Eposta Alıcısında (MX'te) Spam Engelleme

Gelen SMTP bağlantılarında istenmeyen posta nasıl reddedilir?

Yazan:
Tor Slettnes
<tor (at) slett.net>

Düzenleyen:
Joost De Cock
Teknik gözden geçirme
<joost.decock (at) astrid.be>

Düzenleyen:
Devdas Bhagat
Teknik gözden geçirme
<devdas (at) dvb.homelinux.org>

Düzenleyen:
Tom Wright
Dil bakımından gözden geçirme
<tom (at) maladmin.com>

Çeviren:
Nilgün Belma Bugüner
<nilgun (at) belgeler·gen·tr>

Düzenleyen:
Yücel Haluk Bugüner
Çeviri bakımından gözden geçirme
<haluk (at) buguner.name.tr>

2005–11–02

Özet

Bu belgede bir Posta Alıcısına, SMTP gelişi sırasında Dolaylı Spama yol açmadan spam ve kötücül yazılımdan kurtulabilme konusunda düşük tesirlilerden üstün tesirlilere kadar çeşitli yollar açıklanmış ve Exim üzerinde bunlar gerçeklenmiştir. Eğer siz bir son kullanıcı iseniz, yani Evolution, Thunderbird veya KMail gibi bir eposta istemcisi kullanarak sadece eposta okuyan ve spamdan kurtulmanın yollarını arayan biriyseniz bu belge size göre değildir.

Konu Başlıkları

Bu çevirinin sürüm bilgileri:

1.0	2 Kasım 2005	NBB
İlk çeviri		

Özgün belgenin sürüm bilgileri:

1.0	08 Eylül 2005	TS
Halka açık sürüm		

Yasal Açıklamalar

1. Giriş

1.1. Belgenin Amacı

Bu belgede bir *Posta Alıcısı* (sayfa: 72)na, SMTP gelişi sırasında *Dolaylı Spam* (sayfa: 71)a yol açmadan spam ve kötücül yazılımdan kurtulabilme konusunda düşük tesirlilerden üstün tesirlilere kadar çeşitli yollar açıklanmıştır.

Açıklamalar doğal olarak kavramsaldır, ancak Exim MTA ve diğer konuya özgü yazılım araçları kullanılarak örnek bir gerçeklenim de sağlanmıştır. Bir takım toleranssızlıklara belge içinde yer yer değinilmiştir.

1.2. Belgenin hedef kitlesi

Bu belgenin hedef kitlesi SMTP, MTA/MDA/MUA, DNS/rDNS ve MX kayıtları gibi kısaltmalara zaten aşina olan eposta sistemi yöneticileridir. Eğer siz bir son kullanıcı iseniz, yani Evolution, Thunderbird veya Outlook Express gibi bir eposta istemcisi kullanarak sadece eposta okuyan ve spamdan kurtulmanın yollarını arayan biriyseniz bu belge size göre değildir; ama sizin eposta sisteminizin (okul, şirket, servis sağlayıcı, vs.) yöneticisinin böyle bir belgenin varlığından haberdar olmasını sağlamak isteyebilirsiniz.

1.3. Belgenin güncel sürümleri

Bu belgenin en yeni sürümününü `http://slett.net/spam-«filtering-«for-«mx/` adresinde bulabilirsiniz. Lütfen düzeltmeler ve eklemeler için düzenli aralıklarla bu adrese bakmayı ihmal etmeyin.

1.4. Sürüm Tarihçesi

Ayrıntılı sürüm tarihçesi için özgün belgeyi[B6] inceleyebilirsiniz.

1.5. Teşekkür

Sürüm tarihçesi[B7]nde de bahsedildiği gibi bu belgeye geri bildirimde bulunarak, düzeltmeler yollayarak, yazımına yardım ederek bir çok kişi katkıda bulundu.

Bu belgeye sağladıkları araçlarla ve fikirleriyle katkıda bulunan bazı kişiler ve gruplar (belli bir sıra gözetilmeksizin):

- *Grilisteleme* (sayfa: 18) tasarımını yapan ve belgeleyen Evan Harris'e <eharris (at) puremagic.com>,
- *katran çukuru* (sayfa: 11)nun (teergrube) tasarımını yapan Axel Zinser'e <fifi (at) hiss.org>,
- SPF[B10], RMX++[B11] ve diğer *Gönderici Yetkilendirme Şemaları* (sayfa: 20) geliştiricilerine,
- DCC[B13], Razor[B14] ve Pyzor[B15] gibi istenmeyen eposta imza depolarını işbirliği içinde sürdüren, dağıtan ve bunları oluşturanlarlara,
- DNS kara ve ak listelerini oluşturan ve bunları sürdüren SpamCop[B16], SpamHaus[B17], SORBS[B18], CBL[B19] ve birçok diğerleri[B20]ne.
- Oldukça karmaşık ve ampirik yaklaşımlarla çeşitli spam filtreleme tekniklerini birleştirip geliştirerek büyük bir atak yapan SpamAssassin[B21] geliştiricileri[B22]ne,
- SpamAssassin ile kullanmak için *Hatalı Virüs Uyarıları Filtresi* (sayfa: 24) listesini düzenleyen ve sürdüren Tim Jackson'a <tim (at) timj.co.uk>,

- Mükkemel *Exim* (sayfa: 30) MTA'sını geliştiren zeki insanlar: Geliştirici Philip Hazel'e <ph10 (at) cus.cam.ac.uk>, SMTP sırasında içerik sınaması yapan Exiscan–ACL yamasını yazan Tom Kistner'e <tom (at) duncanthrax.net>, Exim 4 Debian paketlerini hazırlayarak gerçekten iyi bir iş çıkaran Andreas Metzler'e <ametzler (at) debian.org>,
- Bu spam salgınına karşı çıkmak için gerek fikirleri gerek yazılımları ve gerekse teknikleriyle katkıda bulunan daha bir çok kişiye,
- Bu belgeyi okuyup uygulayarak e–postaları yararlı bir iletişim aracı olarak iyileştirecek biri olabileceğiniz için size

Teşekkür ederim.

1.6. Geri bildirim

Bu belgede bahsedilen tekniklerle ilgili deneyimlerinizi, önerilerinizi, sorularınızı, yorumlarınızı ve/veya katkıda bulunma isteklerinizi duymaktan mutlu olurum. Lütfen bu gibi durumlarda bana bir eposta gönderin: <tor (at) slett.net>.

Eğer bu teknikleri Sendmail veya Postfix gibi diğer *posta aktarımcıları* (sayfa: 72) üzerinde de uygulayabiliyorsanız, lütfen bunları bizimle paylaşın.

1.7. Neye İhtiyacınız var?

Bu belgede açıklanan teknikler epostalarınızı aldığınız internet alan adınız için yapılandırılmış *posta alıcılarına* (sayfa: 72) sistem erişimi üzerinedir. Esasen, ihtiyacınız olan şey, sisteminiz üzerinde çalıştırdığınız *Posta Aktarımcısı* (sayfa: 72) için yazılım kurabilme ve yapılandırma dosyalarını değiştirebilme yetkisine sahip olmanızdır.

Bu belgede açıklananlar doğasında kavramsal olmasına rağmen bir çok farklı posta aktarımcısına uyarlanabilir. Belgedeki örnekler Exim 4 gerçeklenimine göre verilmiştir. Bu gerçeklenim, SpamAssassin[B28] gibi başka yazılım araçları ile birlikte çalışabilir. Ayrıntılı bilgi için *Exim Gerçeklenimi* (sayfa: 30) bölümüne bakınız.

1.8. Bu belgede kullanılan uzlaşımlar

Bu metinde kullanılan sözdizimsel uzlaşımlar:

Sözdizimsel Uzlaşımlar

Metin türü	Anlamı
"Tırnak içine alınmış metin"	Başkalarının yorumları ve bilgisayar çıktıları tırnak içine alınmıştır.
`ekran görüntüsü`	Uçbirimden kopyalanmış bilgisayar girdileri ve çıktılarının birebir görüntüsü. Çerçeveli açık renk bir zemin kullanılmıştır.
komut	Komut satırından girilebilen komutlar.
değişken	Bir değişken veya bir değişkenin değerine gösterici ismi.
`seçenek`	Komut seçenekleri; **ls** gibi bir komuta argüman olarak verilen `-a` gibi bir seçenek böyle gösterilmiştir.
argüman	Bir komuta verilen argümünlar; "**man** *ls* yazınız" gibi.
komut seçenek *argüman* `...` argüman	Komut kullanım şablonları
`dosyaismi`	Dosya ve dizin isimleri, örn. "`/usr/bin` dizinine geçiniz"

<**Tuş**>	Klavyedeki tuşlar, örn. "çıkmak için <**Q**> tuşuna basınız"
[Düğme]	**[Tamam]** gibi tıklanacak bir çizgesel düğme.
[Menü] -> **[Seçim]**	Bir çizgesel menüden nelerin seçileceği belirtilirken kullanılır. Örneğin, "İstemciden **[Yardım]** -> **[Mozilla hakkında...]** seçiniz."
Terminoloji	Bir kavram veya terimin ismi; örnek: "***Çekirdek*** sistemin kalbidir."
Terimler Sözlüğü (sayfa: 69) bölümüne bakınız.	Belge içindeki bir konu başlığına bağ.
Yazar[B31]	Bir dış kaynağa erişim için bağ.

1.9. Belge içeriği hakkında

Bu belge özetle şu bölümleri içerir:

***Altyapı* (sayfa: 7)**
SMTP'ye ve SMTP sırasında yapılan filtrelemeye genel bir bakış.

***Teknikler* (sayfa: 11)**
Bir SMTP aktarımında istenmeyen postanın engellenmesinin yolları

***Dikkate Alınacak Diğer Hususlar* (sayfa: 26)**
Aktarım sırasında yapılan filtreleme ile ilgili ele alınacaklar.

***Sorular ve Cevaplar* (sayfa: 28)**
Sorularınız ve vermeye çalıştığım yanıtları.

Örnek bir Exim gerçeklenimi *Exim Gerçeklenimi* (sayfa: 30) bölümünde ele alınmıştır.

2. Altyapı

Postaların kabulü ve gönderilmesinde sıklıkla yapıldığı gibi görevi bir şekilde geçiştirmek yerine, gelen SMTP aktarımları sırasında posta filtrelemenin getirileri üzerinde durulmuştur. Ayrıca, SMTP aktarımı konusuna kısa bir giriş hazırladık.

2.1. Posta Filtrelemesi Neden SMTP Aktarımı Sırasında Yapılır?

2.1.1. Mevcut Durum

Spam alıyorsanız, ellerinizi havaya kaldırın. İndirmeyin.

Bilgisayar virüsleri ve kötücül yazılımlar alıyorsanız siz de ellerinizi havaya kaldırın.

"Message Undeliverable" (İleti Teslim Edilemiyor), "Virus found" (Virüs bulundu), "Please confirm delivery" (Lütfen teslimi onaylayın), vs. gibi sizin göndermediğiniz iletilerle ilgili *Teslimat Durum Bildirimi* (sayfa: 73) iletileri alıyorsanız, siz de ellerinizi havaya kaldırın. Bu *Dolaylı Spam* (sayfa: 71) olarak bilinir.

Bu sonuncu türdekiler özellikle sorunludur çünkü istenmeyen veya kötü niyetli iletilere göre temizlenmeleri daha zordur, başlık bölümlerini yorumlamada tanrısal becerilere sahip olmayanlar için bunlarla uğraşmak boşa bir çabadır. Virüs uyarıları söz konusu olduğunda ise alıcılar gereksiz bir endişeye kapılırlar ve daha büyük bir bölümü bunları toptan gözden çıkarır, dolayısıyla *Teslimat Durum Bildirimi* (sayfa: 73) geçerli olan meşru iletiler de bu arada kaybedilir.

Son olarak, spam ya da virüs tarayıcıların hatalı değerlendirmeler yapmalarından dolayı meşru postalarınızı bir karadelikte kaybedenlerdenseniz —— Siz ayaklarınızı da kaldırın.

Eğer hala ayaktaysanız, postalarınızın başına neler geldiği konusunda tam bir bilginiz olmayabileceğini düşünürüm. Bir şekilde spam engellemesi yapıyorsanız, tek başına DNS karalisteleri (SpamHaus, SPEWS, SORBS...) gibi ilkel filtreleme tekniklerinde deneyimli de olsanız, hatta, posta istemcinizde postaları elle bizzat çöp kutusuna taşıyor olsanız bile geçerli postalarınızı kaybetme ihtimaliniz hala vardır.

2.1.2. Sebep

Spam, açgözlülüğün birçok ilkel ürünlerinden biri olarak bir sosyal bozukluktur. Siz buna zenginlik diyebilirsiniz, ne derseniz deyin; daha geniş bir ekosistemi yoketmeye çalışan alt yaşam biçimleri eğer başarılı olurlarsa, aslında eninde sonunda kendi yaşam alanlarını yıkıma uğratırlar.

Toplumsal yaşam, felsefe, vs. bırakın bunları bir kenara: Sen – posta sisteminin yöneticisi – bu döküntüden kurtulmanın bir yolunu bulmak gibi çok somut bir görevin var, bununla yüzleş.

Konumuza dönersek, posta aktarım ve teslimat yazılımlarının ceşitli bileşenleri tarafından işleme sokulan postalarla ilgili üzerinde uzlaşılmış bazı sınırlamalar vardır. Geleneksel yapılandırmada, bir alanın adreslerine gelen posta teslimatlarının çoğu ya da tamamı bir veya daha fazla sayıda *Posta Alıcısı* (sayfa: 72) tarafından kabul edilir. Çoğunlukla, bunlar postayı kullanıcıların posta kutularına teslim edilmek üzere dahili ağdaki bir veya daha fazla sayıda makineyi gönderirler. Eğer bu sunuculardan biri istenen teslimatın veya işlemin gerçekleştirilemeyeceğini saptarsa, özgün postanın göndericisine özdevinimli üretilmiş bir *Teslimat Durum Bildirimi* (sayfa: 73) döndürür.

Spam ve virüs tarayıcıları konuşlandıran organizasyonlar genellikle, *posta alıcılarından* (sayfa: 72) gelen postaları dahili konaklara ya da yazılımlara yönlendirmeden önce iletinin olası en düşük dirençle teslimat yolunu aşmalarını sağlamaya çalışırlar. Spamı filtreleme için tercih edilen yöntem, iletiyi kullanıcının posta kutusuna teslim etmeden önce SpamAssassin veya benzeri bir yazılım üzerinden geçirmek ve/veya kullanıcının *Posta İstemcisi* (sayfa: 73)ndeki spam filtreleme yeteneklerine güvenmek şeklindedir.

Bu noktada spam ya da virüslü olarak tasnif edilmiş postalara uygulanacak işlemler sınırlıdır:

- Göndericiye bir *Teslimat Durum Bildirimi* (sayfa: 73) döndürebilirsiniz. Burada sorun, hemen hemen tüm spamların ya da virüslü postaların taklit edilmiş bir gönderici adresiyle yollanmasıdır. Eğer böyle bir postayı gönderirseniz, posta büyük ihtimalle bunda bir kabahati olmayan kişilere gidecektir. Kimbilir, belki de bilgisayarlar hakkında pek bilgisi olmayan bir ninenin bilgisayarına Blaster kurdunu bulaştırmış olacaksınız. Başka bir deyişle *Dolaylı Spam* (sayfa: 71) üretmiş olacaksınız.
- Göndericiye herhangi bir uyarı döndürmeksizin iletiyi çöpe gönderebilirsiniz. Gerek gönderici gerekse alıcı (böyle bir iletinin varlığından bile haberi olmaz) iletiye ne olduğunu bilmeyeceği için bu işlem, *Hatalı Olumlama* (sayfa: 71) durumuna uyan daha büyük bir soruna dönüşür.
- Kullanıcıların postalarına nasıl eriştiğine bağlı olarak (örneğin, IMAP protokolü üzerinden veya POP–3 ile almaksızın web tabanlı posta okuyucu kullanarak erişenler) ve hesaplarına bir seçenek olarak sunarak, bu tür postaların kullanıcının alanındaki ayrı bir döküntü dizinine konmasını sağlayabilirsiniz.

  Bu, bu üç seçenekten belki de en iyisi. Öyle de olsa, bazı meşru iletiler alıcının kendi döküntü dizinini tarama sıklığına bağlı olarak gözden uzak kalabilir veya silme sırasında gözden kaçabilirler.

2.1.3. Çözüm

Artık sizin de tahmin edebileceğiniz gibi bu sorunun *tek doğru* çözümü, spam ve virüs filtrelemesini alanınıza gelen postaları alan *Posta Alıcısı* (sayfa: 72) üzerinde SMTP diyaloğu sırasında yapmaktır. Bu yolla, eğer posta istenmeyen türdeyse, yukarıda açıklanan çözümsüzlüklere düşmeden, bir SMTP *red* yanıtı yeterli olur. Sonuç olarak:

- Asıl ileti alınmadan önce, SMTP aktarımının başında, istenmeyen postanın çoğunun teslimatını durdurmanız mümkün olur. Böylece, ağ band genişliğini daha verimli kullanmış olmanın yanında işlemcinizi de daha az meşgul etmiş olursunuz.
- Daha sonra yapıldığında yararı olmayan, *SMTP Aktarımının Geciktirilmesi* (sayfa: 11), *Grilisteleme* (sayfa: 18) gibi spam filtreleme tekniklerini devreye sokmak mümkün olur.
- Bir teslimatın mümkün olmadığı durumlarda (alıcı adresin geçersiz olması gibi), *Dolaylı Spam* (sayfa: 71) üretmeksizin göndericiyi uyarmanız mümkün olur.

  Posta listeleri, başka sitelerin posta hesapları gibi güvenilir kaynaklardan yönlendirilen postaların reddedilmesinin bir sonucu olarak dolaylı spama sebep olmaktan nasıl kaçınacağınızı anlatacağız[1].
- Başkalarından kaynaklanan dolaylı spama (antivirüs yazılımlarının "Sizde virüs var" tarzı hatalı uyarıları) karşı kendinizi korumanız mümkün olur.

Artık havadaki eller inebilir. Ayakları da havada olanlar siz de ayağa kalkabilirsiniz.

