Kurumsal Firmalar

için

Postfix+LDAP

halil.agin@gmail.com
Halil AĞIN

İçindekiler

1 GİRİŞ

2 Mail ve Tarihçesi

http://www.coruscant.demon.co.uk/mike/sendmail/history.html

http://en.wikipedia.org/wiki/E-mail

Mail'in tarihçesi günümüzde kullandığımız internetin tarihçesinden daha eskidir. Büyük bilgisayarların (Mainframe'lerin) var olduğu ama henüz internetin olmadığı zamanlarda mail iletme mekanizması mevcuttu. Bu yıllar 1972 yılından önceki yıllardı.

O yıllarda, çoklu kullanıcılı işletim sistemleri mevcuttu ve bilgisayar kullanıcıları bu büyük bilgisayara terminaller aracılığı ile bağlanıyorlardı. Bu zamanlarda kullanıcıların birbirleri ile iletişim ihtiyacı baş gösterdi ve bunun için küçük bir mekanizma üretildi. A kullanıcısı, B kullanıcısına mesaj iletmek istediğinde, bazı özel komutlar yardımı ile B kullanıcısının sabit bir dosyasına bir mesaj ekliyordu. B kullanıcısı da bu dosyayı açarak en son satıra eklenen mesajı okuyordu. Bu sistemin adın '**Mailbox**' idi.

1972 yılına varıldığında, ARPANET aracılığı ile mekansal bağımlılık geride bırakılmış, kullanıcılar ağ üzerinden bilgisayarlara ulaşabilir hale gelmişti. Bu durum mesaj iletim mekanizmasını daha karmaşık hale getirmişti. Artık mesajlar ağ üzerinde iletilmeliydi. Bu durum için ilk akla gelen yöntem FTP (Dosya Transfer Protokolü) idi. Yine Mailbox sistemine benzer bir sistem devrede idi. FTP protokolüne mail iletimini sağlayan bazı özel komutlar eklendi. Bu sayede, ilgili kişiye mesaj iletildiğinde, bu mesaj bilinen bir dosyada saklanıyordu. Alıcı kişi yine bu dosyayı açarak mailine ulaşıyordu. Bu aşamada, gün yüzüne çıkan bir problem vardı: hangi kullanıcı hangi makikenin kullanıcısı idi. Bu iki bilgiyi birbirinden ayırmak için '@' işareti kullanıldı. FTP ile mesaj transferini gerçekleştiren programı yazan ve '@' işaretini ilk kullanan Ray Tomlinson adındaki bir mühendisti.

ARPANET aracılığı ile artık bir çok bilgisayar birbiri ile iletişime geçebiliyordu. Elbette ki bu bilgisayarların hepsinde aynı işletim sistemi yüklü değildi. Farklı işletim sistemleri ağ üzerinden birbiri ile iletişime geçmişti. Bu aşamada mail iletişimi ile ilgili bazı problemler başgösterdi. İşletim sistemlerinin dosya okuma şekilleri farklı olabiliyordu, kaldı ki bazı işletim sistemleri '@' işaretini komut olarak okuyabiliyorlardı. Bu problemleri çözmek için mail transferi için standartlaştırma çalışmaları başladı. Mail başlıklarını (headers) standartlaştıran ilk doküman RFC 680 kodlu doküman idi. Ardından RFC 624 ve RFC 733 anons edildi. Ama hala mail transferi için bir mail protokolü mevcut değildi. Transfer hala FTP üzerinden yapılıyordu.

1979 yılında, Eric Allman FTP üzerinden mail transferi yapabilen bir program geliştirdi. Bu programın adı “delivermail” idi. “Delivermail” programı ARPANET üzerinden FTP aracılığı ile mail transferi yapıyordu. Bu program günümüzde sendmail olarak bilinen mail transfer programının atası olarak bilinir.

Tarih 1982 yılına gelidiğinde, günümüzde TCP/IP olarak bilinen protokol geliştirildi ve ARPANET'in yerini bu protokol aldı. Aynı yıl içinde TCP/IP üzerinde ilk paket transferi yapıldı. Bu protokolün çıkışı ile mail transferi de tamamen değişti. Artık sadece ağlar mevcut değil “Ağ'ların Ağları” var idi. Bu yeni tasarıma göre mail trasnferi tekrar şekillendirildi ve bu transfer şekli için günümüzde SMTP olarak bilinen protokol geliştirildi. Bu protokol FTP'den bağımsızdı, paket transferi için kendi özgün protokolü vardı. 1986 yılına gelindiğinde, DNS protokolünün çıkışı ile artık makineler arasındaki iletişim daha da basitleşmişti. Mail transfer protokolünün DNS protokolü ile entegre edilmesi ile günümüzde modern mail transfer sistemi oluşturulmuş oldu. Erim Allman'ın geliştirmiş olduğu delivermail programı da yeni protokoller dünyasına adapte olarak günümüzde bilinen sendmail ismini aldı. İleriki aşamalarda sendmail projesi baştan tekrar tasarlanıp ayrı bir proje olarak postfix adını alacaktı.

3 Mail Protokolleri

Mail doğası itibari ile içerğinde bir mesaj taşır. Bu mesajı gönderenden alıcısına iletmek istediğimizde, mailin bir dizi prosedürden geçmesi gerekiyor. Mailin sağlıklı bir şekilde gönderenden alıcısına ulaşması için bu prosedürlerin, yani kurallar topluluğunun, önceden belirlenip sabit kurallar silsilesine bağlanması gerekir ki mail gönderenden alıcısına sağlıklı bir şekilde ulaşabilsin. Bu kurallar topluluğuna “Mail Protokolleri“ diyoruz.

Mail protokollerini açıklamadan önce gerçek dünya ile benzerlik kuralım. Elektronik dünyadaki maile benzerlik gösteren en iyi örnek günümüzdeki mektup olgusudur.

Örneğin, Ankarada yaşadığınızı varsayalım ve İngiltere'nin Londra şehrinde yaşayan bir arkadaşınıza mektup göndereceksiniz. İlk yapmanız gereken bir mektup zarfı, kağıt ve pul almak olacaktır. Bunları elde ettikten sonra, mektubunuzu yazarsınız, mektubu zarfa koyar, üzerine pulu yapıştırır ve son olarak gönderen ile alıcı adreslerini yazarsınız. Bunlar tamamlandıktan sonra, mektubunuzu Uluslar arası posta kabul eden bir posta kutusuna atarsınız. Böylelikle mektubunuz seyahtine başlamış olur. İlk olarak, mektubunuz şehir posta merkezine götürülür. Adresinden, Uluslararası bir mektup olduğu görülür ve yurtdışına gidecek mektuplar arasına konur. Ardından, İngiltereye gidecek mektuplar bir yerde toplanır ve büyük bir olasılıkla bir uçak aracılığı ile ingiltereye götürülür. Bu aşamadan sonra Ülkemizin Mektup için yükümlülüğü bitmiş olur. Mektup, İngiltere Posta Sistemine dahil olur. İngiltere'de, mektupun alıcı adresine bakılarak, mektup Londra Posta merkezine gönderilir. Londra'da da, tekrar alıcı adresine bakılarak alıcının tam adresine ulaştırılır. Tabii alıcı yerinde olmayabilir veya adres yanlış olabilir. Bu durumda, mektup yine aynı yoldan ama bu sefer ters istikamette gönderene geri ulaştırılır.

Mektup iletimi en başta basit gibi görülebilir ama gördüğünüz gibi yukardaki gibi uzun bir senaryosu vardır. Hatta bu senaryo daha da dallandırılabilir.

Yukarıdaki senaryoyu inceleyelim. Örneğin Türkiye ve İngiltere arasında posta iletimleri konusunda

bir anlaşma olmasa, mektubumuz yerine ulaşır mı? Tabii ki ulaşmaz. Mektup, posta merkezine ulaştığında adreste İngiltere ibaresi görüldüğünde, işleyiş dışına çıkarılırdı. Peki mektup zarfında pul olmasa mektubumuz yerine ulaşır mıydı? Tabii ki yine ulaşmazdı. Pul, bizim posta iletim sistemine giriş hakkımızı gösterir belge. Bir çeşit bize açılan hesap. Bu hesabımızı gösterir bir pul olmaz ise, mektubumuz posta iletim sistemine dahil olamaz. Sorularımıza devam edelim. Peki mektubu bir zarfa koymak zorunda mıyız? Evet koymak zorundayız. Eğer koymazsak, pul ve adres için bir yer olmaz. Dahası, mektubmuzun bir güvenliği kalmaz. Yani mektubumuzun bir formatı var. Bu formata uymayan mektup yazıları posta iletim sistemine dahil olamıyor. Peki mektup zarfı üzerine yazılan alıcı ve gönderen adresleri. Bu adresleri rastgele yazabilir miyiz? Hayır yazamayız. Alıcı için İngiltere adresleme sistemini kullanmamız gerekir, gönderen için de Türkiye adresleme sistemini kullanmamız gerekir.

Görüldüğü gibi bir mektubun yerine ulaşması için uyulması gereken bir sürü standart ve protokol var. Bu standart ve protokoller elektronik dünyada da geçerli. Hatta daha fazlası var ve daha katı şekilde uygulanıyor.

Şimdi yukarıdaki senaryoyu tekrar analiz edelim ve bu sefer hangi birim ne iş yapıyor sıralayalım. Yukarıdaki senaryoda aşağıdaki rollerin var olduğunu söyleyebiliriz:

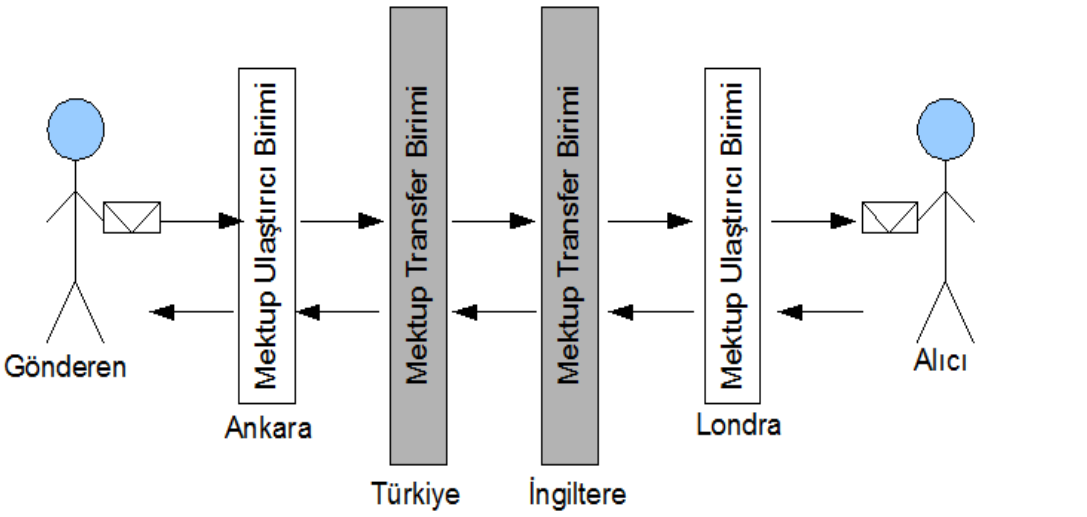
- Gönderen
- Alıcı
- Mektup Transfer birimi (Türkiyeden İngiltereye Gönderen Birim)
- Mektup Alıcı Birimi (İngilterede mektubu Türkiye biriminden alan birim)
- Mektup Toplayıcı Birim (Posta kutusuna koyduğumuz mektubu, Ankara Merkez Posta birimine ulaştıran birim)
- Mektup Dağıtıcı Birim (İngiltereye ulaşan mektubu, Londra Merkez Posta birimine ulaştıran birim)

Yukarıdaki roller Mektup Ankaradan Londraya ulaştırılırken geçerli. Mektubun Londradan Ankaraya geri geldiğini düşünelim. Bu durumda Türkiyedeki Mektup Transfer Birimi İngilteredeki Mektup Alıcı Birimi rolünü üstlenecektir. Aynı durum Mektup Toplayıcı ve Mektup Dağıtıcı birim arasında da geçerlidir. Dolayısı ile Mektup Transfer Birimi ile Mektur Alıcı Birimi aynı rollere sahiptir. Aynı durum Mektup Dağıtıcı Birim ve Mektup Toplayıcı birim için de geçerlidir.

O halde yeni rol dağılımını aşağıdaki gibi yapalım:

- Gönderen
- Alıcı
- Mektup Transfer Birimi (Aynı Zamanda Mektup Alıcı Birimi)
- Mektup Ulaştırıcı Birimi (Mektup Dağıtıcı ve Mektup Toplayıcı Birimi)

Yukarıdaki senaryo ve rolleri göz önünde bulundurursak aşağıdaki şekil senaryomuzu açıklayacaktır.



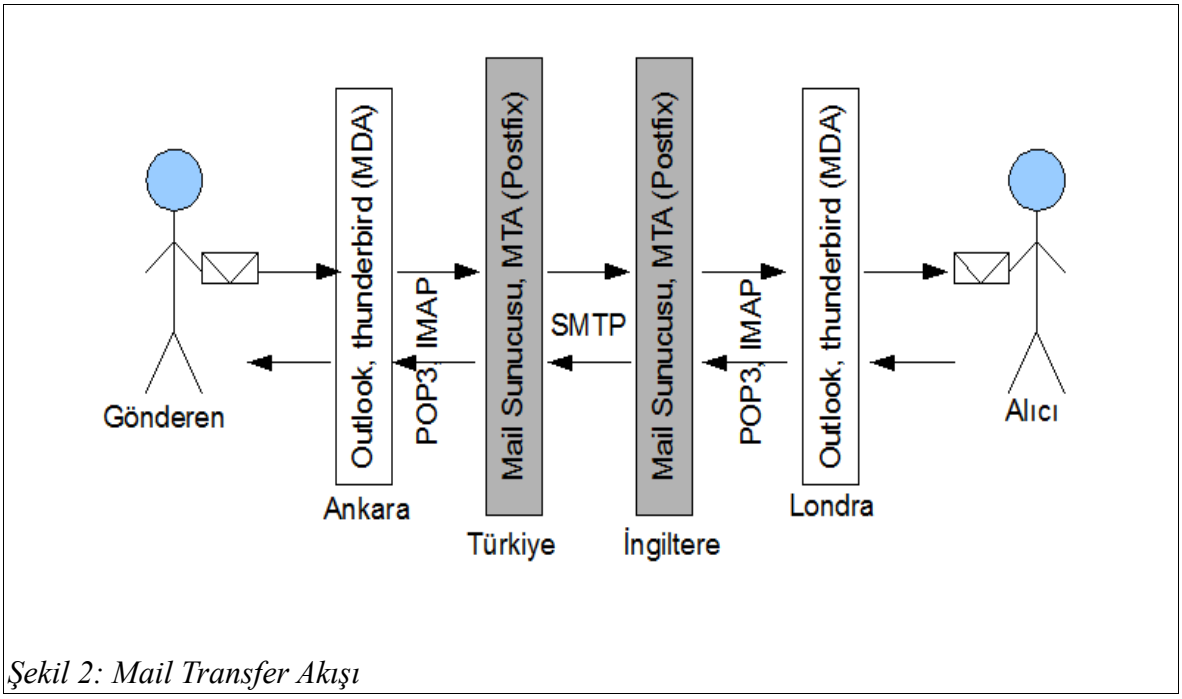
Resim 1: Mektup Transfer Akışı

Yukarıda dikkat edilmesi gereken nokta, mektup transferini Mektup Transfer Biriminin yapmış olmasıdır. Mektup ulaştırıcı birimi, her ne kadar ilk elden mektubu almış olsa da, mektubu kendisi hedefe ulaştırmamaktadır, onun yerine mektubu Mektup Transfer Birimine iletmekte, mektubu transfer etme işini Mektup Transfer Birimi yapmaktadır.

Gerçek hayatta vuku bulan bu senaryo nerdeyse birebir elektronik dünyada da gerçekleşmektedir. Elektronik dünyada, Mektubun yerini Mail, Mektup Ulaştırıcı Biriminin yerini MDA(Mail Devivery Agent), Mektup Transfer Birimi yerini MTA(Mail Transfer Agent) almaktadır.

Burada bahsi geçen MTA'lar Mail Sunuculardır. MDA'lar ise POP3 ve IMAP istemcilerdir. Elektronik Dünyada, Siz mailinizi bir Mail istemcide yazarsınız(Outlook Express veya Thunderbird ile), Outlook Express veya Thunderbird bir MDA görevi görerek mailinizi üye olduğunuz MTA'ya ulaştırır, MTA kendisine ulaşan maili hedef adresine bakarak, DNS yardımı ile hedef MTA makinesini bulur, ve maili hedef MTA'ya iletir. Hedef kullanıcı kendi mail istemcisi ile kendi MDA'sını açtığında (Yani outlook veya Thunderbird'ü açtığında), üye olduğu MTA'ya bir mail ulaşmışsa bu MDA aracılığı ile kendisine ulaşan maili açar ve okur.

Görüldüğü gibi gerçek hayatta vuku bulan senaryo nerdeyse elektronik dünyada da vuku bulmaktadır. O halde elektronik dünyada gerçekleşen bu senaryoyu yeniden çizelim. Şekil 1'deki benzerliğe lütfen dikkat edin.



Şekil 2: Mail Transfer Akışı

Şekil 2'de görüleceği gibi birimler arasındaki Protokoller de belirtilmiştir. 2 MTA arasında iletişim şekli SMTP protokolü ile sağlanmaktadır. Kullanıcı MTA'ya ulaşan maillerini Outlook veya Thunderbird gibi POP3 veya IMAP istemcileri aracılığı ile okumaktadır. Outlook veya Thunderbird MTA ile POP3 veya IMAP protokolü aracılığı ile konuşabilmektedir.

Şekilden de anlaşılacağı gibi mail atmak veya almak istediğinizde MDA birimlerini kullanırsınız. Yani doğrudan Mail sunucusuna(MTA'ya) bağlanmazsınız. Ama bu her zaman gerçekleşen senaryo değiltir. İsterseniz mail atmak için MDA birimini aradan çıkarabilir, doğrudan MTA'ya bağlanarak SMTP protokolü aracılığı ile mail atabilirsiniz. Ama bu çokca tercih edilen bir yöntem değildir. Mail okumak istediğinizde MTA birimini kullanamazsınız. Çünkü MTA sadece mail transferi ile ilgilienir. Eğer size ait bir mail MTA'ya ulaşmışsa, MTA birimi o maili sunucuda bir dosya olarak saklamıştır. Bu dosyaya ulaşmak için MTA'yı yani SMTP protokolünü kullanamazsınız. Size ulaşan bir maili ya POP3 ile ya da IMAP protokolü yardımı ile okuyabilirsiniz. MTA kendisine ulaşan maili sunucuda bir dosya olarak sakladıktan sonra görevini tamamlamış olur. Genellikle MDA ile MTA nın görevleri birbirine karıştırılır. Bu iki birimi şöyle ayırt edebiliriz. MDA Hiçbir zaman mail transferi yapmaz. Sadece iletmek istediğiniz maili MTA'ya iletir. Ya da size ulaşan maili, yani MTA' nın dosya olarak kaydettiği maile ulaşmanıza yardımcı olur. MTA ise size ait olan maili uzaktaki MTA birimine SMTP aracılıoğı ile iletir, ya da uzaktaki MTA size ulaştırmak istediği maili üye olduğunuz MTA'ya ulaştırır, üye olduğunuz MTA da maili sunucuda bir dosya olarak saklar ve görevini tamamlar.

Görüldüğü üzere, bir mail yolculuğu sırasında farklı aşamalardan geçer, ve bu aşamalarda farklı protokoller kullanılabilir. Bu protokoller aşağıdaki gibidir:

- SMTP
- IMAP
- POP3
- SMTP AUTH

SMTP AUTH protokolünden şimdiye kadar bahsetmedik. Ama önemli bir protokol. Aslında kendisi tam anlamıyla bir protokol değildir, bir kimliklendirme mekanizmasıdır. İçeriğinde bir çok farklı kimliklendirme mekanizmasını barındırır. Bu konuyu ilerleyen konularda açacağız.

3.1 SMTP

SMTP protokolü 2 MTA (Mail sunucusu) arasında mail transferi aşamasında kullanılan bir protokoldür. MTA olarak hizmet vermek isteyen her Mail Sunucusu bu protokole uymak zorundadır. Kaynak MTA hedef MTA'ya mail iletmek istediğinde, hedef MTA'nın 25. portuna bağlanır (SMTP sunucusu varsayılan olarak 25. portu dinler) ve böylelikle SMTP protokolü başlamış olur. Mail transferi sonucunda, eğer tüm aşamalar geçilmişse mail hedef MTA'ya ulaşmış olur. Bu aşamadan sonra Hedef MTA, ya maili bir dosya olarak kaydeder ya da başka bir MTA'ya iletir. Başka bir MTA'ya iletme durumunda SMTP protokolü diğer MTA için yeni baştan başlamış olur.

Bu kısa özetten sonra SMTP protokolünü biraz daha detaylı inceleyelim. Bir SMTP sunucusu hem Alıcı MTA hemde gonderen MTA olarak davranabilir. SMTP sunucusu alıcı MTA durumunda, kendisine ulaşan maili, önceden belirtilen bir dizinde saklar ya da başka bir MTA'ya yöneldirir. SMTP sunucusu gönderen MTA durumunda, dışarıya göndermek istediğiniz mailleri uzaktaki alıcı MTA'ya iletir. Öncelikle Alıcı MTA durumunu açalım.

Bir MTA genellikle 25 nolu portu dinler. Bu porta bir istemci bağlandığında, Alıcı MTA eğer herhangi bir problem yok ise, “hazır” olduğu bilgisini istemciye gönderir. Örneğin, aşağıdaki örnek lokalde 25 nolu portu dinleyen MTA'ya bağlantıyı göstermektedir. Telnet programı ile 25 nolu porta bağlanıldığında MTA'nın verdiği cevabı göstermektedir. 220 ile başlayan satır MTA'nın mail kabul edebileceğini göstermektedir.

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 mailserver.interis.com ESMTP Postfix
```

Eğer MTA mail almak için hazır değilse hazır olmadığı bilgisini gönderir. Hazır olduğu durumda genellikle, istemci ilk olarak alıcı MTA'dan neler yapabileceğini sorar. İstemci bu soruyu ehlo ile gerçekleştirmektedir. Ehlo komutu ile istemci veya Gönderen MTA hem kendini tanıtır hem de alıcı MTA'nın neler yapabileceğini sorgular.

```
ehlo abc.com
250-mailserver.interis.com
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-AUTH DIGEST-MD5 LOGIN PLAIN
250-AUTH=DIGEST-MD5 LOGIN PLAIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
```

Alıcı MTA bu durumda, ne büyüklükte mail kabul ettiğini, kimliklendirme için ne tür mekanizmaları desteklediğini istemciye iletir.Yukarıda 250 ile başlayan satırlar bu bilgileri içermektedir. 250-AUTH veya 250-AUTH= ile başlayan satırlar alıcı MTA'nın ne tür kimliklendirme mekanizmalarını desteklediğini göstermektedir. Yukarıdaki alıcı MTA kimliklendirme mekanizması olarak Digest-MD5, Login ve Plain mekanizmalarını kabul etmektedir. Ehlo komutunun çıktısı ile birlikte istemci MTA ne tür bir kimliklendirme mekanizması kullanacağına karar verir. (Bu örneğimizde plain mekanizması kullanılmıştır, daha ayrıntılı bilgi 3.2'de açıklanmıştır.)

Bu aşamadan sonra, istemci alıcı MTA'ya kimliklendirme için bilgilerini gönderir. Bu bilgiler genellikle kimliklendirme yöntemi, kullanıcı adı ve paroladır. Kimliklendirme aşaması aşağıdaki gibi yapılmaktadır.

```
auth plain dXNlcjFAZG9tMS50ZXN0AHVzZXIxQGRvbTEudGVzdABxYXp3c3g=
235 2.7.0 Authentication successful
```

Alıcı MTA sahip olduğu kimliklendirme mekanizmalarına göre kullanıcı adını ve parolayı doğrular. Eğer doğrulama geçerli ise, istemci sırası ile gönderen mail adresi ve alıcı mail adreslerini girer ve ardından mailin içeriğine geçer. Bu aşama aşağıda resmedilmiştir.

```
mail from: user1@dom1.test
250 2.1.0 Ok
rcpt to: user2@dom1.test
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
ilk mail testi.
.
```

Mail içeriği yazıldıktan sonra, yukarıdaki gibi mail içeriğinin en son satırına ".\cr\lf" yazılarak mail sonlandırılır. Böylelikle alıcı MTA, mail içeriğini elde etmiş olur. Alıcı MTA alıcı adresine bakar, eğer alıcı adresi kendi kullanıcısı ise maili kendi sisteminde bir dizinde saklar, eğer alıcı adresi kendi kullanıcı değilse maili ilgili MTA'ya yönlendirir. Böylece alıcı MTA'nın görevi sonlanmış olur.

3.2 SMTP AUTH

SMTP AUTH bir kimliklendirme mekanizmasıdır. Kullanıcı MTA aracılığı ile mailini iletmeden önce, kendini tanıtmak zorundadır. Kendini tanıtma kullanıcı adı ve parola aracılığı ile olmaktadır. Yalnız, kullanıcı adı ve parola doğrulaması bir çok farklı şekillerde olabiliyor. Örneğin, kullanıcı adı ve parola veri tabanında saklanıyorsa, girilen kullanıcı adı ve parola veri tabanındaki kullanıcı adı ve parola ile eşleştirilebilir. Eğer eşleştirme doğru ise doğrulama gerçekleşmiş olur ve kullanıcı kimliklendirilir.

Ama her zaman, kullanıcı adı ve parola veri tabanındaki kullanıcı adı ve parola ile eşleştirilmiyor. Örneğin, kullanıcı adı ve parolanın MD5 alınıp doğrulama yapılabilir. Ya da kullanıcı adı ve parolanın Base64 kodlaması alınıp doğrulama yapılabilir. Dahası, kullanıcı adı ve parola veri tabanında saklanmak zorunda değil. Bu bilgiler, bir dosyada (örneğin /etc/passwd) ya da LDAP dizininde saklanıyor olabilir.

Görüldüğü üzere, doğrulama aşamasında kullanılan yöntemler(MD5, Base64, ya da doğrudan eşleştirme) ve verinin(kullanıcı adı ve parola) saklanma şekli doğrulama mekanizmamızı değiştirebiliyor.

Geçmiştin günümüze, bir çok doğrulama mekanizması geliştirildi, ve bu mekanizmalar hala kullanımda. Eğer SMTP sunucumuzun sadece tek bir doğrulama mekanizmasını kullanmasını istersek, bu şu ana kadar varolan bir çok sistemin fişini çekmemiz anlamına geliyor. Dolayısı ile yapılması gereken, SMTP sunucumuzun aynı anda birden fazla doğrulama mekanizmasını kullanmasını sağlamak.

Şu anda da bir çok Mail Sunucusu birden fazla doğrulama mekanizması kullanmaktadır. Bu mekanizmalardan en çok kullanılanlar aşağıdaki gibidir:

- AUTH PLAIN
- AUTH LOGIN
- AUTH CRAM-MD5
- AUTH DIGEST-MD5

SMTP sunucusunun yukarıdaki 4 mekanizmayı kullanabilmesi için bir güvenlik katmanının inşa edilmesi ve bu 4 mekanizmanın bu katman içinde yönetilmesi gerekmektedir. SMTP Authentication RFC 1869 bu görevi yerine getirecek şekilde tasarlandı. Bu dokuman tasarlanırken, SASL(Simple Authentication and Security Layer, RFC 2222) mekanizmasından yararlanıldı.

Yukarıda tanımlanan 4 mekanizmanın açıklaması aşağıdaki sıralanmıştır.

AUTH PLAIN

AUTH PLAIN mekanizmasında, smtp istemcisi smtp sunucusuna kullanıcı adı ve parolayı base64 kodlaması yaparak gönderir. Örneğin kullanıcı adı user1@dom1.test ve parolası qazwsx olsun. Öncelikle, aşağıdaki tekstin base64 kodlaması alınır.

```
\000user@dom1.test\000qazwsx
```

Ardından aşağıdaki gibi doğrulama işlemi yapılır. (“auth plain” yazısından sonra gelen karakterler topluluğu yukarıdaki kullanıcı ad ve parolanın base64 kodlanmış halidir.)

```
auth plain dXNlcjFAZG9tMS50ZXN0AHVzZXIxQGRvbTEudGVzdABxYXp3c3g=
235 2.7.0 Authentication successful
```

Bu mekanizmada dikkat edilmesi gereken tek bir satırda kullanıcı adı ve parolanın gönderilmesidir.

AUTH LOGIN

Auth Login mekanizması ile Auth Plain mekanizması neredeyse aynıdır. Aralarındaki tek fark auth plain mekanizmasında kullanıcı adı ve parolası tek bir hamlede gönderilirken, auth login mekanizmasında önce kullanıcı adı gönderilir, ardından kullanıcı parolası gönderilir. Bu iki gönderim durumunda da base64 kodlaması kullanılır. Aşağıdaki akış Auht login mekanizmasını resmetmektedir. (S: sunucu cevabını C: istemci girdisini işaret etmektedir)

```
C: auth login
S: 334 VXNlcm5hbWU6
C: avlsdkfj
S: 334 UGFzc3dvcmQ6
C: lkajsdfvlj
S: 535 authentication failed (#5.7.1)
```

AUTH CRAM-MD5

Bu mekanizma Auth Plain ve Auth Login mekanizmalarından biraz daha karışık ve daha güvenlidir. Ana teması ortak şifre kullanma üzerinedir. SMTP istemcisi ilk olarak SMTP sunucusuna bağlandığında, SMTP sunucusu o an rastgele üretilen bir şifreyi istemciye açıktan ve base64 kodlu gönderir. İstemci kullanıcı adını sunucuya gönderir ama parola kısmını bu açıktan gönderilen şifre ile MD5 şifrelemesi yaparak gönderir. Bu şifreleme sonucunda ortaya çıkan karakter dizisi, sunucu tarafında, Orijinal parolanın gönderilen aynı açık şifre ile MD5 şifrelemesi yapılmış hali ile karşılaştırılır. Eğer sunucu tarafında üretilen karakter dizisi, istemci tarafından sunucuya gönderilen karakter dizisi ile aynı ise, doğrulama kabul edilmiş olur. Aksi halde parola yanlış girilmiş sayılır.

AUTH DIGEST-MD5

Bu mekanizma CRAM-MD5 mekanizması ile nerdeyse birebir aynıdır. Tek fark, SMTP sunucusu istemci tarafına açıktan bir şifre göndermez. Dolayısı ile istemci de bu işfre ile parolayı tekrardan şifrelemez. Bunun yerine istemci, standart bir MD5 mekanizması ile kullanıcı parolasını şifreler ve sunucuya gönderir, sunucu da aynı şekilde standart md5 şifrelemesi ile parolayı şifreler ve istemciden gelen şifrelenmiş parolayı kendisininki ile karşılaştırır. Eğer sunucu tarafında üretilen karakter dizisi, istemci tarafından sunucuya gönderilen karakter dizisi ile aynı ise, doğrulama kabul edilmiş olur. Aksi halde parola yanlış girilmiş sayılır.

3.3 POP3

Şimdiye kadar 2 MTA arasında mail transferinden bahsettik. Ama mail hedefteki MTA'ya ulaştıktan sonra neler olduğundan pek bahsetmedik. Maili uzaktaki MTA'ya ulaştırmamızın amacı uzaktaki MTA'ya üye kullanıcının gönderilen maili okuması idi.

Uzaktaki MTA'ya mail ulaştığında, uzaktaki MTA'nın bu maili bir dosya olarak kaydettiğinden bahsetmiştik. Ama şimidye kadar kullanıcı bu dosyaya nasıl ulaşır ve okur ondan bahsetmemiştik. Şimdi gelin bu konu üzerinde düşünelim.

Size ulaşan maili nasıl okursunuz?. Bu sorunun bir çok cevabı mevcut. En basitten ve kronolojik sıra ile bu soruyu cevaplamaya çalışalım. Eğer mail sunucusunun yüklü olduğu işletim sisteminde bir kullanıcı hesabınız var ise, maili okumanız çok kolay. Mail sunucusu size ulaşan maili, sizin ev dizininizdeki bir dizine veya dosyaya mutlaka saklamıştır. Bu dizin genelde Mail veya Maildir ismine sahip olur. Size ulaşan mail bu dizinler altında yer almış olacaktır.

Sisteme login olduğunuzda işletim sistemi konsolda size gelen bir mail olduğunu haber veren bir mesaj gösterecektir. Büyük bir ihtimalle siz de “mail”, “mutt” ya da “pine” adındaki mail istemcileri ile maillerinizi okuyacaksınız. Bu programlar, size mail dizinindeki size ulaşan mailleri listeleyen

üzerinde işlemler yapmanızı sağlayan arayüzler sağlayacaktır.

Gördüğünüz gibi mail size ulaştıktan sonra sizin SMTP veya başka bir protokol ile işiniz kalmayabilir. Doğrudan mailinizi programlar aracılığı ile okuyabilirsiniz. Hatta sizin bu programları kullanmanıza gerek olmayabilir. Mail dizini altına gidip tarih bakımından en yeni dosyayı listelediğinizde size ulaşan mailin o dosya olduğunu anlayabilirsiniz. İstediğiniz metin işlemci ile mail dosyasını açabilir ve mailinizi okuyabilirsiniz.

Ama bu söylediklerimizin hepsi eğer mail sunucusunu barındıran işletim sisteminde bir hesabınız ve konsolunuz var ise geçerli. Peki ya hesabınız yok ise uzaktaki maillerinize nasıl ulaşacaksınız? Bu soruyu cevaplamadan önce şunu farketmeniz gerekmektedir. Problem artık bir mail okuma değil, bir dosyaya ulaşmadan ibarettir. Bu dosyanın tek özelliği mail formatında olmasıdır.

Bu sorunun da bir çok cevabı mevcuttur. İlk akla gelen FTP ile dosyayı almak olabilir. İlk zamanlarda yapıldığı gibi. Sonuçta amaç bir dosyaya ulaşmak. Eğer mail sunucusunu barındıran işletim sisteminde FTP sunucusu mevcutsa ve bu FTP sunucusunda bir FTP hesab ınız var ise, bir konsola ihtiyaç duymadan da maillerinizi okuyabilirsiniz. Tek yapmanız gereken FTP hesabınız ile FTP sunucusuna bağlanmak ve mail dizinindeki dosyaları kendi lokal makinenize çekmek ve ardından mailleri kendi lokal makinenizde okumak. Bu da geçmişte uygulanan bir çözüm.

Ama günümüzde yukarıdaki 2 yöntem de kullanılmamakta. Yukarıda listelenen mail okuma yöntemleri, maile salt dosya olarak yaklaşım sonucunda ortaya çıkan yöntemler. Mail'i diğer dosyalardan ayıran özellikleri mevcut. Bu özellikler de içeriğinde taşıdığı farklı bilgiden kaynaklanmakta. Örneğin, siz maillerinizi tarihe göre sıralamak isteyebilirsiniz. Maillerinizin içinde gönderene göre arama yapmak isteyebilirsiniz. Maillerinizin hangisini cevaplamışsınız görmek isteyebilirsiniz. Bu ihtiyaçların hepsi maili diğer dosyalardan ayırmakta. Bu durumda maillere ulaşma ve onları okuma, üzerinde manipule yapma yöntemlerini değiştirmektedir. Bu sebeple, sizin hesabınıza ulaşan mailleri okumak veya onlara ulaşmak için farklı bir protokol daha geliştirildi. Bu protokolün geliştirilmesindeki amaç uzakta sizin hesabınıza kayıtlı maile ulaşımı ve üzerinde manipulasyonu kolaylaştırmak idi. Bu protokolün adı POP3'tür.

POP3 protokolü, basit bir dosya transfer protokolüdür. Ama mail dosyaları için özelleşmiştir. Her internet protokolü gibi, bir istemci ve bir de sunucu tanımı yapar. POP3 aracılığı ile uzaktaki size ulaşan maili okumak istiyorsanız, uzaktaki makineye POP3 sunucusu kurmalısınız. Bu sunucu ile iletişime geçebilmek için POP3 istemcisi kullanmalısınız. POP3 istemcisi ve POP3 sunucusu aralarında POP3 protokolüne uyarak mail transferi yaparlar.

POP3 protokolü örneğini telnet programını kullanarak uygulayalım. POP3 sunucusu varsayılan olarak 110 portunu dinler. Aşağıdaki komut yardımı ile POP3 sunucusuna bağlanabiliriz.

