasarımından ve Linux dağıtımları arasındaki bazı uyumsuzluklardan dolayı, aktarım konusunda dağıtıma bağlı olarak küçük değişikliklerin yapılması gerektiği anlaşılmıştır. LDAP rehberine veri transferi konusunda PADL yazılım firmasına ait açık kaynak kodlu Perl scriptleri kullanılmıştır.

Kullanıcı hesapları dışında yine bir çok uygulama tarafından (e-posta istemcisi, ajanda vb.) ihtiyaç duyulan adres defteri türündeki bilgileri Microsoft firmasının Outlook, Adres Book gibi uygulamalarından erişebilir hale getirilmiştir. Normal şartlarda Microsoft firmasının ürünleriyle çalışacak durumda olmayan OpenLDAP rehber sunucusunun çekirdek şeması (core schema) üzerinde değişiklikler yapılmıştır. Bu işlem için yine LDAP sunucu sistemi olarak Mandrake Linux 9.2 kullanılmıştır. Bu sistemin üzerinde tamamiyle açık kod (open source) olan OpenLDAP sunucu yazılımı kullanılmıştır.

Tez çalışmasının kapsamı içinde Microsoft firmasına ait Windows işletim sistemleri olmamasına rağmen, Microsoft firmasına ait Active Directory teknolojisi bize bir işletim sistemi ile LDAP rehber sunucusunun entegrasyonunu en iyi şekilde göstermektedir.

Linux ve UNIX dünyasının da bu entegrasyonu en kısa zamanda yapması gerekmektedir. Bu durumda tercih edilecek LDAP sunucu, açık kaynak koda sahip olması bakımından OpenLDAP olabilir. Eğer bu entegrasyon yapılmaz ise her zaman yerel tanımlama dosyaları üzerinde kullanıcı hesapları bulunma olasılığı olacaktır. Parololar yine bu yerel tanımlama dosyaları üzerinde tutulacaktır. Rehber servislerinden beklenen verim tam anlamıyla elde edilemeyecektir.

Neticede rehber hizmet sunucuları ve bu sunucuları işleyen protokollerin önemi her geçen gün artmaktadır. Buna paralel dağıtılmış bir yapıda gerek işletim sistemleri arasında veri paylaşımı gerekse programlar arasındaki veri paylaşımı bu yapılara dayandırılarak, güvenlik temini, veri tutarlılığı, veri tekrarının önlenmesi sağlanabilir.

KAYNAKLAR

[01] ITU-T X.521, ITU-T Recommendation X.521 (1993) | ISO/IEC 9594-7, Information technology – Open Systems Interconnection – The Directory: Selected object classes, 1993

[02]ITU-T X.500, ITU-T Recommendation X.500 (1993) | ISO/IEC 9594-1, Information technology – Open Systems Interconnection – The Directory: Overview of concepts, models and services, 1993

[03]The Lightweight Directory Access Protocol:
X.500 Lite, Timothy A. Howes
Center for Information Technology Integration
University of Michigan

[04]W. Yeong, T. Howes, S. Kille, RFC 1487: X.500 Lightweight Directory Access Protocol (v1), 1993

[05]W. Yeong, T. Howes, S. Kille, RFC 1777: Lightweight Directory Access Protocol (v2), 1995

[06] M. Wahl, T. Howes, S. Kille, RFC 2251: Lightweight Directory Access Protocol (v3), 1997

[07]Acces Nds Edırectory Via Ldap
Assistant Director of Academic Technology
Drew University

[08]ITU-T X.520, ITU-T Recommendation X.520 (1993) | ISO/IEC 9594-6, Information technology – Open Systems Interconnection –The Directory: Selected attribute types, 1993

[09]D. Chadwick.
http://www.salford.ac.uk/its024/X500.htmUnderstanding X.500 - The Directory.
International Thomson Computer Press, 1996.

[10]W. Yeong, T. Howes, and S. Kille.
ftp://ftp.isi.edu/in-notes/rfc1777.txtLightweight Directory Access Protocol.
RFC 1777, IETF, March 1995.