2.2. İyi, Kötü, Çirkin

Filtreleme tekniklerinin bazıları SMTP aktarımı sırasında kullanılmaya diğerlerinden daha elverişlidir. Bunların bazıları da biraz daha iyidir. Hemen hemen hepsinin taraftarları olduğu gibi muhalifleri de vardır.

Yorum yapmadan, tartışmalara yolaçan yöntemleri de burada açıklayalım. Örneğin:

- Bir iddia, *DNS Sınamaları* (sayfa: 12)nın, posta göndericilerini, gönderdikleri iletilerin liyakatine bakmaksızın, tamamen kullandıkları internet hizmet sağlayıcılarından (İSS'lerinden) dolayı cezalandırdığı şeklindedir.
- Bazıları, *SMTP Aktarımının Geciktirilmesi* (sayfa: 11) ve *Grilisteleme* (sayfa: 18) gibi *Kalleş Yazılım* (sayfa: 72) tuzaklarının kolayca üstesinden gelindiğini ve uzun vadede meşru postalar açısından servis kalitesini düşürerek etkilerinin azalacağına işaret etmektedir.
- Bazıları, *Gönderici Yetkilendirme Dizgesi (SPF)* (sayfa: 20) gibi *Gönderici Yetkilendirme Şemaları* (sayfa: 20)nın servis sağlayıcılara, epostalarını farklı ağlar arasında dolaştıran ya da bir konaktan diğerine yönlendiren müşterilerini içerde tutmanın yolunu açacağını savunmaktadır.

Bu tartışmaların çoğundan uzak durararak, kullanılabilir olan çeşitli tekniklerin olası yan etkilerini de dahil ederek işlevsel açıklamalarını yapmaya çalışırken, biraz da bunların kullanımıyla ilgili kendi deneyimlerimden bahsedeceğim.

Bilinçli olarak bu belgenin kapsamına almadığım ve günümüzde hala kullanılmakta olan bazı filtreleme teknikleri var:

- Soru/cevaplı kimlik kanıtlama sistemleri (TMDA[B59] gibi). Postayı alır almaz röleledikten sonra *Zarf Göndericisi* (sayfa: 74)ne bir doğrulama isteği döndürmeleri sebebiyle bu sistemler SMTP sırasında filtrelemeye uygun değildirler. Bu sebepten bu teknik bu belgenin kapsamı dışında tutulmuştur[2].
- *Bayes Filtreleri* (sayfa: 70). Bunların belli bir kullanıcıya ve/veya dile özgü eğitilmeleri gerekir. Bu sebeple bunlar da SMTP aktarımı sırasında kullanılmaya uygun değillerdir (Fakat, yine de *Kullanıcı Verileri ve Ayarları* (sayfa: 27) bölümüne bakın).
- *Gönderici ücretlendirme şemaları* (sayfa: 71), bütün dünyanın meşru postası bir sanal *posta damgası* ile gönderildiği sürece, döküntü postayı ayıklamada aslında bir yararı olmayacaktır. (Orta ya da uzun vadede, tamamen zıt amaçlar için kullanılmaya başlanabilir – normalde reddedilmesi gereken bir postanın sırf damgalı diye kabul edilmesi durumu.)

Genel olarak önerdiğim teknikler somut ve *Hatalı Olumlama* (sayfa: 71) sonucunu doğurmayacak tekniklerdir. Kişilerin epostaları hazırlamak için harcadıkları çabaya ve bu iletilerin onlar için taşıdığı öneme göstereceğimiz saygının gereği olarak büyük miktarda meşru iletinin reddedilmesine yol açabilecek tekniklerden bilinçli bir şekilde kaçınmamız çok önemlidir.[3] Bu özellikle SMTP sırasında sistem çapında filtreleme yapıldığında önem kazanıyor, çünkü postaların son alıcılarının postaları filtrelemede kullanılan kriterler hakkında ya hiç bilgileri yok ya da bunlar üzerinde çok az hakimiyet sağlayabiliyorlar.

2.3. SMTP Aktarımı

SMTP internette posta teslimatı için kullanılan protokolün ismidir. Protokolün ayrıntılı açıklamasını RFC 2821[B69]'de bulabilirsiniz. Ayrıca, Dave Crocker'ın İnternet Postalarının Mimarisi[B70]'ne bakışını da okumanızı öneririm.

Posta teslimatları bağlanan konak (istemci) ile bağlanılan konak (sunucu) arasındaki SMTP işlemleri ile yapılır.

Tipik bir SMTP aktarımında, istemci **EHLO**, **MAIL FROM:**, **RCPT TO:** ve **DATA** gibi komutlar gönderir. Sunucunuz her komuta 3 rakamlık bir sayısal kod (komutun kabul edildiğini belirmek için **2***xx*, geçici bir sorun ya da kısıtlayıcı bir durum için **4***xx*, kesin ve mutlak başarısızlık halinde **5***xx*) ve bunu izleyen insanların anlayabileceği bir açıklama ile yanıt verir. Bu kodların tamamı RFC 2821[B71]'de açıklanmıştır.

SMTP aktarımında eniyi durum senaryosu genel olarak birbirini izleyen şu adımlardan oluşur:

Basit SMTP diyaloğu

İstemci	Sunucu
Sunucuya bir TCP bağlantısı kurar.	SMTP (veya halefi olan ESMTP) diyaloğuna hazır olduğunu belirtmek için **220** koduyla başlayan bir karşılama iletisi gönderir: `220 sunucu.f.q.d.n ESTMP...`

Bir **HELO** (artık atıl) ya da **EHLO** ile başlayan ve kendi *Nitelikli Alan Adı* (sayfa: 72)nı içeren bir selamlaşma komutu ile kendini tanıtır:
`EHLO istemci.f.q.d.n`
	Bir **250** yanıtı ile bu selamı kabul eder. Eğer istemci selamlaşma komutunun gelişmiş sürümünü (**EHLO**) kullanmışsa, sunucunuz onun çok satırlı yanıtları işleme yeterliliğinde olduğunu anlar ve normal olarak kendi yeteneklerini belirten satırları gönderir:
	`250-sunucu.f.q.d.n Hello ...`
	`250-SIZE 52428800`
	`250-8BITMIME`
	`250-PIPELINING`
	`250-STARTTLS`
	`250-AUTH`
	`250 HELP`
	Bu yanıt **PIPELINING** yetisini içeriyorsa, istemci bu noktadan sonra herbiri için ayrı ayrı yanıt beklemeksizin bir çok komutu bir kerede gönderir.

Zarf Göndericisi (sayfa: 74)ni belirterek yeni bir posta aktarımı başlatır: `MAIL FROM:<gönderen@adres>`	Göndericinin kabul edildiğini belirten bir **250** yanıtı gönderir.

İletinin *Zarf Alıcıları* (sayfa: 73)nı bir defada liste halinde yollar: `RCPT TO:<alıcı@adres>`	Her komuta o alıcı için teslimatın kabul mü edildiğine (**2xx**), geçici bir sorun mu oluştuğuna (**4xx**) yoksa red mi edildiğine (**5xx**) dair bir yanıt döndürür.
İletiyi göndermeye hazır olduğunu belirten bir **DATA** komutu gönderir.	Komutun geçici olarak kabul edildiğini belirten **354** yanıtını gönderir.
RFC 2822[B75] uyumlu başlık satırları (`From:`, `To:`, `Subject:`, `Date:`, `Message-ID:` gibi) ile başlayan iletiyi aktarmaya başlar. Başlık ve gövde bir boş satırla ayrılır. İletinin sonunda ileti sonunu belirten ve tek bir nokta içeren ek bir satır göndererek ileti aktarımını bitirir.	İletinin kabul edildiğin belirten **250** yanıtını gönderir.
Eğer teslim edilecek başka iletiler de varsa, bir **MAIL FROM:** komutu gönderir. Aksi takdirde, **QUIT** der ya da yaygın bir durum olarak basitçe bağlantıyı keser.	Bağlantıyı keser.

3. Teknikler

Bu bölümde, uzak konaklardan SMTP aktarımı sırasında döküntü postayı ayıklamanın çeşitli yollarına bakacağız. Ayrıca, bu teknikleri konuşlandırmanın bazı yan etkilerini önceden tahmin etmeye çalışacağız.

3.1. SMTP Aktarımının Geciktirilmesi

Açıkça ortaya çıktığı gibi, spamı durdurmanın en etkin yollarından biri SMTP diyaloğu sırasında aktarıma gecikmeler koymaktır. Bu ilkel bir *katran çukuru* (teergrubing) türüdür (bkz. `http://www.iks-«jena.de/mitarb/lutz/usenet/teergrube.en.html`).

Virüs taşıyan epostaların hemen hepsi ve çoğu spam çok kısa sürede büyük miktarda postayı göndermek üzere eniyilenerek amaca uygun hale getirilmiş bir SMTP istemci yazılımı yoluyla sunucunuza doğrudan doğruya teslim edilir. Böyle istemciler genel olarak *Kalleş Yazılım* (sayfa: 72) olarak bilinir.

Kalleş yazılım yazarları bu işlemi yerine getirmek için RFC 2821[B78] belirtimindekinden birazcık farklı birkaç kısayol kullanırlar. Kalleş yazılımların asıl hedefleri, sabırsızlıklarıyla ve özellikle yavaş yanıt vermeleriyle nam salmış posta sunucularıdır. Sunucu daha SMTP aktarımına hazır olduğunu belirtmeden sunucuya bir **HELO** veya **EHLO** komutu gönderirler ve/veya sunucunun **PIPELINING** yetisini ilan etmesini beklemeksizin çeşitli SMTP komutlarıyla boru hattı oluşturmayı denerler.

Bellibaşlı *posta aktarımcıları* (sayfa: 72) (Exim gibi) özdevinimli olarak SMTP protokolünün bu şekilde hiçe sayılmasını *eşzamanlama hataları* olarak ele alır ve gelen bağlantıyı hemen keser. Eğer böyle bir posta aktarımcısı kullanıyorsanız, günlük kayıt dosyalarında bu tür bir bağlantı reddine rastlamışsınızdır. Aslında, sunucunuza SMTP aktarımının hemen başında aktarıma hazır olduğunu belirtmeden önce, zaman kaybına sebep olan bazı sınamalar yaptırıyorsanız (*DNS Sınamaları* (sayfa: 12) gibi) böyle hatalar sıkça olur ve kalleş yazılımlar sunucunuzun canlanabilmek için biraz zamana ihtiyacı olabileceğini dikkate almadıklarından (spamcılar böyle düşünür) bir şansınız olur.

Ek gecikmeler koyarak da bunun oluşmasına yardımcı olabiliriz. Örneğin biraz bekleme kararı verebiliriz:

- SMTP aktarımına hazır olduğunu bildirmeden önce 20 saniye,
- selamlaşmadan (**EHLO** veya **HELO**) sonra 20 saniye,
- **MAIL FROM:** komutundan sonra 20 saniye ve

- her `RCPT TO:` komutundan sonra 20 saniye.

Nereden çıktı bu 20 saniye diyebilirsiniz. Neden bir dakika değil? Ya da birkaç dakika değil? Herşeyden önce, RFC 2821[B81] gönderici konağın (istemcinin) her SMTP yanıtı için birkaç dakika beklemesini zorunlu kılar. Bazı alıcı konaklar, bilhassa Exim kullananlar, gelen posta teslimat bağlantılarına yanıt olarak *Gönderici Varlık Sınaması* (sayfa: 16) uygularlar. Siz veya kullanıcılarınızdan biri böyle bir konağa posta gönderdiğinde, bu konak sizin alanınız için yetkilendirdiğiniz *Posta Alıcısı* (sayfa: 72)na bağlanıp gönderici adresinin doğrulanmasını sağlamak üzere bir SMTP diyaloğu başlatacaktır. Böyle bir *Gönderici Varlık Sınaması* (sayfa: 16) için öntanımlı zamanaşımı 30 saniyedir. Eğer koyduğunuz gecikme bu sürenin aşılmasına sebep oluyorsa, istemcideki *Gönderici Varlık Sınaması* (sayfa: 16) başarısız olacağından sizin ya da kullanıcınızın istediği posta teslimatı reddedilebilecektir (genellikle, posta teslimatının geri iade edilmeden önce 5 gün boyunca gerçekleştirilmeye çalışılacağını belirten bir geçici başarısızlık olarak).

Başka bir deyişle, meşru posta teslimatı ile ilgili girişimin başlamasını geciktirebileceğiniz en uzun süre 20 saniyedir.

Her SMTP aktarımına böyle gecikmeler uygulamak istemiyorsanız (çok meşgul bir siteniz vardır ve makinenizin kaynakları kıt kanaat yetiyordur), bu gecikmeleri "seçimlik" kullanabilirsiniz. Böyle durumlar:

- İstemcinin DNS yapılandırmasıyla ilgili bir sorun varsa (bkz. *DNS Sınamaları* (sayfa: 12)).
- SMTP aktarımı sırasında bazı sorunların izleri saptandıktan sonra (bkz. *SMTP Sınamaları* (sayfa: 13)).
- Sadece DNS kaydınızdaki en yüksek numaralı yani en düşük öncelikli *Posta Alıcısı* (sayfa: 72) üzerinde. Çoğunlukla, meşru posta göndericilerinin *posta aktarımcıları* (sayfa: 72) en düşük numaralı *posta alıcılarını* (sayfa: 72) denediği halde *Kalleş Yazılım* (sayfa: 72)lar özellikle bu konakları tercih ederler.

Aslında seçimlik aktarım gecikmeleri, bundan sonraki bölümlerde açıklayacağımız sınamalardan daha az kesin sınamalarla birlikte kullanıldığında iyi bir yöntem olabilir. Siz, büyük ihtimalle SPEWS gibi *karalisteler* (sayfa: 13)den kaynaklı olarak posta reddini tercih etmezsiniz, ama bu listeleri gecikmenin uygulanmasında sorunun belirleyicisi olarak kullanabilirsiniz. Tüm bunlardan sonra, aşağılayıcı bir gecikmeye konu olanlar dışında kalan meşru postaların teslimatları bundan etkilenmeyecektir.

Diğer taraftan, spam yapıldığının kesin kanıtını bulursanız (örn. *SMTP Sınamaları* (sayfa: 13) yoluyla) ve sunucunuzun gücü yetiyorsa, teslimatı reddetmeden önce 15 dakikalık veya buna yakın uzunca bir gecikme uygulayabilirsiniz[4]. Bunun spamcının yavaşlatılmasından başka bir yararı yoktur. Ama ne var ki, bunlar DNS karalistelerine ve bunlarla işbirliği yapanlara yakalanmadan önce daha az kişiye ulaşmalarını sağlamış olursunuz ve fedakarlığınızla başbaşa kalırsınız. `:-)`

Benim durumumda, gelen teslimat bağlantılarından reddedilenlerin %50'si, seçimlik aktarım gecikmeleri ve SMTP eşzamanlama hatalarının sonucu reddediliyor. Kabaca bir yaklaşımla, gelen döküntü postanın yaklaşık %50'si tek başına SMTP aktarım gecikmelerinin sonucu olarak durdurulmaktadır, diyebiliriz.

Ayrıca *Soru 5.1.* (sayfa: 28)'e de bakınız.

3.2. DNS Sınamaları

SMTP bağlantısı kurmak isteyen tarafın dürüstlüğünün belirtileri *Alan Adı Sistemi* (sayfa: 70)'nden doğrudan toplanabilir, hatta bu SMTP komutlarının alınmasından önce yapılabilir. Bunun için, belirli koşulları yerine getirenlerin ya da bunlarla çeliştikleri bilinenlerin IP adreslerini bulmak için çeşitli DNS karalistelerine başvurulabilir ve/veya konağın genel olarak dürüstlüğünün ayırıcı bir belirtisi olan normal/ters DNS çifti arasındaki uyuma bakmak için basit bir DNS sorgusu yapılabilir.

Bununla birlikte, SMTP diyaloğu sırasında sunulan çeşitli veri öğeleri (örn, selamlaşmada belirtilen isim) alındıktan sonra bunlar DNS doğrulamasına konu edilebilir. Bu veri öğelerinin açıklamalarını *SMTP Sınamaları* (sayfa: 13) bölümünde bulabilirsiniz.

Yalnız, DNS sınamalarının olumsuz sonuçları tek başına spamın belirtisi olarak ele alınamaz (Sınama için kullanılan sunucu bir sorundan dolayı yanıt vermeyebilir). Bununla birlikte, eğer çok meşgul bir site iseniz, her ileti için harcanacak işlem zamanı pahalıya malolabilir. Bu durumda şu söylenebilir, DNS sınamaları oturum açabilmek için yararlı bilgileri elde edebilmek için ve/veya daha karmaşık dürüstlük sınalamarının bir parçası olarak kullanılabilir.

3.2.1. DNS Karalisteleri

DNS karalisteleri, aktarım sırasında spam engelleme amacıyla kullanılmak üzere oluşturulmuş araçlardır[5]

DNS adresine ("A" kaydı) ek olarak bir girdinin "TXT" kaydına da bakmak isterseniz, bir SMTP red yanıtında kullanılabilecek tek satırlık bir liste açıklaması alırsınız. Bunu denemek isterseniz, çoğu Unix ve Linux sisteminde bulunan "host" komutunu kullanabilirsiniz:

```
host -t txt 2.0.0.127.dnsbl.sorbs.net
```

Bu listelerin farklı listeleme/listelememe kuralları olan yüzlerce çeşidi vardır. Bazı listeler bu farklı listeleme kurallarının bir bütünü olarak tek bir liste oluşturup, ters DNS sorgularına belirtilen adresin etkilendiği koşullara uyan farklı bir veri ile yanıt verirler. Örneğin, `sbl-xbl.spamhaus.org`'a yöneltilen bir ters DNS sorgusuna, SpamHaus kadrosu tarafından spamcılara ve onların İSS'lerine ait olduğunu sanılan IP adresleri için 127.0.0.2, *Zombi Konak* (sayfa: 74)lar için 127.0.0.4 ve *Açık Vekil* (sayfa: 70) sunucuları için 127.0.0.6 yanıtı döner.

Ne yazık ki, bu listelerden bazıları iyi tanımlanmamış listeleme kurallarıyla listelenen adresler hakkında yanlış bilgiler verirler ve iddia ettikleri çelişkilere uygun olmayan büyük IP bloklarını içerirler[6]. Böyle listelere körükörüne güvenmek çoğunlukla *Çevresel Bozunma* (sayfa: 71) olarak bilinen (*Dolaylı Spam* (sayfa: 71) sonucu olmayan) duruma yolaçar.