```
telnet pop3_sunucu_adresi 110
```

POP3 protokolünde kimliklendirme işlemi aşağıdaki gibi yapılmaktadır.(C: istemci komutunu, S. sunucu cevabını göstermektedir.)

```
C:    USER halil
S:     +OK User accepted
C:    PASS parola
S:     +OK Pass accepted
```

Size gelen mailleri listelemek için aşağıdaki “LIST” komutu kullanılmaktadır. Aşağıdaki komut LIST komutunu örneklemektedir. Maillerin 1,2 olarak numaralandırıldığına dikkat ediniz.

```
C:    LIST
S:    +OK 2 messages (620 octets)
S:    1 420
S:    2 200
S:    .
```

Mail okumak için RETR komutu kullanılmaktadır. 1 numaralı maili okumak için aşağıdaki örnek gösterilmektedir.

```
C:    RETR 1
S:    +OK 420 octets
S:    <the POP3 server sends message 1>
S:    .
```

POP3 protokolünde dikkat edilmesi gereken nokta, eğer POP3 ile bir mail okuyorsanız, ilgili mail uzaktaki POP3 sunucusundan silinir ve kendi makinanıza indirilir.

3.4 IMAP

IMAP protokolü POP3 protokolü ile aynı işlevi görür. IMAP ve POP3 aynı kulvarda olan 2 protokoldür. Yalnız IMAP'in, POP3'den daha gelişkin özellikleri vardır. IMAP ile bir çok dizin oluşturabilirsiniz ve bu dizinlerden birini varsayılan mail kutunuz olarak atayabilirsiniz. IMAP ile mail okurken, sunucudan maillerin silinmemeisini sağlayabilirsiniz. Ya da POP3'deki gibi okuma anında makinenize indirilmesini ve sunucudan mailin silinmesini sağlayabilirsiniz.IMAP'in sağladığı dizin yapısından dolayı, mailler üzerinde gruplama veya filtreleme sağlayabilirsiniz.

Görüldüğü gibi IMAP'in bir çok özelliği mevcut. Peki IMAP protokolünün ne tür komutları var bunları inceleyelim.

IMAP sunucusu varsayılan olarak 143 numaralı portta dinleme yapar. Aşağıdaki komut yardımı ile IMAP sunucusuna bağlanabilirsiniz. Eğer sunucu * OK" ile başlayan bir cevap dönmüşse, sunucu bağlantı kurmaya hazır demektir.

```
telnet imap_sunucusu 143
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
libshishi: warning: /root/.shishi/tickets: No such file or directory
* OK [CAPABILITY IMAP4rev1 UIDPLUS CHILDREN NAMESPACE
THREAD=ORDEREDSUBJECT THREAD=REFERENCES SORT QUOTA IDLE ACL
ACL2=UNION] Courier-IMAP ready. Copyright 1998-2005 Double Precision, Inc.  See
COPYING for distribution information.
```

Imap protokolünde sunucuya gönderilen her komutun bir numarası vardır. Sunucu kendisine ulaşan komutları cevaplarken bu komut numarasını kullanır. Örneğin aşağıda kimliklendirme komutu "10" ile numaralandırılmıştır. Sunucunun verdiği cevap da 10 ile başlayacaktır.

```
10 login user1@dom1.test qazwsx
```

Yukarıdaki komuta sunucunun verdiği cevap aşağıdaki gibidir. Kimliklendirme işlemi başarılı olmuştur.

```
10 OK LOGIN Ok.
```

Aşağıdaki komut, mail dizini altındaki tüm mailleri listelemek için kullanılmıştır.

```
* LIST (\NoSelect) "/" Mail
* LIST (\NoInferiors \UnMarked) "/" Mail/dc214
* LIST (\NoInferiors \Marked) "/" Mail/Sent
* LIST (\NoInferiors \UnMarked) "/" Mail/Drafts
* LIST (\NoSelect) "/" Mail/inbox
* LIST (\NoSelect) "/" Mail/inbox/new
* LIST (\NoSelect) "/" Mail/inbox/cur
* LIST (\NoSelect) "/" Mail/inbox/tmp
* LIST (\NoSelect) "/" Mail/outbox
* LIST (\NoSelect) "/" Mail/outbox/new
* LIST (\NoSelect) "/" Mail/outbox/cur
* LIST (\NoInferiors \UnMarked) "/" Mail/.outbox.index
* LIST (\NoInferiors \UnMarked) "/" Mail/.outbox.index.ids
* LIST (\NoSelect) "/" Mail/sent-mail
* LIST (\NoSelect) "/" Mail/sent-mail/new
* LIST (\NoSelect) "/" Mail/sent-mail/cur
* LIST (\NoSelect) "/" Mail/sent-mail/tmp
a02 OK LIST complete
```

3.5 Mailbox ve Maildir Formatları

Mail tarihçesini anlatırken, gelen maillerin tek bir dosyad asaklandığından bahsetmiştik. Bu dosya günümüzde bilinen mbox dosyasıdır. Gelen tüm mailler mbox dosyasında ardı sıra saklanır. Mbox dosyasında, mailleri birbirinden ayıran satır “From :” ile başlayan satırdır. Her “From :” satırı ile başlayan ve en sonunda bir boş satır ve tekrar yeni bir “From :” satırı ile biten satırlar toplamı bir mail demektir.

Mailbox formatı mail tarihi kadar eskidir. Ama günümüzde artık pek kullanılmamaktadır. Çünkü kendisi kullanım sırasında bir çok probleme sebep olmaktadır. Örneğin, bir kullanıcıya aynı anda iki mail gelsin ve 2. mail alınırken bir problem oluşsun ve mail iletimi tamamlanmasın. Eğer işletim sisteminde dosya kitleme sistemi mevcut değilse, ya sadece düzgün ulaşan mail mbox dosyasına yazılacak, ya da hatalı alınan mail sonu boş satır bırakılmadan mbox dosyasına kaydedilecek. İlk durum doğru durum ama 2. oluşan durum problemli bir durum. Hem mailin tamamı alınmamış hem de, mbox formatına uyulmamış durumda. Eğer, işletim sistemi dosya kitleme sistemi kullanırsa, bu durum çözülebilir ama bu sefer de performans problemi başgösterir. Çünki paralel olarak ulaşan tüm mailleri, kaydederken sıralamış oluyorsunuz.

İşte maildir formatı mbox formatının bu problemlerini çözmek için yaratıldı. Maildir formatında, mailler, tek bir dosyaya kaydedilmez. Her dosya ayrı bir dosya olarak kaydedilir. Dosya isimleri sistem tarafından bilinen bir yöntem ile üretilir. Ayrıca, maildir formatında sadece 1 dizin kullanılmaz, 3 dizin kullanılır. Bu dizinler:

- new
- cur
- tmp

dizinleridir. Yeni gelen mailler, mail transfer işlemi tamamlanıncaya kadar tmp dizini altında saklanır. Mail transferi tamamlanınca, mail new dizini altına taşınır. Eğer kullanıcı new dizini altındaki maili okur ise, mail cur dizini altına taşınır. Bu dizinler arasında mail taşıma veya kopyalama yapabilirsiniz. Mail gönderme aşamasında, yine mail transferi tamamlanıncaya kadar, mail tmp dizini altında saklanır. Mail transferi tamamlandığında, tmp dizini altındaki mail dosyası silinir.

3.6 MIME Formatı

Şu ana kadar mail ile sadece mesaj iletiminde bahsettik. Ama maille video, resim, müzik gibi medyalar da iletilebilmekte. Peki bu durum nasıl olabilmekte?

Öncelikle şunu belirtelim: Mail içeriğinde sadece 7-bit ascii karakterler olabilir. Yani sadece

Amerikan ASCII tablosunun tanımladığı karakterler olabilir. Örneğin Türkçe karakterler ya da Çince karakterler olamaz. Ya da ikili sismtle ifade edilen resim, müzik, video dosyaları olamaz. Peki biz mail ile bu medyaları veya karakterleri nasıl iletebiliyoruz. Cevabı basit:MIME formatı ile.

MIME formatının amacı, herhangi bir datayı 7-bit ile ifade edebilmek. Eğer sadece bir birimlik 7-bit ile ifade edilemiyorsa, 2 tane 7-bit ardı ardına sıralanır 14 bit ile ifade edilir. Eğer onunla da ifade edilemiyorsa, tahmin edeceğiniz üzere daha fazla 7-bit ardı ardına sıralanır. Peki bu 1-byte olan 8-bit için de mi geçerli. Nitekim 7-bit ile tüm 8-bitlik datayı ifade edemezsiniz. Evet bu 8-bit için de geçerli. Böylelikle, Türkçe veya Çince karakterlerin nasıl iletildiğine cevap bulabildik.

Peki mail ile resim dosyasını ve beraberinde mail içeriğini nasıl gönderiyoruz? Bunun da cevabı basit, mail içeriğinde bu iki datayı bir ayıraç ile ayırıyoruz. Bu ayıraç da tabiki ascii karakterleri oluyor.

Aşağıda MIME kodlu bir mail içeriği yer almaktadır. Bu mail içeriğinde, mailin gerçek içeriği ile mailin ek dosyası “XXXXayiracXXXX” karakterleri ile ayrılmıştır. Bu bilgi, mailin başlık bilgisinde “Content-Type” başlıpında yer almaktadır. Ve mailde görüldüğü gibi mailin içeriği ile mailin ek dosyası ayırtedici karakter olan “XXXXayiracXXXX” karakterleri ile ayrılmıştır. “muzik.mp3” ikili dosyası da 7-bitlik ascii karakterleri ile ifade edilmiştir. Müzik dosyasının içeriğinin aşağıdaki örnekte yer alması lüzumsuz görüldüğünden, içerik yansıtılmamıştır.

```
From: halil agin <halil.agin@gmail.com>
MIME-Version: 1.0
Content-Type: multipart/mixed;boundary="XXXXayiracXXXX"

This is a multipart message in MIME format.

--XXXXayiracXXXX
Content-Type: text/plain

mime içerikli mail formatinin asil icerigi.

--XXXXayiracXXXX
Content-Type: text/plain;
Content-Disposition: attachment; filename="muzik.mp3"

7-bitlik ascii karakterleri ile ifade edilmis muzik dosyası

--XXXXayiracXXXX--
```

4 Postfix

Postfix, daha önce de belirttiğimiz gibi Sendmail'in yerine geçmek amacı ile yazılmıştır. 1998 yılında Wietse Venema tarafından yazılmıştır. IBM de bu Wietse Venema'ya destek sunmuştur. Wietse'nin projesi IBM tarafından “Secure Mailer” (Güvenli Postacı) olarak adlandırılmıştır.

Postfix'in yazılım amacı, sendmail'in özelliklerini taşıyan ama daha basit ve daha kolay bir yönetime izin veren bir MTA geliştirmektir. Bunu gerçekleştirmek için, Sendmail'den farklı olarak, Postfix bir çok küçük parçacıktan oluşmaktadır. Bu yapı Postfix'i daha kolay yönetilebilir hale getirmektedir. Aynı şekilde, Postfix'in iki ayar dosyasının(main.cf ve master.cf) yapısı da Postfix'in kolay kullanımına katkı sunmaktadır.

Postfix'in modüler yapısı ve kolay kullanılabilir olması kısa süre içerisinde popüler bir MTA olmasına yol açtı. Sendmail gibi hantal ve eski bir sistemin yerini alabilmesi de, sektördeki bir çok Sendmail kullanıcısını heveslendirdi, Postfix kullanmaya teşvik etti.

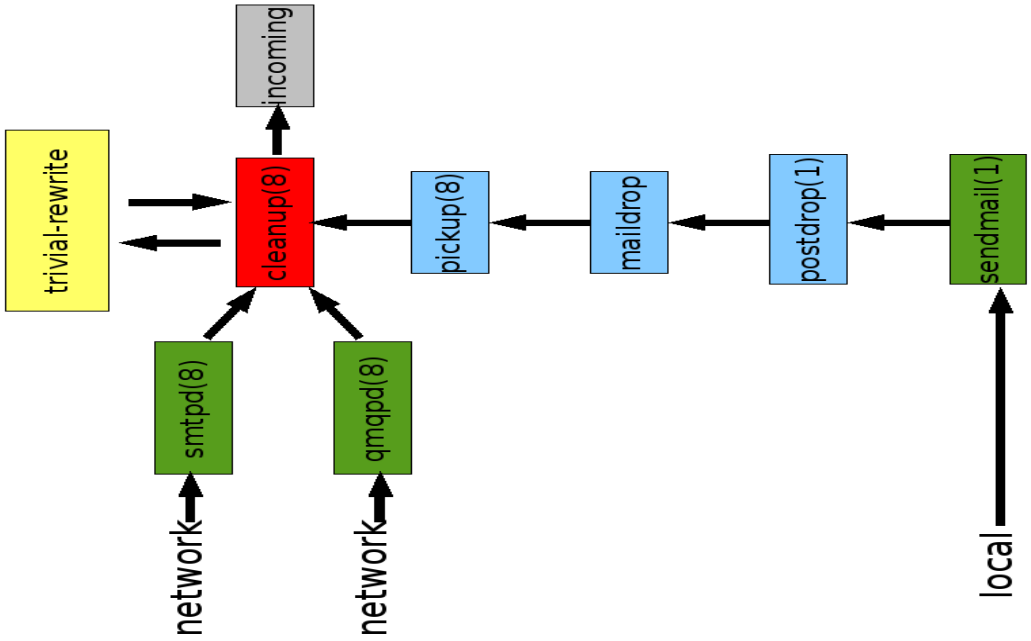
4.1 Postfix'in Bileşenleri

Postfix, modüler yapısı itibari ile birçok bileşene sahiptir. Bazı bileşenler sadece Mail gönderim sırasında aktif iken, bazı bileşenler de sadece Mail alımı sırasında aktiftir. Bunun dışında, hem gönderim sırasında hem de alım sırasında aktif olan bileşenler de mevcuttur.

Postfix'in bileşenleri aşağıdaki tabloda listelenmiştir.

Mail Alımı Sırasında Aktif Olan Bileşenler	Mail Gönderimi Sırasında Aktif Olan Bileşenler
smtpd	incoming
qmqpd	active
pickup	deferred
sendmail	trivial-rewrite
postdrop	qmgr
maildrop	smtpd
cleanup	lmtp
trivial-rewrite	local
incoming	virtual

Postfix Bileşenleri ve Mail Alım Adımları



Şekil 1: Postfix Bileşenleri ve Mail Alım Adımları

Postfix, Mail alırken yukarıda resmedilen bileşenleri kullanır. “Network” ve “local” olarak adlandırılan noktalar mailin geliş noktalarıdır. Eğer bir mail network üzerinden Postfix MTA'sına ulaşıyorsa ya SMTPD ya da QMQPD postfix bileşeni tarafından karşılanır.

QMQPD protokolü gelecekte SMTPD protokolünün yerini alması düşünülen bir protokoldür. SMTPD'ye göre bir çok avantajı vardır. Ama şu an internet dünyasında pek kullanılmamaktadır. Bu sebeple burada bahsedilmeyecektir.

Network üzerinden gelen bir mail SMTPD üzerinden kabul edildiğinde, öncelikle SMTP protokünden kaynaklı bazı işaretlemeler mailden silinir. Ardından, Postfix kendi akışını korumak için elde kalan mail içeriğine bazı güvenlik testleri uygular. Bu testlerden geçen mail cleanup modülüne aktarılır.

Cleanup modulüne ulaşan bir mail, incoming(gelen) kuyruğuna koyulmadan önce bazı aşamalardan geçer. Bu modülde, eğer gelen mailin “From:” ve benzer diğer başlıklarda bir eksiklik veya hata varsa düzeltme yapılır. Ardından, gelen ve giden adresler için adreslerin yeniden düzenlenemesi gerekebilir, bu adımları da Cleanup modulü yerine getirir. Bu adımlar yerine getirilirken Postfix'in main.cf dosyasında tanımlı bazı ayar dosyaları kullanılır. Bu konfigürasyon parametreleri ve nasıl kullanılacağı, postfix ile gelen cleanup(8) man sayfalarında açıklanmıştır.

Cleanup modulü işlemlerini tamamladıktan sonra, işlem gören maili gelen kuyruğuna aktarır ve kuyruk yöneticisini(qmgr) durumdan haberdar eder.

Lokal makineden Postfix'e ulaştırılan mailler, sendmail komutu aracılığı ile iletilirler. Sendmail komutu, sendmail MTA'sı ile karıştırılmamalıdır. Linux dünyasında her MTA'nın bir sendmail isminde komutu vardır ve aynı görevi üstlenir: Lokaldeki maili lokaldeki MTA'ya ulaştırmak. Bu durum Postfix ve Postfix paketi ile gelen sendmail komutu için de geçerlidir.

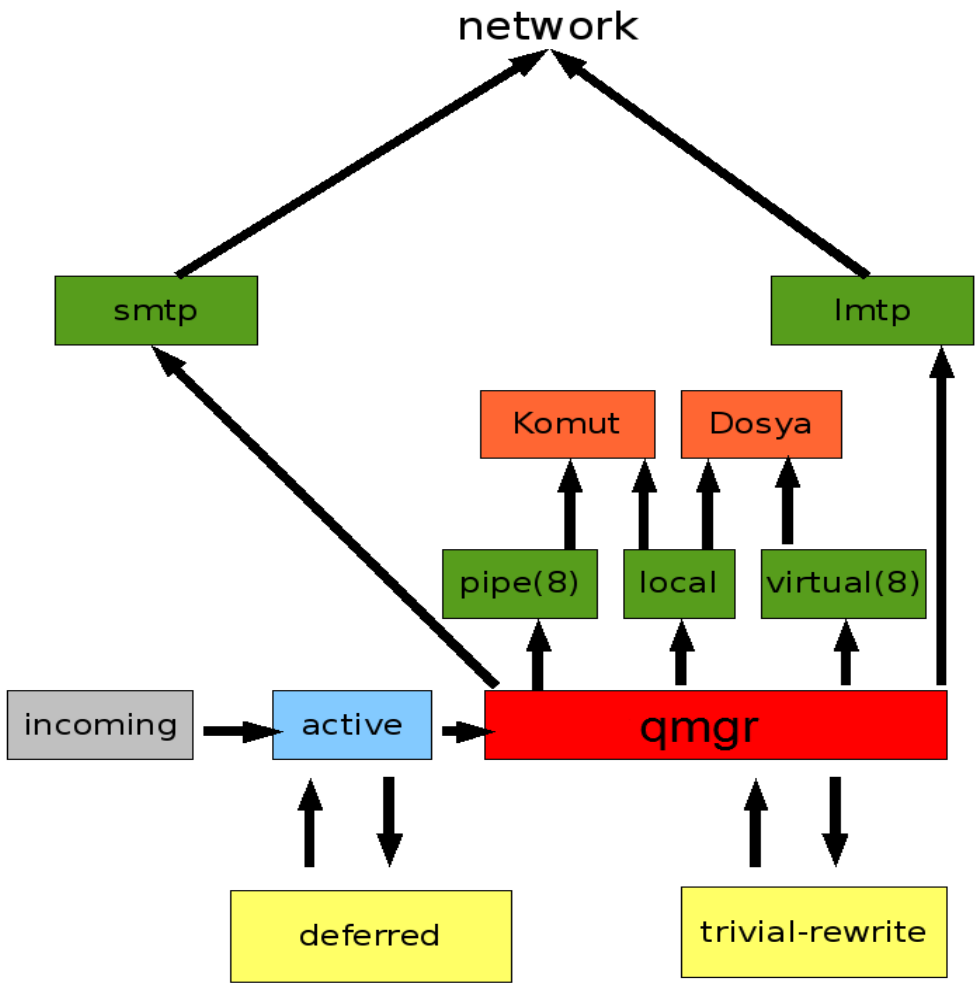
Sendmail ile gönderilen lokal mail öncelikle, postdrop komutu aracılığı ile maildrop kuyruğuna bırakılır. Maildrop kuyruğu, gelen kuyruğundan farklı olarak Postfix o an çalışmasa bile gelen mailleri tutabilmektedir. Postfix ayağa kalktıpında pickup servisi de ayağa kalkmış demektir. Pickup servisi düzenli aralıklarla maildrop kuyruğuna bakar eğer yeni bir mail ulaşmışsa, bu maili cleanup servisine aktarır. Böylelikle, Sendmail-postdrop-maildrop-pickup işbirliği ile lokaldeki mailler Postfix çalışmasa bile hedeflerine ulaştırılabilme imkanına sahip olabiliyorlar.

Trivial-rewrite(Küçük yeniden yazım) servisi, gelen ve giden adreslerinin yeniden yazımından sorumludur. Yeniden yazım işlemi yapılırken Postfix'in konfigürasyon komutlarından yararlanılır. Bu konfigürasyon komutu main.cf ayar dosyasının bir komutudur. Bu komutlardan bazıları aşağıdaki gibidir:

- myorigin
- allow_percent_hack
- append_at_myorigin
- remote_header_rewrite_domain

Daha fazla bilgi için trivial-rewrite(8) man sayfasına bakılabilir.

Postfix Bileşenleri ve Mail Gönderim Adımları



Şekil 3: Postfix Bileşenleri ve Mail Gönderim Adımları

Yukarıdaki şekilden anlaşılacağı gibi qmgr(Kuyruk yöneticisi), Mail gönderim aşamasında Postfix'in kalbini oluşturmaktadır. İncoming(Gelen) kuyuruğuna düşen mailler öncelikle active(aktif) kuyruğuna alınır. Aktif kuyruğundaki mailler qmgr aracılığı ile hedeflerine ulaştırılmaya çalışılır.

Gönderilemeyen mailler, deferred(Ertelenen) kuyruğuna alınır ve bellir bir zaman diliminden sonra tekrar aktifleştirilip qmgr aracılığı ile hedef adrese ulaştırılmaya çalışılır.

Aktif kuyruğunun büyüklüğü ertelenen kuyruğunun büyüklüğünden nispeten daha küçüktür. Bu dizayn, Postfix'in daha performanslı çalışmasına olanak sağlamaktadır. Tersi durumda, qmgr servisi büyük bir aktif kuyruğundaki mailleri sürekli göndermek zorunda kalacaktı. Bu dsizayn, aktif kuyruğunu küçük tutmakta ve gönderilemeyen mailleri aktif kuyruğundan ertelenen kuyruğuna almaktadır. Ertelenen kuyruğundaki mailler belli bir zaman aralığından sonra tekrar aktifleştirilmekte, ve ardı sıra gönderim denemeleri yapılmaktadır. Gönderilemeyen mailler tekrar aktif kuyruğundan ertelenen kuyruğuna geri konmaktadır.

Trivial-rewrite(Küçük yeniden yazma) servisi, mail alım durumumdanki rolü ile aynı role sahiptir. Gönderilmek üzere olan maillerin ”from:” ve “to:” başlıkları bu servis tarafından tekrar yaratılmaktadır. Bu baştan yaratım aşaması, daha önce de bahsedildiği gibi main.cf ayar dosyasında bazı paramtreler ile yönetilmektedir.

Qmgr servisi, iletilmek üzere kendisine ulaşan maili, giden adresine bakarak(“to:”), aşağıdaki beş servisten birine yönlendirir:

- smtp
- lmtp
- local
- virtual
- pipe

Smtp servisi, kendisine ulaşan mailin giden adresine bakıp, hangi SMTP sunucuları ile iletişime geçeceğini belirler ve bunları bir liste içinde saklar. Ardından, listetedi her bir SMTP sunucusu ile iletişime geçip ilgili maili iletmeye çalışır. Uzaktaki SMTP sunucusu ile iletişime geçildikten sonra, smtp servisi gelen ve giden adreslerini yazar, mesaj içeriğini, 7-bit kodlamaya çevirir ve gönderir. Genellikle 8-bit mime türünden 7-bit'lik kodlamaya dönüşüm yapılır.

Lmtp servisi, kendisine ulaşan bir mailin giden adresine bakıp, hangi Mailbox sunucusu ile iletişim kuracağını belirler. Burada mailbox sunucusu basit bir mailbox dosyası, mailbox dizini veya bir program olabilir. Ama genellikle Cyrus gibi mailbox sunucusu(ya da sistemi)'dur.

Lokal servisi, kendisine ulaşan bir mailin giden adresine bakıp, mailin lokaldeki makinede hangi mailbox dosyasına, sistemde tanımlı alias'a, ya da .forward dosyasına gideceğine karar verir. Hedef dosya veya sistem belirlendikten sonra lokal servisi iletimi gerçekleştirir.

Virtual servisi, farklı alan adlarını kullanarak mail iletimi yapmanıza olanak sağlar. Birden fazla mail alan adını hizmet olarak sunmak istiyorsanız virtual servisini kullanmalısınız. Virtual servisi, kendisine ulaşan bir mailin giden adresine bakıp, handi alan adını kullanacağını tespit eder ve o alan adı ile uzaktaki SMTP sunucusuna bağlanıp mail iletimini gerçekleştirir. Mail iletimi lokalde gerçekleşiyor ise, yine mail alan adını giden adresine yazıp lokaldeki mailbox sunucusuna veya dosyaya maili iletir.

Pipe servisi, sürekli internet bağlantısı olmayan uzaktaki hedef alıcı için mail gönderimi yapmak istediğinizde kullanacağınız bir servistir. Pipe servisi ile lokaldesi mailbox dosyalarına da mail bırakabilirsiniz. Uzaktaki hedef adrese mail bırakma işini UUCP protokolü ile yapacaksanız, Pipe servisini kullanmanız ideal çözümdür.

5 Linux'te Sistem Kurulumu Sırasında Asgari bilinmesi gerekenler

Bu bölümde, Mail Sunucusunu kurarken Linux dünyasında asgari bilmeniz gerekenleri açıklayacağız. Kurulum sırasında genellikle aşağıdaki işlemleri yapacağız:

- Yeni paket deposu tanımlamak
- Paket deposunu güncellemek
- Paket Deposundan paket yüklemek
- Tar.gz paketleri ile paket kurmak
  - Konfigüre etmek
  - Derlemek
  - Kurmak
- Kütüphane tanıtmak
- Sembolik link yaratmak
- Kullanıcı ve grup yaratmak
- Dosya haklarını ayarlamak
- Ayar dosyalarını değiştirmek, edit etmek
- Programları açılış aşamasında başlatmak

Paket ve Kurulum İşlemleri

Kurulum sistemi olarak Debian Etch kullanacağız. Debian dağıtımının kendine özgü paket(Program) yönetim sistemi vardır. Eğer bir paket yada program kurmak istiyorsanız, sisteminizde önceden tanımlı paket depolarından çekerek bu paket kurmalısınız. Eğer istediğiniz paket önceden ayarlı paket deposunda tanımlı değilse, ya bu paketi içinde barındıran bir paket deposu tanımlayacaksınız ya da bu paketi tar.gz dosyasından kuracaksınız.

İnternette çeşitli debian paket depoları mevcuttur. En çok kullanılanları Debian dağıtımının kendi siteleridir. Paket depoları bir web adresi olarak tanımlanırlar. Örneğin aşağıdaki gibi:

```
http://http.us.debian.org/debian
```

Yukarıdaki adresi /etc/apt/sources.list dosyasına uygun formatta eklerseniz ve ardından aşağıdaki komutu çalıştırırsanız, sisteminize yeni bir debian paket deposu tanıtmış olursunuz. Apt-get komutu yerine aptitude komutunu da kullanabilirsiniz. İkisi de aynı işlevi görmektedir.

```
#apt-get update
```

Birden fazla paket deposunu sisteme tanıtabilirsiniz. Bunu yapmak için paket deposu adreslerini sources.list dosyasına eklemeniz yeterlidir.

Bazen bir paket kurmak istersiniz ama paketin tam adını bilmezsiniz. Bunun için paket deposu içerisinde arama yapmanız gerekebilir. Örneğin firefox web gezgininin tam paket adını bilmiyorsunuz ve debian paket deposundaki tam adını öğrenmek istiyorsunuz. Bunu başarmak için aşağıdaki komutu kullanabilirsiniz. Bu komut ile içerisinde firefox kelimesi geçen paketleri konsolunuzda listeli bir şekilde göreceksiniz.

```
#aptitude search firefox
```

Eğer paket deposundan bir program kurmak istiyorsanız. Aşağıdaki komut ile kurulum yapabilirsiniz. Örneğin aşağıdaki komut gcc programını sisteminize kuracaktır.

```
#aptitude install gcc
```

Kurulum sırasında genellikle sadece kurmak istediğiniz paket değil beraberinde bağımlı paketleri de kurarsınız. Eğer aptitude bağımlı paketleri kurma aşamasına gelirse size kurup kurmayacağınız yönünde evet/hayır sorusu sorar. Evet derseniz paket bağımlılıkları ile beraber kurulur. Hayır derseniz paket ve bağımlılıkları kurulmaz.

Kaynak Koddan Paket Yüklemek

Bazen paket depolarından paket kurmayız. Onun yerine paketin kaynak kodunu internetten indirir ve kaynak koddan derleme yaparak kurulum yaparız. Eğer kurulumunu yapmak istediğiniz paket, GNU ailesine ait bir paket ise, genellikle bilinen bir yapıya sahiptir.

Örneğin her GNU paketinin bir README(Oku beni) ve AUTHORS dosyası vardır. Birinci dosya ilk okumanız gereken dosyadır. Bu dosya kurulumu nasıl yapacağınızı Ya da kurulum için hangi dokumana ulaşmanız gerektiğini açıklar. İkinci dosya paketi yazan kişileri ve iletişim bilgilerini barındırır.

GNU paketlerinin başka standartları da mevcuttur. Örneğin her GNU paketinin bir src/ dizini vardır. Kaynak kodlar bu dizin altında bulunmaktadır. Bunun dışında, paketin ana dizininde aşağıdaki 2 dosya bulunmaktadır:

- configure
- Makefile.in

“configure” programı, derleme için gerekli olan Makefile dosyasını üretir. Aşağıdaki şekilde çalıştırılır. (Paketin ana dizinine gittikten sonra)

```
#./configure
```

Bu dosyayı üretirken Makefile.in dosyasından yararlanır. “configure” programı paketin derlenebilmesi için gerekli tüm bağımlılıkları, kütüphaneleri, link parametrelerini toplar. “configure” programı sonlandıktan sonra, genellikle “make” komutu çalıştırılır. “make” komutu aşağıdaki gibi çalıştırılır.

```
#make
```

“make” komutu, Makefile dosyasından yararlanarak src/ dizini altındaki kaynak kodları tek tek derler ve paket çıktılarını üretir. “make” aşaması sonlandıktan sonra genellikle “make install” komutu çalıştırılır. Bu komut aşağıdaki gibi çalıştırılır.

```
#make install
```

“make install” komutu derlenen paket programlarını ve ilgili bileşenlerini sisteme yükler. Bu aşama tamamlandıktan sonra, paket sisteminize kurulmuş demektir.

Kütüphane Tanıtmak

Kaynak koddan kurulum yapmışsanız, kurulan paketin kütüphanelerini ayrıca sisteme tanıtmanız gerekmektedir. Bunu gerçekleştirebilmek için /etc/ld.so.conf dosyasına ilgili kütüphanenin tam yolunu tek satıra denk gelecek şekilde eklemeniz gerekmektedir.

Kütüphane yolunu dosyaya ekledikten sonra “ldconfig” komutunu çalıştırarak ilgili kütüphaneyi sisteme tanıtmış olursunuz.

Sembolik Link Yaratmak

Bazen kurulum tamamlandıktan sonra sembolik linkler yaratmak gerekebilir. Sembolik link bir dosyadır. Ama normal bir dosyadan farklı olarak bir içerik barındırmaz başka bir dosyaya işaret eder. Bu link dosyasına “hedefteki dosyaya sembolik link yapan dosya” denebilir. Aşağıdaki komut yardımı ile sembolik link yaratabiliriz.

```
#ln -s hedef_dosya sembolik_dosya
```

Yukarıdaki sembolik_dosya, hedef_dosyaya işaret edern bir sembolik linktir.

Kullanıcı ve Grup Yaratmak

Bazen paket kurulumları için yeni kullanıcılara ihtiyaç olabilir. İlgili paketin güvenlik kaygılarından dolayı yeni yaratılacak kullanıcı ve grup hakları ile çalışması gerekebilir. Bu durumda, kaynak koddan paket kurulumu yapmadan önce bu kullanıcı ve grupları yaratmak gerekebilir.

Kullanıcı yaratmak için “useradd” komutu kullanılır. Aşağıdaki komut “halil” kullanıcısını yaratmaktadır.

```
#>useradd halil
```

Bazen kullanıcı yaratırken, kullanıcıya özel bir kullanıcı numarası ve kullanıcı grubu atamamız gerekebilir. Aşağıdaki komut “halil” kullanıcısını yaratırken, 1001 numaralı kullanıcı numarasını halil kullanıcına atar ve kullanıcıyı wheel grubuna ekler.

```
#>useradd -g wheel -u 1001 halil
```

Grup yaratmak için “groupadd” komutu kullanılır. Aşağıdaki komut “muhasebe” grubunu yaratmaktadır.