[11]Andrew G. Morgan, The Linux-PAM System Administrator Guide, 1999, http://www.kernel.org/pub/linux/libs/pam/

[12]Andrew G. Morgan, The Linux-PAM Application Developer's Guide, 1999, http://www.kernel.org/pub/linux/libs/pam/

[13]Andrew G. Morgan, The Linux-PAM Module Writer's Guide, 1999, http://www.kernel.org/pub/linux/libs/pam/

[14]IBM Corporation, The Library for System Solutions Directory, Naming and Time, IBM Redbooks, 1994

EK 1

Tez Çalışmasında Kullanılan İşletim Sistemleri, Programlama Araçları ve Diğer Yazılımlar

İşletim Sistemleri
Mandrake Linux 9.2 : Bir Fransız firması tarafından geliştirilen Mandrake Linux pazarda büyük paya sahip Linux dağıtımlardan biridir.
http://www.mandrakelinux.com

Suse Linux 9.0 : Bir Alman firması tarafından geliştirilen Suse Linux yine pazarda büyük paya sahip Linux dağıtımlardan biridir. 2003 yılının son çeyreğinde Novell tarafından satın alınmıştır.
http://www.suse.com

MS-Windows XP Pro : Microsoft firmasına ait Windows XP Pro, özellikle iş istasyonu olarak kullanımda en büyük Pazar payına sahip işletim sistemidir.
http://www.microsoft.com

Programlama Araçları
Perl: Larry Wall tarafından geliştirilmiştir. Son derece kararlı (Stable) ve bir çok farklı platformda çalışabilen bir programlama dilidir.
http://www.perl.com

Sunucu Yazılımları
OpenLDAP : Tez çalışması kapsamında, Linux kullanıcı hesaplarının (account) ve kurum personel bilgilerinin saklanması, yönetilmesi için OpenLDAP ürünü kullanılmıştır.

OpenLDAP Rehber Sunucusu, ücretsiz dağıtılan bir sunucu yazılımıdır. Bu sunucu hakkındaki her türlü bilgiye http://www.openldap.org adresinden ulaşılabilir. Sunucu yazılımının en son sürümünün kaynak kodu bu adresten temin edilebilir. OpenLDAP tamamı açık kaynak kod (open source) olan bir yazılımdır. Linux ve Unix işletim sistemleri üzerinde çalışabilir.

LDAP destekli (ldap enabled) uygulama geliştirmek için bir çok dile ait fonksiyon kütüphaneleri ve API setleri mevcuttur. Perl, Java, PHP, C/C++, Python, C# dilleri bu kütüphaneler kullanılabilmektedir. Bu kütüphanelerin bazıları belirli rehber sunuculara hizmet vermek üzere geliştirilmişlerdir. Novell, eDirectory isimli rehber sunucusuna erişmek için kendi Java sınıflarını gerçeklemiştir. Bunun yanında PHP (ldap_* fonksiyonları), Perl (Net::LDAP modülü), Java (JNDI), python (python-ldap API) dilleri her türlü rehber sunucuya erişimi sağlayan programlama kütüphaneleri sunmaktadırlar.

Bu programlama kütüphaneleri, RFC 1823 (The LDAP Application Program Interface) ile uyumlu olarak tanımlanmışlardır. Fakat bu durum, bu kütüphanelerin birbirleriyle uyumlu olmasını garanti etmez.

İstemci Yazılımları

Gq: Linux üzerinde çalışan, ücretsiz, açık kaynak kodlu bir LDAP istemci yazılımıdır. GTK kütüphanesi kullanılarak geliştirilmiştir. LDAP V3 şemalarını desteklemektedir.

Softerra LDAP Browser : MS Windows üzerinde çalışan, ücretsiz bir LDAP istemci yazılımıdır. LDAP V3 şemalarını desteklemektedir.

LDAP Browser/Editor : Java ile geliştirilmiş, ücretsiz, platform bağımsız bir LDAP istemci yazılımıdır. LDAP V3 şemalarını desteklemektedir.