Bu sebeplerden, DNS karalistelerine bağlı tek bir olumlu yanıta bağlı kalarak posta teslimatlarını reddetmek yerine, çoğu yönetici bu listeleri biraz daha ayrıntılı bir inceleminin konusu yaparlar. Çeşitli listelere başvurup her olumlu yanıta bir puan verirler. Toplam puanın önceden belirlenmiş bir eşiği aşması durumunda bu adresten gelen teslimatı reddederler. DNS listelerinin bu şekilde kullanımı daha çok SpamAssassin (bkz. *Spam Tarayıcıları* (sayfa: 24)) gibi filtreleme yazılımlarının kullandıkları yöntemi andırır.

Böyle listelerin bir diğer kullanım şekli de, SMTP aktarımına koşullu gecikmeler koymuşsanız (nam–ı diğer "katran çukuru") bunların tetiklenmesi için kullanımıdır. Eğer bir konak DNS karalistesindeyse, bu konağın gönderdiği her komuta bir gecikme ile (örn, 20 saniye) yanıt vermek tercih edilebilir. Bu tür gecikmeleri tetiklemek için kullanılabilen başka önkoşullar da vardır; *SMTP Aktarımının Geciktirilmesi* (sayfa: 11) bölümüne bakınız.

3.2.2. DNS Düzgünlük Sınamaları

DNS kullanımının diğer bir yolu karşı tarafın IP adresinde bir ters DNS kaydı olup olmadığına bakmaktır. Bu tür sorguların sonucu, böyle bir kayıt varsa, bir alan adı olacaktır. Bu sonuç özgün IP adresini de içeriyorsa, DNS düzgünlüğü doğrulanmış olur. Aksi takdirde, bağlanan konağın DNS bilgileri geçersiz kabul edilir.

Eğer DNS polislerinden biriyseniz, bunu postaların reddedilmesi için bir önkoşul olarak kullanabilirsniz. Kişisel alanınızın posta alıcısını buna göre yapılandırır, meşru posta göndericilerine, sistem yöneticilerinin DNS kayıtlarını düzeltmelerini istemeleri için onlara uyarılar gönderebilirsiniz. Bunun dışında, DNS düzgünlük sınamalarının sonucu daha kapsamlı bir filtrelemenin parametrelerinden biri olarak kullanılabilir. Ancak, yukarıdaki gibi DNS kayıtları yanlış yapılandırılmış konaklar için sırf ters DNS kaydı düzgün değil diye SMTP aktarım gecikmeleri kullanmak iyi bir fikir olmayabilir.

3.3. SMTP Sınamaları

SMTP diyaloğu başladıktan sonra, karşı taraftan gönderilen komutlar ve bunların parametreleri üzerinde çeşitli sınamalar yapabilirsiniz. Örneğin, selamlaşma sırasında karşı tarafın belirttiği ismin geçerli olup olmadığına bakabilirsiniz.

Bununla birlikte, teslimatı reddetmeye daha SMTP aktarımının başlarında karar verseniz bile bunu hemen yapmamak, bir **RCPT TO:** komutu gelene kadar SMTP aktarım gecikmesiyle göndericiyi bekletmek ve **RCPT TO:** komutunu aldıktan sonra reddetmek daha iyidir.

Bunun sebebi, bazı kalleş yazılımların SMTP aktarımının daha başlarında reddedildiklerini ama bekletilmeye çalışıldıklarını anlamamaları içindir. Ayrıca, bunların reddedilme sebebinin **RCPT TO:** başarısızlığından kaynaklandığını sanmaları da sağlanmış olur.

Bu, küçük de olsa bir katran çukuru yapmak için ayrıca hoş bir vesiledir.

3.3.1. Selamlaşma (HELO/EHLO) Sınamaları

RFC 2821[B104]'e göre, istemci tarafından gönderilecek ilk SMTP komutu EHLO (ya da desteklenmiyorsa HELO) olmalı ve komuta parametre olarak kendi birincil *Nitelikli Alan Adı* (sayfa: 72)nı vermelidir. Bu işleme Selamlaşma (Hello greeting) adı verilir. Eğer anlamlı bir nitelikli alan adı veremiyorsa, istemci köşeli ayraç içine alınmış IP adresini belirtebilir: "[1.2.3.4]". Bu biçime IPv4 adresinin "dizgesel" gösterimi denir.

Anlaşılacağı üzere, bir *Kalleş Yazılım* (sayfa: 72) da selamlaşma sırasında kendi nitelikli alan adını genelde sunar. Ama, kalleş yazılım amacına uygun olarak gönderici konağın kimliğini gizlemeye ve/veya karışıklık yaratmaya ve/veya ileti başlığında "Received:" gibi başlıklarla sunucuyu yanıltmaya çalışır. Bu tür selamlaşma örneklerinden bazıları:

- Alıcı adresindeki kullanıcı ismi gibi niteliksiz isimler (noktasız isimler).
- Çıplak IP adresi (köşeli ayraç içine alınmamış olarak); genellikle sizinki, ama rasgele bir adres de olabilir.
- Sizin alan adınız ya da sunucunuzun nitelikli alan adı.
- `yahoo.com`, `hotmail.com` gibi çok bilinen alan adları.
- Mevcut olmayan alan adları veya isim sunucusu olmayan alanların adları.
- Hiç selamlaşmaz.

3.3.1.1. Basit HELO/EHLO sözdizimi sınamaları

Bu RFC 2821[B107] kurallarına uymayanlara karşı ve bazı *Kalleş Yazılım* (sayfa: 72) türlerininin bilinen belirtileri nedeniyle bu sınamalarını yapmak kolaydır. Böyle selamlaşmaları ya hemen ya da **RCPT TO:** komutundan sonra reddedebilirsiniz.

Öncelikle, selamlaşma sırasında çıplak IP adresi belirtenleri gönül rahatlığıyla reddedebilirsiniz. Eğer RFC 2821[B109]'in zorunlu kıldığı, tavsiye ettiği ya da seçiminize bıraktığı herşeye genel anlamda izin vermekten yanaysanız, bir isim yerine belirtildiğinde IP adreslerinin köşeli ayraç içine alınması gerektiğini aklınızdan çıkarmayın[7]

Özellikle, *sizin* IP adresinizi kullanarak selamlaşmaya girişen konakları sert bir dille yazılmış bir iletiyle reddedebilirsiniz. Bunlar açıkça yalancıdır. Hatta, böyle selamlaşmalara girişenleri uzunca süren (saatlerce) SMTP aktarım gecikmeleriyle kapıda bekletirseniz hiç fena olmaz.

Bu konuda benim kendi deneyimlerim, internette kendilerini dizgesel IP adresi belirterek ([x.y.z.w] gösterimiyle) başka sitelere tanıtan bir meşru site olmadığı gibi bunların internete posta gönderen bütün konakları kendilerinin geçerli *Nitelikli Alan Adı* (sayfa: 72)ndan başka bir isim kullanmamaktadırlar. Dizgesel IP adresi kullanımını sadece yerel ağımdan, o da gönderici SMTP sunucusu olarak benim sunucumu kullanmak üzere yapılandırılmış Ximian Evolution gibi posta istemcilerinden gelirse kabul ediyorum. Yani, dizgesel IP adresi kullananları sadece yerel ağımdan geliyorlarsa kabul ediyorum.

Niteliksiz konak isimlerini (nokta içermeyen konak isimleri) reddedip etmemek size kalmış, Bunların yaygın olarak meşru kabul edildiklerini biliyorum (ama her zaman değil – çifte yanlış olumlama sebebi olabilirler).

Benzer şekilde, geçersiz karakter içeren konak isimlerini reddedebilirsiniz. İnternet alan adları için sadece harfler, rakamlar ve tire işareti geçerli karakterlerdir ve tire işaretine ilk karakter olarak izin verilmez. (Ayrıca, altçizgi karakterini de geçerli bir karakter olarak kabul edebilirsiniz, basitçe yanlış yapılandırmanın bir sonucudur ama Windows istemciler için bu bir yanlış değildir.)

Son olarak, sosyal kişilerin ilk yaptığı şeyi yapmayan yani selamlaşmadan bir **`MAIL FROM:`** komutu gönderen bir istemci ile karşı karşıyaysanız bu bağlantıyı da reddedebilirsiniz.

Kendi sunucularımda, bu sözdizimi sınamalarından geçemeyenleri reddediyorum. Yine de reddetme işlemini **`RCPT TO:`** komutunu alana kadar yapmıyorum. Böyle bir durumda, her SMTP komutuna (**`HELO/EHLO`**, **`MAIL FROM:`**, **`RCPT TO:`**) 20 saniyelik bir aktarım gecikmesi uyguluyorum.

3.3.1.2. Selamlaşmanın DNS üzerinden doğrulanması

Konaklar selamlaşmayı gayet yüzeysel bir manada yaparlar. Selamlaşma, bu sırada belirtilen ismi DNS üzerinden doğrulatmak için en doğru zamandır. Şunları yapabilirsiniz:

- Belirtilen ismi DNS sunucusundan sorgulayıp bağlanan konağın IP adresi ile bu ismin eşleşip eşleşmediğine bakabilirsiniz.
- Bağlanan konağın IP adresine bir ters DNS sorgusu yapıp, gelen ismin selamlaşmada belirtilen isim ile eşleşip eşleşmediğine bakarsınız.

Eğer bu iki sınama da başarılı olursa, isim doğrulanmış olur.

Posta aktarımcınız yerleşik bir seçenek olarak bu sınamayı yapabiliyor olabilir. Örneğin *Exim* (sayfa: 30) için "helo_try_verify_hosts= *" atamasını yapıp, "verify= helo" koşuluna göre işlem yapan ACL'ler oluşturabilirsiniz.

Bu sınama, basit sözdizimsel sınamalardan biraz daha fazla ağ özkaynağı tüketir ve biraz daha fazla işlem süresi gerektirir. Bununla birlikte, sözdizimsel sınamaların aksine, bir eşleşmenin olmayışı bir kalleş yazılımın varlığını işaret etmeyebilir. hotmail.com, yahoo.com ve amazon.com gibi büyük internet sitelerinin selamlaşmaları doğrulanabilir türde değildir.

Eğer, sınama öncesi aktarım gecikmeleri ile göndericiyi zaten oyalamıyorsam, sunucularımda selamlaşma sırasında bir DNS doğrulaması yapıyorum. Bu sınama başarısız olduğu takdirde, her SMTP komutuna 20'şer saniyelik gecikmeler uyguluyorum. Ayrıca ileti başlığına bir "X–HELO–Warning:" ekliyorum ve bunu iletinin tamamı alındıktan sonra olası bir red için *SpamAssassin* (sayfa: 24) puanını arttırmakta kullanıyorum.

3.3.2. Gönderici adresi sınamaları

Bağlanan konak **`MAIL FROM:`** <*adres*> komutunu gönderdikten sonra, bu komutla belirtilen *Zarf Göndericisi* (sayfa: 74) adresini doğrulatmaya çalışabilirsiniz[8].

Gönderici adresinin sözdizimsel sınaması

Belirtilen adres <*yerelkısım*@*alan*> biçimine uygun mu? *alan* parçası sözdizimsel olarak geçerli bir *Nitelikli Alan Adı* (sayfa: 72) mı?

Çoğunlukla, posta aktarımcınız bu sınamaları zaten yapar.

Sahtekarlık sınaması

Siz ya da kullanıcılarınız, postalarını sadece belli başlı sunucular üzerinden gönderiyorsa, diğer konaklardan sizin alan adınızı taşıyan zarf göndericisi adresli olarak gelen iletileri reddedebilirsiniz.

Bu sınamayı da kapsayan daha geniş amaçlı bir sınama *Gönderici Yetkilendirme Dizgesi (SPF)* (sayfa: 20)dır.

Basit gönderici adresi doğrulaması

Adres yerelse, adresin yerel kısmı (@ işaretinden önceki isim) sisteminizdeki geçerli posta kutularından birinin ismi mi?

Adres uzaksa, adresin alanadı kısmı (@ işaretinden sonraki parça) mevcut mu?

Gönderici Varlık Sınaması

(Sender Callout Verification)

Exim ve Postfix gibi bazı posta aktarımcıları tarafından uzak gönderici adresindeki "yerel kısmı" doğrulatmakta kullanılan bir mekanizmadır. Postfix terminolojisinde buna "Gönderici Adresinin Doğrulanması" (Sender Address Verification) adı verilir.

Sunucunuz gönderici adreste belirtilen *alan adı*'nın posta alıcısına bağlanır ve bu adrese posta teslim ediyormuş gibi ikincil bir SMTP aktarımı başlatır. Aslında herhangi bir posta göndermez; bir **`RCPT TO:`** komutunun uzak konak tarafından kabul edilip edilmeyeceğine baktıktan sonra bir **`QUIT`** komutu gönderir.

Exim böyle bir varlık sınamasında öntanımlı olarak boş zarf göndericisi adresi kullanır. Bunun amacı, göndericiye döndürülecek olası bir *Teslimat Durum Bildirimi* (sayfa: 73)nin kabul edilip edilmeyeceğini saptamaktır.

Postfix ise, adresi doğrulamak amacıyla öntanımlı kullanıcı adresi olarak <`postmaster@`*alanadı*> adresini kullanır (*alanadı* parçası `$myorigin` değişkeninden alınır). Boş zarf göndericisi adresine yaptığınız gibi bu gönderici adresine aynı uygulamayı yapabilirsiniz (örneğin, *SMTP Aktarımının Geciktirilmesi* (sayfa: 11) veya *Grilisteleme* (sayfa: 18)'tan kaçınmak için, ama alıcı adreslerde *Zarf Gönderici İmleri* (sayfa: 25) gerekir). Daha fazlası için eklerdeki gerçeklenimlere bakınız.

Bu sınamanın tek başına gelen postayı reddetmek bakımından elverişli olmadığını görebilirsiniz. Ara sıra, örneğin, bankanızın yaptığınız bir ödemenin dekontunu göndermesi gibi durumlarda, meşru posta özdevinimli hale getirilmiş bir mekanizma tarafından geçersiz bir dönüş adresi ile gönderilir. Ayrıca, spamın talihsiz yan etkilerinden biri olarak bazı kullanıcılar, giden postalarına dönüş adresi olarak adreslerini biraz bozarak yazarlar (bu daha çok *Zarf Göndericisi* (sayfa: 74)ni değil de, iletinin "From:" başlığını etkileyebilir).

Dolayısıyla, bu sınama sadece bir adresin geçersizliğini sınamaya yarar, iletinin gerçek göndericisini değil (bir de *Zarf Gönderici İmleri* (sayfa: 25) bölümüne bakınız).

Son olarak, "aol.com" gibi bazı sitelerin raporları vardır. Bunlar gönderici varlık sınaması yaptıklarını keşfettikleri her sistemi koşulsuz karalisteye alacaklarını belirtirler. Bu siteler belki de sıkça *Joe İşi* (sayfa: 72) spamın mağduru olmuşlar ve sonuç olarak, gönderici varlık sınaması fırtınalarına maruz kalmış olabilirler. Siz de bu dağıtık servis reddi (DDoS – Distributed Denial–of–Service) saldırılarının bir parçası haline gelerek kendinizi spamcıların elindeki bir piyona dönüşebilirsiniz.

3.3.3. Alıcı adresinin sınanması

Düşündüğünüz gibi bu basit olmalıdır. Bir alıcı adresi ya mevcuttur ya da değildir. Mevcutsa posta teslim alınır, yoksa posta aktarımcı tarafından öntanımlı olarak reddedilir.

Bir bakalım, öyle mi acaba?

Açık Röleye meydan vermemek

Postaları uzak konaklardan uzak adreslere röleleyemezsiniz! (Gönderici kimliğini kanıtlamadıkça).

Bu çoğumuzun farkında olduğu ama belli ki yeterince önem verilmeyen bir konu. Ayrıca, eposta adresleri ve bunların teslim yolları ile ilgili çeşitli internet standartları ("ünlemli teslimat yolları", "yüzde işaretli alan adları" (bang paths, percent hack domains[B125]) gibi) herkesin elinin altında olmayabilir.

Posta aktarımcınızın bir *Açık Röle* (sayfa: 69) gibi davranıp davranmadığını bilmiyorsanız, bunu "relay–test.mail–abuse.org" üzerinden sınayabilirsiniz. Bunun için kabukta şu komutu vermeniz yeterli olacaktır:

```
telnet relay-test.mail-abuse.org
```

Bu, SMTP sunucunuzun uzak posta adreslerine postayı röleleyip rölelemediğini ve/veya bazı adres türlerini kabul edip etmediğini çeşitli denemeler yaparak sınayan bir servistir.

Sunucularınızın birer açık röle gibi davranmasını önlemek fazlasıyla önemlidir. Eğer sunucunuz bir açık röle ise ve spamcılar sizi bulmuşsa, bellibaşlı DNS karalistelerine kalıcı olarak kaydedilirsiniz. Spamcılardan önce bazı DNS karalistelerince farkedilirseniz (rasgele ve/veya şikayet üzerine yoklanarak), uzunca bir süre kalmak üzere bu DNS karalistelerine kaydedilirsiniz.

Alıcı adresine bakılması

Bu da çoğumuza bayağı görünebilir. Ama öyle değil.

Eğer kullanıcılarınızın posta hesapları ve posta kutuları posta alıcınızın çalıştığı makinede saklanıyorsa, alıcı adresinin yerel kısmının bu posta kutularının isimlerinden biri ile aynı olup olmadığına bakmak kolay olur. Burada bir sorun çıkmaz.

Alıcı adresinin doğrulanmasını güçleştiren iki durum vardır:

- Makineniz alıcı alan adı için yedek posta alıcısı olabilir.
- Makineniz aldığı postayı alanınınızdaki (muhtemelen dahili ağınızdaki) diğer makinelere dağıtıyordur.

Bu durumlarda posta alıcısı konak, alıcı adresini doğrulamaksızın, alıcı adreslerin tümünü herbiri kendi alanları içinde kalmak üzere kabul edebilir. Hedef sunucu alıcı adresin geçersiz olması durumunda bir *Teslimat Durum Bildirimi* (sayfa: 73) üretir. Eninde sonunda, bu işlem dolaylı spam üretimine sebep olur.