```
#>groupadd muhasebe
```

Bazen grup yaratırken, gruba özel bir grup numarası atamamız gerekebilir. Aşağıdaki komut "muhasebe" grubunu yaratırken, 1001 numaralı grup numarasını muhasebe grubuna atar.

Dosya Haklarını Ayarlamak

Bazen kurulan paketlerin düzgün çalışabilmesi için ayar dosyalarında veya diğer dosyalarda izin hakları düzenlemesi yapılması gerekebilir. Bu düzenlemeler genellikle dosyanın okuma-yazma-çalıştırma haklarının değiştirilmesi şeklinde olur ya da dosyanın grup ve yaratıcı tanımlarının değiştirilmesi şeklinde olur.

Dosya izin haklarının değiştirilmesi "chmod" komutu ile gerçekleştirilir. Aşağıdaki ornek, a.txt dosyası için dosya yaratıcısına okuma-yazma-çalıştırma, dosya grubu için okuma-çalıştırma ve diğer gruplar için okuma hakkı verir. Daha fazla bilgi için chmod man sayfasına bakabilisiniz.

```
#>chmod  754 a.txt
```

Dosya yaratıcısı ve grubunu değiştirmek için "chown" komutu kullanılır. Aşağıdaki komut a.txt dosyasının yaratıcısını halil kullanıcısı grubunu ise wheel grubu yapar.

```
#>chown  halil:wheel a.txt
```

Daha fazla bilgi için chown man sayfasına bakabilirsiniz.

Dosyaları Edit Etmek

Dosyaları edit edebilmek için bir tekst editore ihtiyacınız vardır. Populer olan tekst editorlerden bazıları aşağıdaki gibidir.

1. vi
2. nano
3. pico

"nano" ve "pico" editorlerinin kullanımları kolaydır. Bu 2 programla bir dosya yaratmak veya var olan dosyayı edit etme işlemini aşağıdaki komut dizisi ile gerçekleştirebilirsiniz.

```
#>nano a.txt
```

```
#>pico a.txt
```

Eğer “a.txt” adında bir dosyanız yok ise bu dosya yaratılır. Eğer “a.txt” dosyası var ise a.txt dosyası edit edilmek için açılır. Daha fazla bilgi için nano ve pico man sayfalarına bakabilirsiniz.

“Vi” editorü de pico ve nano ile aynı işlevi görür. Ama bu editör pico ve nano'ya nazaran çok güçlü ve işlevseldir. Genellikle “vi” editörünü kullanmanızı tavsiye ederim. Yararını bolca göreceğinizi umuyorum. “vi” editörü, geniş içeriğinden dolayı bu kitapta anlatılmayacaktır. Kapsamlı bilgi için http://www.vim.org sitesini ziyaret edebilisiniz.

Servisleri Açılışta Otomatik Olarak Başlatmak

Kurulum aşamasında sisteme bir çok servis yükleyeceğiz. Çoğunlukla da, İşletim Sistemi açılırken bu servislerin otomatik olarak açılmasını isteyeceğiz. Bunu gerçekleştirebilmek için Linux işletim sisteminin servis yönetim sistemi hakkında bilgi sahibi olmamız gerekmektedir.

Linux'te herhangi bir programı açılış aşamasında başlatabiliriz. Bunun için izleyeceğimiz 2 yol vardır:

1. Programı bir servis haline getirmek ve sysV (Servis Yönetim Sistemine) sistemine yü klemek
2. /etc/rc.local dosyasına servis çalıştırma komutunu eklemek

Birinci adım doğru adım ama gerçekleştirme aşamaları uzun olan bir konudur ve kitabımızın konusu değildir. Bu konuyu öğrenmek için Linux dokumanlarına başvurabilirsiniz.

İkinci adım, servisleri otomatik olarak başlatmamızı sağlayan kolay bir aşamadır ve şimdilik işimizi görebilecek durumdadır. Bu sebeple servisleri açılış aşamasında başlatmak için ikinci yöntemi kullanacağız.

İkinci yöntemi gerçekleştirmek çok kolaydır. Tek yapmanız gereken servis çalıştırma komutunu /etc/rc.local dosyasına eklemektir. Linux işletim sistemi tüm servisleri çalıştırdıktan sonra son olarak /etc/rc.local dosyasını çalıştırır. Bu dosyada yazan komutlar bash kabuk programı tarafından okunur ve çalıştırılır. Böylelikle bu dosyada yer alan komutlar da her açılış aşamasında çalıştırılmış olur.

6 Kurulum İçin Gereksinimler

Yazılım Gereksinimleri

Aşağıdaki yazılımların kaynak kodlarının son halleri.

1. Bind
2. Openldap
3. Cyrus Sasl
4. Postfix
5. Courier Authlib
6. Courier Imap
7. Courier POP3
8. ClamAv
9. SpamAssassin
10. Maildrop
11. Gnarwl

İşletim Sistemi olarak;

- Debian Etch

Donanım Gereksinimleri

- Minimum PENTIUM III 450 MHZ
- Minimum 512MB RAM
- 8 GB Harddisk

7 Kuruluma Giriş

Bu doküman tam donanımlı bir Mail sunucusunun nasıl kurulacağını anlatmaktadır. Bu dokümanın sonunda aşağıdaki özelliklere sahip bir Mail sunucusuna sahip olacaksınız.

1. Sanal alan ve sanal kullanıcı desteği
2. Kullanıcı, domain ve alias bilgileri LDAP'ta saklanan bir Mail Sunucu
3. Maildir formatında saklanan mailler
4. SMTP Auth destekli Mail sunucusu
5. Anti virüs ve Anti Spam korumalı mailler
6. Ototmatik tatil mesajı desteği(Vacation, autoreply)
7. Kullanıcılar için Spam dizini

Hedef aldığımız sistemi kurarken sırası ile aşağıdaki paketleri kuracağız.

12. Bind
13. Openldap
14. Cyrus Sasl
15. Postfix
16. Courier Authlib
17. Courier Imap
18. Courier POP3
19. ClamAv
20. SpamAssassin
21. Maildrop
22. Gnarwl

Sanal kullanıcı ve Sanal alan destekli bir Mail sunucusuna sahip olmak için yukarıdaki paketlerden ilk 4'ünün kurulumu yeterli olacaktır. Ama böyle bir sistem elbetteki yeterli olmayacaktır.

Bir mail sunucusu kurmuşsanız bir de beraberinde IMAP ve POP3 sistemi ve mail güvenliği amaçlı Anti virüs ve Anti Spam sistemi kurmanız gerekmektedir. İşte bu özellikler için de yukarıdan 5-9 arasındaki paketleri kurmanız gerekmektedir.

Tabi bu haliyle de sistem sizin için yeterli olmayabilir. Fazladan, kullanıcılar için Spam dizin ve kullanıcıların tatil zamanlarında otomatik cevap verecek bir sisteme ihtiyaç duyacaksınız. Bunun için de 9 ve 10 numaralı paketleri kurup sisteme adapte etmeniz gerekmektedir.

Anlaşılacağı üzere Mail sunucumuzun **Ana kısmını,** Bind+OpenLDAP+Cyrus SASL+Postfix oluşturmaktadır. Bu ana kısım kurulduktan sonra sonradan kurulacak paketler ve tercihleriniz sizin beğeninize kalmıştır. Courier yerine dovecot, spamassassin ve clamav yerine başka sistemler kurabilirsiniz. Yine de, sistemimizin ana kısmı ile bunlar çalışacaktır. Ama bu çalışmalar, bu dokümanımızın içeriğini oluşturmamaktadır. Mail sunucumuzu kurarken yukarıdaki 5. paketten itibaren listelenen paketler bu doküman yazarı tarafından tercih edilen paketlerdir.

Şimdi isterseniz kurulum işlemine başlayalım. İlk olarak ana kısmı oluştururan ilk 4 paketi kuracağız.
Ama bu 4 paketi kurmadan önce Debian Etch Sistemimizi güncel hale getirelim.

8 Sistemin Güncel Hale Getirilmesi

Bunun için lütfen aşağıdaki komutu root kullanıcısı altında çalıştırın:

```
#> apt-get upgrade
```

Bu işlem sonuçlandıktan sonra, aşağıdaki sources.list adreslerini /etc/apt/sources.list dosyasına ekleyin. Tekrar eden web adresi olmadığına ve “deb cdrom” ile başlayan satırın diyezli olduğuna dikkat edin.

```
#
# deb cdrom:[Debian GNU/Linux 4.0 r4a-etchnhalf _Etch-and-a-half_ - Official i386 NETINST
20080804-22:00]/ etch main

#deb cdrom:[Debian GNU/Linux 4.0 r4a-etchnhalf _Etch-and-a-half_ - Official i386 NETINST
20080804-22:00]/ etch main

deb http://http.us.debian.org/debian stable main contrib non-free

deb http://security.debian.org/ etch/updates main
deb-src http://security.debian.org/ etch/updates main
```

Bu aşamadan sonra

```
#> apt-get update
```

Komutu ile sources.list dosyasında tanımlı depoları sisteminize tanıtın.

Şimdi sıra kurulum yapılacak makineye “Fully Qualified Name” vermeye geldi. Bunun için /etc/hosts dosyasında 127.0.0.1 IP'sini aşağıdaki gibi değiştirin.

```
127.0.0.1       mailserver.interis.com  mailserver localhost.localdomain localhost
```

Not:

Gerçekte mailserver.interis.com adresi internet'te tanınan bir adres değildir. Sadece LAN alanı için bu tanımlamayı yapıyoruz. Şu an hizmetter olan bir interis.com adresi mevcuttur. Bu adresin örneğimizle bir ilişkisi bulunmamaktadır.

Bu aşamadan sonra kurulum işine başlayabiliriz.

8.1 Kurulum Ortamının hazırlanması

Kurulum aşamasında bazı paketleri tar.gz (tarball) paketlerinden kuracağız. Bunun için gcc ve make paketlerine ihtiyacımız var. Bunun yanında, paketleri kurarken varsayılan yerlerine değil, belirlediğimiz bir dizin altına kuracağız. Örneğin .tar.gz paketlerini "/usr/local/mailserver_makedir" altında açacağız, kurulum bittiğinde ise paketler "/usr/local/mailserver" dizininde kurulmuş olacak.

Bunları gerçekleştirmek için aşağıdaki komutları çalıştırın.

```
#>mkdir /usr/local/mailserver
#>mkdir /usr/local/mailserver_makedir
```

Kaynak koddan kurulum yaparken, yani tarball paketlerinde kurarken, gcc ve make komutlarına ihtiyacımız olacak. Sisteminize gcc ve make paketlerini kurmak için aşağıdaki komutu çalıştırın.

```
#>aptitude -y install gcc make
```

Bu aşamadan sonra paketlerin kurulumuna başayabiliriz.

Kuruluma başalmadan önce, elinizdeki sistemin bir IP'si olduğunu ve internete erişimi olduğunu varsayıyoruz. Eğer bu durumda değilse, lütfen sisteminizin ethernet ve network ayarlarını yapınız ve daha sonra devam ediniz.

9 Bind Kurulumu

Bind paketinin kurulumu çok kolaydır. Aşağıdaki komut ile Bind9'u kurunuz. Parametre olarak verilen “-y” opsiyonu aptitude programı tarafından sorulan sorulara otomatik olarak evet cevabını vermek içindir.

```
#> aptitude -y install bind9
```

Kurulum bittikten sonra, aptitude programı bind9'u otomatik olarak başlatacaktır. Eğer bind9'un başlayıp başamadığını kontrol etmek istiyorsanız, komut satırında “netstat -ltnp” yazıp enter'a basın. Aşağıdaki gibi bir çıktı ile karşılaşıyor iseniz bind9 kurulu ve çalışıyor demektir.

```
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN      1905/named
```

Şimdi sıra geldi bir alan adı için DNS ayarı yapmaya.

Bind9 kurulunca, /etc/bind dizininin oluşması gerekir. /etc/bind/named.conf.local dosyasını edit edin ve aşağıdaki içeriği bu dosyaya ekleyin.

```
zone "dom1.test" IN {
  type master;
  file "/etc/bind/zones/dom1.test.zone";
};
```

Yukarıdaki tanım, dom1.test alan adının kurduğumuz DNS sunucusu yani bind9 tarafından servis edileceğ,ini göstermektedir. Bu tanım, eğer kurduğumuz DNS sunucusuna dom1.test ve alt alan adları için sorgu gelmişse, dns sunucusunun /etc/bind/zones/dom1.test.zone dosyasına bakmasını söyler. Eğer bahsi geçen dosya doğru bir şekilde tanımlanmış ise DNS sunucumuz dom1.test alan adı için düzgün cevap verir. Şu aşamada böyle bir dosyamız yok, dolayısı ile DNS sunucumuz şu aşamada dom1.test için tam bir şekilde ayarlanmış değil. Bunu başarmak için, öncelike /etc/bind/zones dizinini yaratmalı ve bu dizin altında dom1.test.zoen dosyasını yaratmalıyız.

Bunun için aşağıdaki komut ile zones dizininin yaratın.

```
#> mkdir  /etc/bind/zones
```

Bu aşamadan sonra, sevdiğiniz bir editör ile /etc/bind/zones/dom1.test.zone dosyasını yaratın ve aşağıdaki içeriği bu dosyaya ekleyin.

```
; BIND db file for dom1.test
```

```
$TTL 86400

@       IN      SOA     mailserver.interis.com.      root.dom1.test. (
                        1       ; serial number
                        28800           ; Refresh
                        7200            ; Retry
                        864000          ; Expire
                        86400           ; Min TTL
                        )

              NS      mailserver.interis.com.
              MX      10 mailserver.interis.com.

dom1.test.      IN      A       10.0.5.141

$ORIGIN dom1.test.
```

Yukarıdaki zone kaydı dom1.test için

1. Nameserver'ının mailserver.interis.com olduğunu
2. seri numarasının 1 olduğunu
3. A kaydının mailserver.interis.com sunucusunda tutulduğunu
4. MX kaydının mailserver.interis.com olduğunu

göstermektedir. Burada bahsi geçen “mailserver.interis.com” sunucusu şu an üzerinde kurulum yaptığımız sunucudur.

Yapmamız gereken son bir şey daha var, o da makinemize DNS sunucumuzun kendisi olduğunu yani 127.0.0.1 olduğunu söylemek. Bunun için /etc/resolv.conf dosyasını aşadaki gibi değiştirin.

```
search interis.com
nameserver 127.0.0.1
```

Bu aşamaları geçtikten sonra yapmanız gereken bind9 sunucusunu tekrar başlatmaktır.

Aşağıdaki komut ile Bind9'u yeniden başlatın.

```
#>/etc/init.d/bind9 restart
```

Ardından, Bind9'un düzgün başladığını kontrol etmek için /var/log/syslog dosyasının son satrılarına bakınız. Eğer aşağıdaki gibi bir kayıt görüyorsanız, bind9 düzgün ayarlanmış ve ayakta demektir.

```
Sep 21 13:05:29 mailserver named[2800]: zone dom1.test/IN: loaded serial 1
```

Ama tabii ki bununla yetinmememiz lazım. DNS sunucumuzun dom1.test için DNS sorgularına düzgün cevap vermesi lazım.

Eğer dom1.test için tüm ayarlamaları düzgün yapmış ised komut satırında verdiğimiz “dig dom1.test” komutunun aşağıdakine benzer bir çıktısı olmak durumunda.

```
; <<>> DiG 9.3.4-P1.1 <<>> dom1.test
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12448
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;dom1.test.                     IN      A

;; ANSWER SECTION:
dom1.test.              86400   IN      A       10.0.5.141

;; AUTHORITY SECTION:
dom1.test.              86400   IN      NS      mailserver.interis.com.

;; Query time: 6 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Sep 21 13:09:36 2008
;; MSG SIZE  rcvd: 79
```

Burada dikkat etmemiz gereken “ANSWER:1” çıktısının görülmesidir. Eğer bunu görüyorsanız DNS sunucunuz dom1.test için bir cevap dönüyor demektir, ve cevap dönüyorsa büyük bir ihtimalle düzgün cevap dönüyordur. 2. dikkat etmemiz gereken nokta A kaydının düzgün olması ve ping edilebilir olmasıdır.

Bunun için aşağıdaki komut ile dom1.test alan adına ping atın. Eğer cecap dönüyor ise dom1.test alan adı için A kaydını düzgün ayarlamışsınız demektir.

```
#>ping dom1.test
PING dom1.test (10.0.5.141) 56(84) bytes of data.
64 bytes from 10.0.5.141: icmp_seq=1 ttl=64 time=1.98 ms
64 bytes from 10.0.5.141: icmp_seq=2 ttl=64 time=0.075 ms
64 bytes from 10.0.5.141: icmp_seq=3 ttl=64 time=0.036 ms
```

Hala testlerden şüphemiz var ise son olarak “host dom1.test” komutunun çıktısına bakabiliriz. Bu komutun çıktısı aşağıdaki gibi olmalıdır.

```
dom1.test has address 10.0.5.141
dom1.test mail is handled by 10 mailserver.interis.com.
```

“mail is handled” kısmına özellikle dikkat edin, yazan IP adresi dom1.test için yetkili Mail sunucusunu işaret etmelidir, yani makinemizin kendi IP'sini.

Eğer buraya kadar yazdıklarım ve yaptığınız test sonuçları bu dokümanda yazılan gibi ise büyük bir ihtimal DNS sunucunuz çalışıyor ve dom1.test tanımlı durumda demektir.

Bir sonraki aşamaya geçebilirsiniz.

10 OpenLDAP Kurulumu

OpenLDAP kurulumunu apt-get veya aptitude komutu ile kurulum yerine Tar.gz paketlerinden kuracağız. Bunun sebebi, yaptığım denemelerde depolardan gelen cyrus-sasl, postfix ve openldap paketlerinin birbiri ile tam entegre olmamış olması. Belki bazı versiyonlar birbiri ile düzgün çalışıyodur. Ama ben o versiyonlara rastlamadım. Bunun için bu paketlerin kurulumunu tarball paketlerinden yapacağız. Ayrıca tarball'dan kurulumların bir avantajı var, o da depo paketlerinden bağımsız hale gelmeniz ve Debian'ın ileriki sürümlerinde de kurulum aşamasında bu dokümanın hala geçerlilik kazanması.

Openldap, kurulum için bazı paketlere ihtiyaç duymaktadır, aslında aynı paketler cyrus-sasl, postfix paketleri için öncelik durumunda. Dolayısı ile paket kurulumuna geçmeden önce bu paketleri depodan kuralım.

Aşağıdaki komut ile bağımlı paketleri kurabilirsiniz.

```
#>aptitude -y install libdb4.4 libdb4.4++   libdb4.4++-dev   libdb4.4-dev mysql-server-5.0 mysql-client-5.0  libmysql++-dev  libmysqlclient-dev libmysqlclient15-dev  iodbc libiodbc2 libiodbc2-dev libmyodbc libodbc++-dev libodbc++4   unixodbc unixodbc-dev libsasl2 libsasl2-2 libsasl2-dev libssl-dev libssl0.9.8
```

Şimdi OpenLDAP'ı kurabiliriz. Önce aşağıdaki komut ile openldap'ı bulunduğunuz dizine indiriniz.

```
#>wget ftp://ftp.openldap.org/pub/OpenLDAP/openldap-stable/openldap-stable-20071118.tgz
```

Ardından, aşağıdaki komut ile openldap'ı “/usr/local/mailserver_makedir” dizinine açınız.

```
#>tar xvzf openldap-stable-20071118.tgz -C /usr/local/mailserver_makedir/
```

Bu aşamadan sonra aşağıdaki gibi /usr/local/mailserver_makedir/openldap-2.3.39/ dizinine geçip kurulum komutlarını çalıştırınız.

```
#>cd /usr/local/mailserver_makedir/openldap-2.3.39/
#>./configure --prefix=/usr/local/mailserver/openldap --enable-crypt  --enable-backends --enable-bdb --enable-ldbm --enable-perl=no
#>make depend
#>make
#>make install
#>echo /usr/local/mailserver/openldap/lib>>/etc/ld.so.conf
#>ldconfig
```

Yukarıdaki komutları çalıştırmış ve herhangi bir hata almamış iseniz, configure betiğinin prefix parametresinde belirtildiği gibi /usr/local/mailserver/openldap dizini altında openldap'ın kurulu olması gerekir. Eğer öyle ise başardınız demektir, eğer öyle değilse, bu dokümanı baştan bir daha inceleyin, ya da karşılaştığınız durumu bana mailleyin. Ama bilinen bir durum var, o da hata durumunda bu aşamadan sonra ilerleyemeyeceğiniz.

Eğer herşey yolunda ise bir sonraki aşamaya geçiniz.

11 Jamm Kurulumu

Kuracağımız Mail sunucusu, sanal alanlar ve sanal kullanıcıları LDAP sisteminde saklayacak. LDAP'ta saklayabilmek için openldap'ın bu alanları saklayacak şekilde ayarlanması gerekmekte. LDAP bir veriyi nasıl saklayacağına şemalar(schema) aracılığı ile karar vermekte. Jamm projesi mail kullanıcısı ve alan adlarının saklanma şekli için bir LDAP şeması hazırlamış durumda. Biz de mail sunucumuz için jamm şemasını kullanacağız.

Bunu başarabilmek için Jamm şemasını openldap'a tanıtmamız gerekmekte. Böylelikle, openldap'ın sanal alan ve sanal mail kullanıcılarını nasıl saklayacağına karar vermiş oluyoruz.

Bunun için jamm paketi ile gelen jamm.schema dosyasını openldap'a tanıtacağız.

Bunları başarmak için öncelikle jamm paketini aşağıdaki gibi dizininize indiriniz.

```
#>wget http://prdownloads.sourceforge.net/jamm/jamm-0.9.6-bin.tar.gz
```

Ardından, aşağıdaki komut ile /usr/local/mailserver_makedir dizinine açınız.

```
#>tar xvzf jamm-0.9.6-bin.tar.gz -C /usr/local/mailserver_makedir
```

/usr/local/mailserver_makedir/jamm-0.9.6/ dizinine geçtiğimizde jamm.schema adlı bir dosya ile karşılaşacağız. Bu dosya bizim LDAP şemamız oluyor.

Aşağıdaki komut ile jamm.schema'yı /usr/local/mailserver/openldap/etc/openldap/schema dizinine kopyalayınız.

```
#>cp  /usr/local/mailserver_makedir/jamm-0.9.6/jamm.schema /usr/local/mailserver/openldap/etc/openldap/schema/
```

Bu aşamadan sonra yapmanız gereken openldap'a jamm.schema dosyasını tanıtmak.

/usr/local/mailserver/openldap/etc/openldap/slapd.conf dosyasını edit edip, aşağıdaki içeriği dosyanın en başına ekleyin. OpenLDAP kurulumu ardından sizin için ayarlı bir slapd.conf dosyası hali hazırda olması lazım. Bu dosyada "include" edilmiş bazı şema dosyaları olabilir. Bu şemaların aşağıda eklenen şemalarla çakışmamasına özen gösteriniz. Aynı şema dosyasından 2 tane olmamalı.

```
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
#
include /usr/local/mailserver/openldap/etc/openldap/schema/core.schema
include /usr/local/mailserver/openldap/etc/openldap/schema/cosine.schema
include /usr/local/mailserver/openldap/etc/openldap/schema/inetorgperson.schema
include /usr/local/mailserver/openldap/etc/openldap/schema/nis.schema
include /usr/local/mailserver/openldap/etc/openldap/schema/misc.schema
include /usr/local/mailserver/openldap/etc/openldap/schema/jamm.schema
include /usr/local/mailserver/openldap/etc/openldap/schema/java.schema
```

Yukarıdaki işlemleri yaptınız ama bu haliyle openldap konfigürasyonu bitmiş sayılmaz.

Aşağıda bahsi geçen ayarların da yapılması gerekmektedir. OpenLDAP konfigürasyonu hakkında daha fazla bilgi edinmek istiyorsanız, openldap dokümanlarını okumalısınız.

Openldap'ın Cyrus-sasl ile parola saklama metodlarının aynı olması gerekmektedir. Aksi takdirde, Cyrus-sasl OpenLDAP yardımı ile yapacağı kimliklendirme işlemlerinde başarısız olur. Cyrus-Sasl, parolaları Unix-crypt metodu ile saklar, dolayısı ile OpenLDAP'ın da aynı şekilde saklaması gerekir. Aksi taktirde OpenLDAP'ta saklanan kullanıcı parolaları düzgün bir şekilde okunamaz. Okunsa bile okunan parola orijinal parola olmaz.

Bunun için slapd.conf dosyasına aşağıdaki satırı ekleyin.

```
# Password Hash
password-hash {CRYPT}
```

Kurmuş olduğumuz OpenLDAP versiyonu varsayılan olarak LDAP v3 protokolü ile konuşabilir durumdadır. Eğer LDAP v2 protokolü ile de konuşabilir olmasını istiyorsanız aşağıdaki satırı slapd.conf dosyasına ekleyiniz.

```
# Allow LDAPv2 client connections. This is NOT the default.
allow bind_v2
```

OpenLDAP schema verilerini bir veri tabanında saklar. Veri tabanı yapısı için çeşitli formatlar mevcuttur. Bunlardan 2 tanesi aşağıdadır.

1. BDB
2. LDBM

Bu doküman için BDB veri tabanı yapısını uygun bulduk.

BDB veri tabanı yapısı slapd.conf dosyasında varsayılan olarak seçilmiş olması gerekmekte. Eğer slapd.conf dosyasında aşağıdaki satır mevcut ise değişiklik yapmadan devam edin, ama eğer aşağıdaki gibi "database" direktifi ile başlayan satırda bdb yerine başka bir veri tabanı yapısı tanımlı ise onu "bdb" ile değiştirin.

```
database bdb
```

Veri tabanı yapısını tanımladıktan sonra dikte edeceğimiz her komut bu veri tabanı için geçerli olacaktır. Ta ki yeni bir "database" direktifi ile başlayan bir satıra kadar.

"database bdb" satırından hemen sonra veritabanı yöneticisini tanımlayacağız.
Yönetici kullanıcısı bir LDAP ayırtedici tümcesi olmak zorunda.(Daha fazla bilgi için LDAP dokümanlarını okuyunuz.)

Aşağıdaki içeriği "database bdb" satırının hemen sonrasına ekleyiniz. Aşağıda yönetici olarak "cn=Manager,dc=myhosting,dc=example" kullanıcısını tanımladık ve bir parola atadık.

```
suffix "dc=myhosting,dc=example"
rootdn "cn=Manager,dc=myhosting,dc=example"
rootpw {SSHA}aMtDhFtA6stmoZ4h9SsBzUqjH+wyLkHo
```

Yalnız yukarıdaki "rootpw" direktifi yanındaki karakterler dizisini değiştirmemiz gerekmekte. Çünkü "rootpw" direktifi ile tanımlanan tümce bir parolanın "hash"'lenmiş halidir. Yani bir şekilde parolanın şifreli halidir. Dolayısı ile parolanın orjinal halini bilmiyoruz. Bu şifreli tümceyi, bizim bildiğimiz bir parolanın şifreli hali ile değiştirmemiz gerekmekte. OpenLDAP Paketi ile gelen "slappasswd" adında bir komut mevcuttur. Bu komut aracılığı ile bir parola tümcesinin SSHA algoritması ile şifrelenmiş halini üretebiliriz.

Parolanın SSHA algoritması ile şifrelenmiş halini üretmek için aşağıdaki komutu çalıştırın.

```
#>/usr/local/mailserver/openldap/sbin/slappasswd
```

Yukarıdaki komut, çalışma esnasında sizden bir parola girmenizi isteyecektir. Parolanızı 2 kere girdikten sonra, girdiğiniz parolanın şifrelenmiş halini ekrana basacaktır. Bu ekrana basılan karakterler dizisini aşağıda olduğu gibi "rootpw" direktifinin yanına yapıştırın. Böylelikle yönetici kullanıcınız için parola tanımlamış olursunuz.

Bir sonraki aşama olarak, veri tabanımız için indeks tanımlayacağız. İndex tanımlama LDAP sunucu performansını arttıran bir özelliktir. LDAP indeksleri normal veritabanı indeksleri ile aynı işlevselliğe sahiptir. Slapd.conf dosyasında indekslerin tanımlandığı yeri bulup indeks tanımlarını

aşağıdaki gibi değiştiriniz.

```
# Indices to maintain for this database
index objectClass eq,pres
index ou,cn,mail,surname,givenname eq,pres,sub
index uidNumber,gidNumber,loginShell eq,pres
index uid,memberUid eq,pres,sub
index nisMapName,nisMapEntry eq,pres,sub
```

Şimdi sıra LDAP sunucumuza giriş yetkilerini tanımlamada.

Aşağıda listelenen tanımları tam anlamı ile anlamanıza gerek yoktur. Çünkü hazırlanan her şema için bu tanımlar değişmektedir. Jamm.schema şemasını kullanıyorsanız aşağıdaki gibi giriş hakkı tanımlamaları yapmanız gerekmektedir. Daha fazla bilgi için ***http://jamm.sf.net*** sitesine uğrayınız.

```
# Access Control Statements (ACL)
access to dn.regex=".*,jvd=([^,]+),o=hosting,dc=myhosting,dc=example"
attrs=userPassword
by self write
by
group/jammPostmaster/roleOccupant.expand="cn=postmaster,jvd=$1,o=hosting,dc=myhosting,dc
=example" write
by anonymous auth
by * none
access to dn.regex=".*jvd=([^,]+),o=hosting,dc=myhosting,dc=example"
by self write
by
group/jammPostmaster/roleOccupant.expand="cn=postmaster,jvd=$1,o=hosting,dc=myhosting,dc
=example" write
by * read
access to *
by * read
```

Şu hali ile slapd.conf mevcut isteklerimiz için ayarlanmış durumda. Yapmanız gereken slapd sunucusunu başlatmaktır. Aşağıdaki komut ile slapd'yi başlatabilirsiniz.

```
#>/usr/local/mailserver/openldap/libexec/slapd
```

OpenLDAP sunucusunu durdurmak için aşağıdaki komutu çalıştırın.

```
#>killall slapd
```

OpenLDAP sunucusunu yeniden başlatmak için bir komut mevcut değildir. Dolayısı ile yukarıdaki komutlar yardımı ile önce durdurup sonra başlatmanız gerekmektedir.

OpenLDAP'ın çalışıp çalışmadığını görmek istiyorsanız, komut satırında “netstat -ltnp” yazın. Eğer aşağıdaki gibi bir çıktı ile karşılaşıyorsanız, OpenLDAP çalışıyor demektir.

```
tcp        0      0 0.0.0.0:389             0.0.0.0:*               LISTEN      2275/slapd
```

Makinenizin her açılışında OpenLDAP sunucusunun otomatik olarak çalışmasını istiyorsanız, yukarıdaki komutu /etc/rc.local dosyasına ekleyiniz. Ekleme yaparken, yukarıdaki komutu "exit 0" satırından önce eklemeye özen gösteriniz. Aksi taktirde "exit 0" komutu yukarıdaki komuttan önce çalışacağı ve programı sonlandıracağı için OpenLDAP sunucusu açılışta başlatılmaz.

Aşağıda slapd.conf dosyasının tam içeriği yer almaktadır.

```
#
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
#
include /etc/openldap/schema/core.schema
include /etc/openldap/schema/cosine.schema
include /etc/openldap/schema/inetorgperson.schema
include /etc/openldap/schema/nis.schema
include /etc/openldap/schema/misc.schema
include /etc/openldap/schema/jamm.schema
include /etc/openldap/schema/java.schema
# Password Hash
password-hash {CRYPT}
# Allow LDAPv2 client connections. This is NOT the default.
allow bind_v2
# Do not enable referrals until AFTER you have a working directory
# service AND an understanding of referrals.
pidfile
/var/run/openldap/slapd.pid
argsfile
/var/run/openldap/slapd.args
#######################################################################
# ldbm and/or bdb database definitions
#######################################################################
database
ldbm
suffix
"dc=myhosting,dc=example"
rootdn
"cn=Manager,dc=myhosting,dc=example"
rootpw
{SSHA}aMtDhFtA6stmoZ4h9SsBzUqjH+wyLkHo
# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd and slap tools.
# Mode 700 recommended.
directory /var/lib/ldap

# Indices to maintain for this database
index objectClass
eq,pres
index ou,cn,mail,surname,givenname eq,pres,sub
index uidNumber,gidNumber,loginShell eq,pres
index uid,memberUid
eq,pres,sub
index nisMapName,nisMapEntry
eq,pres,sub
# Access Control Statements (ACL)
access to dn.regex=".*,jvd=([^,]+),o=hosting,dc=myhosting,dc=example"
attrs=userPassword
by self write
by
group/jammPostmaster/roleOccupant.expand="cn=postmaster,jvd=$1,o=hosting,dc=myhosting,dc
```

```
=example" write
by anonymous auth
by * none
access to dn.regex=".*jvd=([^,]+),o=hosting,dc=myhosting,dc=example"
by self write
by
group/jammPostmaster/roleOccupant.expand="cn=postmaster,jvd=$1,o=hosting,dc=myhosting,dc
=example" write
by * read
access to *
by * read
```

Yalnız, henüz openldap'la işimiz bitmiş değil. Openldap'ı ayarladık ama şema altyapısını hazırlamadık.

Bunları gerçekleştirmek için base.ldif adında bir dosya yaratınız ve aşağıdaki içeriği bu dosyaya yapıştırınız.

```
dn: dc=myhosting, dc=example
objectClass: top
objectClass: domain
domainComponent: myhosting

dn: cn=Manager, dc=myhosting, dc=example
objectClass: top
objectClass: organizationalRole
cn: Manager

dn: o=hosting, dc=myhosting, dc=example
objectClass: top
objectClass: organization
o: hosting
```

Ardından aşağıdaki komutu çalıştırarak yukarıdaki veriyi openldap veri tabanına kaydediniz. Aşağıdaki komutu çalıştırmadan önce OpenLDAP sunucusunun çalışıyor olduğundan emin olunuz.

```
#>/usr/local/mailserver/openldap/bin/ldapadd -x -D "cn=Manager,dc=myhosting,dc=example" -W
-f base.ldif
```

Bu komutu çalıştırırken size "cn=Manager,dc=myhosting,dc=example" kullanıcısının parolası sorulacaktır. slapd.conf dosyasında rootpw parameteresi ile verdiğiniz parolanın şifrelenmemiş halini yazıp enter'a basınız. Eğer bir hata ile karşılaşmadı iseniz,OpenLDAP kurulumu bitmiş durumda.

Eğer buraya kadar bir hata ile karşılaşmadıysanız bir sonraki adıma geçebilirsiniz.

12 Jamm Arayüzünün Kurulumu

jamm şemasının OpenLDAP'a tanıtılması bize sanal alan adlarını ve sanal mail kullanıcılarının OpenLDAP'ta saklanabilmesine olanak sağladı. Ama henüz bir alan adı ve bu alan adına ait bir mail kullanıcısını OpenLDAP veritabanına eklemiş değiliz. Bu iş için Jamm geliştiricileri Struts kütüphanesi ile yazılmış bir web uygulaması geliştirmiş durumdalar. Biz de bu web uygulaması ile alan adları ve kullanıcılarımızı sisteme yükleyeceğiz.

Jamm Web uygulaması Apache-Tomcat sunucusunda çalışmakta, bu sebeple tomcat ve dolayısı ile java SDK'yı sistemimize kurmamız gerekmekte.

İşe Java SDK kurmak ile başlayalım.

Java SDK 1.6(diğer adı ile Java 6) paketini http://java.sun.com adresinden indirin. İndirirken binary kurulum yapan paketi özellikle seçin. İndirdiğiniz paketin sonu .bin ile bitmesi gerekmekte.

Ardından aşağıdaki komut ile ".bin" dosyasını çalıştırın.