EK 2

Yararlanılan Internet Siteleri

http://www.openldap.org

http://www.opengroup.org

http://www.umich.edu

http://docs.sun.com/db/doc/816-4856/6mb1q0bib?a=view

http://www.innosoft.com

http://www.itu.int

http://java.sun.com/developer/Books/ldap/chap05.pdf

http://www.ietf.org

http://developer.novell.com/ndk/cldap.htm

http://developer.netscape.com

http://www.linuxjournal.com/article.php?sid=6988

http://www.tarantella.com/knowhow/e3.3/help/en-us/base/standard/attr_scottamembersearch.html

http://www.securityfocus.com/printable/infocus/1428

http://igloo.its.unimelb.edu.au/Webmail/tips/msg00256.html

http://perl-ldap.sourceforge.net

http://www.perldap.org

http://www.sendmail.org/m4/ldap.html

http://www.nntp.perl.org/group/perl.ldap

http://www.iit.edu/~gawojar/ldap/

http://www.surfnet.nl/innovatie/afgesloten/x500/

http://tr.php.net/ldap

http://lcassel.csc.villanova.edu/networks/directorynew/sld018.htm

http://database.sarang.net/database/ldap/X.500/012.GIF

http://www.lions.odu.edu/docs/dce/fig1.15.gif

ÖZGEÇMİŞ

Adı, Soyadı	**:**	Fuat ALTUN
Doğum Tarihi	**:**	1973
Doğum Yeri	**:**	İstanbul
Medeni Hali	**:**	Evli –1996
Adres	**:**	Başbakanlık Toplu konutlar 3. Etap a72/d14
Telefon	**:** :	Ev: (212) 471 61 51 İş: (212) 252 29 00
Lisan	**:**	İngilizce (Orta)
Sürücü Belgesi	**:**	B - Sınıfı

Eğitim Durumu :

1980 - 1985	:	Fikret Yüzatlı İlkokulu- Bahçelievler /İstanbul
1985 - 1988	:	Bahçelievler Ortaokulu- Bahçelievler /İstanbul
1988 - 1991	:	Bahçelievler Lisesi- Bahçelievler /İstanbul
1991 - 1993	:	İstanbul Üniversitesi Bilgisayar Programcılığı Yüksek Okulu (Ön lisans)
1994 -2000	:	Anadolu Üniversitesi İktisat Bölümü (Lisans)
2001-	:	İKÜ Bilgisayar Mühendisliği (Yüksek Lisans)

İş Tecrübeleri – Görevler :

1992	:	İstanbul Sanayi Odası-Bilgi İşlem Merkezi **Yardımcı programcı**
1998	:	İstanbul Sanayi Odası-Bilgi İşlem Merkezi **Veri Tabanı Yöneticisi**
2004	**:**	İstanbul Sanayi Odası-Bilgi İşlem Merkezi **Sorumlusu**

Alınan Eğitimler :

- Adabas C / Natural (Unix) Sistem Yönetimi

- Natural Ligtstorm Implememtation-I
- Natural Programlama I-II
- Natural Programlama I-II (UNIX)
- Adabas Database Fundamentals
- Natural/ Adabas ile Programlamada Performans
- Predict Sistem Kullanımı ve Yönetimi

Acc. Train. For Windows NT 4.0

- Windows NT 4.0 Core Tecn.
- Internetworking &TCP/IP
- Internetworking with TCP/IP on MS Win NT 4.0
- HP-UX 10.0 Fundamentals
- MS Exchange Server 5.5 Des. & Impl.
- MS Exchange Server 5.5 Con. & Adm.
- Internet/Intranet Security
- Visual Basic 6.0 Advanced
- C++ Programing & Object Oriented Programing
- Cisco Router Conf.
- Solaris 7 Sistem Administration
- XML, Tamino Administration
- Bolero(Java) & OOP
- Bolero(Java) & Businnes App. ()
- Şirket İçi İletişim (Prometheus)
- Temel Yöneticilik (Prometheus)
- Kalite Bilinçlendirme ve ISO 9000 (SGS)
- Hizmet sektöründe ISO 9000 Uygulamaları (SGS)
- Kalite Güvence Sistem Dökümantasyonu (SGS)
- Kuruluş İçi Kalite Sistem Tetkikçisi (SGS)