Niyetimizi aklımızdan çıkarmadan, bu iki durumda alıcıyı nasıl doğrulayabileceğimize bakalım.

Alıcı Varlık Sınaması

(Recipient Callout Verification)

Bir uzak alıcı adresin yerel kısmını doğrulamak için kullanılan bu mekanizma Exim ve Postfix gibi bazı posta aktarımcılarında mevcuttur bunun nasıl çalıştığı *Gönderici Varlık Sınaması* (sayfa: 16) bölümünde açıklanmıştır. Postfix terminolojisinde bu mekanizmaya "Alıcı Adresi Doğrulaması" (Recipient Address Verification) adı verilir.

Bu durumda, sunucu karşı taraftan **`RCPT TO:`** komutuyla aldığı her alıcı adresini doğrulatmak için hedef sunucuya bağlanmaya çalışır.

Bu çözüm basit ve şıktır. Herhangi bir rehber hizmetine erişmeksizin hedef konakta çalışabilecek herhangi bir posta aktarımcısı ile bu gerçekleştirilebilir. Bununla birlikte, eğer bu posta aktarımcısı alıcı adreslerde bir bulanık eşleşme uyguluyorsa (Lotus Domino sunucuların yaptığı gibi), bu sınama alıcı adresin neticede kabul edilip edilmeyeceğini tam olarak yansıtır ama aşağıda açıklanan mekanizmalar açısından birşeyler yanlış gidebilir.

Özgün *Zarf Göndericisi* (sayfa: 74)nin alıcı varlık sınamaları için değişmeden kalmasına, aksi takdirde, hedef sunucudan dönen yanıtın doğruyu yansıtmayabileceğine dikkat edin. Örneğin, hedef sunucu *Göndericisi olmayan postaları sadece gerçek kullanıcılar için kabul edin* (sayfa: 26) bölümünde açıklandığı gibi sistem kullanıcıları ve takma adları için gönderilen göndericisiz (örn, zarf göndericisi olmayan) postaları reddebilir.

Bellibaşlı posta aktarımcılarından Exim ve Postfix bu mekanizmayı destekler.

Adres Rehberi Hizmetleri

Posta aktarımcınızın sorgulayabileceği bir rehber hizmetinin olması (örn, bir veya daha fazla LDAP sunucusu) diğer bir iyi çözüm olurdu. Çoğu posta aktarımcısı kullanıcı hesap bilgilerini sağlayan LDAP, NIS gibi artalan uygulamalarını kullanabilmektedir.

Asıl can alıcı nokta, epostanın hedef konağının kullanıcı isimleri ile posta kutularını eşleştirmek için böyle bir rehber hizmetini kullanmaması halinde bazı karışıklıkların ortaya çıkabileceğidir (Hem posta alıcısı hem de hedef konak sınamayı aynı kaynaktan yapmalı).

Posta Kutusu Listeleri

Eğer yukarıdaki seçeneklerin hiçbiri uygulanabilir değilse, son çare olarak "yoksul işi bir rehber hizmeti" kullanabilirsiniz. Düzenli aralıklarla posta kutularının listesini, bulundukları makinelerden posta alıcısı makinelerinize kopyalayabilir ve bu listeyi `RCPT TO:` komutlarında belirtilen alıcıları doğrulamak için kullanabilirsiniz.

Eğer, posta kutularını içeren makinelerinizde bir UNIX veya Linux çalışıyorsa, böyle bir listeyi muhtemelen /etc/passwd[B131] dosyasından üretecek ve OpenSSH[B132] paketindeki scp[B133] komutunu kullanarak bu listeyi posta alıcınızın bulunduğu makineye kopyalayacak bir betik yazabilirsiniz. Sonra da bir cron[B134] işi olarak bu betiğin belli aralıklarla çalıştırılmasını sağlayabilirsiniz.

Sözlük Saldırılarının Önlenmesi

Sözlük Saldırısı (Dictionary Attack), çok kullanılan isimleri bazan alfabetik, bazen ters alfabetik bazan da rasgele seçilmiş isimler şeklinde `RCPT TO:` komutlarıyla deneyerek alıcı adreslerinin saptanması şeklinde gelişen SMTP aktarımlarını açıklamakta kullanılan bir terimdir. Böyle bir adresin kabul edilmesi halinde, bu adres spamcının cephaneliğinde yerini alır.

Bazı siteler, özellikle büyük olanları sıklıkla böyle saldırıların hedefi haline gelirler. Spamcılar açısından, çok sayıda kullanıcısı olan sitelerde bir ismin bulunabilme şansı bir kaç kullanıcısı olanlardan daha yüksektir.

Sözlük saldırılarıyla mücadele etmenin tek etkin yolu, her başarısız adreste aktarım gecikmesini arttırmaktır. Örneğin, mevcut olmayan ilk alıcı adresi için bekleme süresi 20 saniye, ikincisinde 30 saniye, 3. için 40 saniye, ... gibi.

Teslimat durum bildirimlerini tek alıcı için kabul edin

Meşru *Teslimat Durum Bildirimi* (sayfa: 73) tek bir alıcı adrese – bildirimi tetikleyen özgün iletiyi yazana – gönderilmiş olmalıdır. *Zarf Göndericisi* (sayfa: 74) adresi boş olan ve birden fazla alıcıya teslimat yapmaya çalışan bağlantıları kesebilirsiniz (drop).

3.4. Grilisteleme

Griliste kavramı Evan Harris tarafından `http://projects.puremagic.com/greylisting/` adresinde açıklanmıştır.

3.4.1. Nasıl Çalışır

Grilisteleme, *Kalleş Yazılım* (sayfa: 72) üzerinden teslim edilmeye çalışılan iletileri ayıklamak için *SMTP Aktarımının Geciktirilmesi* (sayfa: 11) gibi basit ama oldukça etkili bir mekanizmadır. Ana fikir. bir iletinin göndericisi ile alıcısı arasında bir ilişkinin mevcudiyetini sağlamaktır. Çoğu meşru posta için böyle bir ilişki kurulabilir ve teslimat normal şekilde gerçekleşir.

Diğer yandan, böyle bir ilişki evvelce mevcut değilse, teslimat geçici olarak reddedilir (bir `451` SMTP yanıtı ile). Meşru posta aktarımcıları böyle bir durumda biraz gecikmeyle teslimatı yinelerler[9]. Kalleş yazılımlar ise, tersine ya teslimatı hemen yinelerler ya da basitçe vazgeçip adres listelerindeki sonraki hedefe yönelirler.

Bir teslimat sırasında verilen bilgilerden üçü, bir *üçlü* olarak, bir gönderici ile alıcı arasındaki eşsiz ilişkiyi tanımlamak için kullanılır:

- *Zarf Göndericisi* (sayfa: 74).
- Gönderen konağın IP adresi.
- *Zarf Alıcısı* (sayfa: 73).

Bir teslimat reddedilmişse bu üçlü belli bir süre (normalde 1 saat) grilisteli olarak saklanır ve bu sürenin sonunda aklisteye alınır. Belli bir sürenin sonunda (normalde 4 saat) bu üçlü için bir teslimat gerçekleşmemişse, bu üçlü listeden silinir.

Eğer aklisteye alınmış bir üçlüye uzunca bir süre (aylık faturalama dönemli hesaplar düşünülerek, en az bir aylık bir süre) teslimat olmazsa, bu üçlü listeden silinir. Bu işlem listenin sınırsız büyümemesi için yapılır.

Bu zamanaşımları Evan Harris'in grilistelemeyi açıkladığı belgesinden alınmıştır. Bazıları için, grilisteye alınmış üçlülere daha uzun zamanaşımları gerekebilir, çünkü bazı İSS'ler (*earthlink.net* gibi) teslimatlarını her 6 saat ya da buna yakın aralıklarla yinelerler[10].

3.4.2. Çok sayıda posta alıcısı olması durumu

Birden fazla posta alıcısı kullanıyorsanız ve her sunucu kendi grilistesini oluşturuyorsa:

- Belli bir göndericiden kullanıcılarınızdan birine gelen ilk teslimatlar teorik olarak, N posta alıcısı konakların sayısını göstermek üzere N çarpı 1 saatlik ilk gecikme süresi kadar geciktirilir. Bunun sebebi, farklı bir sunucuya yapılan yinelenmiş bir teslimatın bu sunucu açısından ilk teslimat olması ve `451` yanıtıyla reddedilmesidir. En kötü durumda, gönderici konak başa dönüp teslimatı ilk posta alıcısına teslim etmeye çalışması sırasında 4 saatlik süre veya grilistenin ikinci zamanaşımı süresi dolar ve bu kısır döngü gönderici postayı teslim etmekten vazgeçinceye kadar (genelde bu süre 4 gün civarındadır) sürer.

  Uygulamada bu böyle olmaz. Bir teslimat geçici bir başarısızlıkla reddedilirse, gönderici konak hemen diğer posta alıcısından teslimatı gerçekleştirmeye çalışır. Böylece 1 saat sonra bu posta alıcılarından biri bu teslimatı kabul eder.

- Bir üçlü, posta alıcılarınızdan biri tarafından aklisteye alınmış olsa bile, aynı üçlü farklı bir posta alıcısında teslimat için kullanıldığında o sunucu açısından grilisteli olarak işlem görür.

Bu sebeplerden, posta alıcılarınız arasında paylaşılan bir grilisteleme veritabanı gerçekleştirmek iyi bir çözüm olabilir. Ancak, bu veritabanını tutan makinenin başına gelecek bir talihsizlik bütün posta alıcıları için teslimatların başarısız olmasına sebep olacaktır. Bunun yerine veritabanlarının birebir kopyalanması tekniklerinden birini kullanılabilir ve bu sunuculardan birini sorgulara yanıt verecek bir son çare SMTP sunucu yapabilirsiniz.

3.4.3. Sonuç

Şahsi deneyimlerime göre, evvelce açıklanan *SMTP Sınamaları* (sayfa: 13) uygulandıktan sonra grilistelemenin uygulanması halinde bu iki yöntem birlikte döküntü postanın %90'ından kurtulmayı sağlıyor. Eğer grilistelemeyi ilk savunma mekanizması olarak kullanırsanız, gelen döküntü postanın önemli bir kısmını tek başına yakalayacaktır.

Buna karşın, bu tekniğin kullanımından sıfıra yakın *Hatalı Olumlama* (sayfa: 71) ortaya çıkar. Bellibaşlı *posta aktarımcılarının* (sayfa: 72) tümü bir geçici başarısızlıktan sonra eninde sonunda başarılı bir teslimatı gerçekleştirmek amacıyla teslimat yinelemeleri yaparlar.

Grilistelemenin perde arkasında, birinin belli bir alıcıya bir saatlik gecikmeye konu olarak hemen teslim alınmamış bir meşru posta vardır (bu bir saatlik gecikme, çok sayıda posta alıcısı kullanıyorsanız bir kaç saat de olabilir).

Ayrıca, *Soru 5.1.* (sayfa: 28)'in yanıtına da bakınız.

3.5. Gönderici Yetkilendirme Şemaları

Gönderici adresinin sınanması anlamında, sadece göndericinin varlığının değil, ayrıca kimliğinin de kanıtlanmasını sağlayacak çeşitli kullanıcı doğrulama şemaları geliştirilmiştir. İnternet alanının sahibi kendi alanındaki göndericilerden teslimata yetkili olanları belirleyen bazı kurallar belirtir.

Bu şemaların ilklerinden iki örnek:

- `MAIL-FROM` MX kayıtları, Paul Vixie <paul (at) vix.com> tarafından tasarlanmıştır.
- Ters Posta Alıcısı (RMX – Reverse Mail Exchanger) kayıtları (DNS'ye ek olarak); Hadmut Danisch <hadmut (at) danisch.de> tarafından tasarlanmış ve yayınlanmıştır.

Bu iki şema altında, `kullanıcı@alanadı.dom` adresli tüm postalar `alanadı.dom`'un DNS kayıtlarında bulunan konaklardan gelmek zorundadır.

Bu iki şema gelişmiş, hatta benzer çalışmalara çatallanmıştır.

3.5.1. Gönderici Yetkilendirme Dizgesi (SPF)

SPF ("Sender Policy Framework" veya "Sender Permitted From" kısaltması), gönderici yetkilendirme için en iyi bilinen şemalardan biridir. Yukarıdaki şemalardan hareketle ortaya çıkmıştır ama kuralları belirtmek bakımından biraz daha esnek bir yapıdadır.

SPF bilgisi bir alan adının üst düzey DNS kayıtları arasında bir `TXT` kaydı olarak görünür. Bu kayıt bu alanın kullanıcısının ağzından basitçe şunu der: "Ben postamı sadece bu makinelerden gönderiyorum. Eğer başka bir makine benim postamı oradan gönderdiğini iddia ediyorsa, o bir yalancıdır." Bu kayıtta şunlar belirtilmiş olabilir:

- bu alandan posta göndermesine izin verilen makineler
- bu alandan giden postada zorunlu bir GPG imzasının varlığı
- diğer kurallar; ayrıntılar için `http://www.openspf.org/` adresine bakınız.

Bu **`TXT`** kaydının geliştirilmesi hala sürmektedir, yine de temel özellikler yukarıda bahsedildiği gibidir. Bir `v=spf1` dizgesi ile başlar ve şu belirteçlerden bazıları ya da hepsi kullanılabilir:

- `a` – geçerli gönderici makine bu alanın kendi IP adresidir.
- `mx` – bu alanın posta alıcıları, ayrıca geçerli göndericilerdir.
- `ptr` – eğer gönderenin IP adresi için ters DNS kaydındaki isim, gönderici adresin alan adı kısmındaki isimle eşleşiyorsa, gönderen konak geçerli göndericidir.

Bu belirteçlerin herbirinin önüne bir yetkili kaynak olduğunu belirtmek için bir artı işareti (bu öntanımlıdır), yetkisiz olduğunu belirtmek için bir eksi işareti, yetki bakımından nötr olduğunu belirtmek için soru işareti veya yetkisiz olarak değerlendirilebilineceğini belirtmek üzere bir yaklaşık işareti (~) konabilir.

Her belirteç bir ikinokta üstüste işaretinden sonra bir alan adı belirtmek üzere kullanılabilir. Örneğin, bir Comcast müşterisiyseniz, sizin DNS kayıtlarınız arasında "`v=spf1 -ptr:client.comcast.net`

`ptr:comcast.net -all`" şeklinde bir **TXT** kaydı olabilir. Bu kayıt, bu alandan posta gönderen makinenin IP adresi çözümlendiğinde elde edilen isim *birşey*.client.comcast.net şeklindeyse bu adres yetkisizdir, *birşey*.comcast.net şeklindeyse yetkilidir, belirtilenler dışında kalanlar da yetkisizdir ("–all") anlamına gelir.

Her alan adı için bir SPF kaydı bulunmalıdır. Bazı büyük siteler artık bu kaydın bulunmadığı alanlardan posta kabul etmemektedir.

Gönderici yetkilendirme şemaları genelde kabul görmemiş olmasına rağmen SPF evrensel olarak büyük oranda kabul görmüştür. SPF'ye karşı çıkanlar, alan adı sahiplerinin posta gönderen müşterileri/kullanıcıları üzerinde bir tekel kurmak için bunu kullanabileceklerini ileri sürmektedirle[11].

Diğer bir idda, SPF'nin geleneksel eposta yönlendirmesini bozduğu şeklindedir; yönlendiren konak, zarf göndericisinin alan adındaki SPF kaydında yetkisiz olabilir. Bu sorun, Göndericiyi Yeniden Yazma Şeması[B148] (SRS – Sender Rewriting Scheme) ile kısmen halledilebilir. SRS'de postanın yönlendiricisi *Zarf Göndericisi* (sayfa: 74) adresinin biçimini değiştirir:

```
kullanıcı=kaynak.alanadı@yönlendirici.alanadı
```

3.5.2. Epostalar için Microsoft Çağrı Kimliği

Kurallarının gönderici alan adının DNS bilgileri arasında bir TXT kaydı olarak görünmesi bakımından SPF'ye benzer. Ancak, basit anahtar sözcükler yerine, XML olarak kodlanmış oldukça geniş kapsamlı MS CIDE bilgilerinden oluşur. Bu XML şeması Microsoft tarafından bir lisans altında yayınlanmıştır.

SPF, bir postanın sadece *Zarf Göndericisi* (sayfa: 74) adresine bakarak çalışırken, MS CIDE iletinin RFC 2822[B151] başlıklarını değerlendiren bir araç olarak karşımıza çıkar. Böyle bir sınamanın SMTP aktarımında yapılabileceği en erken nokta, ileti verisi alındıktan sonra ve son **250** yanıtını göndermeden öncedir.

Dobra dobra ölü doğmuş denebilir. Patentiyle ve karmaşıklığıyla bir engelli olarak doğmuş da denebilir.

`http://www.openspf.org/`'da SPF'ye ek olarak MS Çağrı kimliğini de (MS CIDE) sınayacak araçlar bulabilirsiniz.

3.5.3. RMX++

Basit Çağrıcı Yetkilendirme Yapısının (SCAF – Simple Caller Authorization Framework) bir parçası. Bu şema, zaten özgün RMX'in tasarımcısı olan Hadmut Danisch tarafından geliştirilmiştir.

RMX++, HTTP sunucular üzerinden özdevimli yetkilendirmeyi mümkün kılar. Alanadı sahibi DNS üzerinden bir sunucu belirtir ve posta alıcısı konak bu sunucuya bağlanarak göndericinin geçerliliğini saptamak için oradan bir yetkilendirme kaydı elde etmeye çalışır.

Bu şema alanadı sahibine gönderici adreslerini yetkilendirmede kullanılacak kuralları daha ayrıntılı belirleme imkanı verir (SPF kayıtlarıyla, ağının yapısını kamuya açık alanlarda ilan etmeksizin). Hadmut'tan bir örnek: Hergün iş saatleri dışında belli bir adresten beş iletiden fazlasına izin vermeyen bir yetkilendirme sunucusu bu sınır aşıldığında bir uyarı verecektir.

Keza, SCAF epostalarla sınırlı değildir, ayrıca IP üzerinden sesli iletişim (VoIP) gibi hizmetler için çağrıcı yetkilendirmesi için kullanılabilir.