```
#>sh <java paketi>.bin
```

Yukarıdaki komut sizden kabul etmenizi istediği sözleşmeyi gösterecektir. Sözleşmeyi "yes" yazarak kabul edin. Java SDK bulunduğunuz dizinde bir dizin altında kurulmuş olmalı. Yapmanız gereken, bu dizini /usr dizini altında jdk16 olarak isim değiştirerek taşımanız. Aşağıdaki komut ile bunu gerçekleştirebilirsiniz.

```
#>mv <java dizini> /usr/jdk16
```

Java SDK'nın ana dizininin /usr/jdk16 olmasına özen gösteriniz.

Bir sonraki aşama, kurulu java SDK'yı root kullanıcısına tanıtmak.

/root/.bashrc dosyasını edit edin (Kabuk programı olarak bash kullandığınızı varsayıyoruz.) ve aşağıdaki içeriği .bashrc dosyasına ekleyip kaydedin.

```
export JAVA_HOME=/usr/jdk16
export PATH=$JAVA_HOME/bin:$PATH:.
```

Ardından aşağıdaki komut aracılığı ile .bashrc ayarlarını etkinleştirin.

```
#>source /root/.bashrc
```

Şu an itibari ile Java SDK sisteminizde kurulmuş durumda.

Şimdi sıra Tomcat kurulumunda. Tomcat kurulumu da Java kurulumu kadar basit.

Aşağıdaki komut ile Tomcat'i bulunduğunuz dizine indirin.

```
#>wget http://ftp.itu.edu.tr/Mirror/Apache/tomcat/tomcat-6/v6.0.18/bin/apache-tomcat-6.0.18.tar.gz
```

Ardından aşağıdaki komutlar aracılığı ile tomcat paketini /usr/local dizinine açın ve ismini tomcat60 olaral değiştirin.

```
#>tar  xvzf apache-tomcat-6.0.18.tar.gz -C /usr/local
#>mv /usr/local/apache-tomcat-6.0.18 /usr/local/tomcat60
```

Son olarak /root/.bashrc dosyasında tomcat ana dizinini aşağıdaki satırı ekleyeret tanıtın.

```
export CATALINA_HOME=/usr/local/tomcat60
```

Ve aşağıdaki komut ile tomcat ayarlarını etkinleştirin.

```
#>source /root/.bashrc
```

12.1 Jamm Web uygulamasını yüklemek

Tomcat kurulumu tamamlandı. Şimdi sıra Jamm web uygulamasının Tomcat'e yüklenmesinde.

Aşağıdaki komut ile /usr/local/tomcat60/webapps dizini altında jamm dizini oluşturun.

```
#>mkdir /usr/local/tomcat60/webapps/jamm
```

Daha önceden jamm paketini /usr/local/jamm-0.9.6 dizinine açmıştık. Bu dizin altında bulunan jamm-0.9.6.war dosyasını /usr/local/tomcat60/webapps/jamm dizinine aşağıdaki komut ile kopyalayın.

```
#>cp  /usr/local/mailserver_makedir/jamm-0.9.6/jamm-0.9.6.war
/usr/local/tomcat60/webapps/jamm
```

Şimdi /usr/local/tomcat60/webapps/jamm dizini altına geçip aşağıdaki komut ile war paketini açın.

```
#>jar -xf jamm-0.9.6.war
```

Yukarıdaki komuttan sonra bulunduğunuz dizinde bir dizi dosya ve dizin oluşması gerekmekte. Burada önemli olan WEB-INF dizini altındaki jamm.properties.dist dosyası. Aşağıdaki Komut ile bu dosyanın bir kopyasını oluşturun. Böylelikle jamm ayar dosyasını oluşturmuş olacağız.

```
#>cp WEB-INF/jamm.properties.dist WEB-INF/jamm.properties
```

Ve ardından WEB-INF/ jamm.properties dosyasını edit edin. Aşağıdaki satırları bu dosyaya ekleyin ve dosyayı kaydedin.

```
jamm.ldap.search_base = o=hosting, dc=myhosting, dc=example
jamm.ldap.root_dn = cn=Manager, dc=myhosting, dc=example
```

Şu an itibari ile Jamm Web uygulaması ve Tomcat kurulumu bitmiş durumda yapmanız gereken Tomcat'i başlatmak.

Aşağıdaki komut ile tomcat'i başlatın.

```
#>/usr/local/tomcat60/bin/startup.sh
```

Tomcat'in çalışıp çalışmadığını “netstat” komutu ile kontrol edin.
Komut satırında “netstat -ltnp” yaptığınızda aşağıdaki'ne benzer bir çıktı almanız gerekmekte.

```
tcp6       0      0 :::8080                 :::*                    LISTEN      2950/java
```

8080 portunun çalıştığını görüyorsaız Tomcat çalışmış demektir.

Tomcat'i durdurmak için aşağıdaki komutu çalıştırın.

```
#>/usr/local/tomcat60/bin/shutdown.sh
```

Tomcat'i yenided başlatmak için bir komut mevcut değildir. Dolayısı ile tomcat'i yeniden başlatmak için, önce tomcat'i durdurmak ardından başlatmak gerekmektedir. Bu işlemler için yukarıdaki komutları kullanabilirsiniz.

Makineniz her açıldığında, Tomcat'in otomatik olarak çalışmasını istiyorsanız, tomcat çalıştırma komutunu /etc/.rc.local dosyasına eklemeniz yeterlidir. Çalıştırma komutunun “exit 0” satırından önce geldiğine özen gösteriniz.

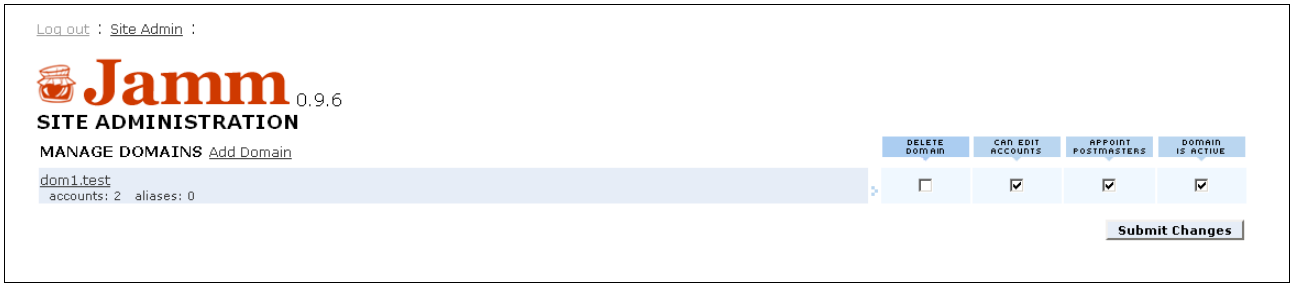
12.2 Jamm ile Alan Adı ve Mail Kullanıcısı Yaratmak

Jamm web uygulamasını kurduğumuza göre artık bu web uygulaması aracılığı ile sanal alan ve sanal mail kullanıcısı ekleyebiliriz.

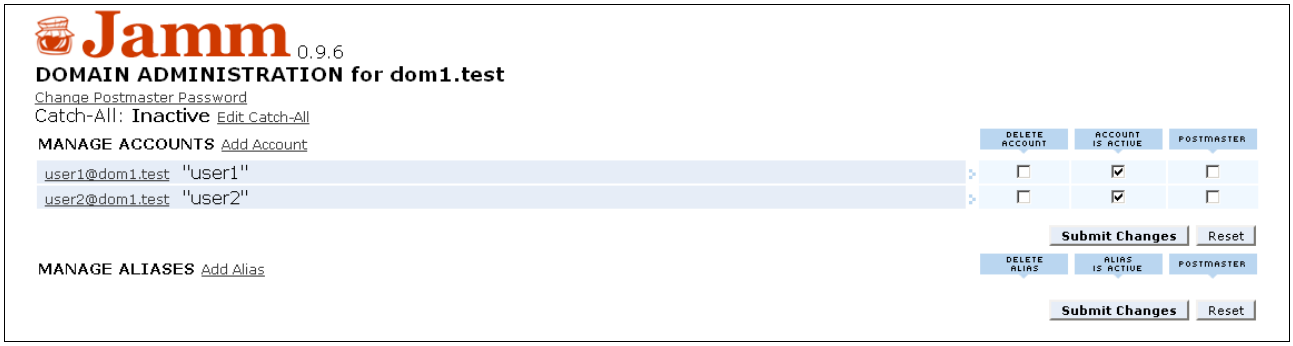
İnternet gezgininiz(Browser) ile http://<sunucu ip adresi>:8080/jamm adresini açınız. “sunucu ip adresi” sizin kurulum yaptığınız makinenin IP'sidir.

Ekranınıza aşağıdaki gibi kulanıcı ve parola soran bir form gelecek. Burada kullanıcı adına “root”, parola kısmına ise slapd.cong dosyasında “rootpw” direktifi ile girdiğiniz parolanın şifrelenmemiş halini giriniz.

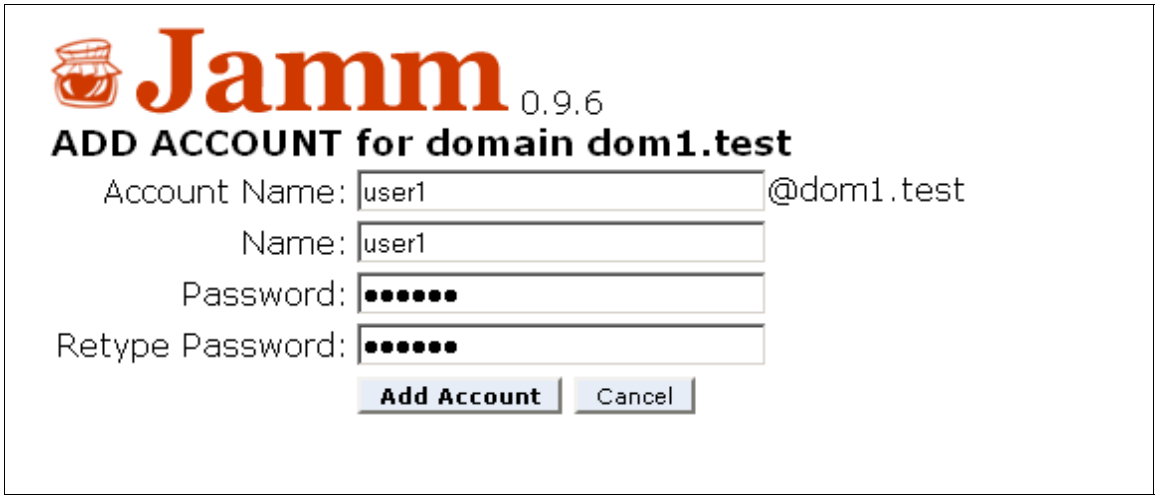
Jamm uygulamasına girdiğinizde aşağıdakine benzer bir arayüz ile karşılaşmanız gerekmekte.



Yukarıdaki arayüzden “Add Domain” linkine tıklayınız. Gelen ekranda “dom1.test” alan adını ve dom1.test postmaster kullanıcısı için parolayı kayıt ediniz. Kayıttan sonra tekrar yukarıdaki alan adlarını listeleyen ekrana geri döneceksiniz. Bu ekranda “dom1.test” linkine tıklayınız. Aşağıdaki gibi bir ekranla karşılaşacaksınız.



Bu ekranda “Add Account” linkine tıklayınız. Aşağıdaki gibi bir ekran gelecek. Bu ekranda kulanıcı adı olarak user1 ve parola olarak “qazwsx” giriniz ve kaydediniz.



Böylelikle bir alan adı (dom1.test) ve bir mail kullanıcı (user1@dom1.test) yaratmış olduk. İlerde test aşamasında lazım olacağı için bir de user2@dom1.test kullanıcısı yaratınız.

13 Postfix Kurulumu

Postfix her ne kadar Mail sunucumuzun ana parçası olarak büyük bir kısmını oluştursa da, kurulumu çok basittir. Ama zor olan postfix'i isteğinize göre konfigüre etmektir. Çünkü inanılmaz derece çok konfigür parametresi mevcuttur. Biz bu parametrelerden sadece bize lazım olanları kullanacağız.

Postfix'i kurmadan önce sisteminizde başka bir MTA mevcut mu bir kontrol edin. Sisteminizde exim4 veya sendmail yüklü olabilir. Mümkünse onları kaldırın, eğer kaldırmak istemiyorsanız stop edin.

Postfix kurulumuna postfix'in ihtiyaç duyacağı kullanıcı ve grupların kurulumu ile başlayalım. Postfix çalışma esnasında “postfix” kullanıcısına ve “postdrop” grubuna ihtiyaç duyar. Kullanıcı ve grup adları “postfix” ve “postdrop” olmak zorunda değil. Ama postfix ve postdrop kullacıları çalışırken bu kullanıcı ve gruplara ihtiyaç duyacağı için bu adla anılırlar.

Aşağıdaki komutlar aracılığı ile “postfix” kullanıcısını ve “postdrop” grubunu yaratınız. “adduser” komutu sizden postfix için parola isteyecektir. Bunun için güçlü bir parola giriniz.

```
#>adduser postfix
#>groupadd postdrop
```

Postfix'i paketini aşağıdaki komut aracılığı ile bulunduğunuz dizine indiriniz.

```
#>wget http://www.tigertech.net/mirrors/postfix-release/official/postfix-2.5.5.tar.gz
```

Ardından aşağıdaki komut aracılığı ile paketi /usr/local/mailserver_makedir dizini altına açın ve ana dizine geçin.

```
#>tar xvzf postfix-2.5.5.tar.gz -C /usr/local/mailserver_makedir
#>cd /usr/local/mailserver_makedir/postfix-2.5.5
```

Sıra derleme aşamasında. Aşağıdaki komutlar yardımı ile Postfix'i derleyin.

```
#>make makefiles CCARGS="-DUSE_SASL_AUTH -DUSE_CYRUS_SASL -DHAS_LDAP
-I/usr/include -I/usr/include/sasl  -I/usr/local/mailserver/openldap/include"
AUXLIBS="-L/usr/local/mailserver/openldap/lib  -lldap -llber -lsasl2"
#>make
#>make install
```

“make install” komutu size bir dizi soru soracaktır. Tüm sorulara “enter”'a basarak cevap verin, bu

Postfix'i varsayılan ayarları ile sisteminize kuracaktır.

Yalnız kurulum henüz bitmemiştir. Hala kurulum ile ilgili yapmamız gerekn birkaç adım vardır.

Postfix kurulduğunda /usr/sbin/sendmail adında bir dosya yaratır. Ayrıca /usr/lib/sendmail sembolik linki /usr/sbin/sendmail dosyasını işaret etmelidir. Ama eğer sisteminizde Postfix kurulumundan önce başka bir MTA mevcut ise /usr/lib/sendmail sembolik linki başka bir sendmail programına işaret ediyor olabilir. Yapmamız gereken /usr/lib/sendmail sembolik linkini silmek ve aynı isimde, /usr/sbin/sendmail dosyasına işaret eden yeni bir sembolik link yaratmaktır.

Bunun için önce aşağıdaki komut yardımı ile /usr/lib/sendmail sembolik linkini silin.

```
#>rm -f /usr/lib/sendmail
```

Ardından, aşağıdaki komut yardımı ile /usr/lib/sendmail sembolik linkini /usr/sbin/sendmail programına işaret edecek şekilde yaratın.

```
#>ln -s /usr/sbin/sendmail /usr/lib/sendmail
```

Postfix, mail takma adlarını /etc/postfix/aliases dosyasında saklar. Ama takma adlar için, Postfix doğrudan /etc/postfix/aliases dosyasını okumaz. Onun yerine /etc/postfix/aliases.db dosyasını okur. /etc/postfix/aliases.db dosyası /etc/postfix/aliases dosyası ile aynı içeriğe sahiptir. Ama bir farkla; /etc/postfix/aliases dosyası içeriğini tekst olarak saklar, /etc/postfix/aliases.db ise içeriğini ikili sayı sisteminde saklar. /etc/postfix/aliases.db dosyasının takma adları saklama formatı, poxtfix'e performans kazandırır. Çünkü ikili dosyaları okumak, tekst dosyalarını okumaktan daha hızlı bir şekilde gerçekleşir.

Yalnız, postfix /etc/postfix/aliases.db dosyasını okuduğu halde, biz postfix'e takma ad tanımlarken /etc/postfix/aliases.db dosyasına tanımlama yapmayız. Onun yerine takma adları /etc/postfix/aliases dosyasına tanımlar, ve “newaliases” komutu yardımı ile /etc/postfix/aliases.db dosyasını ürettiririz.

Kurulum bittikten sonra /etc/postfix/aliases dosyası yaratılır ama /etc/postfix/aliases.db dosyası yaratılmaz. Aliases.db dosyasının “newaliases” komutu ile yaratılması gerekmektedir. Aşağıdaki komutu çalıştırarak aliases.db dosyasını yaratabilirsiniz.

```
#> newaliases
```

Kurulum bitti, şimdi sıra postfix ayarlamalarında.

Postfix'in 2 tane ayar dosyası vardır:

1. /etc/postfix/master.cf
2. /etc/postfix/main.cf

main.cf ayar dosyası Postfix'in kendisini ayarlamak içindir. Master.cf ayar dosyası postfix ile entegre çalışan programları, postfix'e entegre etmek için kullanılan ayar dosyasıdır.

Bir de Postfix'in dizaynından kaynaklı entegre olduğu bir çok sistem vardır. Bunlardan biri de Cyrus-Sasl'dır. Postfix ve Cyrus-Sasl'ın uyumlu bir şekilde çalışmasını sağlayan ayar dosyası ise

3. /usr/lib/sasl2/smtpd.conf

dosyasıdır.

Öncelikle main.cf dosyasını ayarlayalım, master.cf dosyasını ayarlamaya şu an itibari ile gerek yoktur, smtpd.conf dosyasının ayarlanmasına Cyrus-Sasl kurulumunda bahsedilecektir.

/etc/postfix/main.cf dosyasının bir yedeğini alın ve /etc/postfix/main.cf içindeki tüm verileri silin. Aşağıdaki main.cf içeriğini /etc/postfix/main.cf dosyasına yapıştırın.

```
queue_directory = /var/spool/postfix
command_directory = /usr/sbin
daemon_directory = /usr/libexec/postfix
data_directory = /var/lib/postfix
mail_owner = postfix
myhostname = mailserver.interis.com
mydomain = interis.com
myorigin = $myhostname
inet_interfaces = all
unknown_local_recipient_reject_code = 550
mynetworks = 10.0.5.0/24, 127.0.0.0/8
alias_maps = hash:/etc/postfix/aliases
alias_database = hash:/etc/postfix/aliases
home_mailbox = Maildir/
mailbox_command = /usr/bin/procmail
debug_peer_level = 2
debugger_command =
       PATH=/bin:/usr/bin:/usr/local/bin:/usr/X11R6/bin
       ddd $daemon_directory/$process_name $process_id & sleep 5
sendmail_path = /usr/sbin/sendmail
newaliases_path = /usr/bin/newaliases
mailq_path = /usr/bin/mailq
setgid_group = postdrop
html_directory = no
manpage_directory = /usr/local/man
sample_directory = /etc/postfix
readme_directory = no
# Virtual Domain Information
domains_server_host = localhost
domains_search_base = o=hosting,dc=myhosting,dc=example
domains_query_filter = (&(objectClass=JammVirtualDomain)(jvd=%s)(accountActive=TRUE)
(delete=FALSE))
domains_result_attribute = jvd
domains_bind = no
domains_scope = one

# User Aliases
aliases_server_host = localhost
aliases_search_base = o=hosting,dc=myhosting,dc=example
aliases_query_filter = (&(objectClass=JammMailAlias)(mail=%s)(accountActive=TRUE))
aliases_result_attribute = maildrop
aliases_bind = no

# User Accounts
accounts_server_host = localhost
accounts_search_base = o=hosting,dc=myhosting,dc=example
accounts_query_filter = (&(objectClass=JammMailAccount)(mail=%s)(accountActive=TRUE)
```

```
(delete=FALSE))
accounts_result_attribute = mailbox
accounts_bind = no

accountsmap_server_host = localhost
accountsmap_search_base = o=hosting,dc=myhosting,dc=example
accountsmap_query_filter = (&(objectClass=JammMailAccount)(mail=%s)(accountActive=TRUE)
(delete=FALSE))
accountsmap_result_attribute = mail
accountsmap_bind = no

virtual_alias_maps = ldap:accountsmap, ldap:aliases
virtual_transport = virtual
virtual_mailbox_base = /home/vmail/domains
virtual_mailbox_maps = ldap:accounts
virtual_mailbox_domains = ldap:domains
virtual_minimum_uid = 101
virtual_uid_maps = static:101
virtual_gid_maps = static:101

# SASL Support
smtpd_sasl_auth_enable = yes
smtpd_sasl_local_domain =
smtpd_sasl_authenticated_header = yes
broken_sasl_auth_clients = yes
smtpd_sasl_path = smtpd
smtpd_recipient_restrictions = permit_sasl_authenticated, check_relay_domains,
reject_unauth_destination
smtpd_sasl_security_options = noanonymous
smtp_sasl_auth_enable = no
```

Yukarıdaki içeriği main.cf dosyasına yapıştırdığınızda, dikkat etmeniz gereken bir nokta vardır. “PATH” ve “ddd” ile başlayan satırlar “tab” ile başlamak zorundadır. Lütfen buna özen gösterin.

Main.cf dosyasında sistemden sisteme değişen bazı parameterler vardır. Aşağıdaki parametreleri sizin sisteminize uyarlamak gerekmektedir.

```
myhostname = mailserver.interis.com
mydomain = interis.com
```

Kurulum aşamasında makine adı ve alan adını farklı girmişseniz, yukarıdaki iki parameteryi ona göre ayarlayın.

```
mynetworks = 10.0.5.0/24, 127.0.0.0/8
```

Makinenizin bulunduğu network farklı olabilir. Kendi makinenizin ve Postfix'in kendi ağı olarak görmesini istediğiniz ağ'ları mynetworks parametresi ile yerleştiriniz.

```
virtual_mailbox_base = /home/vmail/domains
```

virtual_mailbox_base parametresi kullanıcılara gelen maillerin hangi dizin altında saklanacağını belirtir. Yukarıdaki ayar maillerin “/home/vmail/domains” dizini altında saklanacağını göstermektedir. Fakat şu an itibari ile böyle bir dizin yaratmadık. Dikkat ederseniz bu dizin için bir de “vmail” kullanıcısı yaratılmış. Böyle bir kullanıcıya ihtiyaç var çünkü mailler bu dizin altına

bırakılırken vmail kullanıcısının hakları kullanılarak bırakılıyor. İlerleyen aşamalarda bundan bahsedeceğiz.

Şimdi yapmamız gereken bir "vmail" kullanıcısı yaratmak ve "vmail" ev dizini altında "domains" dizini yaratmak.

Aşağıdaki komut aracılığı ile "vmail" grubunu ve kullanıcısını yaratınız. Aşağıda grup numarası olarak 1001 ve kullanıcı numarası olarak yine 1001 verdik. Eğer bu numaralar sistem tarafından başka bir kullanıcı ve gruba verilmişse 1001 numarasını sisteminizde boş bir numara ile değiştirin. 1024 ile 65535 arasında bir numara kullanabilirsiniz. Grup ve kullanıcı numaralarını aynı verebilirsiniz.

```
#>groupadd -g 1001 vmail
#>useradd -g vmail -u 1001 vmail
#>mkdir /home/vmail
#>chown -R vmail:vmail /home/vmail
```

Ardından aşağıdaki komutlar aracılığı ile vmail kullanıcısı altında "domains" dizinini yaratın. "domains" dizininin okuma ve yazma haklarının "vmail" kullanıcı ve grubuna ait olduğuna özen gösteriniz.

```
#> mkdir /home/vmail/domains
#>chown -R vmail:vmail /home/vmail/domains
```

Böylelikle virtual_mailbox_base ayarlamasını yapmış oluyoruz..

Yukarıdaki ayarlamalar vmail kullanıcısı ile ilgili ayarlamalardı. Aşağıdaki 3 parametre de vmail kullanıcısı ile alakalı.

virtual_minimum_uid parametresi vmail kullanıcısının Uid'sine işaret etmektedir, yani yukarıdaki örnekteki 1001 sayısına. virtual_uid_maps parametresi yine vmail kullanıcısının Uid'sine işaret etmektedir, yani yukarıdaki örnekteki 1001 sayısına. virtual_gid_maps parametresi vmail grup numarasına işaret etmektedir, yani yukarıdaki örnekteki 1001 sayısına.

Yapmanız gereken, aşağıda 101 olarak görülen numarları 1001 ile değiştirmek, veya vmail kullanıcı ve grup numaraları için verdiğiniz sayı ile değiştirmek..

```
virtual_minimum_uid = 101
virtual_uid_maps = static:101
virtual_gid_maps = static:101
```

Bu hali ile postfix kurulumu bitmiş durumda, bir sonraki aşamaya geçebilirsiniz.

13.1 Postifx'i başlatmak ve durdurmak.

Öncelikle şu uyarıyı yapalım, Cyrus-Sasl kurulumu ve Postfix ile entegrasyonu tamamlanmadan, postfix'i başlatmayın.

Postfix'i başlatmak için aşağıdaki komutu kullanabilirsiniz.

```
#>postfix start
```

Postfix'i durdurmak için aşağıdaki komutu kullanabilirsiniz.

```
#>postfix start
```

Postfix'i yeniden başlatabilmek için bir komut mevcut değildir. Dolayısı ile, postfix'i yeniden başlatmak için öncelikde postfix'i durdurmak ardında başlatmak zorundasınız. Yukarıdaki komutlar yardımı ile bunu başarabilirsiniz.

Postfix'in çalışıp çalışmadığını görmek istiyorsanız, komut satırında “netstat -ltnp” yazın. Eğer aşağıdaki gibi bir çıktı iel karşılaşıyorsanız, Postfix çalışıyor demektir.

```
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN      2330/master
```

Makineniz her açıldığında, Postfix'in otomatik olarak çalışmasını istiyorsanız, Postfix çalıştırma komutunu “/etc/rc.local” dosyasına ekleyiniz. Bu komutun “exit 0” satırından önce gelmesine özen gösteriniz.

14 Cyrus-Sasl Kurulumu

Cyrus-Sasl mail kullanıcıları için kimliklendirme işlemini yapan birimdir. Diğer adı ile, SMTP Auth işlevselliğini uygulayan birimdir. Cyrus-Sasl'ın düzgün çalışabilmesi için OpenLDAP kütüphanesi ile birlikte derlenmesi gerekmektedir. Çünkü Cyrus-Sasl kullanıcı bilgilerini OpenLDAP dizinlerinden okuyacaktır. Bu nedenle OpenLDAP istemcisi gibi davranabilmelidir. Ek olarak, Cyrus-Sasl ldap sunucusunun adresinin ve LDAP sunucusuna bağlantı için kullanıcı ve parola bilgilerine ihtiyaç duymaktadır. Bu durum için de Cyrus-sasl'ın ayrıca bir konfigürasyon dosyası mevcuttur.

Bunlara ek olarak, Postfix, mail kullanıcılarını kimliklendirirken, yani SMTP Auth yaparken, Cyrus-Sasl sunucusu olan saslauthd programı ile entegre çalışmaktadır. Bu sebeple, Cyrus-Sasl kurulumundan sonra, Postfix ve Cyrus-Sasl ın birlikte çalışabilmesi için Cyrus-Sasl'ın ve postfix'in saslauthd üzerinden ayarlanması gerekmektedir. Bu ayarları kurulum bittikten sonra yapacağız.

Şimdi kuruluma başlayalım. Aşağıdaki komut aracılığı ile Cyrus-Sasl paketini bulunduğunuz dizine

indiriniz.

```
#>wget http://ftp.andrew.cmu.edu/pub/cyrus-mail/cyrus-sasl-2.1.22.tar.gz
```

Ardından aşağıdaki komut yardımı ile paketi /usr/local/mailserver_makedir dizinine açınız ve Cyrus-Sasl ana dizinine geçiniz.

```
#>tar xvzf cyrus-sasl-2.1.22.tar.gz -C /usr/local/mailserver_makedir/
#>cd /usr/local/mailserver_makedir/cyrus-sasl-2.1.22
```

Aşağıdaki komutlar yardımı ile Cyrus-Sasl paketini sisteminize kurunuz.

```
#>./configure --prefix=/usr/local/mailserver/cyrus_sasl --with-ldap=/usr/local/mailserver/openldap
#>make
#>make install
```

Kurulum ardından hala yapmamız gereen işler vardır. Cyrus-Sasl çalışma esnasında unix-soketi yaratır. Ama henüz bu soket için bir yer belirlemiş değiliz. Çoğu linux dağıtımında bu soketin yeri "/var/run/saslauthd" yada "/var/state/saslauthd" dizinleri altındadır. Biz de bu standatlara uymak adına /var/run/saslauthd dizinini yaratacak ve /var/run/saslauthd sembolik linkini /var/run/saslauthd dizinini gösterecek şekilde yaratacağız. Bunu gerçekleştirmek için aşağıdaki komutları çalıştırın.

```
#>mkdir -p /var/run/saslauthd
#>mkdir -p /var/state
#>ln -s /var/run/saslauthd /var/state/saslauthd
```

Cyrus-sasl kurulumu bitti, şimdi sıra ayarlamalarda.

Önce, Cyrus-Sasl'ı OpenLDAP ile entegre edeceğiz, ardından Postfix ile.

Cyrus-Sasl, OpenLDAP ile konuşurken /etc/saslauthd.conf dosyasındaki ayar parametrelerinden yararlanır. Yani aslında, Cyrus-Sasl'ın OpenLDAP ile bağlantı bilgileri bu dosyada saklanmakta. Cyrus-Sasl bu bağlantı bilgileri ile OpenLDAP'a bağlanabilmektedir.

Bizim yapmamız gereken, /etc/saslauthd.conf dosyasını öyle ayarlamak ki Cyrus-Sasl saslauthd programı yardımı sayesinde OpenLDAP ile düzgün bir şekilde konuşabilsin.

Şu an /etc/saslauthd.conf adında bir dosya mevcut değil. Bu dosyayı yaratın ve aşağıdaki içeriği içine yapıştırın.

```
ldap_servers: ldap://127.0.0.1/
ldap_port: 389
ldap_mech: simple
ldap_search_base: o=hosting,dc=myhosting,dc=example
```

```
ldap_auth_method: bind
ldap_bind_dn: cn=Manager,dc=myhosting,dc=example
ldap_password: ldapldap1
ldap_filter: (&(objectClass=JammMailAccount)(mail=%u@%r)(accountActive=TRUE)
(delete=FALSE))
```

”ldap_password:” directifine dikkat ediniz. Bu direktif “`cn=Manager,dc=myhosting,dc=example`” kullanıcısının parolasına işaret etmektedir. Hatırlayacak olursanız, biz bu parolayı slapd.conf dosyasında “rootpw” direktifi ile vermiştik. Siz de “rootpw” direktifi ile verdiğiniz parolanın şifrelenmemiş halini aşağıdaki gibi “ldap_password:” direktifi yanına yazınız.

/etc/saslauthd.conf dosyasında dikkat etmemiz gereken Birkaç nokta daha vardır. Bunlar aşağıdaki gibidir.

1. Satırlar boşlu ile başlamamalı, eğer varsa silin
2. Satır sonlarında boşluklar olmamalı, eğer varsa silin
3. “:” karakterinden sonra sadece bir boşluk bırakınız, eğer birden fazla boşluk varsa, fazla boşlukları silin.

Yukarıdaki 3 hususu yerine getirmezseniz Cyrus-Sasl ve OpenLDAP entegre bir şekilde çalışmaz.

Böylelikle OpenLDAP ile Cyrus-Sasl'ı entegre etmiş olduk.

Şimdi sıra Cyrus-Sasl ve Postfix entegrasyonunda.

Postix SMTP Auth yöntemleri için saslauthd programına başvurur. Cyrus-Sasl ana programı olan saslauthd programı hangi kimliklendirme yöntemlerini destekliyorsa Postfix de aynı kimliklendirme yöntemlerini destekliyor demektir. Bu yöntemler, /usr/lib/sasl2/smtpd.conf dosyasında belirtilir. Smptd.conf dosyasını ayarlama konusunda daha fazla bilgi için Cyrus-Sasl ev sayfasını ziyaret ediniz.

Şu an itibari ile /usr/lib/sasl2/smtpd.conf dosyası mevcut değil. Bu dosyayı yaratın ve aşağıdaki içeriği dosyaya yapıştırın.

```
pwcheck_method: saslauthd
mech_list: AUTH LOGIN PLAIN DIGEST-MD5
```

Smtpd.conf dosyası için, saslauthd.conf dosyasına dair yaptığımız uyarılar geçerlidir. Satırlar boşlukla başlamamalı, boşlukla bitmemeli ve “:” karakterinden sonra sadece 1 boşluk olmalı. Cyrus-Sasl ana programı saslauthd programını çalıştırmak için aşağıdaki komutu kullanabilirsiniz.

```
#> /usr/local/mailserver/cyrus_sasl/sbin/saslauthd -a ldap -m /var/run/saslauthd -O
/etc/saslauthd.conf
```

Çalışan bir saslauthd uygulamasını sonlandırmak istiyorsanız aşağıdaki komutu çalıştırın.

```
#>killall saslauthd
```

Cyrus-Sasl ana programı saslauthd programını yeniden başlatmak için bir komut öevcut değildir. Bu komutu yeniden başlatmak için öncelikde saslauthd programını durdurmak ardından programı başlatmak zorundasınız. Yukarıdaki komutlar yardımı ile bunu gerçekleştirebilirsiniz.

Saslauthd uygulamasının çalışıp çalışmadığını görmek istiyorsanız, komut satırında “ps aux |grep saslauthd” yazın. Eğer aşağıdaki gibi bir çıktı ile karşılaşıyorsanız, Saslauthd çalışıyor demektir.