Rick Stewart <rick.stewart (at) theinternetco.net> RMX++'nın makine ve ağ kaynaklarına etkisine dikkat çekerek RMX++'in perde arkasında kalan bir olasılıktan sözetmiştir: HTTP sunucularının yanıtları DNS sunucularının ki gibi geniş çapta önbelleklenmediğinden bir HTTP isteği yapmak bir DNS isteğinden kat kat pahalıya malolacaktır.

Rick devam ediyor, RMX++'nın özdevimli doğası bir başarısızlığın nedenlerinin bulunmasını da zorlaştıracaktır. Eğer günlük beş iletilik bir sınır varsa, bu sınır, tek bir iletinin beş kere sınanması ile dolacaktır. Yani şema, bir iletinin defalarca sınanmasına imkan vermiyor.

RMX, RMX++ ve SCAF hakkında daha fazla bilgi edinmek için
http://www.danisch.de/work/security/antispam.html adresine bakınız.

3.6. İleti verisinin sınanması

İletinin içeriğine bakmanın zamanı geldi. Bu, ileti tamamen kabul edildikten sonra spam ve virüs tarayıcılarının yaptıklarına benzer bir işlemdir. Ancak, bizim durumumuzda bu sınamaları, posta kabul edilmeden yani, sonuncu `250` yanıtından önce yapacağız. Böylece, daha sonra reddederek *Dolaylı Spam* (sayfa: 71) üretmeksizin postayı daha SMTP aktarımı bitmeden reddetme şansımız olacak.

Eğer posta alıcılarınız çok meşgulse (büyük bir site ve bir kaç tane MX), bu makinelerde bu sınamaların bir kaçını bile uygulamak oldukça pahalıya malolabilir. Özellikle, *Virüs Tarayıcıları* (sayfa: 23) ve *Spam Tarayıcıları* (sayfa: 24)'nın çalıştırılması büyük miktarda işlemci zamanı harcanmasına sebep olur.

Böyle bir durumda, bu tarama işlemlerine adanmış bir makine ayarlamak iyi bir çözüm olabilir. Sunucu tarafında çalışabilen çoğu anti–spam ve anti–virus yazılım ağ üzerinden çalıştırılabilmektedir.

3.6.1. İleti başlıklarının sınanması

Eksik başlık satırları

RFC 2822[B157] bir iletinin en azından şu başlıkları içermesini *zorunlu* kılar:

```
From: ...
To: ...
Subject: ...
Message-ID: ...
Date: ...
```

Bu satırlardan herhangi birinin yokluğu iletinin bir *Posta İstemcisi* (sayfa: 73) tarafından üretilmediğini ve büyük olasılıkla bir döküntü posta olduğunu gösterir[12].

Başlık Adresinin Sözdizimi Sınaması

İleti başlığındaki adresler (`To:`, `Cc:`, `From:` ... başlıkları), sözdizimsel olarak geçerli olmak zorundadır. Daha fazla birşey söylemeye gerek yok.

Basit Başlık Adresi Doğrulaması

İleti başlığındaki her adres için:

- Eğer adres yerelse, *yerel kısım* (@ işaretinden önceki parça) geçerli bir posta kutusu ismi mi?
- Eğer adres uzaksa, *alanadı parçası* (@ işaretinden sonraki parça) mevcut mu?

Başlık Adresi Varlık Doğrulaması

Bu, *Gönderici Varlık Sınaması* (sayfa: 16) ve *Alıcı Varlık Sınaması* (sayfa: 17)'na benzer şekilde çalışır. Bir *Teslimat Durum Bildirimi* (sayfa: 73)'nın kabul edilip edilmeyeceğini saptamak için her gönderici adresin birincil posta alıcısına erişilerek adres doğrulatılmaya çalışılır.

3.6.2. Döküntü Posta İmza Depoları

Döküntü postayı diğerlerinden ayıran bir özellik, çok sayıda adrese gönderilmiş olmasıdır. Eğer 50 alıcı belli bir iletiyi spam olarak nitelemişse, posta size teslim edilirken, iletiyi kabul edip etmemek noktasında neden bu fiili durumu kullanmayasınız? Daha da iyisi, spamcıları bilen kamuya açık bir havuzu beslemek için neden bir *Spam Tuzağı* (sayfa: 73) ayarlamayasınız?

Böyle havuzlar var:

- Razor[B165]
- Pyzor[B166]
- Distributed Checksum Clearinghouse (DCC)[B167]

Bu araçlar sadece, döküntü posta olarak bilinen bir iletinin bir eşdeğer kopyasını aldığınızda tetiklenen basit imza sınamaları yaparak çalışırlar. Bunlar ileti içinde bilinen kalıpları arayarak değil, ileti başlığındaki ve gövdesindeki belli değişiklikleri hesaba katarak değerlendirme yaparlar.

3.6.3. Baskı karakteri olmayan karakterlerin varlığı

Baskı karakteri olmayan karakterleri içeren iletilere nadir de olsa rastlanır. Böyle bir ileti, hemen hemen daima bir virüs ya da uygun bir MIME kodlaması olmaksızın batı dillerinde yazılmamış bir spam türü olarak karşımıza çıkar.

Özel bir durum, iletinin boş karakter (sıfırıncı karakter – '\0') içermesi durumudur. Baskı karakteri olmayan karakterlerin karmaşıklığı karşısında bunun yararından çok zararı olacağını düşünseniz bile en azından bu karakteri sınamayı düşünebilirsiniz. Çünkü, Cyrus Posta Araçları[B168] gibi bazı *posta teslimatçıları* (sayfa: 73) bu karakteri içeren postaları eninde sonunda reddedecektir[13]. Eğer böyle bir yazılım kullanıyorsanız, boş karakterlerden kurtulmayı kesinlikle hesaba katmalısınız.

Diğer taraftan, RFC 822 belirtimi (artık atıl) iletilerde boş karakteri açıkça yasaklamamıştır. Bu sebeple, bu tür postaları reddetmek yerine, iletiden bu karakterleri ayıkladıktan sonra postayı Cyrus'a teslim etmek daha iyi bir yol olabilir.

3.6.4. MIME sınamaları

Benzer şekilde, gelen iletinin MIME yapısı değerlendirmeye alınırsa bu işlem çekilen zahmete değebilir. MIME çözümleme hataları veya tutarsızlıkları çok sık ortaya çıkmaz; fakat bu olursa, ileti kesinlikle döküntüdür. Üstelik böyle hatalar bundan sonraki *Eklenti Sınamaları* (sayfa: 23), *Virüs Tarayıcıları* (sayfa: 23), *Spam Tarayıcıları* (sayfa: 24) gibi sınamalar açısından ortaya çıkabilecek sorunların habercisi olabilir.

Başka bir deyişle, eğer MIME kodlaması kuraldışıysa iletiyi reddedin.

3.6.5. Eklenti Sınamaları

Son zamanlarda istemediğiniz halde birileri size bir Windows ekran koruyucusu (".scr") veya bir Windows Program bilgi dosyası (".pif") gönderdi mi?

Ekinde "Windows çalıştırılabilirleri" – yukarıdaki gibi üç harfli uzantıları olan dosyalar – olan iletileri önlemeyi gözönünde bulundurun. Bu sınama *Virüs Tarayıcıları* (sayfa: 23)nın tükettiğinden daha az sistem kaynağı tüketeceği gibi ayrıca, anti–virus tarayıcınızda henüz imzası bulunmayan yeni virüsleri de yakalamanızı sağlayacaktır.

Bu tür dosya uzantılarının az çok kapsamlı bir listesini `http://support.microsoft.com/default.aspx?scid=kb;EN-«US;290497` adresinde bulabilirsiniz.

3.6.6. Virüs Tarayıcıları

Sunucu tarafında çalışan bir miktar virüs tarayıcı mevcuttur. Bazıları:

- Sophie[B175]
- KAVDaemon[B176]
- ClamAV[B177]

- DrWeb[B178]

Tehlike arzetmesi olası dosyaları (".zip" dosyaları gibi) sadece isimlerine bakarak engellemek istemiyorsanız böyle tarayıcılar işe yarar. Ayrıca bunlar bir posta eki olmayan "Bagle.R" virüsü (2004 Mart'ında ortaya çıkmıştı) gibi virüsleri de yakalayabilir.

Çoğu durumda, virüs taraması yapan makinenin sizin posta alıcınız olması gerekmez. Bu virüs tarayıcıların çoğu bir ağ bağlantısı üzerinden başka bir konakta çalıştırılabilir.

Antivirüs yazılımları genellikle bilinen virüsleri imzalarından ya da *virüs tanımlarından* saptar. Yeni virüsler geliştirildiğinden bunların düzenli aralıklarla güncellenmeleri gerekir. Ayrıca, olası en yüksek doğruluğun sağlanması için yazılımın kendisinin de güncel olması gerekir.

3.6.7. Spam Tarayıcıları

Benzer şekilde, anti–spam yazılımlar iletileri standartlara uyumluluk açısından değerlendirerek, *DNS Karalisteleri* (sayfa: 13) ve *Döküntü Posta İmza Depoları* (sayfa: 22) gibi sınamalar yaparak ve içeriklerini büyükçe bir ampirik kümeye göre değerlendirerek sınıflandırmakta kullanılabilir. Sonuçta, bu tür yazılımlar iletinin ne derece spam olabileceğini belirten bir puan verirler. Eğer bu puan belli bir eşik değeri aşmışsa iletinin spam olabileceğine karar verilebilir.

Sunucu tarafında çalışan ampirik antispam filtrelerinden çok tanınmış ikisi:

- SpamAssassin[B181]
- BrightMail[B182]

Spamcılar bu araçların kullandıkları çeşitli sınamaları alt edecek yöntemler (örn, "GR0W IO 1NCH35" gibi oldukça yaratıcı dizgeler) buldukça bu araçlar da sürekli geliştirilmektedir. Antivirüs yazılımlarında olduğu gibi bu yazılımlar da olası en yüksek doğruluğu sağlamak için sıkça güncellenmelidir.

Ben SpamAssassin kullanıyorum, makine özkaynaklarına olan etkisini en aza indirmek için onu savunma hattımın önüne yerleştirmedim. Benim kişisel adresime günde 500 civarında döküntü posta teslim edilmeye çalışılıyor. Bunlardan yaklaşık 50 tanesi SpamAssassin tarafından sınanacağı noktaya ulaşabiliyorlar. Bu 50 iletinin dışında posta kutuma her iki ya da üç günde bir, tek bir ileti düşüyor.

3.7. Dolaylı Spamın Engellenmesi

Dolaylı Spam (sayfa: 71)ın şimdiye kadar açıklanan tekniklerle engellenmesi zordur, çünkü bunlar normalde standart posta aktarımcılarını (Sendmail, Postfix veya Exim gibi) kullanan meşru sitelerden gelirler. Asıl sorun, kendi kullanıcılarınızın gönderdikleri postalara yanıt olarak gelen geçerli *teslimat durum bildirimleri* (sayfa: 73)nden bu iletileri ayırmaktır. Bu ayrımı yapanların kullandıkları yöntemlerden bazıları:

3.7.1. Hatalı Virüs Uyarıları Filtresi

Çoğu zaman, dolaylı spam antivirüs tarayıcılarının ürettiği virüs uyarıları şeklinde karşımıza çıkar[14]. Bu virüs uyarılarının `Subject:` satırı dahil birçok karakteristik özelliği antivirüs yazılımının kendisi tarafından oluşturulur. Dolayısıyla, ortak karakteristik özelliklerin bir listesini yapıp böyle hatalı virüs uyarılarını filtreleyebilirsiniz.

Ne yazık ki, şansınız yok — birileri bunu zaten sizin için yapmış. `:-)`

Tim Jackson <tim (at) timj.co.uk>, *SpamAssassin* (sayfa: 24) ile kullanmak için bir hatalı virüs uyarı listesi yapmış bile. Bu listeyi `http://www.timj.co.uk/linux/bogus-virus-warnings.cf` adresinden edinebilirsiniz.

3.7.2. Alanınız için SPF kaydı oluşturun

Gönderici Yetkilendirme Dizgesi (SPF) (sayfa: 20)'nin amacı özellikle *Joe İşi* (sayfa: 72)nden korunmaktır. Yani geçerli bir eposta adresinin taklit edilmesini önlemektir.

Eğer alanadınızın DNS bilgileri arasında bir SPF kaydı varsa, SPF sınamaları yapan alıcı konaklar, taklit edilmiş adreslerle gönderilmiş postaları kabul etmeyecektir. Böyle bir durumda da, size bir *Teslimat Durum Bildirimi* (sayfa: 73) gönderilmeyecektir.

3.7.3. Zarf Gönderici İmleri

Benim kendim için de denemekte olduğum bir başka farklı yaklaşım, giden postanın *Zarf Göndericisi* (sayfa: 74) adresinin yerel kısmına bir dizgecik eklemek ve *teslimat durum bildirimleri* (sayfa: 73)ni kabul etmeden önce *Zarf Alıcısı* (sayfa: 73) adresinde bu dizgeciğin varlığına bakmaktır. Örnek olarak, böyle bir gönderici adresinin biçimi şöyle olabilir:

yerelkısım=dizgecik@alanadı

Normal ileti yanıtları bundan etkilenmez. Çünkü bu yanıtlar, bu işlem sırasında içeriğine dokunulmayan `From:` veya `Reply-To:` başlıklarındaki adreslerle yapılır.

Söylemesi kolay, değil mi? Maalesef, bu amaca uygun bir imleme biraz karmaşık bir işlem. Hesaba katılması gereken bir takım olumsuz durumlar mevcut:

- Bu yöntemin yararlı olabilmesi için, imli gönderici adresini spamcıların kullanamaması lazım. İm olarak bir zaman damgası kullanıp bir süre sonra kullanışsız hale gelecek bir adres oluşturulabilir:
  gönderen=zamandamgası=dizgecik@alanadı
- Postanızın *Grilisteleme* (sayfa: 18) yapan bir siteye de gidebileceğini gözönüne alırsak, zarf gönderici adresinizin belli bir alıcı için değişmez olması gerekir, aksi halde sürekli grilistede kalırsınız.

  Bu durumda, *Zarf Alıcısı* (sayfa: 73)na uygun bir *Zarf Göndericisi* (sayfa: 74) üretebilirsiniz:
  gönderen=alıcı=alıcının.alanadı=dizgecik@alanadı

  Bu adres zamanaşımına uğramayacağından, bu adresle ilgili döküntü postalar görmeye başlarsanız, en azından kaçağın kaynağını öğrenmiş olursunuz. Bununla birlikte, aynı alıcıdan size gelen normal teslimatları etkilemeksizin, imli adresinize bu alıcıdan gelen postaları kolayca reddebilirsiniz.
- Posta listelerinin sunucuları ile ilgili iki durum vardır. Genellikle, sunuculara yapılan isteklerde ("subscribe"/"unsubscribe" gibi), yanıtlar zarf gönderici adresi boş bırakılarak gönderilir.
  - İlk durum, sunucunun istek postasının *Zarf Göndericisi* (sayfa: 74) adresine gönderdiği yanıt ile ilgilidir. Posta listesi sunucusu ile ilgili sorun, komutların (**`subscribe`** veya **`unsubscribe`** gibi) genellikle farklı adreslere (<discuss (at) en.tldp.org> listesi için <discuss–subscribe (at) en.tldp.org> ve <discuss–unsubscribe (at) en.tldp.org> gibi) gönderilmesidir. Böyle bir durumda, üyelik adresi ile listeye gönderilecek iletinin gönderici adresi farklı olacaktır — ve bu örnekte, ayrıca, üyelikten çıkma isteğinde kullanılan adresten de farklı olacaktır. Sonuçta, ne listeye posta göndermek ne de üyelikten çıkmak mümkün olacaktır.

    Bu durumda uzlaşma ancak gönderici adresinde imleme için sadece alıcının alan adının bulunmasıyla sağlanabilir. Yani, gönderici adresi şöyle üretilebilir:
    üyeismi=en.tldp.org=dizgecik@üyelik.alanadı
  - İkinci durum, yanıtların, istek postasının (<spam–l–request (at) peach.ease.lsoft.com> gibi bir adrese istek yapılması gibi) ileti başlığındaki yanıtlama adresine gönderilmesi ile ilgilidir. Bu adres imli olmayacağından, liste sunucusundan gelen yanıt sunucunuz tarafından reddedilecektir.

Postayı imsiz alıcı adresine yollayan bu tür sunucuları "aklisteye" almaktan başka yapabileceğiniz pek birşey yoktur.

Bu noktada bu yaklaşım kenarından köşesinden kırılmaya başlar. Bununla birlikte, özgün postası sizin sunucunuzdan gönderilmemiş meşru teslimat durum bildirimleri de reddedilir. Bu durumda, postalarını sadece sizin denetiminizde olan sunucular üzerinden gönderen kullanıcılarınız için bunu yapmayı düşünebilirsiniz.

Sonuç olarak, yukarıda bahsedilen durumların hiçbirinin mevcut olmadığı durumlarda, bu yöntem dolaylı spamı engelleme imkanı vermekten başka, bunları üreten site sahiplerini eğitme imkanı da verir. Bunun yanında, yararlı bir yan etki olarak, *Gönderici Varlık Sınaması* (sayfa: 16) yapan siteler, özgün postanın sadece sizin sunucunuzdan gitmiş olması durumunda bir olumlu yanıt alacaklardır. Özünde, spamcılar tarafından gönderici adreslerin taklit edilmesine maruz kalma şansını düşürürsünüz.

Kullanıcılarınızın giden postalarında imli adresler belirtmek isteyip istemediklerine bağlı olarak, adreslerinin imsiz türlerine dönen postalara izin verecek konakları belirleyebilirsiniz. Örneğin, bu kullanıcılar posta sunucunuzun aynı zamanda sisteme kayıtlı kullanıcıları iseler, onların ev dizinlerindeki belli bir dosyanın içeriğine bakarak bunu yapabilirsiniz.

3.7.4. Göndericisi olmayan postaları sadece gerçek kullanıcılar için kabul edin

Zarf gönderici imleri ile ilgili sınamalar yapıyor olsanız bile, göndericisiz postaların istenmeyenleri arasından kaçanlar olabilir. Özellikle, bu şemayı yeğleyen kullanıcılarınız varsa, `postmaster` veya `mailer-daemon` gibi takma adlara gönderilmiş postalarda böyle bir imin varlığına bakmazsınız. Mantıken, bu takma adlı kullanıcılar için giden bir posta olmayacağından, bunlara göndericisiz postalar da gelmemesi gerekir.

Bu tür takma adlı kullanıcılar için hatta, alıcı adresi için bir posta kutusu olmayan postaları reddedebilirsiniz.

4. Dikkate Alınacak Diğer Hususlar

Sistem çapında filtrelemenin bir sonucu olarak devreye giren bazı özel durumlar vardır. Burada bunlar üzerinde duracağız.

4.1. Çok Sayıda Posta Alıcısı

Çoğu DNS kaydı birden fazla *Posta Alıcısı* (sayfa: 72) (MX) belirtir. Siz de birden fazla posta alıcısı kullanıyorsanız, birincil posta alıcınızda uyguladığınız filtrelemeleri diğerleri için de yapmalısınız. Aksi takdirde, postayı birincil posta alıcınıza teslim edemeyen gönderici konak teslimatı yedek posta alıcılarınızda yapacaktır.

Eğer yedek posta alıcılarınız sizin denetiminiz altında değilse, *çok sayıda posta alıcısına gerçekten ihtiyacım var mı* diye kendi kendinize sormalısınız. Yedek posta sunucularınız, gelen postaları alıp sizin birincil posta alıcınıza göndermekten başka bir şey yapmıyorsa, *gereksiz* posta sunucuları olarak davranıyorlar demektir. Böyle bir durum varsa, şüphesiz onlara ihtiyacınız yok demektir. Birincil posta sunucunuz bir süre için devre dışı kalacaksa, bunlar devreye sokulabilir, yine de işini iyi yapan konakların postayı teslim etmek için günlerce işlemi yineleyeceklerini aklınızdan çıkarmayın[14].

Çok sayıda posta alıcısı gerektiren bir durum da çeşitli sunucular arasında yükün dağıtılmak istendiği durumdur. Örneğin o kadar çok posta geliyordur ki, bir makine tek başına bunlara yetmiyordur. Bu durumda bu gereksinimi ortadan kaldırmak için bazı görevleri (*virus* (sayfa: 23) ve *spam* (sayfa: 24) tarayıcıları gibi) diğer makinelere paylaştırabilirsiniz.

Tekrar belirtelim, eğer çok sayıda posta alıcısı kullanacaksanız, yedek sunucularınız da en azından birincil sunucu kadar filtreleme yapmıyorsa, birincil sunucunun yaptığı filtrelemenin bir faydası olmayacaktır.

Çok sayıda posta alıcısı kullanımı ile ilgili diğer durumlar için *Grilisteleme* (sayfa: 18) bölümüne de bakınız.

4.2. Diğer SMTP Sunucularına Erişimin Engellenmesi

DNS kaydınızda *Posta Alıcısı* (sayfa: 72) olarak listelenmemiş SMTP sunucularınız varsa, bunlara internetten gelen bağlantıları kabul etmemelisiniz. İnternetten gelen tüm posta trafiğini sadece posta alıcılarınız kabul etmelidir.

Bu husus SMTP sunucuları ile sınırlı değildir. Eğer sitenizde sadece dahili kullanıma yönelik hizmet sunan makineler varsa, bunlara erişimi güvenlik duvarı kullanarak sınırlamalısınız.

Bu bir kural olmakla beraber istisnaları da vardır. Eğer bunların neler olduğunu dair bir fikriniz yoksa bu kural tam size göredir.

4.3. Yönlendirilen Postalar

Aşağıdaki gibi "dost" kaynaklardan yönlendirilmiş postaları spam filtrelemenin bir sonucu olarak reddetme konusunda dikkatli olmalısınız:

- Varsa, yedek posta alıcılarınız.Bunların döküntü postanın çoğunu zaten filtrelediğini varsayarak (bkz. *Çok Sayıda Posta Alıcısı* (sayfa: 26)).
- Sizin ya da kullanıcılarınızın üyesi oldukları posta listeleri. Böyle postaları yine de filtreleyebilirsiniz (bir kara delikte kaybolmaları hayati önemde olmayabilir). Ancak, postayı reddetmeniz, liste sunucusunun bu üyeyi üyelikten çıkarmasına sebep olabilir.
- Alıcının diğer hesapları. Tekrar belirtelim, postaların reddedilmesi dolaylı spam üretecek ve/veya posta yönlendiren konaklarda sorunlara yolaçacaktır.

Son iki kaynak ile ilgili bir mantıksal çıkarım yapabiliriz: Bu iki kaynak kullanıcıya özeldir. Dolayısıyla, şu sorular akla gelir: Aklisteye alınmasını istedikleri konakları belirtebilmeleri için kullanıcılara nasıl bir imkan sunabilirim ve böyle ayrı ayrı aklisteleri sistem çapında SMTP sırasında filtreleme yaparken nasıl kullanabilirim? Eğer, çeşitli alıcılarıma bu iletiler bir yönlendirmenin sonucu olarak geliyorsa (posta listeleri buna bir örnektir), kullanacağım aklisteye nasıl karar vereceğim?

Sihirli bir reçetemiz yok. Bu durumlardan her biri birazcık çalışma yapmayı gerektirir. Alıcılarınızın herhangi birinin aklistesindeki konaklardan posta geldikçe, bunları spam sınıflamasına sokmaksızın kabul etmeye karar verebilirsiniz. Örneğin, her `RCPT TO:` komutunda bu alıcının aklistesinde gönderici konağın bulunup bulunmadığına bakarsınız. Eğer varsa, daha sonra reddedilmesini önlemek için bir bayrak kullanırsınız. Daha da verimli olması açısından, alıcıların aklistelerinden üretilmiş bir *ortak* akliste kullanabilirsiniz.

Eklerdeki gerçeklenimlerde daha fazla ayrıntı bulabilirsiniz.

4.4. Kullanıcı Verileri ve Ayarları

Sitedeki her kullanıcı için verileri ve ayarları desteklemek isteyebileceğiniz başka durumlar da mevcuttur. Örneğin, gelen postayı SpamAssassin (bkz. *Spam Tarayıcıları* (sayfa: 24)) ile tarıyorsanız, spam eşikleri, kabul edilecek dil ve karakter kodlamaları ve Bayes eğitimi/verileri gibi ayarları kullanıcıya özel hale getirmeyi ve bu alıcılara gelen postaları bu ayarlara göre taramayı düşünebilirsiniz.

Bu sadece, SMTP sırasında gelen posta belli bir alıcı için teslim alınmadan önce sistem seviyesinde yapılabilir ve böyle bir işlem kişisel tercihlere çok iyi uyum sağlamaz. Tek bir iletinin çok sayıda alıcısı olabilir; ve *Yönlendirilen Postalar* (sayfa: 27)daki durumun aksine, alıcıların tercihlerinin bir toplamı olarak ortak bir aklistenin kullanımı iyi bir seçenek olmaz. Kullanıcılarınızın anadillerinin farklı farklı olduğu bir senaryo mevcut olabilir.

Anlaşılacağı üzere, bu duruma uygun bir değişiklik yapılabilir. İşin püf noktası, gelen iletinin alıcılarının sayısını birle sınırlamaktır, böylece ileti kullanıcının ayarlarına ve verilerine uygun olarak analiz edilebilir.

Bunu yapmak için, ilk `RCPT TO:` komutunu aldıktan sonra diğer komutlara bir SMTP `451` (defer) yanıtı verilir. Eğer gönderici posta aktarımcısı işini bilen bir aktarımcı ise bu yanıtı nasıl yorumlayacağını bilir ve teslimatı daha sonra yineler. (Eğer bunu yapmazsa, büyük ihtimalle bu postanın göndericisi sizin kendisinden posta almak istemediklerinizden biridir.)

Açıkça bu durumu kurtarmaya çalışmaktan başka bir şey değildir. Çok sayıda alıcıya gönderilmiş postalar her alıcı için yarım saat veya daha uzun bir süre gecikir. Bilhassa postanın zamanında ulaşmasının önemli olduğu şirket ortamlarında, gerek şirket için de gerekse dışında postaların teslim alınıp alınmaması dolayısıyla işlerin zamanında yapılması ile ilgili tartışmalar görülmeye başlar. Bu gibi ortamlar sözkonusu olduğunda bu iyi bir çözüm değildir.

Özellikle tüzel girişimler ve çok büyük sitelerle ilgili diğer bir durum, gelen postanın teslimat için dahili makinelere yönlendirildiği durumdur. Bu durumda, posta alıcısında kullanıcıların hesapları bulunmaz. Ama hala veritabanı ya da LDAP sorguları gibi işlemlerle kullanıcıya özel verileri desteklemek mümkündür. Yine de, attığınız taş ürküttüğünüz kurbağaya değer mi, bu tartışılır.

Ama şu olur, küçük bir siteyseniz ve teslimatların gecikmesinden korkunuz yoksa, her kullanıcının kendi filtreleme koşullarına uygun filtreleme yapabilirsiniz.

5. Sorular ve Cevaplar

Bu bölümde, sizlerden gelen sorulara yanıt vermeye çalışacağım. Eğer burada yanıtlarını bulamadığınız sorularınız varsa ya da bu yanıtlara ekleyebileceğiniz bilgiler varsa, lütfen bunları bana *bildirin* (sayfa: 6) ki, onları da buraya ekleyebileyim.

Spamcılar Uyum Sağlarsa

5.1. Spamcılar uyum sağlayıp bu belgede açıklanan teknikleri aşmanın çarelerini bulurlarsa ne olacak?

Olur mu olur, bakalım neler olur. `:-)`

Açıklanan bazı sınamalar (*SMTP Sınamaları* (sayfa: 13) ve *Grilisteleme* (sayfa: 18) gibi) özellikle *kalleş yazılımların* davranışlarını hedef alır. Eğer yeterince site bu sınamaları yapmaya başlarsa bu davranışların değişeceğini düşünmek elbette mümkündür. Hatmut Danisch bu konuda şöyle diyor:

> Kalleş yazılımlar SMTP protokolünü kendilerine uydurarak kullanırlar, çünkü daha iyisine ihtiyaçları yoktur. Onlar bu şekilde çalışır, hem niye fazla zaman kaybetsinler ki? Bu arada kalleş yazılımlarda kalite yükselmiştir, hatta spam iletilerinin kalitesinde bile önemli bir iyileşme göze çarpmaktadır. Kötü SMTP protokolü nedeniyle spam iletilerini reddeden kullanıcı sayısı yeterli bir seviyeye ulaştığında spam yazarları da yazılımlarını iyileştirecektir.

Böyle kalleş yazılımların uyum sağlamaya çalışacakları durumlara bir bakalım:

- *SMTP aktarımındaki gecikmeleri* (sayfa: 11) kandırmak için alıcı durumundaki SMTP sunucusundan gelecek her yanıtı beklemek zorundalar. Bu noktada, spam yapan konağın birim zamanda teslim edebileceği posta sayısını bu gecikmeleri birarada kullanarak önemli ölçüde düşürebiliyoruz. Spamcılar, DNS karalistelerine ve içerik filtrelerine yakalanmadan önce mümkün olduğunca çok postayı teslim edebilmek için zamana karşı yarıştıklarından, bu araçların etkinliğini olabildiğince arttırmaya çalışıyoruz.

Gönderici ücretlendirme şemaları (sayfa: 71)na benzer bir etki sağlamak üzere, gönderici, postanın her alıcısı için hesaplamalı bir kimlik denetim dizgesi hazırlamak için birkaç saniye harcayıp bunu alıcının doğrulayabilmesi için ileti başlığına ekleyebilir. Bu şemaların karmaşıklığı, uygulanabilirliğini zorlaştıran faktörlerden biri gibi görünse de, asıl önemlisi, dünyadaki herkesin katılımını gerektirmesidir. Halbuki SMTP gecikmelerinin etkinliği daha uygulandığı ilk makinede ortaya çıkmaya başlar.

- *Selamlaşma (HELO/EHLO) Sınamaları* (sayfa: 14)nı kandırmak için selamlaşmayı olması gerektiği gibi yapmak, yani kendilerini geçerli bir *Nitelikli Alan Adı* (sayfa: 72) ile tanıtmak zorundalar. Özellikle, kendiliklerinden rDNS sorgusunun sonucunu iletinin Received: başlığına eklemeyen *posta aktarımcıları* (sayfa: 72) açısından böyle bir selamlaşma izlenebilirliği arttıracaktır.

- *Gönderici adresi sınamaları* (sayfa: 15)nı kandırmak için ise, sürekli geçerliliği olan bir gönderici adresi belirtmek zorundalar. Bunu yaparlar mı?

- *Grilisteleme* (sayfa: 18)yi kandırmak için geçici olarak başarısızlığa uğramış teslimatları bir saat sonra (ama dört saat geçmeden) yinelemek zorundalar. (İyi bir gerçeklenimin yaptığı gibi başarısız olmuş her postanın bir kopyasını tutmak yerine, başarısız olmuş alıcıların bir listesini tutmak kalleş yazılımlar için yeterli olacağından bu adreslere bir veya iki saat sonra bir atak başlatabilirler.)

  Öyle bile olsa, *spam tuzakları* (sayfa: 73) ile beslenen *DNS Karalisteleri* (sayfa: 13) ile birlikte *grilisteleme* hala etkinliğini koruyacaktır. Bu bir saatlik zorunlu gecikme, gönderici konağı listelerine almak için bu karalistelere yeterli süreyi verecektir.

Spam Tarayıcıları (sayfa: 24) ve *Virüs Tarayıcıları* (sayfa: 23) gibi yazılım araçları sürekli geliştirilmektedir. Spamcılar kendilerini geliştirdikçe bunlar da geliştirilmektedir, dolayısıyla bu araçların daima son sürümlerini kullanarak bunların etkin kalmasını sağlayabilirsiniz.

Sonuç olarak, bu belge de gelişmeye ve değişikliklere açık. Döküntü postaların değişken tabiatına uygun olarak, bunları engellemek isteyenler de hep yeni ve oldukça yaratıcı fikirlerle ortaya çıkmaktalar.

A. Exim Gerçeklenimi

Burada, bu belgede açıklanan tekniklerin ve araçların Exim *Posta Aktarımcısı* (sayfa: 72)na uyarlanması üzerinde duracağız.

A.1. Öngereksinimler

Bu örnekler için, `Exim` *Posta Aktarımcısı* (sayfa: 72)na ihtiyacınız olacak, Tom Kistner'in `Exiscan-ACL` yamasının uygulanmış olması tercih edilmelidir. İkisi `Exim+Exiscan-ACL` olarak tek bir paket halinde çoğu Linux dağıtımında, hatta FreeBSD'de bulunmaktadır. Daha fazla bilgi edinmek için Exiscan–ACL[B222] ana sayfasına bakınız[15].

Son gerçeklenim örneğinin birlikte çalışabildiği ek araçlar:

- SpamAssassin[B224] – posta içeriğini ampirik yaklaşımlardan oluşmuş çok sayıda ve oldukça karmaşık araçlarla analiz eden çok popüler bir spam filtreleme aracı.
- greylistd[B225] – Exim ile kullanmak için bu belgenin yazarı tarafından geliştirilmiş basit bir grilisteleme çözümü.

Örnekler üzerinde başka araçlar da kullanılmıştır.

A.2. Exim Yapılandırma Dosyası

Exim yapılandırma dosyası genel tanımlarla başlar (buna *ana bölüm* diyeceğiz) ve çeşitli alt bölümlerden oluşur[16]. Bu alt bölümlerin her biri şöyle bir satırla başlar:

```
begin bölümismi
```

Zamanımızın çoğunu `acl` bölümünde (`begin acl` ile başlayan bölüm) harcayacağız; ama dosyanın başındaki ana bölüm ile `transports` ve `routers` bölümlerine de bir kaç öğe ekleyecek ya da bazılarında değişiklik yapacağız.

A.2.1. Erişim Denetim Listeleri

(ACL – Access Control Lists)

4.xx sürümünden itibaren Exim, SMTP sırasındaki filtrelemeler için *Erişim Denetim Listeleri* (ACL'ler) adı verilen oldukça karmaşık ve esnek bir mekanizma kullanmaktadır.

Bir ACL, SMTP aktarımı sırasında gelen iletinin red mi yoksa kabul mü edileceğine karar verebilmek için, uzak konak ilk bağlantıyı kurduğunda veya **`HELO/EHLO`**, **`MAIL FROM:`**, **`RCPT TO:`** gibi aktarımın çeşitli aşamalarından birinde değerlendirmeler yapabilmek amacıyla kullanılabilir. Örneğin, karşıdan gelen her **`RCPT TO:`** komutunda değerlendirmeler yapmak için `acl_rcpt_to` isimli ACL'yi kullanabilirsiniz.

Bir ACL, *deyimler*den (veya *kurallar*dan) oluşur. Her deyim eylem belirten `accept` (kabul et), `warn` (uyar), `require` (gerekir), `defer` (ertele) veya `deny` (reddet) gibi bir emir ile başlar ve bunu koşullar, seçenekler ve diğer ayarlamalardan oluşan bir liste izler. Her deyim, tanımlayıcı bir eyleme rastlanıncaya kadar (`warn` hariç) sırayla değerlendirilir. Her ACL'nin sonunda örtük bir `deny` vardır.

`acl_rcpt_to` ACL'sinden örnek bir deyim:

```
  deny
    message  = relay not permitted
   !hosts    = +relay_from_hosts
   !domains  = +local_domains : +relay_to_domains
    delay    = 1m
```

Bu deyim, **RCPT TO:** komutu +relay_from_hosts (röleleme yapmasına izin verilen konaklar) listesindeki konaklardan birinden alınmamışsa ve alıcı konak "+local_domains" (yerel alanlar) veya "+relay_to_domains" (röleleme yapılacak alanlar) listelerindeki konaklardan biri değilse, postayı reddedecektir. Bu arada, "550" yanıtını vererek reddetmeden önce karşı sunucuyu bir dakika bekletecektir.

SMTP aktarımının belli bir aşamasında bir ACL'nin değerlendirmeye alınabilmesi için Exim'in *kural denetimleri*nde bu ACL'yi belirtmek gerekir. Örneğin, önceki örnekteki acl_rcpt_to ACL'sinin **RCPT TO:** komutunda değerlendirmeye alınabilmesi için Exim yapılandırma dosyasının ana bölümüne (begin ile başlayan ilk satırdan önceki bölüm) şöyle bir satır eklemek gerekir:

```
acl_smtp_rcpt = acl_rcpt_to
```

Bu tür *kural denetimleri*nin tam listesini Exim belirtiminin 14.11 bölümünde[B227] bulabilirsiniz.