```
root     2278  0.0  0.2   3840   732 ?        Ss   20:56   0:00
/usr/local/mailserver/cyrus_sasl/sbin/saslauthd -a ldap -m /var/run/saslauthd -O /etc/saslauthd.conf
root     2280  0.0  0.1   3840   408 ?        S    20:56   0:00
/usr/local/mailserver/cyrus_sasl/sbin/saslauthd -a ldap -m /var/run/saslauthd -O /etc/saslauthd.conf
root     2281  0.0  0.1   3840   372 ?        S    20:56   0:00
/usr/local/mailserver/cyrus_sasl/sbin/saslauthd -a ldap -m /var/run/saslauthd -O /etc/saslauthd.conf
root     2282  0.0  0.1   3840   372 ?        S    20:56   0:00
/usr/local/mailserver/cyrus_sasl/sbin/saslauthd -a ldap -m /var/run/saslauthd -O /etc/saslauthd.conf
root     2283  0.0  0.1   3840   372 ?        S    20:56   0:00
/usr/local/mailserver/cyrus_sasl/sbin/saslauthd -a ldap -m /var/run/saslauthd -O /etc/saslauthd.conf
```

Cyrus-Sasl saslauthd'yi çalıştırırken kullandığımız komutu anlamak istiyorsanız birkaç noktayı açıklığa kavuşturmak gerekir.

Cyrus-Sasl bir kimliklendirme uygulamasıdır ve bir kullanıcıyı çeşitli yöntemlerle kimliklendirebilir. Cyrus-Sasl, kullanıcı ve parola bilgileriniz bir veri tabanında saklanıyor ise PAM kimliklendirme yöntem ile, ya da kullanıcı bilgileri bir dosyada saklanıyor ise getpwent metodu ile kimliklendirme yapabilir. Biz kullanıcı ve parola bilgilerini LDAP dizinlerinde saklıyoruz. Dolayısı ile kimliklendirme mekanizmamız “ldap”'dır. Bu sebeple saslauthd programına parametre olarak “-a ldap” verdik. Kimliklendirme mekanizması olarak ldap kullandığımız için saslauthd'nin ldap sunucusuna ulaşım ve giriş bilgilerine ihtiyacı vardır. Bu bilgileri de “-O” parametresi ile sağlıyoruz. “-O” paramtresi LDAP iletişim bilgilerini saklayan dosyaya işaret etmektedir. Bu da /etc/saslauthd.conf dosyasıdır.

Saslauthd çalışırken başka birimlerle de iletişim kurmaktadır. Bu iletişim Unix-Socket aracılığı ile olmaktadır. Saslauthd çalışırker bir soket yaratır ve diğer birimlerle iletişimi bu soket aracılığı ile sağlar. Diğer birimlerin saslauthd ile iletişim kurabilmesi için saslauthd soketinin nerede olduğunu bilmesi gerekmektedir. Bu dosya yolu saslauthd programına “-m” parametresi ile verilmektedir. Yukarıdaki örnekte, saslauthd çalıştığında dizin yolu /var/run/saslauthd olan dizin altına “mux” adında bir soket yaratır. Böylece diğer birimler saslauthd ile iletişime geçebilir.

Makineniz her açıldığında, Cyrus-Sasl'ın otomatik olarak çalışmasını istiyorsanız, Cyrus-Sasl çalıştırma komutunu “/etc/rc.local” dosyasına ekleyiniz. Bu komutun “exit 0” satırından önce

gelmesine özen gösteriniz.

Böylelikle Cyrus-Sasl ile Postfix arasındaki entegrasyonunu tamamlamış olduk.

Bind+OpenLDAP+Cyrus-Sasl+Postfix kurulumu ile Mail sunucumuzun belkemiğini yapılandırmış olduk.

Şimdi sıra, kurduğumuz sistemi ayağa kaldırmak ve test etmekte.

14.1 Sistemin Ayağa Kaldırılması

Mail sunucumuzun Ana kısmı için kurulumu tamamlamış bulunuyoruz. Artık sistemi ayağa kaldırabiliriz. Şu an itibari ile 4 adet sunucumuz bulunmakta. Bunlar

1. Bind
2. OpenLDAP
3. Cyrus-Sasl
4. Postfix

uygulamalarıdır. Sırası ile bu programları çalıştıracağız.

Önce Bind. Eğer bind programını daha önce başlatmamış iseniz aşağıdaki komut ile bind'ı başlatınız.

```
#>/etc/init.d/bind start
```

Eğer bind'ın çalışıp çalışmadığını kontrol etmek istiyorsanız, komut satırına “netstat -ltnp” yazıp enter'a basınız. Aşağıdakine benzer bir çıktı alıyorsanız bind çalışıyor demektir.

```
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN      3050/named
```

Eğer OpenLDAP'ı daha önce başlatmadı iseniz aşağıdaki komut ile OpenLDAP'ı başlatabilirsiniz.

```
#>/usr/local/mailserver/openldap/libexec/slapd
```

OpenLDAP'ın çalışıp çalışmadığını görmek istiyorsanız, komut satırında “netstat -ltnp” yazın. Eğer aşağıdaki gibi bir çıktı ile karşılaşıyorsanız, OpenLDAP çalışıyor demektir.

```
tcp        0      0 0.0.0.0:389             0.0.0.0:*               LISTEN      2275/slapd
```

Eğer Cyrus-Sasl ana programı olan saslauthd programını daha önce başlatmadı iseniz, aşağıdaki

komut ve parametreler ile Cyrus-Sasl uygulamasını başlatabilirsiniz.

```
#> /usr/local/mailserver/cyrus_sasl/sbin/saslauthd -a ldap -m /var/run/saslauthd -O
/etc/saslauthd.conf
```

Saslauthd uygulamasının çalışıp çalışmadığını görmek istiyorsanız, komut satırında “ps aux |grep saslauthd” yazın. Eğer aşağıdaki gibi bir çıktı ile karşılaşıyorsanız, Saslauthd çalışıyor demektir.

```
root     2278  0.0  0.2   3840   732 ?        Ss   20:56   0:00
/usr/local/mailserver/cyrus_sasl/sbin/saslauthd -a ldap -m /var/run/saslauthd -O /etc/saslauthd.conf
root     2280  0.0  0.1   3840   408 ?        S    20:56   0:00
/usr/local/mailserver/cyrus_sasl/sbin/saslauthd -a ldap -m /var/run/saslauthd -O /etc/saslauthd.conf
root     2281  0.0  0.1   3840   372 ?        S    20:56   0:00
/usr/local/mailserver/cyrus_sasl/sbin/saslauthd -a ldap -m /var/run/saslauthd -O /etc/saslauthd.conf
root     2282  0.0  0.1   3840   372 ?        S    20:56   0:00
/usr/local/mailserver/cyrus_sasl/sbin/saslauthd -a ldap -m /var/run/saslauthd -O /etc/saslauthd.conf
root     2283  0.0  0.1   3840   372 ?        S    20:56   0:00
/usr/local/mailserver/cyrus_sasl/sbin/saslauthd -a ldap -m /var/run/saslauthd -O /etc/saslauthd.conf
```

Şimdi sıra en önemli aşamada: Postfix MTA biriminin yani mail sunucusunun kalbinin çalıştırılmasında.

Eğer Postfix'i daha önce başlatmadı iseniz, aşağıdaki komut yardımı ile postfix'i çalıştırabilirsiniz.

```
#>postfix start
```

Postfix'in çalışıp çalışmadığını görmek istiyorsanız, komut satırında “netstat -ltnp” yazın. Eğer aşağıdaki gibi bir çıktı iel karşılaşıyorsanız, Postfix çalışıyor demektir.

```
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN      2330/master
```

15 Sistemin Test Edilmesi

Mail sunucumuz şu an ayakta ve çalışıyor. Ama henüz düzgün çalışıp çalışmadığını bilmiyoruz. Bunu öğrenmek için Postfix'e bağlanıp bir kullanıcımıza mail atmamız gerekmekte. Mail atarken Cyrus-Sasl'ın, ve OpenLDAP'ın düzgün açlışıp çalışmadığını da test etmiş olacağız. Mail atma işini Postfix sunucusuna “telnet” komutu ile bağlanıp yapacağız. Dolayısı ile eğer sisteminizde telnet programı yüklü değil ise aşağıdaki komut ile telnet programını yükleyiniz.

```
#>aptitude install inetutils-telnet
```

Aşağıdaki komut ile makinenizin 25. portuna(Postfix'in dinlediği porta) bağlanın.

```
#>telnet localhost 25
```

Aşağıkdakine benzer bir çıktı almanız gerekmekte.

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
libshishi: warning: /root/.shishi/tickets: No such file or directory
220 mailserver.interis.com ESMTP Postfix
```

“libshishi” ile başalayan uyarının bir önemi yoktur. Şimdilik yok sayabiliriz.

Eğer Postfix'in verdiği cevaplar arasında “220” ile başlayan bir satır yok ise, Postfix'iniz çalışmıyor demektir. Bu aşamada lütfen dokümanı tekrar gözden geçirin.

Eğer yukarıdaki gibi bir “220” çıktısı görüyorsanız postfix'iniz çalışıyor demektir. Şimdi kursor'ün beklediği yerden “ehlo abc.com” yazıp enter'a basın. Bu komut ile kendinizi Postfix sunucusuna “abc.com” olarak tanıtmış oluyorsunuz. Aşağıdakine benzer bir çıktı almanız gerekmekte.

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
libshishi: warning: /root/.shishi/tickets: No such file or directory
220 mailserver.interis.com ESMTP Postfix
ehlo abc.com
250-mailserver.interis.com
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-AUTH DIGEST-MD5 LOGIN PLAIN
```

```
250-AUTH=DIGEST-MD5 LOGIN PLAIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
```

Postfix'in “ehlo” komutuna karşılık verdiği cevap postfix'in sizin için neler yapabildiğini göstermektedir. Burada bizim için önemli olan “AUTH” ile başlayan satırlardır. AUTH satırında Postfix, kimliklendirme yöntemi için 3 yöntem kullandığını söylemekte. Dikkat ederseniz bu 3 yöntem de Cyrus-Sasl'ı ayarlarken /usr/lib/sasl2/smtpd.conf dosyasına yazdığımız metodlarla aynı. Bu demek oluyor ki, Postfix Cyrus-Sasl ile entegre olmuş durumda ve saslauthd programından aldığı kimliklendirme metodlarını bize cevap olarak verebilmekte.

Her 3 yöntemin kendine özgü protokolü vardır. Biz kimliklendirme için plain metodunu kullanacağız.

Burada biraz plain metodundan bahsetmek gerekiyor. Plain metodu ile kimliklendirme yaparken, kullanıcı adı ve parola Base64 kodlamasına göre yapılmaktadır. Yani Postfix'e kimliklendirme aşamasında kullanıcı adı ve parolamızı söylerken kullanıcı adı ve parolamızın Base64 kodu alınmış halini söyleriz. Dahası, kullanıcı adı ve parolamız aşağıdaki formatta olmak zorunda.

```
kulanıcıadı\0kullanıcıadı\0parola
```

Şimdi isterseniz uygulamaya geçelim. Daha önceden user1@dom1.test ve user2@dom1.test mail kullanıcılarını yaratmış ve bunlara parola atamıştık. Gelin user@dom1.test kullanıcısını plain metodu ile kimliklendirelim. Aşağıdaki komut ile kullanıcı adı ve parolamızın base64 kodunu üretelim.

```
#>perl -MMIME::Base64 -e 'print
encode_base64("user1\@dom1.test\0user1\@dom1.test\0qazwsx");'
```

Yukarıdaki komut aşağıdaki gibi ibr çıktı üretecektir.

```
dXNlcjFAZG9tMS50ZXN0AHVzZXIxQGRvbTEudGVzdABxYXp3c3g=
```

Bu çıktıyı kullanarak “telnet localhost 25” komutu ile açtığımız bağlatıya aşağıdaki gibi komutumuzu yazalım.

```
auth plain dXNlcjFAZG9tMS50ZXN0AHVzZXIxQGRvbTEudGVzdABxYXp3c3g=
```

Aşağıdaki gibi 235 ile başlayan bir satır alıyorsanız, user1@dom1.test Postfix sunucumuzda kimliklendirilmiş demektir.

```
auth plain dXNlcjFAZG9tMS50ZXN0AHVzZXIxQGRvbTEudGVzdABxYXp3c3g=
```

```
235 2.7.0 Authentication successful
```

Eğer buraya kadar anlatılan şekilde gelmiş iseniz, Postfix-OpenLDAP-Cyrus-Sasl üçlüsü düzgün bir şekilde çalışıyor demektir.

İsterseniz bir de user@dom1.test kullanıcısından user2£@dom1.test kullanıcısna mail atalım.

“telnet localhost 25” bağlantımızda kaldığımız yerden devam edelim.

Aşağıkdaki komut ile mailin gönderen adresini belirtin.

```
mail from:user1@dom1.test
```

Aşağıdaki komut ile mailin alıcı adresini belirtin.

```
mail from:user2@dom1.test
```

Aşağıdaki komut ile mailin içeriğine başladığınızı belirtin.

```
data
```

Son olarak normal bir tekst yazar gibi mailinizi yazın ve önce “enter” ardından “.” ve enter yazın ve maili sonlandırın. Mail protokolünde mail sonlandırmak için en son satırınıza “.” (nokta) yazar ve enter a basarsınız. Böylelikle maili göndermiş olursunuz.

Daha açıklayıcı olması için “telnet localhost 25” komutundan sonraki çıktıların tamamı aşağıdaki gibidir.

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
libshishi: warning: /root/.shishi/tickets: No such file or directory
220 mailserver.interis.com ESMTP Postfix
ehlo abc.com
250-mailserver.interis.com
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-AUTH DIGEST-MD5 LOGIN PLAIN
250-AUTH=DIGEST-MD5 LOGIN PLAIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
auth plain dXNlcjFAZG9tMS50ZXN0AHVzZXIxQGRvbTEudGVzdABxYXp3c3g=
235 2.7.0 Authentication successful
mail from:user1@dom1.test
250 2.1.0 Ok
rcpt to:user2@dom1.test
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
```

```
ilk mail testi.
.
```

Eğer mailin user2@dom1.test kullancısına ulaşıp ulaşmadığını görmek istiyorsanız, "/home/vmail/domains/dom1.test/user2/new" dizinine bakınız. Eğer orada bir dosya duruyor ise içini açıp bakınız, yukarıda gönderdiğimiz mail olmalı.

16 Mesaj Erişim Birimi Kurulumu (POP3, IMAP)

Mail sunucumuzun ana kısmını bitirdik. Artık kullanıcılarımız mail atabiliyor ve alabiliyorlar. Ama gayet ilkel yöntemlerle, yani telnek yaparak. Dikkat ederseniz user2@dom1.test kullanıcısına gelen maili de konsolda bit tekst editörle okuduk. Bu sistem yöneticileri için pek irrite edici bir yöntem olmayabilir, ama kullanıcılarınızı bu yöntemle mail atma ve okumaya ikna etmeniz imkansız. Bu problem sadece bizim kurduğumuz Mail sunucumuzda değil, internette mail oldusu oluştuğundan beri var. Tabi internet dünyasında Hiçbir problem çözümsüz kalmaz. İnternet guruları bu duruma hemen bir çözüm buldular: Maillerinize erişimi sağlayan sunucular. Tabi sadece maillere ulaşımı sağlamakla kalmıyor aynı zamanda bu sunucular SMTP sunucunuza bağlanıp mail de atabiliyor.

Şu an internet teknolojilerinde kabul görmüş 2 adet mail erişim protokolü mevcut. Bunlar

- IMAP(Internet Message Access Protocol, Internet Mail Erişim Protokolü)
- POP3 (Post Office Protokolü, Posta Şubesi Protokolü)

protokolleridir.

Öncelikle şunu hemen açıklığa kavuşturalım: IMAP ve POP3 maillerinizi uzaktaki bir adrese ulaştırmak için kullanılmazlar. Yani postfix'in MTA görevini üstlenmezler. Çoğunlukla bu karıştırılır. Gerçek hayattan örnek verelim.

Örneğin bir postanız var ve bunu İngilterenin Londra şehrinde oturan bir arkadaşınıza postalamak istiyorsunuz. Yapmanız gereken postanızı, mahallenizdeki PTT ofisine bırakmak. Peki postanızı mahallenizdeki PTT ofisi mi Londraya ulaştırıyor. Hayır. Eğer öyle olsaydı, PTT ofisindeki çalışanın postanızı beraberinde alıp, uçak ile Londraya uçup arkadaşınıza iletmesi gerekirdi. Bunun yerine, PTT çalışanı postanızı şehrinizdeki giden postaların biriktiği bir merkezi ofise aktarır ve orada işi biter. Burdan sonrası şehrin merkezi PTT biriminin işidir. Merkezi PTT de postanızı İngilterenin Yetkili PTT birimine teslim eder ve şehrimizdeki PTT merkezi biriminin işi burada bitmiş olur. Bu aşamdan sonrası uzaktaki PTT biriminin yani İngiltere çalışanlarının işidir.

Bu örnekte resmedilen mahallemizdeki PTT ofisi, IMAP ve POP3 birimine denk gelmektedir. Şehrimizdeki Merkezi PTT birimi ise bizim Postfix'imiz yani MTA'mızdır. Uzaktaki İngiltere PTT birimi ise, uzaktaki MTA'dır. Dikkat ederseniz burada posta transferi Uzaktaki İngiltere PTT birimi ile şehrimizdeki PTT birimi arasındadır. Bu senaryo tamamen internetteki mail transferleri için de aynıdır.

Peki, yüzümüzü tekrar teknik konumuza dönelim. IMAP ve PO3'ün Mail sunucumuzdaki rolü tam olarak nedir?

IMAP ve POP3 sunucuları, MTA ile posta kutumuza düşen mailleri okumamızı sağlar. Mail gönderriken de bizim yerimize MTA nın 25'inci portuna bağlanıp maillerimizi göndermemize yardımcı olur(Dikkat edin mail gönderir demiyorum, yardımcı olur diyorum.)

Öncelikle maillerimizi IMAP ve POP3 ile nasıl okuduğumuzu açalım. İlk dikkat etmemiz gereken şey, MTA sunucumuz, yani postfix, nerede kurulu ise IMAP ve POP3'de aynı makinede kurulu olmalıdır. IMAP ve POP3 sunucularının mail kullanıcılarının adreslerini ve maillerinin hangi

dizinde sakladığını bilmesi gerekmektedir. Eğer maillerin nerede saklandıklarını bilemezlerse, bizim yerimize bu maillere ulaşıp bize iletemezler. Örneğin, mail sunucumuzda mailler /home/vmail/domains dizini altında <alan_adı>/<kullanıcı adı> formatında saklanmaktadır. Bu bilgiyi bir şekilde IMAP ve POP3'e tanıtmak gerekiyor.
Aynı durum kullanıcı adları ve parolalar için de geçerli. IMAP ve POP3 elbetteki herkese açık olmayacaktır. Bunun yanında bir kullanıcı sadece kendi mail dizinine ulaşabilmelidir, başkalarının mail dizinlerine ulaşamamalıdır. Bizm mail sunucumuz mail kullanıcısı, parolası ve mail dizinlerini LDAP'ta saklamaktadır. Bu bilgilerin bir şekilde IMAP ve POP3 sunucuları tarafından ulaşılabilir olması gerekmektedir.

Yukarıda saydığımız gekensinimlerin hepsi bizim mail sunucumuz için de geçerlidir. Bu şu anlama geliyor: IMAP-POP3 sunucuları OpenLDAP sunucumuzla entegre olacak.

Şimdi isterseniz kuruluma başlayalım.

17 Courier Authlib

İnternette bir çok IMAP ve POP3 protokolünü uygulayan sunucu mevcuttur. Biz bunlardan courier-IMAP ve courier-POP3 paketlerini seçtik. Ek olarak, kimliklendirme işlemleri için Courier-Authlib paketini seçtik. Courier-Authlib paketi LDAP'ta saklanan kullanıcı adı, parola ve kullanıcı mail dizinlerini Courier-IMAP ve Courier-POP3 sunucularına ulaştıran güvenlik katmanı rolünü üstlenecek.

Şimdi paket kurulumlarına başlayalım. İlk Courier-Authlib paketini kuracağız. Aşağıdaki komut yardımı ile Courier-Authlib paketini dizininize indiriniz.

```
#> wget http://prdownloads.sourceforge.net/courier/courier-
authlib-0.61.0.tar.bz2
```

Ardından aşağıdaki komut yardımı ile paketi açın.

```
#>tar xvjf courier-authlib-0.61.0.tar.bz2 -C /usr/local/mailserver_makedir/
```

Paketimiz /usr/local/mailserver_makedir/courier-authlib-0.61.0 dizini altında açılmış durumda. Burdan itibaren kurulum adımlarımızı biraz değiştireceğiz. Daha önce kurlumların hepsini root kullanıcısı ile yapıyorduk. Ama Courier-Authlib paketi kurulumun bazı aşamalarının normal bir kullanıcı ile yapılmasını istiyor. Courier-authlib kurulum adımlarına göre sadece “make install” komutu root olarak çalıştırılacak.

Biz de bu yoldan gidelim. Eğer normal bir kullanıcınız yok ise, hemn “adduser” komutu ile bir tane yaratın, örneğin “normuser”. Aşağıdaki komut yardımı ile kullanici k'ml'['n'z' “normuser” olarak değiştirin. Eğer aşağıdaki komut sonunda “$>” ile biten bir komut satırınına sahipseniz “normuser” olmuşsunuz demektir.

```
#>su – normuser
$>
```

Ama biz yine tedbiri elden bırakmayalım ve “whoami” komutu ile kim olduğumuzu öğrenelim.

Eğer cevap “normuser” ise devam deebiliriz.

```
$>whoami
normuser
$>
```

Aşağıdaki komut yardımı ile /usr/local/mailserver_makedir/courier-authlib-0.61.0 dizinine geçin.

```
$>cd /usr/local/mailserver_makedir/courier-authlib-0.61.0
```

Ardından aşağıdaki komut yardımı ile OpenLDAP kütüphane ve C başlık dosyalarını kabuk programının ortam değişkenlerine atayın. “configure” komutu OpenLDAP bilgilerine bu ortam değişkenleri ile ulaşacak.

```
$>export CFLAGS="-I/usr/local/mailserver/openldap/include"
$>export CPPFLAGS="-I/usr/local/mailserver/openldap/include"
$>export LDFLAGS="-L/usr/local/mailserver/openldap/lib"
```

Ortam değişkenlerini atadıktan sonra “configure” komutunu çalıştırabiliriz. Aşağıdaki komut yardımı ile “configure” komutunu çalıştırın.

```
$> ./configure --prefix=/usr/local/mailserver/courier_authlib --with-
mailuser=vmail --with-mailgroup=vmail    --with-
authdaemonrc=/usr/local/mailserver/courier_authlib/etc/authlib/authdaemonrc
--with-authldaprc=/usr/local/mailserver/courier_authlib/etc/authlib/authldaprc
--with-db=db
```

“configure” komutu sonlandığında aşağıdaki komutu çalıştırarak derleme işlemine başlayın.

```
$>make
```

Derleme işlemi sonlandığında, artık “normuser” komutundan çıkabilirsiniz. “normuser” ile yapacaklarımız bitmiş durumda. Bundan sonra, işlemlerimizi tekrar “root” olarak yapacağız. Aşağıdaki komut yardımı ile “normuser” kullanıcısından çıkın.

```
$>exit
```

Komut satırından “whoami” yazın eğer cevap olarak “root” alıyorsanız, tekrar “root” olmuşsunuz demektir. Root kullanıcısı ile tekrar /usr/local/mailserver_makedir/courier-authlib-0.61.0 diznine geçin. Aşağıdaki komut bu işlemi yapacaktır.

```
$>cd /usr/local/mailserver_makedir/courier-authlib-0.61.0
```

Ve son olarak aağıdaki komutu çalıştırarak, Courier-Authlib kurulumunu tamamlayın. Courier-Authlib paketinin /usr/local/mailserver/courier_authlib dizini altında kurulmuş olması gerekli.

```
#>make install
```

17.1 Courier-Authlib Konfigürasyonu

Courier-Authlib paketinin 2 adet ayar dosyası mevcuttur. Bunlar

- /usr/local/mailserver/courier_authlib/etc/authlib/authdaemonrc
- /usr/local/mailserver/courier_authlib/etc/authlib/authldaprc

dosyalarıdır. Ama aşağıdaki komut ile bu dizine geçiş yaptığınızda bu dosyaları bulamayacaksınız.

```
#>cd /usr/local/mailserver/courier_authlib/etc/authlib
#>ls
```

Onun yerine aşağıdaki dosyaları bulacaksınız.

- authdaemonrc.dist
- authldaprc.dist

“authdaemonrc” ve “authldaprc” dosyaları kurulum sonunda gelmezler, onun yerine bu 2 dosyanın varsayılan ayarlarını içeren yukarıdaki 2 dosya gelir. Yapmamız gerek bu 2 dosyadan “authdaemonrc” ve “authldaprc” dosyalarını yaratmak. Aşağıdaki komut ile bu dosyaları yaratın.

```
#>cp authdaemonrc.dist authdaemonrc
#>cp authldaprc.dist authldaprc
```

Henüz işimiz bitmiş değil. Bu dosyaları ayarlamamız gerekmekte. Önce “authdaemonrc” dosyasından başlayalım.

Bu dosyayı edit edin ve “authmodulelist” ile başlayan satırı bulun. Bu satırı aşağıdaki gibi değiştirin.

```
authmodulelist="authldap"
```

Eğer kullanıcıları kimliklendirirken olay kayıtlarını /var/log/syslog dosyasında görmek istiyorsak, yine authdaemonrc dosyasında DEBUG_LOGIN satırını bulun ve aşağıdaki gibi değiştirin.

```
DEBUG_LOGIN=2
```

Authdaemonrc dosyası ayarlanmış durumda. Bir sonraki ayar dosyası “authldaprc” dosyasına geçebiliriz.

“authldaprc” dosyasını edit edin ve içindeki tüm satırları silin. Ardından aşağıdaki içeriği bu dosyaya yapıştırın. Satır sonlarında boşluk kalmamasına özen gösterin. Eğer vmail kullanıcısına

için 10001 dışında bir numara atamışsanız, LDAP_GLOB_UID ve LDAP_GLOB_GID değişkenlerini uygun bir şekilde değiştiriniz.

```
LDAP_URI ldap://127.0.0.1
LDAP_PROTOCOL_VERSION 3
LDAP_BASEDN dc=myhosting,dc=example
LDAP_BINDDN cn=Manager,dc=myhosting,dc=example
LDAP_BINDPW qazwsx
LDAP_TIMEOUT 15
LDAP_AUTHBIND 1
LDAP_GLOB_UID 10001
LDAP_GLOB_GID 10001
LDAP_FILTER (objectClass=JammMailAccount)
LDAP_MAIL mail
LDAP_HOMEDIR homeDirectory
LDAP_DEFAULTDELIVERY defaultDelivery
LDAP_FULLNAME cn
LDAP_CRYPTPW userPassword
#LDAP_MAILROOT /home/vmail/domains
LDAP_DEREF never
LDAP_TLS 0
LDAP_MAILDIR mailbox
```

Şu an itibari ile courier-Authlib konfigürasyon işlemleri bitmiş durumda. Artık test edebiliriz.

Courier-Authlib Testi

Teste başlamdan önce OpenLDAP ve Courier-authlib programlarını çalıştırmalıyız. OpenLDAP'ı daha önce anlatıldığı gibi başlatın ve çalışıp çalışmadığını kontrol edin.

Courier-Authlib paketini çalıştırmak için aşağıdaki komutu çalıştırın.

```
#>/usr/local/mailserver/courier_authlib/sbin/authdaemond start
```

Durdurmak için aşağıdaki komutu çalıştırın.

```
#>/usr/local/mailserver/courier_authlib/sbin/authdaemond stop
```

Auhtdaemond programının çalışıp çalışmadığını kontrol etmek için komut satırına “ps aux |grep authdaemond” yazın. Eğer aşaıdaki gibi bir çıktı veriyor ise authdaemond çalışıyor demektir.

```
root     25658  0.0  0.1   1572   392 ?        S    11:38   0:00
/usr/local/mailserver/courier_authlib/sbin/courierlogger -pid=/usr/local/mailserver/courier_authlib/
var/spool/authdaemon/pid -start /usr/local/mailserver/courier_authlib/libexec/courier-
```

```
authlib/authdaemond
root     25659  0.0  0.4   3932  1124 ?       S    11:38   0:00
/usr/local/mailserver/courier_authlib/libexec/courier-authlib/authdaemond
root     25660  0.0  0.4   3980  1216 ?       S    11:38   0:00
/usr/local/mailserver/courier_authlib/libexec/courier-authlib/authdaemond
root     25661  0.0  0.1   3932   396 ?       S    11:38   0:00
/usr/local/mailserver/courier_authlib/libexec/courier-authlib/authdaemond
root     25662  0.0  0.1   3932   396 ?       S    11:38   0:00
/usr/local/mailserver/courier_authlib/libexec/courier-authlib/authdaemond
root     25663  0.0  0.1   3932   396 ?       S    11:38   0:00
/usr/local/mailserver/courier_authlib/libexec/courier-authlib/authdaemond
root     25664  0.0  0.1   3932   396 ?       S    11:38   0:00
/usr/local/mailserver/courier_authlib/libexec/courier-authlib/authdaemond
root     25705  0.0  0.2   2848   704 pts/1    R+   12:17   0:00 grep authdaemond
```

Artık test işlemine başlayabiliriz.

Testi yaparken /usr/local/mailserver/courier_authlib/sbin/authtest programını kullanacağız. Bu komut courier-authdaemond'nin LDAP'tan kullanıcı bilgilerini düzgün alıp almadığını kontrol etmek içindir. Hatırlarsanız, Jamm aracılığı ile LDAP veri tabanımıza dom1.test adında bir alan adı ve bu alan adına ait user1 ve user2 adlarında 2 tane mail kullanıcı yaratmıştık. En nihayetinde elimizde "user1@dom1.test" ve "user2@dom1.test" adında 2 adet mail adresimiz vardı. Bu 2 adresin de parolaları "qazwsx" idi. Eğer Courier-authlib konfigürasyon dosyalarını düzgün ayarlamışsak, aşağıdaki komutun çıktı olarak bize user1@dom1.test kullanıcı bilgilerini dönmesi gerekmekte.

```
#> /usr/local/mailserver/courier_authlib/sbin/authtest user1@dom1.test qazwsx
```

Yukarıdaki komutu çalıştırdığımızda, aşağıdaki gibi bir çıktı ile karşılaşmamız lazım. Aşağıdaki çıktı, bize user1@dom1.test kullanıcısının kimliklendirildiğini, ev dizininin "/home/vmail/domains" olduğunu mail dizininin "/home/vmail/domains" dizini altında "dom1.test/user1" dizini olduğunu, ve parolasının "qazwsx" olduğunu söylemekte. Eğer aşağıdaki çıktı dışında bir çıktı ile karşılaşırsanız, courier-Authlib düzgün ayarlanmamış demektir.

```
Authentication succeeded.

     Authenticated: user1@dom1.test  (uid 10001, gid 10001)
    Home Directory: /home/vmail/domains
           Maildir: dom1.test/user1/
             Quota: (none)
Encrypted Password: {CRYPT}HS707TEjBwlRg
Cleartext Password: qazwsx
           Options: (none)
```

Eğer yukarıdaki senaryo anlatıldığı gibi işlemişse Courier-Authlib kurulumu ve konfigürasyonu bitmiş demektir. Artık Courier-IMAP kurulumuna geçebiliriz.

18 Courier IMAP Kurulumu

Courier-IMAP kurulumuna başlamak için aşağıdaki komut yardımı ile courier-imap paketini indirin.

```
#>wget http://prdownloads.sourceforge.net/courier/courier-imap-4.4.1.tar.bz2
```

Ardından aşağıdaki komut yardımı ile paketi açın ve paket ana dizinine geçiş yapın.

```
#>tar xvjf courier-imap-4.4.1.tar.bz2 -C /usr/local/mailserver_makedir/
```

Paketimiz /usr/local/mailserver_makedir/courier-imap-4.4.1 dizini altında açılmış durumda. Burdan itibaren kurulum adımlarımızı biraz değiştireceğiz. Courier-Authlib paket kurulumuna benzer bir kurulum yapacağız. Courier-IMAP paketi kurulumun bazı aşamalarının normal bir kullanıcı ile yapılmasını istiyor. Courier-IMAP kurulum adımlarına göre sadece "make install" komutu root olarak çalıştırılacak.

Aşağıdaki komut yardımı ile kullanici kimliğinizi "normuser" olarak değiştirin. Eğer aşağıdaki komut sonunda "$>" ile biten bir komut satırınına sahipseniz "normuser" olmuşsunuz demektir.

```
#>su - normuser
$>
```

Ama biz yine tedbiri elden bırakmayalım ve "whoami" komutu ile kim olduğumuzu öğrenelim. Eğer cevap "normuser" ise devam deebiliriz.

```
$>whoami
normuser
$>
```

Aşağıdaki komut yardımı ile /usr/local/mailserver_makedir/courier-imap-4.4.1 dizinine geçin.

```
$>cd /usr/local/mailserver_makedir/courier-imap-4.4.1
```

Ardından aşağıdaki komut yardımı ile Courier-Authlib kütüphane ve C başlık dosyalarını kabuk programının ortam değişkenlerine atayın. "configure" komutu Courier-Authlib bilgilerine bu ortam değişkenleri ile ulaşacak.

```
$>export CFLAGS="-I/usr/local/mailserver/openldap/include"
$>export CPPFLAGS="-I/usr/local/mailserver/openldap/include"
```

```
$>export LDFLAGS="-L/usr/local/mailserver/openldap/lib"
```

Courier-IMAP kurulum aşamasında Courier-authlib paketi ile gelen courierauthconfig programına ihtiyaç duyar. Bu programın tam yolunu aşağıdaki gibi Kabuk ortam değişkenlerine atamamız gerekmektedir.

```
$> export
COURIERAUTHCONFIG=/usr/local/mailserver/courier_authlib/bin/courierauthconfig
```

Ortam değişkenlerini atadıktan sonra “configure” komutunu çalıştırabiliriz. Aşağıdaki komut yardımı ile “configure” komutunu çalıştırın.

```
$>./configure --prefix=/usr/local/mailserver/courier_imap --enable-workarounds-
for-imap-client-bugs  --enable-unicode --with-trashquota && make
```

“configure” komutu sonlandığında aşağıdaki komutu çalıştırarak derleme işlemine başlayın.

```
$>make
```

Derleme işlemi sonlandığında, artık “normuser” kullanıcısından çıkabilirsiniz. “normuser” ile yapacaklarımız bitmiş durumda. Bundan sonra, işlemlerimizi tekrar “root” olarak yapacağız. Aşağıdaki komut yardımı ile “normuser” kullanıcısından çıkın.

```
$>exit
```

Komut satırından “whoami” yazın eğer cevap olarak “root” alıyorsanız, tekrar “root” olmuşsunuz demektir. Root kullanıcısı ile tekrar /usr/local/mailserver_makedir/courier-imap-4.4.1 diznine geçin. Aşağıdaki komut bu işlemi yapacaktır.

```
$>cd /usr/local/mailserver_makedir/courier-imap-4.4.1
```

Ve son olarak aşağıdaki komutu çalıştırarak, Courier-IMAP kurulumunu tamamlayın. Courier-IMAP paketinin /usr/local/mailserver/courier_imap dizini altında kurulmuş olması gerekli.

```
#>make install
```

18.1 Courier-IMAP Konfigürasyonu

Courier-IMAP konfigürasyonu diğer paket konfigürasyonlarına göre çok kolaydır.