A.2.2. Yerleşikler

Çalışma anı değişkenleri, sorgu işlevleri, dizgeler ve düzenli ifadeler, konak ve alan adı listeleri, vs. gibi çok sayıda oluşum desteklenmektedir. Bunların kapsamlı bir listesini son x.x0 sürümleriyle gelen "spec.txt" dosyasında bulabilirsiniz. ACL'ler ise Exim belirtiminin 39. bölümünde[B228] açıklanmıştır.

Özellikle, ACL deyimlerinde değer atamak için kullanılmak üzere genel amaçlı yirmi yerleşik değişken tanımlıdır:

- *$acl_c0* – *$acl_c9* değşkenlerine SMTP bağlantısı boyunca kalıcı olacak değerler atanabilir.
- *$acl_m0* – *$acl_m9* değişkenlerine bir ileti alınırken değer atanabilir, fakat alım sonunda bunlara yeniden değer atanmaz. Bunlara, **HELO**, **EHLO**, **MAIL** ve **RSET** komutları tarafından da yeniden değer atanabilir.

A.3. Seçenekler ve Ayarlar

Exim yapılandırma dosyasının ana bölümü (begin ile başlayan ilk satırdan önceki bölüm) çeşitli makrolar, kural denetimleri ile genel ayarları içerir. Daha sonra kullanacağımız bazı makroları tanımlayarak başlayalım:

```
# İleti boyutunun sınırı; bunu DATA ACL içinde kullanacağız.
MESSAGE_SIZE_LIMIT = 10M

# Spam veya Virus taraması için azami ileti boyutu.
# Büyük iletiler sunucuyu aşırı yüklemesin diye düşük tutuyoruz.
MESSAGE_SIZE_SPAM_MAX = 1M

# Çırpılama için kullanmak üzere gizemli bir dizge tanımlayan makro.
# BUNU KENDİNİZE GÖRE DÜZENLEYİN!.
SECRET = gizemli–dizge
```

Genel Exim ayarlarının bazılarını yapalım:

```
# DNS başarısızlıklarını (SERVFAIL) sorgu başarısızlığı olarak
# ele alacağız. Böylece, daha sonra mevcut olmayan alanlara veya
# alanadı sunucusu olmayan alanlara ait gönderici adreslerden
# gelen teslimatları reddedebileceğiz. (Bu tanıma göre, yerel
# alanlar ve röleleme yapılan alanlar için sorgu yapılmayacak.)
dns_again_means_nonexist = !+local_domains : !+relay_to_domains

# Tüm konaklar için ACL'lerde HELO doğrulamasını etkinleştirelim.
helo_try_verify_hosts = *
```

```
# Bir defada hizmet sunacağımız gelen bağlantı sayısına bir
# sınırlama koymuyoruz. Daha sonra spamcılara SMTP aktarım
# gecikmeleri uygulayacağımız için bu sırada yeni gelen bağlantıları
# böylece kabul edebileceğiz.
smtp_accept_max = 0

# Ama, sistemin yükü de 10'nun üzerini çıkmamalı.
smtp_load_reserve = 10

# Hiçbir konağa ESMTP "PIPELINING" yapabileceğimizi söylemeyeceğiz.
# Bu, kalleş yazılımlar için gerekli, boruhattı açmaya bayılırlar.
pipelining_advertise_hosts =:
```

Son olarak, gelen bir SMTP aktarımının çeşitli aşamalarında değerlendirmeye alınmasını istediğimiz beş ACL'yi Exim kural denetimcisine tanıtacağız:

```
acl_smtp_connect = acl_connect
acl_smtp_helo    = acl_helo
acl_smtp_mail    = acl_mail_from
acl_smtp_rcpt    = acl_rcpt_to
acl_smtp_data    = acl_data
```

A.4. ACL'lerin Hazırlanması – İlk Aşama

ACL bölümünde (`begin acl` ile başlayan bölüm) ihtiyacımız olacak ACL'leri tanımlayacağız. Bunu yaparken bu belgede daha önce açıkladığımız *DNS Sınamaları* (sayfa: 12) ve *SMTP Sınamaları* (sayfa: 13) gibi temel *Teknikler* (sayfa: 11)den bazılarını da kullanacağız.

Bu aşamada, sınamaların çoğunu *acl_rcpt_to* (sayfa: 33) altında yapacağız ve diğer ACL'leri ise büyük oranda boş bırakacağız. Bunu böyle yapmamızın sebebi kalleş yazılımların daha SMTP aktarımının başlarında keşfedildiklerini anlamamaları içindir – aksi takdirde tekrar gelirler. Diğer taraftan, çoğu kalleş yazılım **`RCPT TO:`** başarısızlıklarında teslimattan vazgeçer.

Bu ACL'leri daha sonra kullanmak üzere şimdiden oluşturacağız.

A.4.1. `acl_connect`

```
# Bu erişim denetim listesi gelen bağlantının başlangıcında
# kullanılır. Bu sınamalar bağlantı kabul ya da red edilinceye
# kadar sırayla yapılır.

acl_connect:

  # Bu aşamada, herhangi bir sınama yapmıyoruz ve bağlantıyı
  # kabul ediyoruz.
  accept
```

A.4.2. `acl_helo`

```
# Bu erişim denetim listesi gelen bir SMTP aktarımında HELO veya EHLO
# komutları için kullanılır. Bu sınamalar selamlaşma kabul ya da
# red edilinceye kadar sırayla yapılır.

acl_helo:

  # Bu aşamada, herhangi bir sınama yapmıyoruz.
```

```
  accept
```

A.4.3. acl_mail_from

```
# Bu erişim denetim listesi gelen bir SMTP aktarımında MAIL FROM:
# komutu için kullanılır. Bu sınamalar gönderici adresi kabul
# ya da red edilinceye kadar sırayla yapılır.

acl_mail_from:

  # Bu aşamada, herhangi bir sınama yapmıyoruz.
  accept
```

A.4.4. acl_rcpt_to

```
# Bu erişim denetim listesi gelen bir SMTP aktarımında RCPT
# komutu için kullanılır. Bu sınamalar alıcı adresi kabul
# ya da red edilinceye kadar sırayla yapılır.

acl_rcpt_to:

  # Posta yerel SMTP üzerinden alınmışsa (yani, TCP/IP bağlantısı
  # ile gelmiyorsa) kabul et. Bunu boş bir konak alanını
  # sınayarak yapacağız. Ayrıca, postalarını rölelediğimiz
  # konaklardan gelen postaları da kabul edeceğiz.
  #
  # Alıcı doğrulamasını burada atlıyoruz, çünkü çoğu durumda
  # istemciler kullanıcıların posta istemcileridir ve SMTP
  # hata iletileri ile ne yapacaklarını bilmezler.
  #
  accept
    hosts       =: +relay_from_hosts


  # İleti, kimlik kanıtlaması yapılan bir bağlantı üzerinden
  # geliyorsa kabul ediyoruz. Yine, bu iletiler kullanıcıların posta
  # istemcilerinden geldiklerinden alıcı doğrulaması yapmıyoruz.
  #
  accept
    authenticated = *


  ############################################################################
  # DNS sınamaları
  ############################################################################
  #
  # Bu sınamaların sonuçları arabelleğe alınır, böylece çok sayıda alıcı
  # olduğunda çok sayıda DNS sorgusu yapılmasına gerek kalmaz.
  #

  # Eğer bağlanan konak seçtiğimiz birkaç DNS karalistesinde kayıtlı
  # ise iletiyi reddediyoruz. Bu listeleri seçerken dikkatli olun,
  # çoğu yanlış olumlama yapar ve/veya kara listeden silme konusunda
  # kuralları iyi belirlenmemiştir.
  #
  deny
```

```
  dnslists    = dnsbl.sorbs.net : \
                dnsbl.njabl.org : \
                cbl.abuseat.org : \
                bl.spamcop.net
  message     = $sender_host_address is listed in $dnslist_domain\
                ${if def:dnslist_text { ($dnslist_text)}}


# Eğer gönderici konağın ters DNS sorgusu başarısız olursa
# (rDNS kaydı yoksa veya sonuçlar normal DNS sorgusuyla eşleşmiyorsa)
# iletiyi reddediyoruz.
#
deny
  message     = Reverse DNS lookup failed for host $sender_host_address.
 !verify      = reverse_host_lookup



#############################################################################
# Selamlaşma sınamaları
#############################################################################

# Bağlanan konak selamlaşma sırasında bir IP adresi belirtmişse,
# postayı reddediyoruz.
#
deny
  message     = Message was delivered by ratware
  log_message = remote host used IP address in HELO/EHLO greeting
  condition   = ${if isip {$sender_helo_name}{true}{false}}


# Bağlanan konak selamlaşma sırasında bizim isimlerimizden birini
# kullanmışsa reddediyoruz.
#
deny
  message     = Message was delivered by ratware
  log_message = remote host used our name in HELO/EHLO greeting.
  condition   = ${if match_domain{$sender_helo_name}\
                {$primary_hostname:+local_domains:+relay_to_domains}\
                {true}{false}}


# Bağlanan konak selamlaşma sırasında kendini tanıtmamışsa
# reddediyoruz.
#
deny
  message     = Message was delivered by ratware
  log_message = remote host did not present HELO/EHLO greeting.
  condition   = ${if def:sender_helo_name {false}{true}}


# HELO doğrulaması başarısız olmuşsa, ileti başlığına bir
# X-HELO-Warning: satırı ekliyoruz.
#
warn
  message     = X-HELO-Warning: Remote host $sender_host_address \
                ${if def:sender_host_name {($sender_host_name) }}\
```

```
                incorrectly presented itself as $sender_helo_name
  log_message = remote host presented unverifiable HELO/EHLO greeting.
  !verify     = helo



##############################################################################
# Gönderici adresi sınamaları
##############################################################################

# Gönderici adresini doğrulatamazsak iletiyi reddedeceğiz.
#
# "callout" seçeneğini isterseniz silebilirsiniz. Özellikle, postanızı
# doğrudan değil de göstermelik sunucu (smarthost) olarak
# gönderiyorsanız, bu seçenek anlamsız olacaktır.
#
# Başarısız varlık doğrulamalarının ayrıntıları genelde 550 yanıtları
# içerir; bunları yoksaymak için "sender/callout" dizgesini
# "sender/callout,no_details" olarak değiştirebilirsiniz.
#
deny
  message     = <$sender_address> does not appear to be a \
                valid sender address.
  !verify     = sender/callout



##############################################################################
# Alıcı adresi sınamaları
##############################################################################

# Yerel kısım @  %  / |! karakterlerinden birini içeriyorsa,
# iletiyi reddediyoruz. Bunlar normal yerel kısımlarda çok nadir
# görülür, çoğunlukla röleleme sınırlamalarını aşmaya çalışanlarca
# kullanılır.
#
# Ayrıca, yerel kısım bir nokta ile başlıyorsa da reddediyoruz.
# Boş bileşenler RFC 2822'de kuraldışıdır, fakat Exim bu yaygın
# olduğundan bunlara izin verir. Buna rağmen, bir nokta ile
# başlayan bir yerel kısım bir dosya ismi olarak kullanılmışsa
# (örneğin, bir posta listesi), sorunlara yol açabilir.
#
deny
  local_parts = ^.*[@%!/|]: ^\\.


# Zarf göndericisi adresi boş olduğu halde iletinin birden fazla alıcısı
# varsa, bağlantıyı kesiyoruz. Meşru teslimat durum bildirimleri asla
# birden fazla adrese gönderilmez.
#
drop
  message      = Legitimate bounces are never sent to more than one \
                 recipient.
  senders      = : postmaster@*
  condition    = $recipients_count
```

```
  # Alıcı adres bizim postalarını kabul ettiğimiz alanlardan birine
  # ait değilse, iletiyi reddediyoruz.
  #
  deny
    message     = relay not permitted
    !domains    = +local_domains : +relay_to_domains


  # Alıcının geçerli bir posta kutusu yoksa iletiyi reddediyoruz.
  # Eğer posta kutuları sistemimizde bulunmuyorsa (alıcı alanadı
  # için yedek posta alıcısı isek), bir varlık sınaması yaparız;
  # ama hedef sunucu yanıt vermezse postayı mecburen kabul edeceğiz.
  #
  deny
    message     = unknown user
    !verify     = recipient/callout=20s,defer_ok


  # Aksi takdirde, alıcı adres geçerlidir.
  #
  accept
```

A.4.5. acl_data

```
# Bu erişim denetim listesi gelen bir SMTP aktarımında ileti
# tamamen alındıktan sonra kullanılır. Bu sınamalar alıcı adresi
# kabul ya da red edilinceye kadar sırayla yapılır.

acl_data:

  # İleti kendi konaklarımızdan alınmış ve Message-ID başlığını
  # içermiyorsa, onu biz ekleyeceğiz.
  #
  warn
    condition   = ${if !def:h_Message-ID: {1}}
    hosts       = : +relay_from_hosts
    message     = Message-ID: <E$message_id@$primary_hostname>


  # Posta yerel SMTP üzerinden alınmışsa (yani, TCP/IP bağlantısı
  # ile gelmiyorsa), kabul ediyoruz. Bunu boş bir konak alanını
  # sınayarak yapacağız. Ayrıca, postalarını rölelediğimiz
  # konaklardan gelen postaları da kabul edeceğiz.
  #
  accept
    hosts       = : +relay_from_hosts

  # İleti, kimlik kanıtlaması yapılan bir bağlantı üzerinden
  # geliyorsa kabul ediyoruz.
  #
  accept
    authenticated = *


  # İleti boyutu ile ilgili sınırlama aşılıyorsa iletiyi reddediyoruz.
  #
```

```
deny
  message      = Message size $message_size is larger than limit of \
                 MESSAGE_SIZE_LIMIT
  condition    = ${if >{$message_size}{MESSAGE_SIZE_LIMIT}{true}{false}}


# Adres listesinin sözdizimi hatalıysa reddediyoruz.
#
deny
  message      = Your message does not conform to RFC2822 standard
  log_message = message header fail syntax check
  !verify      = header_syntax


# Dışardan gelen ve Message-ID veya Date başlığı bulunmayan postaları
# reddediyoruz.
#
# Bazı özelleştirilmiş posta aktarımcılarının, örneğin posta listesi
# sunucularının boy gönderici adresi ile gönderdikleri postalara
# kendiliklerinden bir Message-ID üretmedikleri bilinmektedir;
# böyle durumlar için boş bir gönderici adresin varlığına da bakacağız.
#
deny
  message      = Your message does not conform to RFC2822 standard
  log_message = missing header lines
  !hosts       = +relay_from_hosts
  !senders     = : postmaster@*
  condition    = ${if or {{!def:h_Message-ID:}\
                          {!def:h_Date:}\
                          {!def:h_Subject:}} {true}{false}}


# "Sender:", "Reply-To:" veya "From:" satırlarından en azından birindeki
# gönderici adres doğrulanabilir değilse, bir uyarı veriyoruz.
#
warn
  message      = X-Sender-Verify-Failed: No valid sender in message header
  log_message = No valid sender in message header
  !verify      = header_sender


# İletiyi kabul ediyoruz
#
accept
```

A.5. SMTP aktarım gecikmelerinin eklenmesi

A.5.1. Basit yöntem

SMTP aktarım gecikmeleri uygulamanın en basit yolu hazırladığımız ACL'lerin sonundaki `accept` deyimine bir `delay` denetimi eklemektir:

```
accept
  delay = 20s
```

Buna ek olarak, *acl_rcpt_to* (sayfa: 33) içindeki geçersiz kullanıcı ("unknown user") ile ilgili `deny` deyimine

arttırımlı gecikmeler de ekleyebilirsiniz. Bu, sözlük saldırılarını yavaşlatmak için oldukça yararlıdır. Örnek:

```
  deny
    message      = unknown user
   !verify       = recipient/callout=20s,defer_ok,use_sender
    delay        = ${eval:$rcpt_fail_count*10 + 20}s
```

Bu noktada birşeye dikkatinizi çekmek isterim, ileti verisi alındıktan bunu yapmanın yararı yoktur, bunun yapılabileceği tek yer *acl_rcpt_to* (sayfa: 33) ACL'sidir. Kalleş yazılımlar genellikle ileti verisini aktardıktan sonra sunucunuzun yanıtını beklemeden bağlantıyı keserler. İstemci bağlantıyı kessin ya da kesmesin bu noktada Exim'in iletinin teslimatı ile ilgili ne işlem yapacağının artık bir önemi kalmaz.

A.5.2. Seçimlik Gecikmeler

Benim gibi, SMTP aktarım gecikmelerine konu edeceğiniz konaklar için biraz daha seçici davranmak isteyebilirsiniz. Örneğin, bu belgede daha önce açıklandığı gibi, DNS karalistelerinde bulunma durumunda ya da kesin bir redde konu olmayan ama doğrulama da yapılamayan selamlaşmalar sonucunda bu gecikmeleri uygulamaya karar verebilirsiniz.

Seçimlik gecikmelerin uygulayabilmek için *acl_rcpt_to* (sayfa: 33) ACL'sindeki bazı sınamaları diğer ACL'lere taşımak gerekir. Böylelikle, sorunların işaretlerini gördüğümüz anda gecikmeleri uygulayabilir ve kalleş yazılımların eşzamanlama hataları verme ve başka sorunlarla karşılaşma talihsizliğini arttırmış oluruz.

Özellikle yapacaklarımız:

- DNS sınamalarını *acl_connect* (sayfa: 58) ACL'sine taşıyacağız.
- Selamlaşma sınamalarını *acl_helo* (sayfa: 59) ACL'sine taşıyacağız. Bir istisna: Bu noktada henüz selamlaşmanın olmayışını sınayamayız, çünkü bu ACL bir EHLO veya HELO komutunun varlığı halinde devreye girer. Bu sınamayı *acl_mail_from* (sayfa: 60) ACL'sinde yapacağız.
- Gönderici adresi sınamalarını *acl_mail_from* (sayfa: 60) ACL'sine taşıyacağız.