Courier-IMAP'in tek 2 ayar dosyası vardır, bu dosyalar aşağıdaki gibidir.

- /usr/local/mailserver/courier_imap/etc/imapd
- /usr/local/mailserver/courier_imap/etc/imapd-ssl

```
ama aşağıdaki komutlar yardımı ile /usr/local/mailserver/courier_imap/etc/
dizinine geçip dosyaları listelediğinizde yukarıdaki iki dosyayı
bulamayacaksınız.
```

```
#>cd /usr/local/mailserver/courier_imap/etc/
#>ls
```

Onun yerine aşağıdaki iki dosyayı bulacaksınız.

- /usr/local/mailserver/courier_imap/etc/imapd.dist
- /usr/local/mailserver/courier_imap/etc/imapd-ssl.dist

İlk yapmamız gereken yukarıdaki dosyaları kullanarak ayar dosyalarımızı yaratmak. Bunun için aşağıdaki komutları çalıştırın.

```
#>cp  /usr/local/mailserver/courier_imap/etc/imapd.dist /usr/local/mailserver/courier_imap/etc/imapd
#>cp /usr/local/mailserver/courier_imap/etc/imapd-ssl.dist /usr/local/mailserver/courier_imap/etc/imapd-ssl
```

Şimdi Courier-IMAP'i ayarlayabiliriz. Courier-IMAP2in ana ayar dosyası “imapd” dosyasıdır. Burada sadece bir değişkeni değiştireceğiz ve ayarlamamız bitecek.

“imapd” dosyasını açın ve MAILDIRPATH ile başlayan satırı bulun ve bu satırı aşağıdaki gibi değiştirin. Eğer MAILDIRPATH ile başlayan bir satır mevcut değilse, en son satıra aşağıdaki içeriği ekleyin.

```
MAILDIRPATH=Maildir
```

Courier-IMAP ayarlamaları bitmiş durumda. Artık Courier-IMAP'i çalıştırabiliriz.

18.2 Courier-IMAP'in Calıştırılması

Courier-IMAP aşağıdaki komut yardımı ile çalıştırılır.

```
#> /usr/local/mailserver/courier_imap/libexec/imapd.rc start
```

Ve aşağıdaki komut yardımı ile durdurulur.

```
#> /usr/local/mailserver/courier_imap/libexec/imapd.rc stop
```

Courier-IMAP'in çalışıp çalışmadığını kontrol etmek istiyorsanız, komut satırında “netstat -ltnp” yazın. Bu komutun çıktısından aşağıdakine benzer bir satır olmalı.

```
tcp6       0      0 :::143                  :::*                    LISTEN      2352/couriertcpd
```

Eğer yukarıdaki satırı görüyorsanız, Courier-IMAP çalışıyor demektir. Şimdi test aşamasına geçelim.

18.3 Courier-Authlib ve Courier-IMAP testi

Teste başalamdan önce, OpenLDAP, Courier-Authlibve Courier-IMAP'in çalışıyor olduğunu teyyit edin. Bunun nasıl yapılacağını daha önce açıklamıştık.

Eğer bu 3 program çalışıyorsa, aşağıdaki komut ile courier-authlib'in çalışıp çalışmadığını test edin. Bu komut courier-authdaemond'nin LDAP'tan kullanıcı bilgilerini düzgün alıp almadığını kontrol etmek içindir. Hatırlarsanız, Jamm aracılığı ile LDAP veri tabanımıza dom1.test adında bir alan adı ve bu alan adına ait user1 ve user2 adlarında 2 tane mail kullanıcı yaratmıştık. En nihayetinde elimizde “user1@dom1.test” ve “user2@dom1.test” adında 2 adet mail adresimiz vardı. Bu 2 adresin de parolaları “qazwsx” idi. Eğer Courier-authlib konfigürasyon dosyalarını düzgün ayarlamışsak, aşağıdaki komutun çıktı olarak bize user1@dom1.test kullanıcı bilgilerini dönmesi gerekmekte.

```
#> /usr/local/mailserver/courier_authlib/sbin/authtest user1@dom1.test qazwsx
```

Yukarıdaki komutu çalıştırdığımızda, aşağıdaki gibi bir çıktı ile karşılaşmamız lazım. Aşağıdaki çıktı, bize user1@dom1.test kullanıcısının kimliklendirildiğini, ev dizininin “/home/vmail/domains” olduğunu mail dizininin “/home/vmail/domains” dizini altında “dom1.test/user1” dizini olduğunu, ve parolasının “qazwsx” olduğunu söylemekte. Eğer aşağıdaki çıktı dışında bir çıktı ile karşılaşırsanız, courier-Authlib düzgün ayarlanmamış demektir.

```
Authentication succeeded.

     Authenticated: user1@dom1.test  (uid 10001, gid 10001)
    Home Directory: /home/vmail/domains
           Maildir: dom1.test/user1/
             Quota: (none)
Encrypted Password: {CRYPT}HS707TEjBwlRg
Cleartext Password: qazwsx
           Options: (none)
```

18.4 Courier-IMAP Testi

Courier-Atuhlib testi bitti. Şimdi sıra Courier-IMAP testinde. IMAP testi için aşağıdaki komut ile IMAP sunucusuna bağlanın. Imap sunucusunun dinlediği port 143'tür, bu sebepten telnet yardımı ile

makinemizin 143 nolu portuna bağlanıyoruz.

```
#> telnet localhost 143
```

Yukarıdaki komutu yazıp enter'a tıkladıktan sonra aşağıdaki gibi bir çıktı ile karşılaşmalıyız. “* OK” ile başlayan satır her ne kadar yukarıda 4 satıra sığmış ise de, siz konsolunuzda bunu tek satır olarak göreceksiniz. Eğer “* OK” ile başlayan bir satır ile karşılaşmadı iseniz IMAP ayarlarında bir problem var demektir.

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
libshishi: warning: /root/.shishi/tickets: No such file or directory
* OK [CAPABILITY IMAP4rev1 UIDPLUS CHILDREN NAMESPACE
THREAD=ORDEREDSUBJECT THREAD=REFERENCES SORT QUOTA IDLE ACL
ACL2=UNION] Courier-IMAP ready. Copyright 1998-2005 Double Precision, Inc.  See
COPYING for distribution information.
```

Eğer yukarıdaki gibi “* OK” satırını görmüşseniz kursorün yanıp söndüğü yerden başlayarak ekrana aşağıdaki satırı yazın. Bu satır imap sunucusuna user1@dom1.test kullanıcı olarak Login olmak istediğinizi belirtir.

```
10 login user2@dom1.test qazwsx
```

Cevap olarak aşağıdaki gibi bir çıktı almalısınız.

```
10 OK LOGIN Ok.
```

Eğer yukarıdaki gibi bir çıktı almıyorsanız, ya courier-authlib ya da courier-imap ayarlarında bir problem var demektir. Ya da user2@dom1.test kullanıcısının mail dizini /home/vmail/domains/dom1.test/user2 dizini yaratılmamış demektir.

/home/vmail/domains/dom1.test/user2 dizininin olmayışı konusu çok önemli. Çünkü yukarıdaki testi user1@dom1.test ile yaparsanız, hata alırsınız. Çünkü user1@dom1.test kullanıcısının henüz dizini yaratılmamıştır. Yani /home/vmail/domains/dom1.test/user1 dizini henüz mevcut değiltir.

Peki niye user2@dom1.test kullanıcısının mail dizini var da user1@dom1.test kullanıcısının mail dizini yok. Çünkü, hatırlarsanız telnet aracılığı ile user1@dom1.test kullanıcısından user2@dom1.test kullanıcısına mail attık. Mail postfix'e ulaştığında, postfix maili

/home/vmail/domains/dom1.test/user2 dizinine bırakmak istedi ama öyle bir dizin mevcut değildi. Dolayısı ile postfix ilgili dizini yaratttı, ardından o dizin altında Maildir yapısını oluşturdu ve maili / home/vmail/domains/dom1.test/user2/new dizini altına bıraktı. Ama gönderici adres olan user1@dom1.test için böyle bir işlem yapmadı. Gerek de yoktu. Bu durum, mail sunucumuzda herhangi bir probleme yol açmaz, ama imap ile kimliklendirme yapmak istersek her kullanıcının mail dizininin olması gerekir. Bu da şu demek: her kullanıcıya ilk yaratıldığında bir mail atalım ki mail dizinleri yaratılsın. Bu Mail yönetimi açısından da iyi bir hareket olur. Her kullanıcıya, ilk yaratıldığında "Hoşgeldin" mesajı göndermiş oluruz.

İsterseniz user2@dom1.test ile yaptığım testi bir de user1@dom1.test ile yapın. Bu sefer "OK Login Ok." cevabı almamanız gerekiyor. Onun yerine aşağıdaki gibi bir cevap alacaksınız.

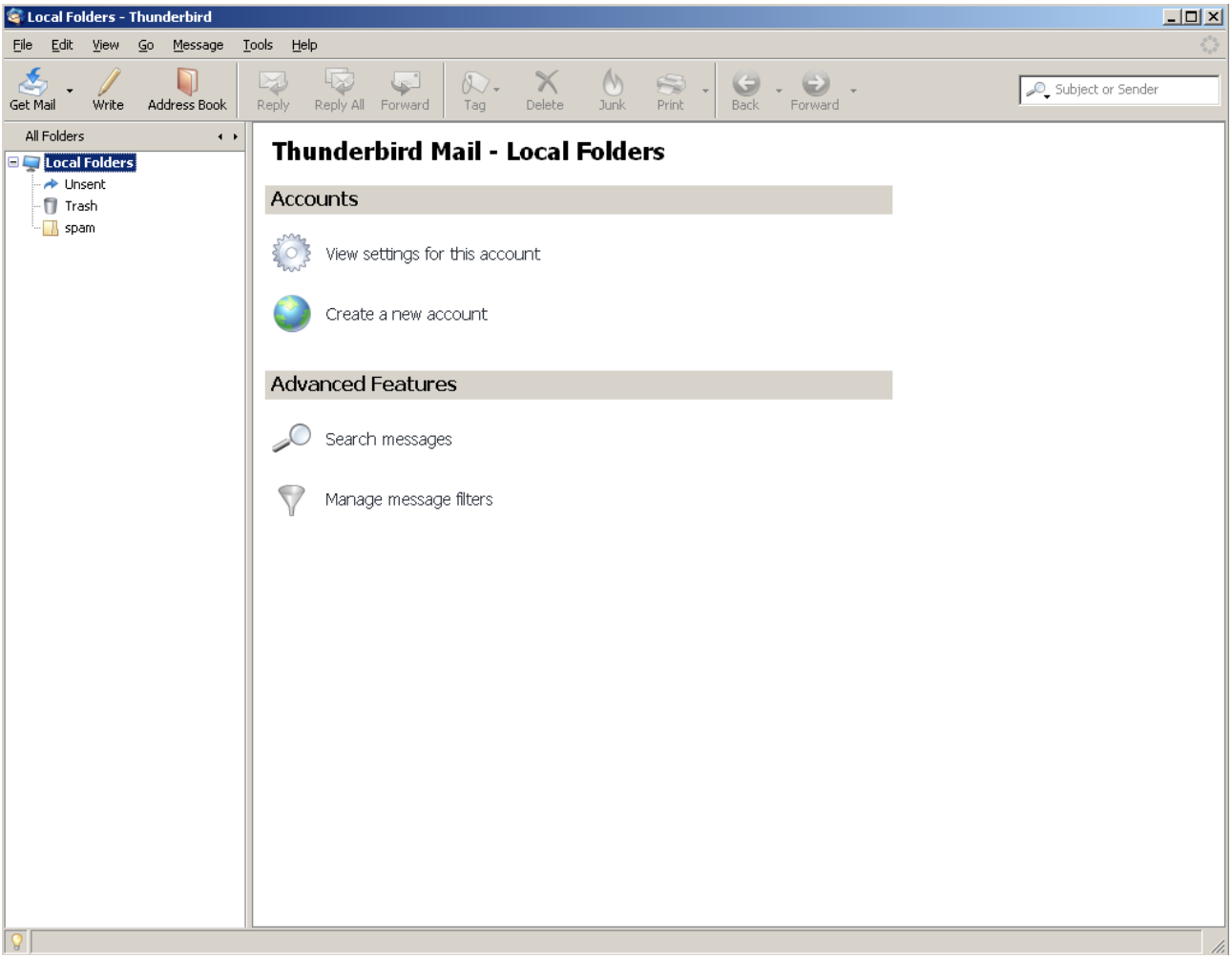
```
10 NO Login failed.
```

Şu an itibari ile testler bitmiş durumda. Courier-Authlib ve Courier-IMAP çalışıyor. Ama isterseniz bir de outlook veya Thunderbird gibi modern Mail istermcileri ile test edelim. En nihayetinde kullanıcılarımız, Thunderbird veya Outlook ile maillerini okuyacaklar.

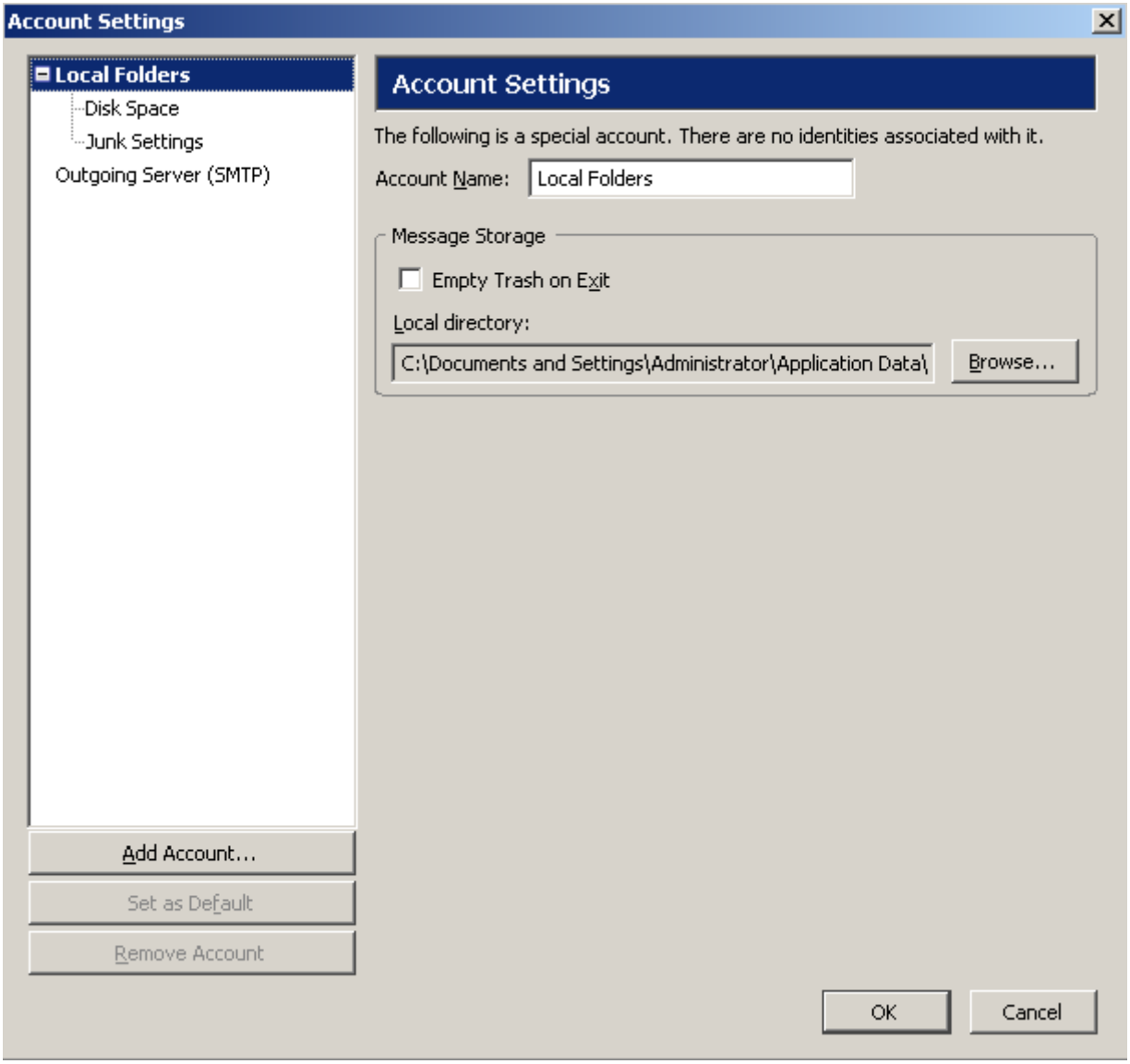
Test için, Thunderbird ve Outlook arasından Thunderbird'ü seçtik. Aşağıdaki adresten thunderbird'ü indirip kurun.

```
http://www.mozilla.com/en-US/thunderbird/
```

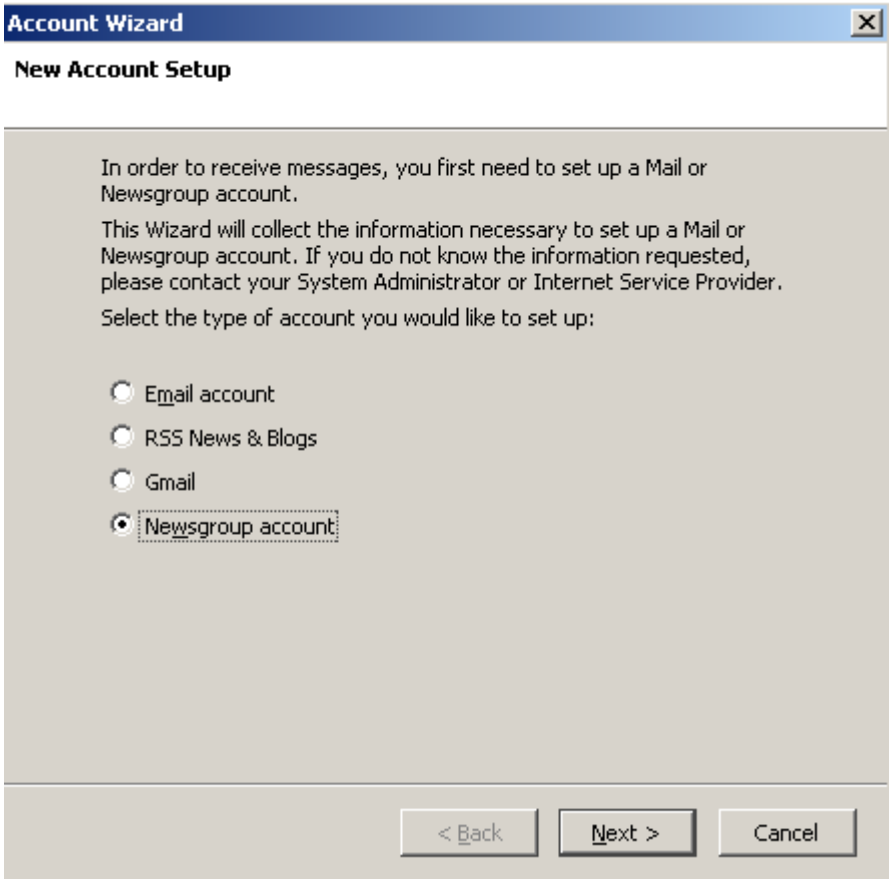
Ardından, thunderbird'ü çalıştırın, aşağıdaki gibi bir ekranla karşılaşmanız gerekmekte.



Sol alanda üst menüde Tools -> Account Settings... seçeneğine tıklayın. Aşağıdaki gibi bir ekranla karşılaşmanız gerekmekte. Bu ekranda, "Add Account..." düğmesine tıklayın.

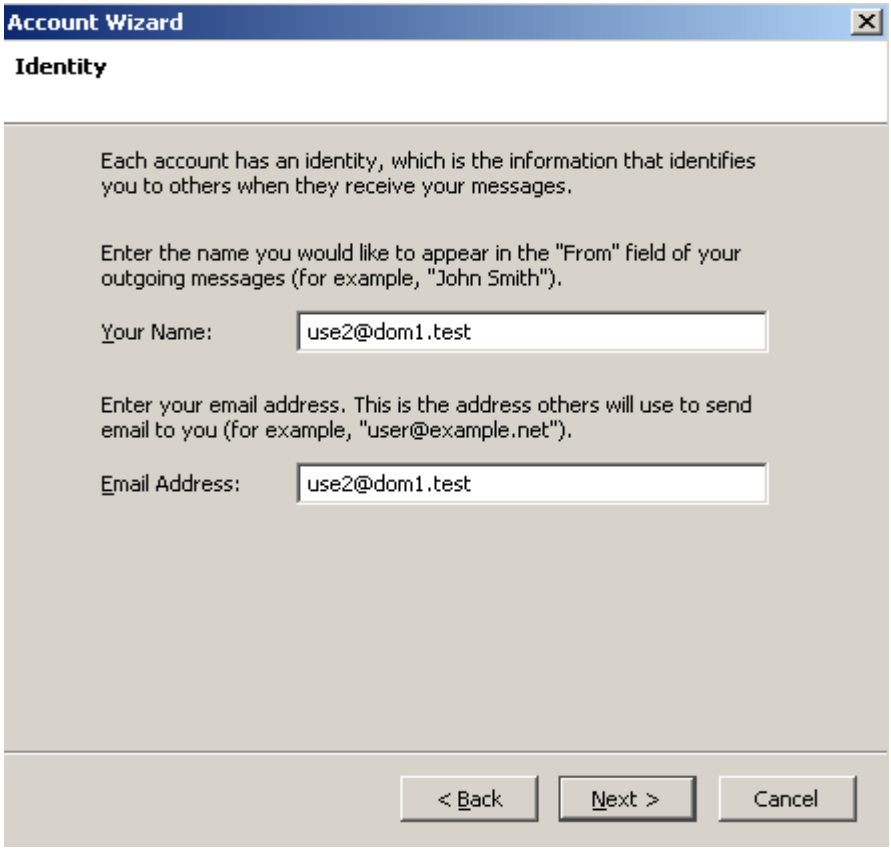


"Add Account..." düğmesine tıkayınca aşağıdaki gibi bir ekran ile karşılaşmalısınız.
Bu ekranda "Email Account" seçeneğini işaretleyip, "next" düğmesine tıklayınız.

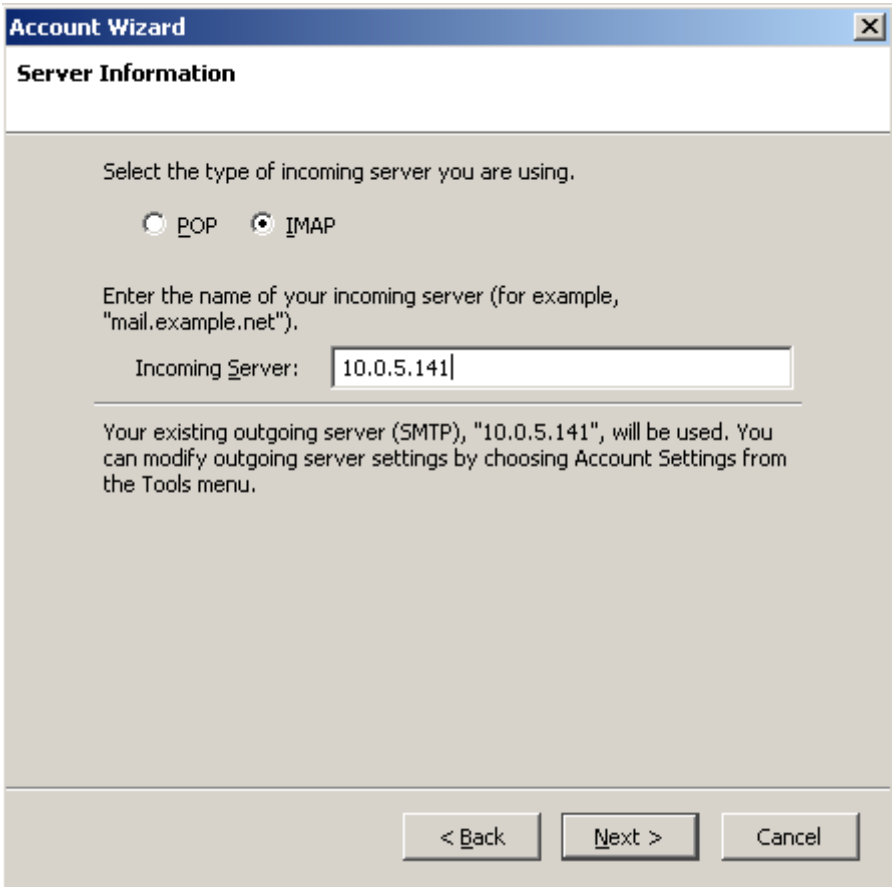


"Next" düğmesine tıkladıktan sonra aşağıdaki gibi bir ekranla karşılaşmanız gerekmekte. Bu ekranda kullanıcı adı ve Mail adresini ekrandaki gibi giriniz. Ardından "next" düğmesine tıklayınız.

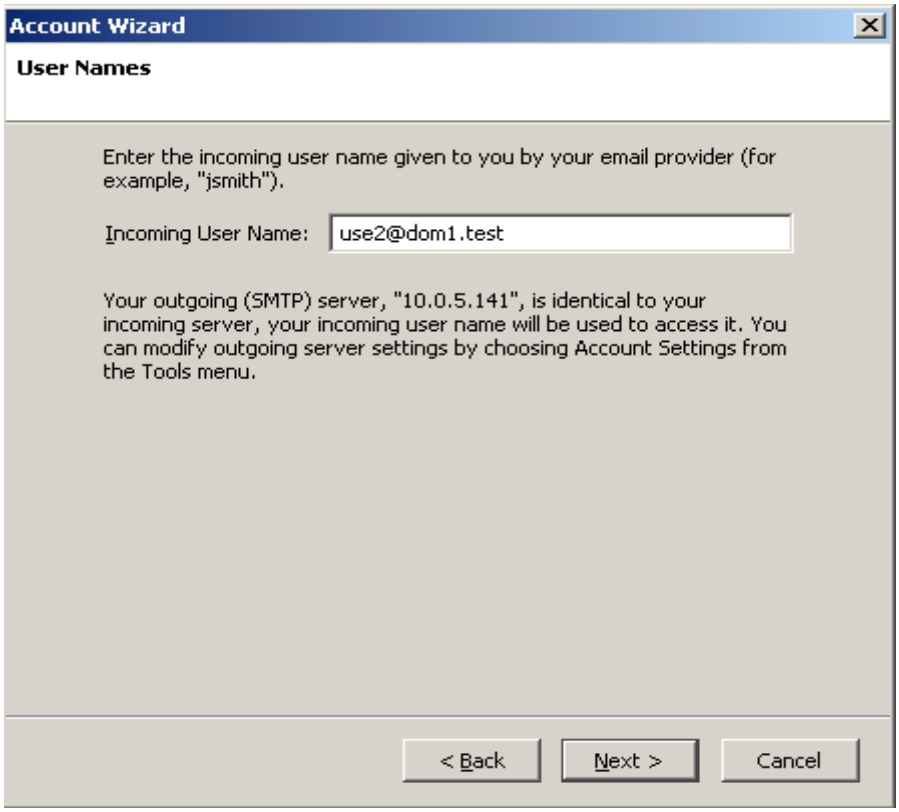
Dikkat edersek user2@dom1.test adresini tanımlıyoruz. Çünkü şu anda user1@dom1.test kullanıcmızın imap sunucusuna login olamayacağını biliyoruz. Çünkü henüz bu kullanıcının mail dizini yaratılmış durumda değil.



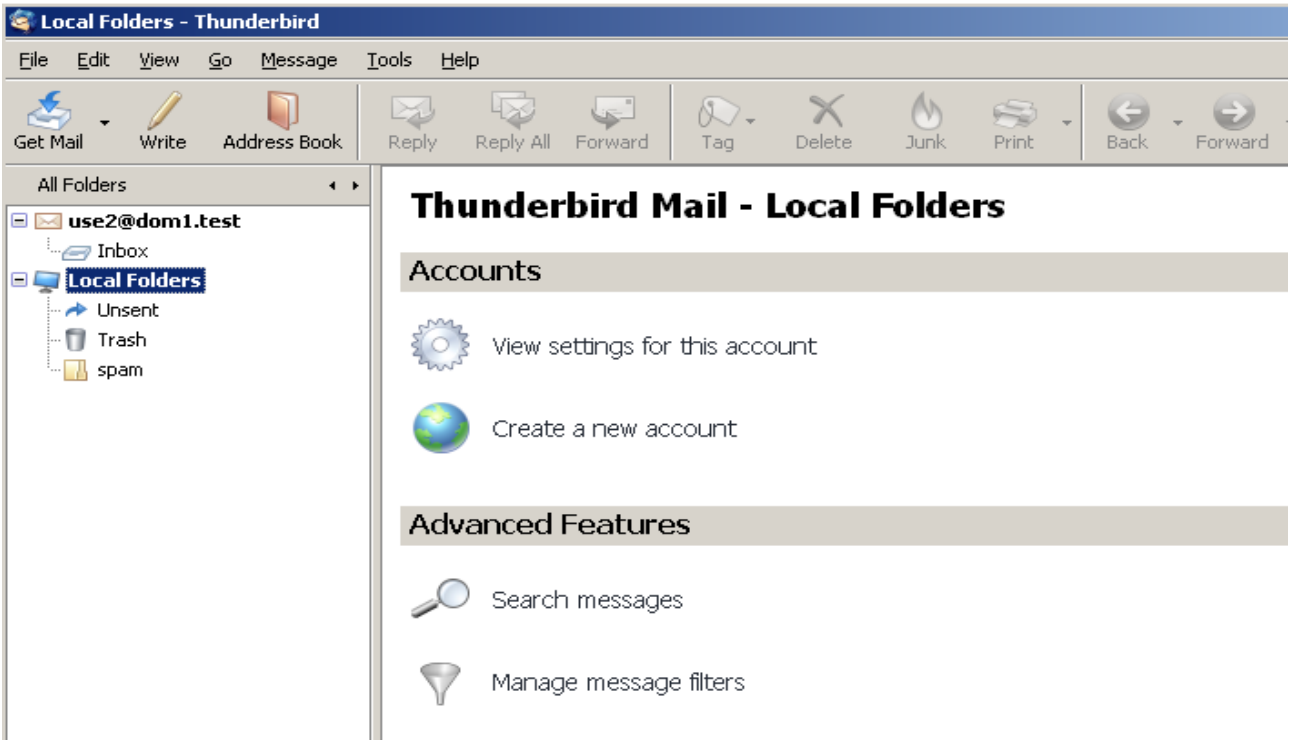
"Next" düğmesine tıkladıktan sonra aşağıdaki gibi bir ekran ile karşılaşmamız gerekmekte. Bu ekranda, POP3 ve IMAP arasından IMAP seçeneğini seçin ve "Incoming Server" kısmına makinenizin IP'sini yazın ve "Next" düğmesine tıklayın.



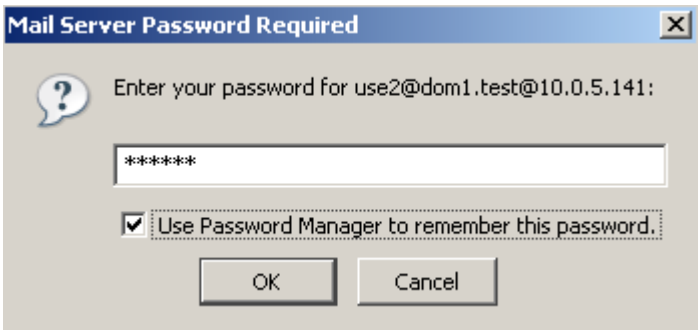
"Next" düğmesine tıklayınca aşağıdaki gibi bir ekran ile karşılaşmalısınız. Bu ekranda da "Incoming User Name" girdisine "user2@dom1.test" adresini yazıp "next" düğmesine tıklayın.



"Next" düğmesine tıkladıktan sonra geriye 2 ekran işe daha karşılaşacaksınız. Bu ekranlarda sırası ile "Next" ve "Finish" düğmelerine tıklayıp "user2@dom1.test" kullanıcı adını tanımlama işini sonlandırın. En sonunda aşağıdaki gibi sol listenizde user2@dom1.test kullanıcısının tanımlı olduğunu görmelisiniz.



Sol listede “user2@dom1.test” seçeneği altındaki “Inbox” ikonuna tıkladığınızda, Thunderbird IMAP sunucumuz olan 10.0.5.141 makinesine bağlanmak isteyecektir. Bağlanma esnasında user2@dom1.test kullanıcısını sisteme login etmeye çalışacaktır. Bu sebeple aşağıdaki gibi önünüze bir parola sorma ekranı gelecektir. “user2@dom1.test” için parolayı girin, isterseniz alttaki “User Password Manager to remember this password.” seçeneğini de işaretleyin. Bu her seferinde parola girmenizi engeller ve son olarak OK düğmesine basın.



OK düğmesine tıkladıktan sonra, user1@dom1.test'ten gelen maili ekranınızda görmelisiniz. Şimdi yapmanız gereken, user1@dom1.test kullanıcısna bir mail atmak. Bunu yapmamızın amacı, user1@dom1.test kullanıcısının mail dizin yapısını oluşturmak. Hatırlarsanız, user2@dom1.test kullanıcısının mail dizini oluşmuşken, user1@dom1.test kullanıcısının dizin yapısı henüz oluşmamıştı, kendisine mail atmadığımız için.

Böylelikle Courier-authlib ve Courier-IMAP paketlerinin testlerini tamamlamış oluyoruz.

19 Maildrop

Maildrop daha önce belirttiğimiz gibi bir MDA(Message Delivery Agent, Mesaj Ulaştırma Birimi) aracıdır. MDA birimi, postfix maili elde edip kullanıcı dizinine ilettiği anda devreye girer. Yani Postfix kuyruğundaki maili MDA'ya iletir, MDA mail bilgilerine bakarak mailin hangi kullanıcıya ait olduğunu, dolayısı ile hangi dizine yerleştireceğini anlar. Bu bilgileri elde ettikten sonra MDA maili ilgili dizine bırakır.

Burda dikkat etmemiz gereken bir nokta daha var. MDA birimi, daha özelde maildrop programı, mail bilgilerinden mailin hangi dizine gönderileceğini nasıl öğrenecek?

Daha önce courier-authlib testini yaparken “authtest” programını kullanmıştık ve authtest ile aşağıdaki komutu çalıştırdığımızda,

```
#>authtest user1@dom1.test qazwsx
```

aşağıdaki gibi bir çıktı almıştık. Bu çıktıda mail kullanıcısının dizin bilgisi mevcut.

```
Authentication succeeded.

     Authenticated: user1@dom1.test  (uid 10001, gid 10001)
    Home Directory: /home/vmail/domains
           Maildir: dom1.test/user1/
             Quota: (none)
Encrypted Password: {CRYPT}HS707TEjBwlRg
Cleartext Password: qazwsx
           Options: (none)
```

Maildrop programının da aynı işlevselliğe ihtiyacı var. Maildrop da elindeki mail içeriğini kullanarak mail alıcısının mail dizinini elde etmek durumunda. Madem ki bu bilgi "authtest" programı ile elde edebiliyor maildrop programını bu sisteme entegre etmemiz yeterli olacaktır.

Courier-Authib sistemini OpenLDAP'a entegre etmiştik. Courier-authlib ile gelen authtest programı Courier-authlib kütüphanelerini kullanarak ldap'ta saklı bilgilere ulaşabiliyor. Biz de eğer maildrop'u kurarken courier-auhtlib kütüphanesi ile entegre edersek, maildrop programı da bu kütüphaneleri kullanarak mail kullanıcısının dizin yapısını elde edebilir ve eline ulaşan maili hedef dizine yerleştirebilir.

Maildrop'u kurarken yapacağımız şey de tamamen budur. Madilrop'u courier-authlib ile entegre etmek. Böylelikle Maildrop'un LDAP'ta saklı mail dizini bilgilerine ulaşmasını sağlamak.

Artık maildrop kurulumuna başlayabiliriz.

Maildrop Kurulumu

Maildrop, kurulum aşamasında bazı paketlere ihityaç duyar. Bu paketler

- libpcre3-dev
- g++

paketleridir. "g++", C++ dilini derleme programıdır. Maildrop'un bir kısmı C++ ile yazılmış durumda, bu sebeple g++ paketine ihtiyacımız var..

Kuruluma maildrop paketini indirerek başlayabiliriz. Aşağıdaki komut bu işlevi yerine getirecektir.

```
#>wget http://prdownloads.sourceforge.net/courier/maildrop-2.0.4.tar.bz2
```

Ardından aşağıdaki komut yardımı ile paketi /usr/local/mailserver_makedir dizinine açın ve paket ana dizinine geçiş yapın.

```
#>tar xvjf maildrop-2.0.4.tar.bz2 -C /usr/local/mailserver_makedir
#>cd /usr/local/mailserver_makedir/maildrop-2.0.4/
```

Configure betiğini çalıştırmadan önce aşağıdaki komutları çalıştırarak, Courier-Authlib kütüphane ve C başlık dosyası bilgilerini kabuk programı değişkenlerine atayın. Böylelikle Configure script'i çalıştığında Courier-Authlib ile ilgili bilgilere ulaşabilecek Eğer bu işlemi yapmazsak, maildrop paketimiz courier-Authlib paketi ile entegre olmaz.

```
#>export export PATH=$PATH:/usr/local/mailserver/courier_authlib/bin:.

#>export LDFLAGS="-L/usr/local/mailserver/courier_authlib/lib
-L/usr/local/mailserver/courier_authlib/lib/courier-authlib"

#>export CPPFLAGS="-I/usr/local/mailserver/courier_authlib/include"
#>export CFLAGS="-I/usr/local/mailserver/courier_authlib/include"
```

Ve aşağıdaki komutlar ile kurulumu gerçekleştirin.

```
#>./configure --prefix=/usr/local/mailserver/maildrop --enable-authlib
--enable-maildirquota --enable-maildrop-uid=vmail --enable-maildrop-gid=vmail
--with-trashquota --with-db=db --enable-authlib-tempreject=1 --with-
etcdir=/usr/local/mailserver/maildrop/etc --enable-trusted-users='vmail postfix
root' --enable-syslog=1

#>make
#>make install
```

Yukarıdaki komutlar bittiğinde “/usr/local/mailserver/maildrop” dizini altında maildrop programının kurulu olması gerekiyor.

Maildrop kurulumu bitmiş durumda, ama henüz Courier-Authlib ile düzgün entegre olup olmadığını bilmiyoruz. Komut satırında “/usr/local/mailserver/maildrop/bin/maildrop -v” yazın. Bu komutun çıktısı aşağıdaki gibi olmalıdır.

```
maildrop 2.0.4 Copyright 1998-2005 Double Precision, Inc.
GDBM extensions enabled.
Courier Authentication Library extension enabled.
Enabled Berkeley DB instead of GDBM extensions.
Maildir quota extension enabled.
This program is distributed under the terms of the GNU General Public
License. See COPYING for additional information.
```

Özellikle “*Courier Authentication Library extension enabled.*” satırına dikkat edin. Eğer böyle bir satır görüyorsanız, maildrop Courier-Authlib ile entegre durumda demektir. Eğer böyle bir satır yok ise, maildrop kurulumu gerçekleşmemiş anlamına geliyor.

Artık maildrop konfigürasyonuna geçebiliriz.

19.1 Maildrop Konfigürasyonu

Öncelikle Maildrop ve Postfix'i entegre edelim. Entegrasyon için /etc/postfix/master.cf dosyasını açın ardından dosyanın sonuna aşağıdaki içerği ekleyin ve kaydedin.

```
maildrop unix - n n - - pipe
  flags=DRhu user=vmail argv=/usr/local/mailserver/maildrop/bin/maildrop -d $
{recipient}
```

Bir sonraki adım olarak, /etc/postfix/main.cf dosyasını açın ve mailbox_command ile başlayan satırı bulun. Aşağıdaki gibi bir satır olmalı.

```
mailbox_command = /usr/bin/procmail
```

Yukarıdaki satırı aşağıdaki gibi değiştirin.

```
mailbox_command = /usr/local/mailserver/maildrop/bin/maildrop
```

Ve aşağıdaki içeriği main.cf dosyasının sonuna ekleyin.

```
maildrop_destination_recipient_limit = 1
maildrop_destination_concurrency_limit = 1
owner_request_special = no
```

Şu an itibari ile maildrop ve Postfix entegrasyonu bitmiş durumda. Artık MDA olarak procmail değil maildrop kullanıyoruz. Her ne kadar maildrop'un procmail'e birçok üstünlüğü olsa da procmailin güzel bir yanı vardı. O da, hatırlarsanız "user2@dom1.test" kullanıcısına ilk mailimizi attığımızda /home/vmail/domains dizini altında dom1.test ve dom1.test/user2 dizinlerini otomatik oluşturması ve dom1.test/user2 dizini altına Maildir yapısını yerleştirmesi idi. Bunu sağlayan procmail idi. Maalesef maildrop'un böyle bir özelliği yok. Eğer yeni bir kullanıcı yaratmışsak, bu yeni kullanıcının dizinini ve maildir yapısın manuel olarak oluşturmamız gerekmekte.

İsterseniz şimdi yeni bir mail kullanıcısı ve Maildir yapısı yaratalım. Mail kullanıcımızın adı: user3@dom1.test olsun. Bu mail kullanıcısının dizini /home/vmail/domains/dom1.test/user3 olması gerekmekte. Aşağıdaki komut ile bu dizini yaratın. Bu komut aynı zamanda Maildir yapısını da yaratacaktır.

```
#>/usr/local/mailserver/maildrop/bin/maildirmake /home/vmail/domains/dom1.test/
user3
```

Yalnız yarattığımız dizinlerin yazma ve okuma hakları "vmail" kullanıcısına ait olmak zorunda. Çünkü maildrop kullanıcısı çalışırken, "vmail" kullanıcısının haklarını kullanarak çalışır. Eğer "vmail" kullanıcısı yukarıda yarattığımız dizinlere yazamaz ise maildrop gelen mailleri mail

kullanıcı dizinine bırakamaz. Dolayısı ile aşağıdaki komut yardımı ile bu problemi aşalım.

```
#>chown -R vmail:vmail /home/vmail/domains/dom1.test/user3
```

Şu an itibari ile dizin yapımız oluşmuş durumda, ama henüz işimiz bitmedi. LDAP veri tabanına user3@dom1.test kaydını yaptırmamız gerekiyor. Daha önce yaptığınız gibi Jamm Web arayüzünden bu mail kullanıcı kaydını yapabilirsiniz.
Böylelikle maildrop ayarlarını tamamen bitirmiş oluyoruz. Şimdi test zamanı.

19.2 Maildrop Testi

Maildrop, mailleri "/usr/local/mailserver/maildrop/bin/maildrop" komutu ile kullanıcı dizinine ulaştırır. Eğer bu komut düzgün bir şekilde maili kullanıcı dizinine ulaştırabiliyorsa, maildrop testimiz başarı ile tamamlanmış demektir. Şimdi, user3@dom1.test kullanıcsıına maildrop ile bir mail atalım. Bunun için aşağıdaki komut ile mail atma işlemine başlayın.

```
#> /usr/local/mailserver/maildrop/bin/maildrop -V 4 -d user3@dom1.test
```

Yukarıdaki komutun çıktısı aşağıdaki gibi olmalıdır. "maildrop: authlib" ile başlayan satırlara dikkat edin. Bu satırlar, maildrop'un courier-authlib ile düzgün bir şekilde iletişebildiğini ve user3@dom1.test kullanıcısının dizin bilgilerine düzgün bir şekilde ulaşabildiğini göstermekte.

```
maildrop: authlib: groupid=10001
maildrop: authlib: userid=10001
maildrop: authlib: logname=user3@dom1.test, home=/home/vmail/domains,
mail=dom1.test/user3/
maildrop: Changing to /home/vmail/domains
```

Kursor'ün bulunduğu yerden itibaren bir mesaj yazın ve "ctrl+d" tuş kombinasyonu ile işlemi sonlandırın. Maildrop testinin tüm adımları aşağıdaki gibidir.

```
#> /usr/local/mailserver/maildrop/bin/maildrop -V 4 -d user3@dom1.test
maildrop: authlib: groupid=10001
maildrop: authlib: userid=10001
maildrop: authlib: logname=user3@dom1.test, home=/home/vmail/domains,
mail=dom1.test/user3/
maildrop: Changing to /home/vmail/domains

al sana mail user3
Message start at 1 bytes, envelope sender=user3@dom1.test
maildrop: Attempting .mailfilter
#>
```

Mailin user3@dom1.test kullanıcısına ulaşıp ulaşmadığını görmek için

/home/vmail/domains/dom1.test/user3/new dizinine bakın. Eğer orada garip isimde bir dosya duruyorsa, mailiniz ulaşmış demektir. İsterseniz bu garip isimli dosyayı açıp bakabilirsiniz.

Böylelikle Maildrop kurulumu tamamen bitmiş oluyor.

20 Otomatik Cevap Sistemi

Mail sunucumuz neredeyse tamamlandı. Artık mail alıp gönderebiliyoruz. Gelen mailleri spam ve virüs kontrollerinde geçirebiliyoruz. Peki kulanıclarımızın bazıları, izin günlerinde kendilerine ulaşan maillere otomatik cevap yazmak istiyorlarsa, kendilerinin o an maillere cevap yazamayacağını gönderen kişiye haber vermek istiyorlarsa ne yapacağız. Tabii ki Postfix'de bunun için de çözüm var. “Otomatik Cevap Sistemi” kuracağız.

Postfix ve LDAP entegrasyonu birlikte çalışan gnarwl paketi bizim ihtiyacımızı karşılayacaktır. Gnarwl ldap ve postfix ile entegre olabilen bir sistem. LDAP kısmı için gnarwl ile birlikte gelen ldap şemasını openldap sistemimize entegre edeceğiz. Gnarwl LDAP şemasını kendi ldap sistemimize entegre ettiğimizde, her bir mail kullanıcısı için “o an izinde olup olmadığına” dair bir bilgi tutmuş olacağız. Bu bir çeşit işaretçi olacak. Eğer mail kullanıcımız tatile çıkmış ise, LDAP sistemimizde mail kullanıcı bilgilerine ulaşıp, gnarwl ile gelen “o an izinde olup olmadığı” bilgisini tutan işaretçiyi aktif yapacağız. Eğer aktif ise, o mail kullanıcısı izinde demektir.

İzinde olduğunu söyledik, bir de izin mesajını tutmamız lazım. Yine gnarwl ile gelen ldap şeması bir “izin metni” tutmamıza olanak sağlıyor. Yine mail kullanıcısının gnarwl şeması yardımı ile izinde olduğunu belirttiğimizde beraberinde bir “izin metni” doldurabiliyoruz. Mail kullanıcısı izinde olduğu süre içerisinde, eğer kendisine bir mail ulaşırsa, gnarwl sistemi o “izin metni”ni gönderen kişiye mail olarak atacak.

Böyle bir mekanizma ile “Otomatik Cevap Sistemi”'i gerçekleştirmiş oluyoruz.

Şimdi isterseniz kuruluma geçelim.

Gnarwl Kurulumu

Gnarwl paketi kurulum aşamasında bir pakete ihtiyaç duyuyor. Bu paket libgdbm-dev paketidir. Bu paketi aşağıdaki komut yardımı ile sistemimize yükleyin.

```
#>aptitude install libgdbm-dev
```

Ardından aşağıdaki komut yardımı ile gnarwl paketini indiriniz.

```
#>wget http://www.onyxbits.de/sites/default/files/gnarwl-3.4.tgz
```

İndirme işlemi tamamlandıktan sonra, aşağıdaki komutları çalıştırarak gnarwl paketinin ana

dizinine geçin.

```
#>tar xvzf gnarwl-3.4.tgz -C /usr/local/mailserver_makedir/
#>cd /usr/local/mailserver_makedir/gnarwl-3.4/
```

Daha önce belirttiğimiz gibi, gnarwl OpenLDAP sistemimiz ile entegre çalışacak. Bu sebeple gnarwl'ın kurulum aşamasında OpenLDAP kütüphanelerinin ve C başlık dosyalarının nerede olduğunu bilmesi gerekir. Aşağıdaki komutlar yardımı ile bu bilgileri kabuk programının ortam değişkenlerine yükleyin. “configure” komutu çalışırken bu değişkenler yardımı ile OpenLDAP bilgilerine ulaşacaktır.

```
#>export CFLAGS="-I/usr/local/mailserver/openldap/include"
#>export CPPFLAGS="-I/usr/local/mailserver/openldap/include"
#>export LDFLAGS="-L/usr/local/mailserver/openldap/lib"
```

Artık kurulum işlemine başlayabiliriz. Ama bu kurulum diğer kurulumlardan biraz farklı olacak, çünkü “configure” komutu kendisinden beklenen gibi düzgün çalışmıyor. “configure” komutu Makefile dosyasını düzgün üretmekle mükellef. Ama maalesef gnarwl paketi ile gelen “configure” paketi bu işi layıkıyla yerine getiremiyor. Ama yine de “configure” komutunu çalıştırmamız gerekiyor. Ardından, “configure” komutunun Makefile dosyasında ilgili değişiklikleri yapamadığı kısımları manuel olarak yapacağız.

Kurulum işlemine aşağıdaki komut ile başlayabilirsiniz.

```
#>./configure --with-ldap=/usr/local/mailserver/openldap
```

“configure” komutu sonlandıktan sonra Makefile dosyasını değiştirmeye başlayabiliriz. Öncelikle Makefile dosyasını edit edin ve “CFLAGS=” ile başlayan satırı bulun. Bu satırın sonuna "-I/usr/local/mailserver/openldap/include" parametresini ekleyin.(tırnak karakterlerini silipparametreyi yerleştirin)

Yine aynı şekilde “LDFLAGS=” ile başlayan satırı bulun. Bu satırın sonuna "-L/usr/local/mailserver/openldap/lib" parametresini ekleyin.(tırnak karakterlerini silip parametreyi yerleştirin)

Ardından dosyayı kaydedin ve aşağıdaki komutları çalıştırarak kurulumu tamamlayın.,

```
#>make
#>make install
#>make perm
```

Kurulum bitti. şimdi Sıra konfigürasyonda.

20.1 Gnarwl Konfigürasyonu

Gnarwl programının tek bir konfigürasyon dosyası vardır. Bu konfigürasyon dosyası, /usr/local/etc/gnarwl.cfg dosyasıdır.

Bu dosyayı aşağıdaki komut ile yedekleyin.

```
#>mv /usr/local/etc/gnarwl.cfg /usr/local/etc/gnarwl.cfg.ori
```

Ardından, /usr/local/etc/gnarwl.cfg dosyasını tekrar yaratın ve aşağıdaki içeriği içerisine yapıştırın. Aşağıdaki içerikte dikkat etmeniz gereken parametre “password” parametresidir. Eğer login kullanıcısı(cn=aManager,dc=myhosting, dc=example) için farklı bir parola vermişseniz verdiğiniz parolayı “password” parametresi ile belirtin.

```
map_receiver $recepient
map_subject $subject
map_field $begin vacationStart
map_field $end vacationEnd
map_field $fullname cn
map_field $deputy vacationForward
server localhost
port 389
scope sub
login cn=Manager,dc=myhosting, dc=example
password qazwsx
protocol 0
base dc=myhosting, dc=example
queryfilter (&(mail=$recepient)(vacationActive=TRUE))
result vacationInfo
blockfiles /usr/local/var/lib/gnarwl/block/
umask 0644
blockexpire 0
mta /usr/sbin/sendmail -F $recepient -t
maxreceivers 64
maxheader 512
charset ISO8859-1
badheaders /usr/local/var/lib/gnarwl/badheaders.db
blacklist /usr/local/var/lib/gnarwl/blacklist.db
forceheader /usr/local/var/lib/gnarwl/header.txt
forcefooter /usr/local/var/lib/gnarwl/footer.txt
recvheader To Cc
loglevel 3
```

Gnarwl kurulumu ve entegrasyonu bitmiş durumda.Ama henüz gnarwl paketini LDAP ve Postfix'e entegre etmedik. Bu entegrasyon çalışmalrından sonra gnarwl ayarlaması tamamen bitmiş olacak.

LDAP entegrasyonu ile devam edelim.

20.2 Gnarwl LDAP Entegrasyonu

Gnarwl ve LDAP entegrasyonu, gnarwl ile gelen LDAP şemasını OpenLDAP'a tanıtmaktan ibarettir. Gnarwl ile gelen farklı LDAP şemaları mevcuttur. Bu şema dosyaları kurulum dizininde bulununa "doc" dizini altındadırlar. Biz en son sürüm garwl şeması ISPEnv2.schema şeması ile çalışacağız. Ama yine "configure" komutunun eksik çalışması gibi ISPEnv2.schema şema dosyasında da bir problem vardır. Bu şema dosyasında "mailHost" değişkeni OpenLDAP'taki diğer şemalarla çakışmaktadır. "mailHost" ile aynı isme sahip bir değişken diğer OpenLDAP şemalarında da mevcuttur. Dolayısı ile ISPEnv2.schema dosyasındaki "mailHost" değişken ismini değiştirmemiz gerekmektedir. Bunu gerçekleştirmek için "doc/ISPEnv2.schema" dosyasını edit edin ve "mailHost" karakter dizisini bulun ve bu değişkeni "mailHostVacation" olarak değiştirip kaydedin.

Ardından aşağıdaki komut yardımı ile OpenLDAP şemalarının bulunduğu dizine kopyalayın.

```
#>cp /usr/local/mailserver_makedir/gnarwl-3.4/doc/ISPEnv2.schema
/usr/local/mailserver/openldap/etc/openldap/schema/
```

Gnarwl LDAP entegrasyonu henüz bitmiş durumda değil. ISPEnv2.schema şeması henüz OpenLDAP tanınmış durumda değil. Daha önce Jamm şemasını nasıl tanıtmışsak, bu şemayı da aynı şekilde tanıtacağız.

Gnarwl şemasını tanıtmak için /usr/local/mailserver/openldap/etc/openldap/slapd.conf dosyasını açın ve "include" ile başlayan satırları bulun. Genellikle dosyanın en başında olurlar. "include" satırlarının bittiği satırın ardına aşağıdaki satırı ekleyin ve dosyayı kaydedin.

```
include /usr/local/mailserver/openldap/etc/openldap/schema/ISPEnv2.schema
```

Böylelikle gnarwl LDAP entegrasyonunu tamamlamış oluyoruz.

20.2.1 Gnarwl Postfix Entegrasyonu

Gnarwl ve Postfix entegrasyonu çok basittir. Tek yapmanız gereken aşağıdaki satırı /etc/postfix/main.cf dosyasının sonuna eklemektir.

```
always_bcc=gnarwl
```

20.3 Gnarwl Testi

Gnarwl testini yapabilmemiz için, öncelikle LDAP'ta yer alan kullanıcılarımıza gnarwl ile gelen şemayı objesini eklememiz gerekli. Bu objeyi kullanıclara tanıttıktan sonra, LDAP'ta "Otomatik Cevap Sistemi" isteyen kullanıcılara bu hizmeti sunabiliriz. Yalnız, bunu becerebilmemiz için bir LDAP yönetim arayüzüne ihtiyacımız var.

Debian depolarına "phpldapadmin" ismindeki paket bir LDAP yönetim arayüzü. Bu sistemi kurarsak gnarwl testini rahatlıkla uygulayabiliriz.

Bu sebeple önce phpldapadmin paketini kuralım. Aşağıdaki komut bu işi görecektir. Aşağıdaki komut, aynı zamanda makinenizde Apache Wen sunucusunu ve PHP dilini de yükler. Kurulum sonunda web tabanlı çalışan bir LDAP yönetim sisteminiz olur.

```
#>aptitude -y install phpldapadmin
```

Kurulum bittikten sonra phpldapadmin paketini ayarlamamız gerekmekte. Bu paketin ayar dosyası / etc/phpldapadmin/config.php dosyasıdır. Dosya isminden de anlayacağınız gibi bu bir php dosyasıdır. Öncelikle aşağıdaki komut yardımı ile bu dosyanın bir yedeğini alın.

```
#>mv /etc/phpldapadmin/config.php /etc/phpldapadmin/config.php.ori
```

Ardından /etc/phpldapadmin/config.php dosyasını yeniden yaratın ve aşağıdaki içeriği içerisine yapıştırın.

```
<?php
$config->custom->session['blowfish'] = '';

$i=0;
$ldapservers = new LDAPServers;

$ldapservers->SetValue($i,'server','name','My LDAP Server');
$ldapservers->SetValue($i,'server','host','127.0.0.1');
$ldapservers->SetValue($i,'server','base',array('dc=myhosting,dc=example'));
$ldapservers->SetValue($i,'server','auth_type','session');
$ldapservers->SetValue($i,'login','dn','cn=Manager,dc=myhosting,dc=example');

$friendly_attrs = array();

$friendly_attrs['facsimileTelephoneNumber'] = 'Fax';
$friendly_attrs['telephoneNumber']                      = 'Phone';

$q=0;
$queries = array();
$queries[$q]['name'] = 'User List';
$queries[$q]['base'] = 'dc=example,dc=com';
$queries[$q]['scope'] = 'sub';
$queries[$q]['filter'] = '(&(objectClass=posixAccount)(uid=*))';
$queries[$q]['attributes'] = 'cn, uid, homeDirectory';

$q++;
$queries[$q]['name'] = 'Samba Users';
$queries[$q]['base'] = 'dc=example,dc=com';
$queries[$q]['scope'] = 'sub';
```

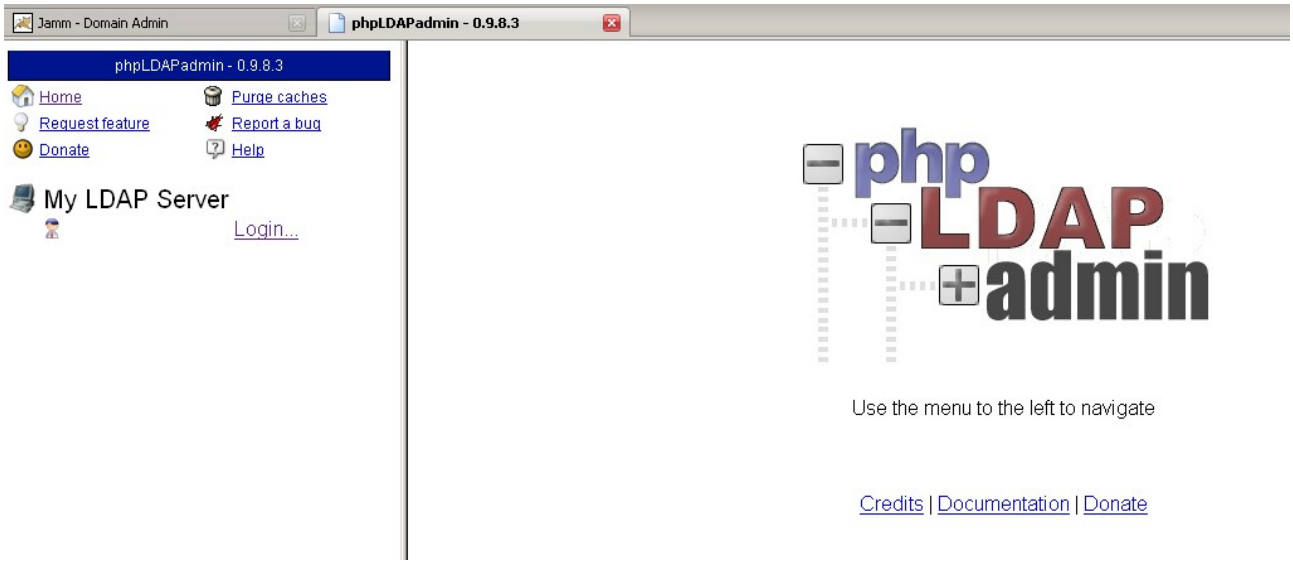
```
$queries[$q]['filter'] = '(&(|(objectClass=sambaAccount)
(objectClass=sambaSamAccount))(objectClass=posixAccount)(!(uid=*$)))';
$queries[$q]['attributes'] = 'uid, smbHome, uidNumber';

$q++;
$queries[$q]['name'] = 'Samba Computers';
$queries[$q]['base'] = 'dc=example,dc=com';
$queries[$q]['scope'] = 'sub';
$queries[$q]['filter'] = '(&(objectClass=sambaAccount)(uid=*$))';
$queries[$q]['attributes'] = 'uid, homeDirectory';
?>
```

Ardından web sunucusunun bu dosyayı okuyabilmesi için aşağıdaki komutları çalıştırın.

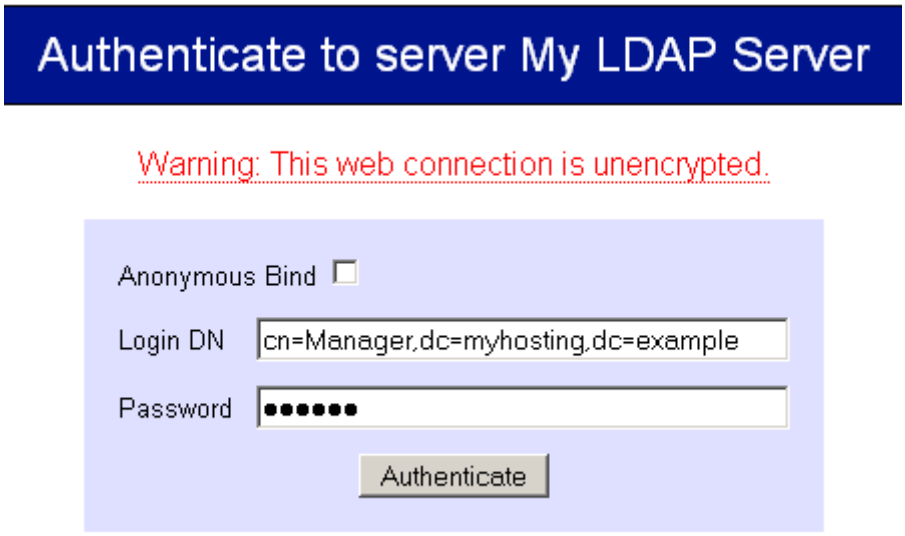
```
#>chmod 755 /etc/phpldapadmin/config.php
#>chgrp www-data /etc/phpldapadmin/config.php
```

Şu an itibari ile phpldapadmin kurulmuş olmalı. Bir internet gezgini aracılığı ile http://<makine_ipsi>/phpldapadmin adresini açın.(<makine_ipsi> karakter dizisi mail sunucunuzun IP'si olmalı)

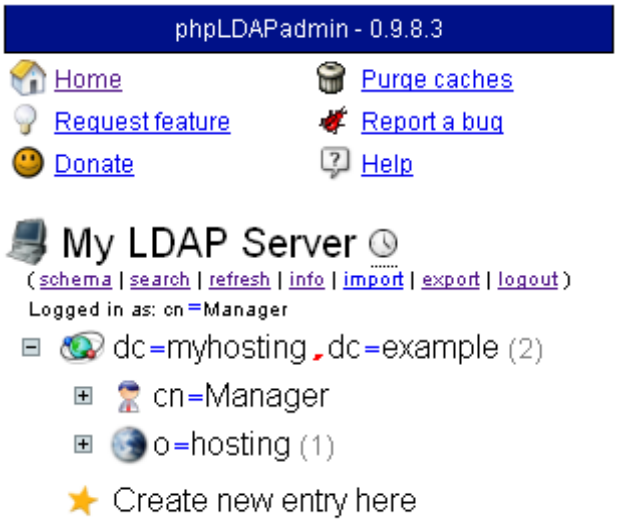
Örnek olarak geliştirdiğimizin mail sunucumuzun IP si 10.0.5.141, http://10.0.5.141/phpldapadmin adresini açtığımızda aşağıdaki gibi bir ekran geliyor. Siz de sisteminizde kurduğunuz phpldapdmin sayfasını açtığınızda aşağıdaki gibi bir ekran gelmeli. Bu ekranda "Login..." linkine tıklayın.



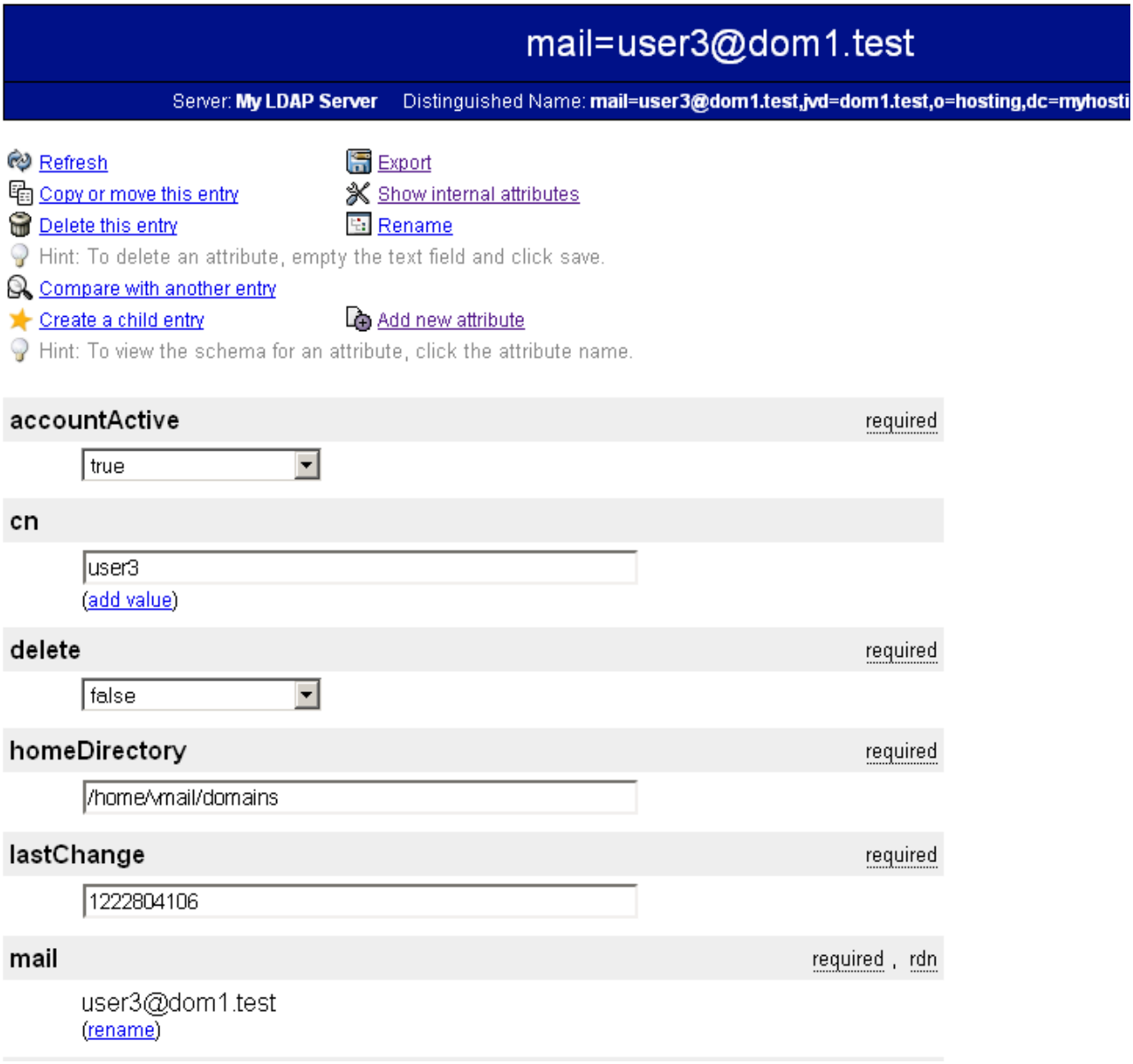
"Login..." linkine tıklayınca aşağıdaki gibi bir ekran gelmeli. Bu ekranda "Login DN", yani kullanıcı adı, olarak "cn=Manager,dc=myhosting,dc=example" girin. "Password" kısmına kullanıcının parolasını girin ve "Authenticate" düğmesine tıklayın. Eğer kullanıcı ve parola doğru ise sisteme giriş yapmalısınız.



Sisteme giriş yaptığınızda sol menüde aşağıdaki gibi bir görüntü olmalı. Bu ekranda "dc=myhosting, dc=example" yazısının solundaki "+" işaretine tıklayın. Mail sunucu şemamızın ağaç şeklideki yapısını göreceksiniz. Tüm mail ve alan adlarımız "o=hosting" başlığı altında yer alıyor.



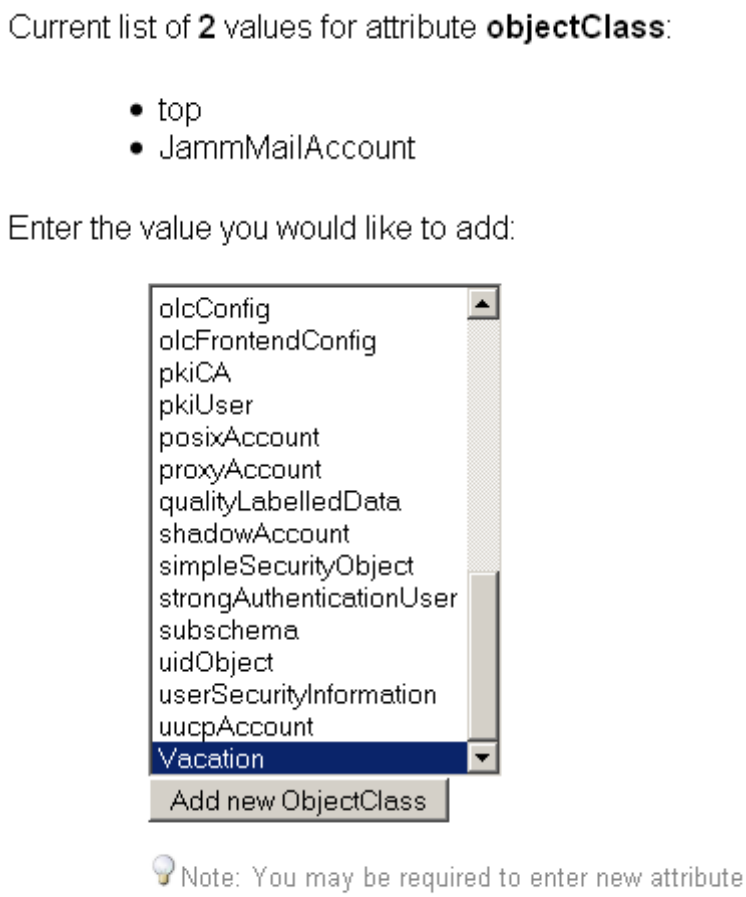
"o=hosting" alanını "+" düğmesine tıklayrak dallandırın ve "mail=user3@dom1.test" alanına ulaşın. Ardından "mail=user3@dom1.test" alanına tıklayın, aşağıdaki gibi bir görüntü elde etmeniz gerekmekte. Sağ alan, aşağıdaki gibi, user3@dom1.test mail kullanıcısı için tüm alan bilgilerini listeliyor olmalı.



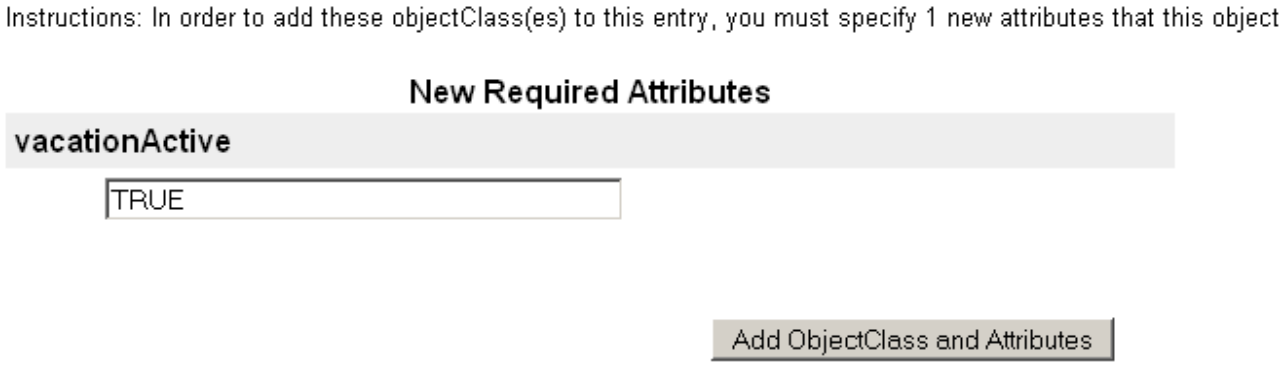
Yukarıdaki ekranda "ObjectClass" başlıklı alanı bulun. Bu alan, aşağıdaki gibi olmalı. ObjectClass alanına Gnarwl şeması ile gelen "Vacation" objesini ekleyeceğiz. Bu objeyi ekleyince user3@dom1.test kulanıcısı için "Otomatik Cevap Sistemi" ayarlanabilir olacak. Bunu gerçekleştirmek için aşağıda resmedilen "add value" linkine tıklayın.

"add value" linkine tıklayınca aşağıdaki gibi bir ekran belirecek. Bu ekranda "Vacation" objesini

bulun ve “Add New ObjectClass” düğmesine tıklayın.



“Add new ObjectClass” düğmesine tıkladığınızda aşağıdaki gibi bir ekran belirecek. Bu ekran “Otomatik Cevap Sistemi”"nin user3@dom1.test kullanıcısı için aktifleştirilip aktifleştirilmeyeceğini soruyor. Bu ekrandaki tekst alanına “TRUE” yazıp “Add ObjectClass and Attributes” düğmesine tıklayın.

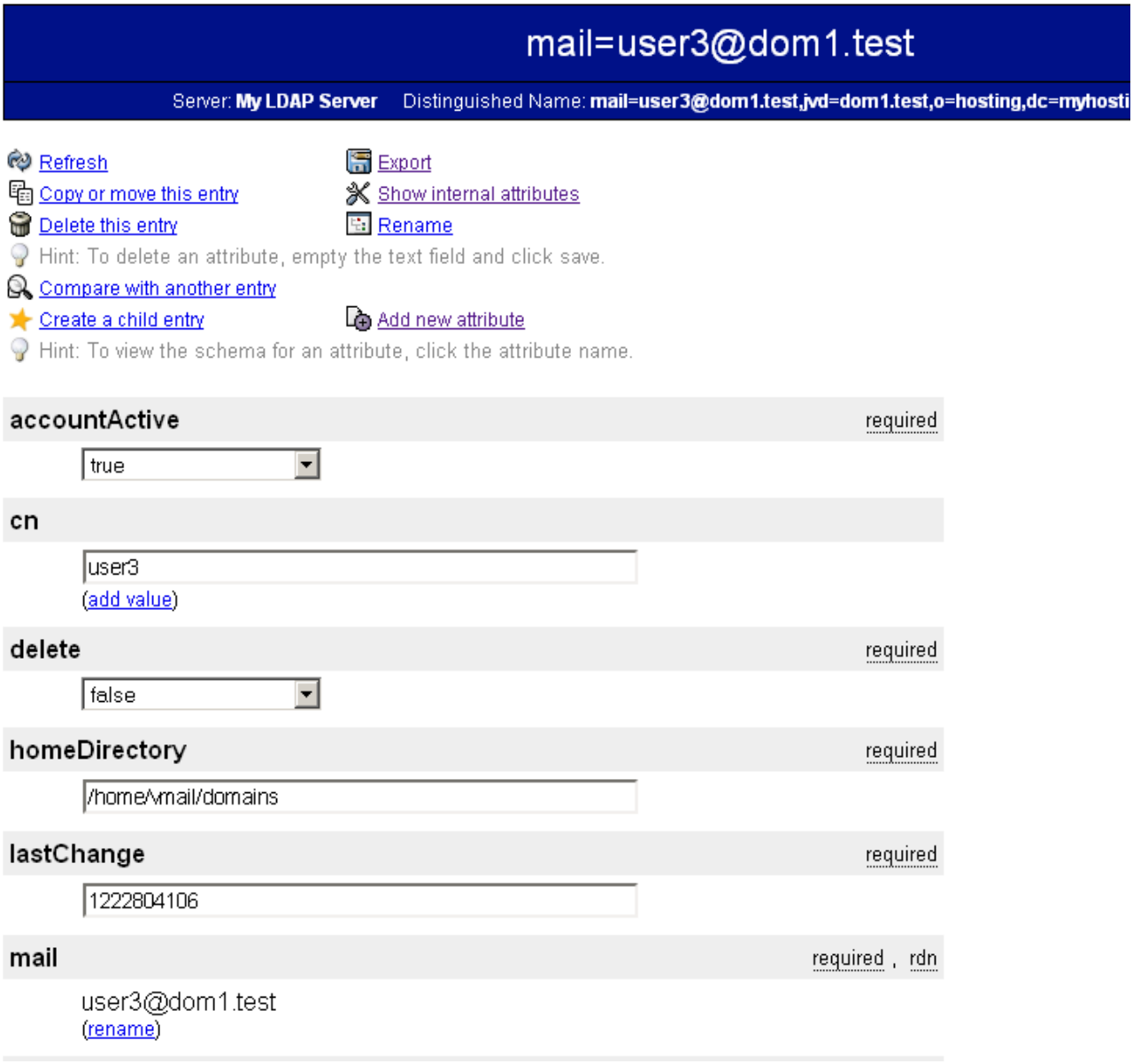


Böylelikle “Vacation” objesini user3@dom1.test için tanıtmış olduk. Bu tanıtma işlemi ile, artık

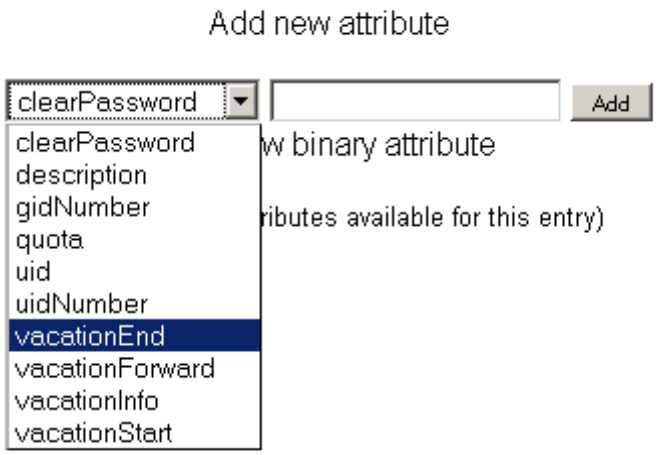
user3@dom1.test kullanıcısı için “Otomatik Cevap Sistemini” aktifleştirebiliriz.

Bunu başarmak için şimdi user3@dom1.test kullanıcısını tatile çıkaralım ve mail adresi için “Vacation” objesini phpldapadmin ile tanımlayalım.

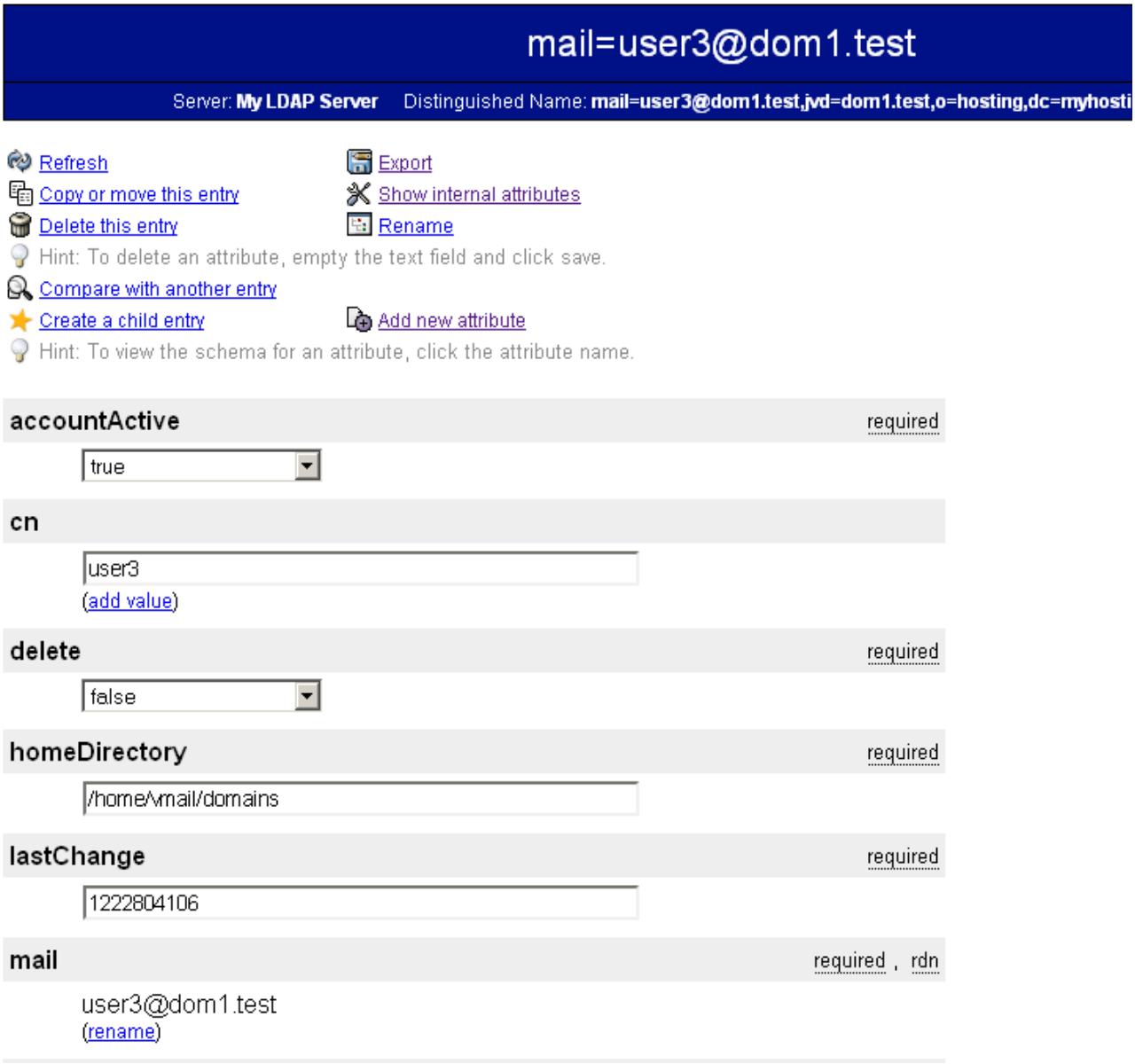
Aşağıdaki ekranda “Add New Attribute” düğmesine tıklayın.



Ardından aşağıdaki ekranda, “vacation” ile başlayan değişkenleri seçin ve hemen yanındaki alana değerini yazıp “Add” düğmesine tıklayın. Örneğin, Önce vacationStart değişkeninin seçin artından seçilebilir menünğn yanındaki alana “10.10.2008” yazın ve “Add” düğmesine tıklayın. Bu sizi ana menüye tekrar götürecektir. Ana menüde tekrar “Add New Attribute” linkine tıklayın ve aşağıdaki ekrana tekrar ulaşın. Bu sefer de geriye kalan vacationEnd(bitiş tarihini içerir, örneğin 15.10.2008) ve vacationInfo alanlarını doldurun. “vacationInfo” alanı tatildeki kullanıcıya gelen maillere cevap olarak gönderilecek metni içermektedir. Dolayısı ile “vacationInfo” alanına bir tatil mesajı yazın.



user3@dom1.test kullanıcısı için “vacation” değişkenlerini atadıktan sonra aşağıdaki ana ekrana dönün. Bu ekranın en sonundaki “Save Changes” düğmesine tılayarak değişiklikleri aktifleştirin.



Değişiklikleri aktifleştirmek istediğinizde,phpldapadmin aşağıdaki gibi, sizden bunları onaylamanızı isteyecek. Bu ekranda "commit" düğmesine tıklayarak değişiklikleri onaylayın.

Şu an itibari ile user3@dom1.test tatilde. Kendisine gelen maillere cevap olarak "vacationInfo"

değişkeninde atanan mesajı yollayacak. Bunu test etmek için user2@dom1.test kullanıcısından user3@dom1.test kullanıcısına bir mail atın. user2@dom1.test kullanıcısına cevap olarak "vacationInfo" alanında kaydedilen mesaj dönmeli.

21 Clamav ve SpamAssassin Kurulumu

Mail sunucumuz şu an hizmet verebilecek durumda. Ama, maalesef, tehlikelere de açık. Bu tehlikeler maillerle gelen virüs ve SPAM içerikli yazılar. Bunlar için henüz bir önlem almış değiliz. Dokümanımızın bu kısmı bu önlemler için kurcağımız antivirüs ve Antispam paket kurulumlarını içermekte.

Antivirüs olarak clamav, AntiSpam olarak Spamassassin paketlerini kuracağız. Elbetteki antivirüs ve antispam için farklı seçenekler var. Ama kendini en iyi ispatlayan paketler bunlar.

Clamav ve Spamassassin'i kurduktan sonra bunları postfix sistemimize entegre edeceğiz. Bu aşamada araya bir paket daha giriyor: Amavis. Amavis, Clamav ve Spamassassini ve daha başka antivirüs ve antispam programlarını yönetebilen ve Postfix'e entegrasyonu sağlayan bir program. Clamav ve Spamassassin postfix'e direkt entegre olmuyor. Onun yerine, Clamav ve Spamassasin amavis ile entegre oluyor, Amavis de Postfix ile. Yani amavis, clamav-Spamassassin ile Postfix arasında entegrasyon amaçlı duran bir program.

Peki Amavais bunu nasıl yapıyor? Amavis, çalıştığında 2 portu dinler. Birinci port, Postfix'den gelen mailleri dinler, ikinci port postfix'e giden mailler için dinlenir. 1.porttan mail alır, virüs ve spam kontrolü yapar ve maili tekrar 2. port aracılığı ile postfix'e gönderir.

Bunu bir örnekle açıklayalım. Örneğin, 10024 portu posftixden gelenler için, 10025 portu da postfix'e gidenler için açılmış olsun.

Postfix kendisine bir mail ulaştığında, bu mailin spamlı mı yoksa virüslümü olduğunu bilmez. Bunu öğrenmek için, maili Amavis'in 10024 nolu portuna gönderir. Amavis'in 10024 nolu portu normal bir MTA gibi çalışır. Yani SMTP protokolüne uyar. Amavis 10024 portundan maili aldığında, sisteminde yüklü antivirüs ve antispam modülleri aracılığı ile mail üzerinde antispam ve antivirüs testleri yapar. Eğer virüslü veya spamlı bir mail olduğuna karar verirse, Mail başlık bilgilerine (Header'larına) spamlı veya virüslü olduğunu yazar ve 10025 portundan postfix'e maili geri gönderir. Bu aşamadan sonra tekrar Postfix, mailin spamlı veya virüslü olduğuna bakmaksızın, maili Mail Ulaştırma Birimine(MUB,Mail Delivery Agent[MDA]) aktarır. Spamlı veya Virüslü maile ne yapılacağına Mail Ulaştırma Birimi karar verir.

Biz, Mail sunucumuzda şu ana kadar MDA olarak procmail kullandık. Eğer, spamlı ve virüslü maillerle ilgili ne yapılacağına dair bir politika belirlemek istiyorsak procmail ayar dosyasını güncellememiz gerekiyor. Ama hemen söyleyelim, bu aşamadan sonra procmail kullanmayacağız, onun yerine daha yetenekli olan maildrop'u kullanacağız. Maildrop kurulumu ve konfigürasyonu ileriki aşamalarda anlatılacak.

Şimdi kuruluma başlayalım. Aşağıdaki komut ile clamav ve Spamassassin'i ve bağımlı paketlerini kurun.

```
#> aptitude -y install amavisd-new spamassassin clamav clamav-daemon zoo unzip bzip2 unzoo
libnet-ph-perl libnet-snpp-perl libnet-telnet-perl nomarch lzop
```

21.1 Amavis ve SpamAssassin ayarlamaları

Ardından "/etc/amavis/conf.d/15-content_filter_mode" dosyasını edit edin ve @bypass_virus_checks_maps satırını bulun, bu satır ve bir altındaki satırdan '#' karakterini silin. Aynı işlemi @bypass_spam_checks_maps ile başlayan satır için de yapın. "/etc/amavis/conf.d/15-content_filter_mode" dosyasının son hali aşağıdaki gibi olmalıdır. Bu değişiklik ile, Amavis'in, Clamav'ı ve Spamassasin'i kullanabileceğini belirtmiş olduk.

```
use strict;

# You can modify this file to re-enable SPAM checking through spamassassin
# and to re-enable antivirus checking.

#
# Default antivirus checking mode
# Uncomment the two lines below to enable it back
#

@bypass_virus_checks_maps = (
   \%bypass_virus_checks, \@bypass_virus_checks_acl, \$bypass_virus_checks_re);


#
# Default SPAM checking mode
# Uncomment the two lines below to enable it back
#

@bypass_spam_checks_maps = (
  \%bypass_spam_checks, \@bypass_spam_checks_acl, \$bypass_spam_checks_re);

1;  # insure a defined return
```

Şimdi sıra gelen Amavis ayarlarını yapmakta. "/etc/amavis/conf.d/20-debian_defaults" dosyasını edit edin ve aşağıdaki değişkenleri gösterildiği şekilde değiştirin. ("/etc/amavis/conf.d/20-debian_defaults" dosyası büyük bir dosya, bu sebeple sadece değiştirilecek olan parametreleri gösteriyoruz.)

QUARANTINEDIR parametresi ilk kurulumda, açık olarak gelir. Aşaıdaki gibi satır başına '#' işaretini koyarak, karantina işlemini devre dışı bırakın. Bunu yapmamızın amacı, velen maillerin spamlı veya virüslü olması durumunda, kullanıcının kendi dizin yapısında saklamak. İleride kullanıcı dizin yapısına "spam" ve "virüs" dizinleri ekleyeceğiz. Gelen mailler Postfix tarafından Maildrop'a aktarıldığında, maildrop gelen mailin spamlı veya virüslü olma durumuna göre maili

“virüs” veya “spam” dizinleri aldında saklayacak Bu senaryoyu gerçekleştirebilmek için amavis'in gelen virüslü ve spamlı mailleri bir dizin altında yani karantina dizinialtında saklamsını engellememiz gerekiyor. Eğer bunu engellemezsek, spamlı veya virüslü mailler amavisde işlenir iken karantina altına alınır ve dolayısı ile bu mailler postfix'e geri verilmez, böylece mailler spam ve virüs dizinlerine ulaşmaz.

```
#$QUARANTINEDIR = "$MYHOME/virusmails";
```

Amavisde, karantina işlemini devre dışı bırakmak için bir işlem daha yapmamız gerekmekte. “/etc/amavis/conf.d/20-debian_defaults” dosyasında virüslü ve spamlı maillerin son durağının Amavis mi yoksa amavisten sonraki birim mi olacağına $final_spam_destiny değişkeni karar verir. Eğer bu değişken D_BOUNCE'a eşitlenmiş ise kötü mailler için son durak amavis demektir. İlk kurulumda $final_spam_destiny=D_BOUNCE olarak gelir. Ama bizim isteğimiz, son durağın amavis olması değil. Biz kötü maillerin amavisten geçip tekrar postfix'e ulaşmasını istiyoruz. Bu sebeple bu değişkeni aşağıdaki gibi değiştirmemiz gerekmekte.

```
$final_spam_destiny       = D_PASS;
```

Amavis, kendi ayar dosyalarında tanımlı alan adları için spam ve virüs kontrolü yapar. Henüz, amavis'in hangi alan adları için spam ve virü kontrolü yapacağını belitmedik. Örneğin dom1.test ve dom2.test için spam ve virüs kontrolleri yapsın. Bunu gerçekleştirmek için “/etc/amavis/conf.d/50-user” dosyasını açın ve içine aşağıdaki satırı ekleyin. Dikkat ederseniz ”dom1.test” ve “dom2.test” yazıları önünde bir nokta(.) var. Bu nokta, dom1.test alt alan adları için de virüs ve spam kontrolü yapmak istediğimizi belirtir. Eğer gerekli görmüyorsanız silebilirsiniz.

```
@local_domains_maps = ([ '.' ]);
```

Şu an itibari ile amavis konfigürasyonu bitmiş durumda. Sırada amavisi postfix iel entegre etmek var.

21.2 Amavis ve Clamav ayarlamaları

Sistemimizde şu an clamav kurulu durumda. Ama henüz Amavis'in clamav'ı kullanması yönünde bir ayarlama yapmış değiliz.

Amavis, anti-virüs sistemlerini /etc/amavis/conf.d/15-av_scanners dosyasında tutar. Clamav'ı Amavis'e tanıtmak için bu dosyayı açın ve 'ClamAV-clamd' kelimesini içeren satırı bulun. Bu satır ve aşağısındaki 3 satır aşağıdaki gibi olmalıdır. Yani aşağıdaki 4 satır önünde '#' işareti olmamalıdır. Eğer '#' karakteri varsa, bu karakteri silin ve dosyayı kaydedin. Böylelikle clamav antivirüs programını amavis'e tanıtmış oluyoruz.

```
['ClamAV-clamd',
  \&ask_daemon, ["CONTSCAN {}\n", "/var/run/clamav/clamd.ctl"],
  qr/\bOK$/, qr/\bFOUND$/,
  qr/^.*?: (?!Infected Archive)(.*) FOUND$/ ],
```

21.3 Postfix Amavis Entegrasyonu

Postfix ve amavisi entegre etmek için aşağıdaki 2 dosyayı edit etmemiz gerekmekte.

- /etc/postfix/main.cf
- /etc/postfix/master.cf

Daha önce bahsettiğimiz gibi amavisin dinlediği 2 port vardır. Bu portları 10024 ve 10025 olarak kararlaştırmıştık. 10024 portu postfix'in mailleri amavis'e aktardığı port olsun. 10025 portu ise amavis'in mailleri postfix'e geri verdiği port olsun.

Öncelikle, postfix'e mailleri kontrol etmek için 10024 portuna göndermesini belirtelim. Bunun için aşağıdaki içeriği /etc/postfix/main.cf dosyasının sonuna ekleyin.

```
content_filter = amavis:[127.0.0.1]:10024
receive_override_options = no_address_mappings
```

Şimdi postfix'in mailleri 10025 portundan almasını sağlayalım. Bunun için aşağıdaki içeriği /etc/postfix/master.cf dosyasının sonuna ekleyin.

```
amavis unix - - - - 2 smtp
      -o smtp_data_done_timeout=1200
      -o smtp_send_xforward_command=yes

127.0.0.1:10025 inet n - - - - smtpd
      -o content_filter=
      -o local_recipient_maps=
      -o relay_recipient_maps=
      -o smtpd_restriction_classes=
      -o smtpd_client_restrictions=
      -o smtpd_helo_restrictions=
      -o smtpd_sender_restrictions=
      -o smtpd_recipient_restrictions=permit_mynetworks,reject
      -o mynetworks=127.0.0.0/8
      -o strict_rfc821_envelopes=yes
      -o receive_override_options=no_unknown_recipient_checks,no_header_body_checks
      -o smtpd_bind_address=127.0.0.1
```

Ve postfix'i aşağıdaki komut yardımı ile yeniden başlatın.

```
#>postfix stop && postfix start
```

Postfix yenıden başladığında “netstat -ltnp” komutunun çıktısı aşağıdaki gibi olmalıdır. Aşağıdaki çıktı, amavis'in 10024 portunu dinlediği, postfix master programının ise 25 ve 10025 portlarını dinlediği görülmektedir.Eğer öyle değilse, lütfen ayar dosyalarını kontrol edin.

```
tcp        0      0 127.0.0.1:10024         0.0.0.0:*               LISTEN      1982/amavisd (maste
tcp        0      0 127.0.0.1:10025         0.0.0.0:*               LISTEN      2921/master
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN      2921/master
```

Amavis ve postfix entegrasyonu bitmiş durumda. Şimdi sıra Anti-Spam modulü Spamassassin'i ayarlamakta.

21.4 SpamAssassin Konfigürasyonu

Spamasssassin daha önce belirttiğimiz gibi bi rAnti-spam programı. Girdi olarak bir mail içeriği alır ve çıktı olarak size bu mailin spam içerikli olup olmadığını söyler. Bunu söylerken, spam bilgilerini mail başlıklarına(Mail Header) yazar.

Spamassassin, kendisine girdi olarak verilen mailleri test ederken bir çok alt modül kullanır. Bu alt modüller spamassasin'e eklenti olarak entegre olurlar. Biz de mail sunucumuzda bu alt modüllerden aşağıda listelenen modülleri sistemimizde kullanacağız.

- `razor`
- `pyzor`

Öncelikle yukarıdaki iki modülü sistemimize yükleyelim. Bunun için aşağıdaki komutu çalıştırın.

```
#>aptitude -y install razor pyzor
```

Şimdi SpamAssassin2i ayarlayalım. Bunun için /etc/spamassassin/local.cf dosyasını edit edin ve aşağıdaki içeriği en son satıra yapıştırın.

```
#pyzor
use_pyzor 1
pyzor_path /usr/bin/pyzor
pyzor_add_header 1

#razor
use_razor2 1
razor_config /etc/razor/razor-agent.conf

#bayes
use_bayes 1
use_bayes_rules 1
bayes_auto_learn 1
```

Yukarıdaki içeriği adım adım açıklayalım.

Aşağıdaki içerik, SpamAssassin Alt modülü olan Pyzor Anti-Spam sistemini SpamAssassin sisteminde etkinleştirmek içindir. “use_pyzor 1” direktifi Pyzor'u etkinleştirir. “pyzor_path” ile başlayan direktif, sistemde kurulu olan pyzor komutunu işaret eder. “pyzor_add_header 1” direktifi, eğer pyzor programı mail'i spam olarak tanırsa, spam bilgisini mail başlığına eklemesini dikte eder.

```
#pyzor
use_pyzor 1
pyzor_path /usr/bin/pyzor
pyzor_add_header 1
```

Aşağıdaki içerik, SpamAssassin Alt modülü olan Razor2 Anti-Spam sistemini SpamAssassin sisteminde etkinleştirmek içindir. “use_razor2 1” direktifi Razor2'yi etkinleştirir. “razor_config” ile başlayan direktif, sistemde kurulu olan razor2 ayar dosyasını işaret eder. SpamAssassin Razor2 ile ilgili tüm bilgileri bu ayar dosyasından elde eder.

```
#razor
use_razor2 1
razor_config /etc/razor/razor-agent.conf
```

Aşağıdaki içerik, SpamAssassin'in kendisi için direktifler içerir. “use_bayes 1” direktifi SpamAssassin'in spam tanıma işlemlerinde Bayes modulünü kullanmasını işaret eder. Bu modül spam tanıma işlemlerini istatistik kurallarına göre yapar. “use_bayes_rules 1” direktigi Bayes istatistik kurallarını kullanmayı işaret eder. “bayes_auto_learn 1” direktifi Bayes otomatik öğrenme modülünü aktifleştirir. Böylelikle, Bayes, mailleri okudukça spam mailleri tanımada daha da hassaslaşır. Tanıdığı her spam maili kendi öğrenme sürecine dahil eder. Böylelikle spam mailleri tanıma olasılığı artar.

```
#bayes
use_bayes 1
use_bayes_rules 1
bayes_auto_learn 1
```

21.5 SpamAssassin ve ClamAV Paketlerinin Güncel Tutulması

AntiVirüs ve AntiSpam sistemler Mail Sunucumuz için kritik önemde olan sistemlerdir. Bunları sürekli güncel tutmalıyız. Eğer tutamazsak, kulanıcılarımızdan kötü geri bildirimler alabiliriz.

İsterseniz bu paketleri nasıl güncel tutabililiriz bir bakalım. Debian, paketlerini kararlı ve test paketleri olarak farklı depolarda tutar. Kararlı paketler, günümüzü daha geriden takip ederler, bu yüzden de daha eski versiyon numaralarına sahiptirler. Buna karşın, test deposundaki paketler günümüzü daha yakından takip ederler ve bu sebeple kararlı depodaki paketlere nazaran daha güncel verisyon numaralarına sahiptirler. Ama bazı paketler vardır ki, anında güncellenmesi gerekmektedir. Anti-spam veri şablonları, virüs veri tabanları bunlara örnektir. Bu paketler sistemimizin kararlılığını etkilemezler ama bunun yanında anında güncellenmesi gerekmektedir. Debian, bu tür paketleri volatile deposunda saklar. Bizde bu tür paketleri, güncellendiği andan

itibaren sistemimize yüklemek istiyorsak, volatile deposunu sistemimize tanıtmalıyız.

Paketlerimizi güncel tutmak için Aşağıdaki içeriği /etc/apt/sources.list dosyasının sonuna ekleyin.

```
deb http://volatile.debian.net/debian-volatile sarge/volatile main contrib non-
free
```

Ardından aşağıdaki komut ile volatile deposunu sistemimize tanıtın.

```
#>apt-get update
```

Eğer yeni paketlerle tekrar kurulum yapmak istiyorsanız **Clamav ve SpamAssassin Kurulumu** başlıklı konumuza geri dönün ve tüm kurulum aşamalarını tekrar gösden geçirin.

Şu an itibari ile Amavis, SpamAssassin ve Clamav paketleri sistemimizde kurulu ve ayarlanmış durumda. Ama hala spamlı mailleri yakalayacak durumda değiliz. Çünkü, hatırlarsanız, amavis'e spamlı mailleri "D_PASS" parametresini verdirerek kullanıcıya ulaştırmasını, spamlı maillere dokunmamasını söylemiştik. Ama bunun yanında da, eğer spamlı veya virüslü mail ile karşılaşırsan mail başlık bilgilerine bu bilgileri yazmasını direktif etmiştik.

Dolayısı ile Antivirüs ve AntiSpam sistemimiz için hala yapmamız gerekenler var. Spamlı veya virüslü maillere asıl işlemi, mailler Amavis'ten geçtikten sonra yapacağız. Hatırlarsanız, mailler Amavis'ten geçtikten sonra postfix aracılığı ile MDA birimine aktarılıyordu. Bu MDA birimde, spamlı ve virüslü maillere, mailin başlık bilgilerine bakarak işlemde bulunacaktık. MDA birimimiz olan maildrop, bu başlık bilgilerine göre spamlı mailleri spam dizinine, virüslü mailleri virus dizinine, normal mailleri de nor mail dizinine aktaracaktı.

Şimdi bu bahsetiğimiz konfigürasyonu gerçekleştirme aşamasına gelmiş durumdayız. Bunun için maildrop ayar dosyasını ayarlayacağız.

Mailrop kurulumu aşamasında dikkat ederseniz "configure" komutuna "*--with-etcdir=/usr/local/mailserver/maildrop/etc*" parametresi vermiştik. Bu, maildrop ayar dosyasının /usr/local/mailserver/maildrop/etc/ dizini altında saklanacağı anlamına geliyor. Maildrop programının ayar dosyası maildroprc adındaki dosyadır. Henüz /usr/local/mailserver/maildrop/etc/ dizini altında maildroprc dosyasını yaratmış değiliz. Maildrop'u ayarlamaya bu dizin altında maildroprc dosyasını yaratarak başlayalım.

Öncelikle /usr/local/mailserver/maildrop/etc/maildroprc dosyasını yaratın ve aşağıdaki içeriği içerisine yapıştırın. Aşağıdaki maildroprc içeriği spamlı mailleri kullanıcın mail dizinindeki "spam" dizinine, virüslü mailleri ise kullanıcının mail dizinindeki virus dizinine taşır. Eğer gelen mail spamlı veya virüslü değil ise, normail mail kutusuna taşır.

```
DEFAULT="$HOME"
LOGDIR="$HOME"

# Log File
```

```
#
logfile "$LOGDIR/maildroprc.log"

# Drop anything listed as Spam into .Spam
if (/^X-Spam-Flag: *YES/)
{
        to "$DEFAULT/spam/"
}
else
{
    if(/^X-Virus-Status:.*INFECTED/)
    {
        to "$DEFAULT/virus/"
    }
    else
    {
      to "$DEFAULT"
    }
}
```

Böylelikle Anti-Virüs ve Anti-Spam için tüm ayarlamaları bitirmiş durumdayız. Postfix'i baştan başlattığımızda, Mail sunucumuz spamlı ve virüslü mailleri için gereken davranışı sergileyecektir.