Bununla birlikte, evvelce açıkladığımız sebeplerle, asıl reddi **RCPT TO:** komutunu alana dek yapmayacağız. Bunu gerçekleştirmek için önceki ACL'lerdeki `deny` deyimlerini `warn` deyimlerine dönüştüreceğiz ve **RCPT TO:** komutunu alana kadar hata iletilerini ve uyarıları saklamak için Exim'in genel amaçlı ACL değişkenlerini kullanacağız. Şöyle ki:

- Teslimatı reddetmeye karar verirsek, gönderilecek **550** yanıtlarında kullanmak üzere hata iletisini saklamak için `$acl_c0` veya `$acl_m0` değişkenini kullanacağız:
  - Eğer gerekli koşullar bir posta teslimatından önce sağlanmışsa (örn, *acl_connect* (sayfa: 58) veya *acl_helo* (sayfa: 59) ACL'sinde), bağlantı boyunca değer saklayabilen `$acl_c0` değişkenini kullanacağız.
  - Posta aktarımı başladıktan sonra (örn. **MAIL FROM:** komutundan sonra), `$acl_c0` içeriğini iletilere özel değişken olan `$acl_m0`'a kopyalayacağız ve bu noktadan sonra bu değişkeni kullanacağız. Böylece, bu ileti ile belirlenmiş bir durum, aynı bağlantıdan alınan daha sonraki iletilerden etkilenmemiş olacak.

  Ayrıca, benzer şekilde, *günlükleme iletilerini* `$acl_c1` veya `$acl_m1` değişkeninde saklayacağız.
- Eğer, kesin reddine karar verilecek yeterli koşulların sağlanmadığı bir durumla karşılaşırsak, `$acl_c1` veya `$acl_m1` değişkeninde sadece bir uyarı iletisi saklayacağız. Posta aktarımı başladığında (örn, *acl_mail_from* (sayfa: 60) ACL'sinde), bu değişkenin içeriğini ileti başlığına da ekleyeceğiz.

- Daha sonraki sınamaların (SpamAssassin taraması gibi) sonuçlarına bakmaksızın bir iletiyi *kabul etmeye* karar verirsek, durum belirtecini `$acl_c0` veya `$acl_m0` değişkeninde saklayacak ama `$acl_c1` ve `$acl_m1` değişkenlerini boş bırakacağız.
- Her ACL'nin başlangıcında ve *acl_mail_from* (sayfa: 60) ACL'sinde, o anki zaman damgasını `$acl_m2` değişkenine atayacağız. ACL'nin sonunda ise, `$acl_c1` veya `$acl_m1` değişkeninin varlığına bakıp zaman damgasındaki değerden başlayarak 20 saniyelik gecikmeyi dolduracak şekilde SMTP aktarım gecikmesini uygulayacağız.

Kullandığımız değişkenleri bir tablo halinde özetlersek:

ACL bağlantı/ileti değişkenlerinin kullanımı

Değişkenler:	`$acl_[cm]0` boş	`$acl_[cm]0` dolu
`$acl_[cm]1` boş	(Henüz bir karar yok)	Postayı kabul ediyoruz
`$acl_[cm]1` dolu	Başlığa bir uyarı ekliyoruz	Postayı reddediyoruz

Bu yaklaşıma bir örnek olarak, selamlaşma ile ilgili iki sınama yapacağız; birinde karşı taraf kendini IP adresi ile tanıtırsa postayı reddedeceğiz, diğerinde ise doğrulanabilir olmayan bir isim belirtirse bir uyarıyı kayda alacağız. Önceden, bu iki sınamayı *acl_rcpt_to* (sayfa: 33) ACL'sinde yapmıştık, şimdi *acl_helo* (sayfa: 59) ACL'sine alacağız.

```
acl_helo:
  # Gecikme uygularken başlangıç olarak kullanmak üzere o anki zaman
  # bilgisini kaydediyoruz.
  warn
    set acl_m2  = $tod_epoch

  # Posta yerel SMTP üzerinden alınmışsa (yani, TCP/IP bağlantısı
  # ile gelmiyorsa), kabul ediyoruz. Bunu boş bir konak alanını
  # sınayarak yapacağız. Ayrıca, postalarını rölelediğimiz
  # konaklardan gelen postaları da kabul edeceğiz.
  #
  accept
    hosts       =: +relay_from_hosts

  # Karşı taraf selamlaşmayı IP adresi ile yaparsa, $acl_c0'a red
  # iletisini, $acl_c1'e ise günlük iletisini kaydedeceğiz. Bunları
  # daha sonra "deny" deyiminde kullanacağız. Bu değişkenlerin varlığı
  # göndericinin oyalanacağını belirtecek.
  #
  warn
    condition   = ${if isip {$sender_helo_name}{true}{false}}
    set acl_c0  = Message was delivered by ratware
    set acl_c1  = remote host used IP address in HELO/EHLO greeting

  # Selamlaşmada doğrulaması başarısız olursa, acl_c1'e bir uyarı iletisi
  # kaydedeceğiz. Bu iletiyi daha sonra ileti başlığına ekleyeceğiz.
  # Bu değişkenin varlığı göndericinin oyalanacağını belirtecek.
  #
  warn
    condition   = ${if !def:acl_c1 {true}{false}}
   !verify     = helo
    set acl_c1  = X-HELO-Warning: Remote host $sender_host_address \
                  ${if def:sender_host_name {($sender_host_name) }}\
                  incorrectly presented itself as $sender_helo_name
```

```
    log_message = remote host presented unverifiable HELO/EHLO greeting.

  #
  # ... bu örnek için diğer sınamaları atlıyoruz ...
  #

  # Bağlantıyı kabul ediyoruz ama $acl_c1'de bir ileti varsa, göndericiyi
  # 20 saniye oyalıyoruz.
  accept
    set acl_m2  = ${if def:acl_c1 {${eval:20 + $acl_m2 - $tod_epoch}}{0}}
    delay       = ${if >{$acl_m2}{0}{$acl_m2}{0}}s
```

Sonra, *acl_mail_from* (sayfa: 60) ACL'sinde iletileri `$acl_c{0,1}`'den `$acl_m{0,1}` değişkenlerine aktaracağız. Ayrıca, `$acl_c1` içeriğini ileti başlığına ekleyeceğiz.

```
acl_mail_from:
  # Gecikme uygularken başlangıç olarak kullanmak üzere o anki zaman
  # bilgisini kaydediyoruz.
  warn
    set acl_m2  = $tod_epoch

  # Posta yerel SMTP üzerinden alınmışsa (yani, TCP/IP bağlantısı
  # ile gelmiyorsa), kabul ediyoruz. Bunu boş bir konak alanını
  # sınayarak yapacağız. Ayrıca, postalarını rölelediğimiz
  # konaklardan gelen postaları da kabul edeceğiz.
  #
  accept
    hosts     =: +relay_from_hosts

  # $acl_c0 ve $acl_c1 değişkenleri bu SMTP aktarımı sırasında yapılmaya
  # çalışılacak her teslimat için kullanılacak red ve/veya uyarı iletilerini
  # içeriyor olacak (böyle bir durum varsa). Bu değişkenlerin içeriklerini
  # $acl_m{0,1} iletiye özel değişkenlere aktaracağız ve $acl_m1'deki uyarıyı
  # ileti başlığına ekleyeceğiz. (Bir red durumunda, $acl_m1 artık bir günlük
  # kaydı içeriyor olacak, ama iletiyi bununla değil başlığındaki kayıtla
  # reddedeceğiz.)
  #
  warn
    set acl_m0  = $acl_c0
    set acl_m1  = $acl_c1
    message     = $acl_c1


  #
  # ... bu örnek için diğer sınamaları atlıyoruz ...
  #

  # Bağlantıyı kabul ediyoruz ama $acl_c1'de bir ileti varsa, göndericiyi
  # 20 saniye oyalıyoruz.
  accept
    set acl_m2  = ${if def:acl_c1 {${eval:20 + $acl_m2 - $tod_epoch}}{0}}
    delay       = ${if >{$acl_m2}{0}{$acl_m2}{0}}s
```

Kalan değişiklikler için *Tamamlanmış ACL'ler* (sayfa: 58) bölümüne bakınız.

A.6. Grilisteleme Desteğinin Eklenmesi

Exim ile kullanmak üzere hazırlanmış çeşitli grilisteleme gerçeklenimleri vardır. Burada bunların bir kısmına değineceğiz.

A.6.1. greylistd

Bu, belgenin yazarı tarafından bir Python gerçeklenimi olarak geliştirilmiş bir grilisteleme gerçeklenimidir. (Doğal olarak, bu gerçeklenim *Tamamlanmış ACL'ler* (sayfa: 58) bölümünde uygulanacak.) Tek başına bir artalan süreci olarak çalışır ve herhangi bir veritabanını kullanmaz. Grilesteleme verisi verimlilik açısından 32 bitlik basit bir çırpı olarak saklanır.

Paketi `http://packages.debian.org/unstable/mail/greylistd` adresinde bulabilirsiniz. Debian kullanıcıları APT ile kolayca kurabilir:

```
# apt-get install greylistd
```

`greylistd`'yi devreye sokmak için, *acl_rcpt_to* (sayfa: 62) ACL'sinde sonuncu `accept` deyiminden hemen önceye iki deyim ekleyeceğiz.

```
  # Belli bir sunucu/gönderici/alıcı üçlüsü için grilisteleme durumunu
  # belirleyecek "greylistd"'yi devreye sokacağız.
  #
  # Grilisteleme iletilerini bir boş gönderici için yapmıyoruz, çünkü
  # boş gönderici adresli varlık doğrulaması işimize yaramaz (gerçek
  # göndericinin varlığını sınamak için bir konağa posta gönderemeyiz).
  #
  defer
    message     = $sender_host_address is not yet authorized to deliver mail \
                  from <$sender_address> to <$local_part@$domain>. \
                  Please try later.
    log_message = greylisted.
    domains     = +local_domains : +relay_to_domains
   !senders     =: postmaster@*
    set acl_m9  = $sender_host_address $sender_address $local_part@$domain
    set acl_m9  = ${readsocket{/var/run/greylistd/socket}{$acl_m9}{5s}{}{}}
    condition   = ${if eq {$acl_m9}{grey}{true}{false}}
```

Hatalı *teslimat durum bildirimlerini* (sayfa: 73) engellemek için *zarf gönderici imlerini* (sayfa: 51) kullanmıyorsanız, bunun benzeri bir deyimi *acl_data* (sayfa: 66) ACL'sine ayrıca boş göndericili grilisteleme iletileri olarak ekleyebilirsiniz.

Grilistelemenin amaçlarına uygun olarak veriyi burada yukarıdakinden biraz farklı olarak kullanıyoruz. Boş olan `$sender_address`'e ek olarak, bu noktada ne `$local_part` ne de `$domain` tanımlıdır. Ancak, `$recipients`, tüm alıcı adreslerinin virgüllerle ayrılmış bir listesini içerir. Meşru teslimat durum bildirimleri için bu değişken sadece bir adres içermelidir.

```
  # Burada, zarf gönderici adresi olmayan iletilere grilisteleme
  # uygulayacağız. Bunları RCPT TO:'dan sonra grilistelemeye konu
  # etmeyeceğiz, çünkü gönderici varlık doğrulamaları yaparken
  # karşı konaklarla olumsuz etkileşime girilebilir.
  #
  defer
    message     = $sender_host_address is not yet authorized to send \
                  delivery status reports to <$recipients>. \
                  Please try later.
    log_message = greylisted.
```

```
    senders      =: postmaster@*
    set acl_m9   = $sender_host_address $recipients
    set acl_m9   = ${readsocket{/var/run/greylistd/socket}{$acl_m9}{5s}{}{}}
    condition    = ${if eq {$acl_m9}{grey}{true}{false}}
```

A.6.2. MySQL gerçeklenimi

Bu gerçeklenim, aşağıdaki gerçeklenimlerin üzerine inşa edilerek Johannes Berg <johannes (at) sipsolutions.net> tarafından geliştirilmiştir:

- İlk olarak bir Postgres gerçeklenimi olarak Tollef Fog Heen <tfheen (at) raw.no> tarafından oluşturuldu. http://raw.no/personal/blog/tech/Debian/2004-«03-«14-«15-«55_greylisting adresinden edinilebilir.
- Ardından, Rick Stewart <rick.stewart (at) theinternetco.net> üzerinde çalıştı ve çalışmasını http://theinternetco.net/projects/exim/greylist adresinde yayınladı.

Başka bir pograma ihtiyaç duymaz – gerçeklenim tamamen MySQL veritabanında yapılandırılmıştır.

Yapılandırmanın gelişimini içeren bir arşiv ve bir README dosyası http://johannes.sipsolutions.net/wiki/Projects/exim-«greylist adresinde mevcuttur.

Sisteminizde MySQL kurulu olmalıdır. MySQL komut satırında exim_greylist ve exim_greylist_log isimli iki tablo ile exim4 veritabanı oluşturulur:

```
CREATE DATABASE exim4;
use exim4;

CREATE TABLE exim_greylist (
   id bigint(20) NOT NULL auto_increment,
   relay_ip varchar(80) default NULL,
   sender varchar(255) default NULL,
   recipient varchar(255) default NULL,
   block_expires datetime NOT NULL default '0000-00-00 00:00:00',
   record_expires datetime NOT NULL default '9999-12-31 23:59:59',
   create_time datetime NOT NULL default '0000-00-00 00:00:00',
   type enum('AUTO','MANUAL') NOT NULL default 'MANUAL',
   passcount bigint(20) NOT NULL default '0',
   blockcount bigint(20) NOT NULL default '0',
   PRIMARY KEY  (id)
);

CREATE TABLE exim_greylist_log (
   id bigint(20) NOT NULL auto_increment,
   listid bigint(20) NOT NULL,
   timestamp datetime NOT NULL default '0000-00-00 00:00:00',
   kind enum('deferred', 'accepted') NOT NULL,
   PRIMARY KEY (id)
);
```

Exim yapılandırma dosyasının ana bölümüne bazı makrolar eklenir:

```
# Eğer başka veritabanları da kullanıyorsanız, bu veritabanına erişimi
# mysql_servers = localhost/exim4/kullanıcı/parola şeklinde sağlayabilirsiniz.

# seçenekler
```

```
# bunlar, xxx olarak mysql'in DATE_ADD(..,INTERVAL xxx) deyiminde
# geçerli olacak şekilde belirtilmelidir, örneğin çoğul olarak
# "2 HOUR" yerine "2 HOURS" belirtilirse geçersiz olacaktır.
GREYLIST_INITIAL_DELAY = 1 HOUR
GREYLIST_INITIAL_LIFETIME = 4 HOUR
GREYLIST_WHITE_LIFETIME = 36 DAY
GREYLIST_BOUNCE_LIFETIME = 0 HOUR

# tablo isimlerini değiştirebilirsiniz
GREYLIST_TABLE=exim_greylist
GREYLIST_LOG_TABLE=exim_greylist_log

# grilistelemeyi (geçici olarak) iptal etmek için bu satırı açıklama
# haline getirin
GREYLIST_ENABLED=

# günlük kayıtlarını etkinleştirmek için bu satırın başındaki # işaretini
# kaldırın
#GREYLIST_LOG_ENABLED=

# bundan sonrasında normalde bir düzenleme yapılmamalıdır

.ifdef GREYLIST_ENABLED
# veritabanı makroları
GREYLIST_TEST = SELECT CASE \
   WHEN now() > block_expires THEN "accepted" \
   ELSE "deferred" \
 END AS result, id \
 FROM GREYLIST_TABLE \
 WHERE (now() < record_expires) \
   AND (sender      = '${quote_mysql:$sender_address}' \
     OR (type='MANUAL' \
       AND (    sender IS NULL \
         OR sender = '${quote_mysql:@$sender_address_domain}' \
           ) \
        ) \
      ) \
   AND (recipient   = '${quote_mysql:$local_part@$domain}' \
     OR (type = 'MANUAL' \
       AND (    recipient IS NULL \
         OR recipient = '${quote_mysql:$local_part@}' \
          OR recipient = '${quote_mysql:@$domain}' \
            ) \
         ) \
       ) \
   AND (relay_ip    = '${quote_mysql:$sender_host_address}' \
     OR (type='MANUAL' \
       AND (    relay_ip IS NULL \
         OR relay_ip = \
         substring('${quote_mysql:$sender_host_address}',1,length(relay_ip)) \
           ) \
         ) \
      ) \
 ORDER BY result DESC LIMIT 1

GREYLIST_ADD = INSERT INTO GREYLIST_TABLE \
  (relay_ip, sender, recipient, block_expires, \
```

```
   record_expires, create_time, type) \
 VALUES ( '${quote_mysql:$sender_host_address}', \
  '${quote_mysql:$sender_address}', \
  '${quote_mysql:$local_part@$domain}', \
  DATE_ADD(now(), INTERVAL GREYLIST_INITIAL_DELAY), \
  DATE_ADD(now(), INTERVAL GREYLIST_INITIAL_LIFETIME), \
  now(), \
  'AUTO' \
)

GREYLIST_DEFER_HIT = UPDATE GREYLIST_TABLE \
                     SET blockcount=blockcount+1 \
                     WHERE id = $acl_m9

GREYLIST_OK_COUNT = UPDATE GREYLIST_TABLE \
                    SET passcount=passcount+1 \
                    WHERE id = $acl_m9

GREYLIST_OK_NEWTIME = UPDATE GREYLIST_TABLE \
                      SET record_expires = DATE_ADD(now(), \
                      INTERVAL GREYLIST_WHITE_LIFETIME) \
                      WHERE id = $acl_m9 AND type='AUTO'

GREYLIST_OK_BOUNCE = UPDATE GREYLIST_TABLE \
                     SET record_expires = DATE_ADD(now(), \
                     INTERVAL GREYLIST_BOUNCE_LIFETIME) \
                     WHERE id = $acl_m9 AND type='AUTO'

GREYLIST_LOG = INSERT INTO GREYLIST_LOG_TABLE \
               (listid, timestamp, kind) \
               VALUES ($acl_m9, now(), '$acl_m8')
.endif
```

Artık, ACL bölümünde (`begin acl` satırından sonra) "greylist_acl" ismiyle yeni bir ACL tanımlayabiliriz:

```
.ifdef GREYLIST_ENABLED
# Bu acl ya deny ya da accept döndürecek.
# acl = greylist_acl'yi ber deger ile kullandığımızdan,
# bir accept, kuralı DOĞRU yapacak, dolayısıyla bir erteleme olacak;
# bir deny ise kuralı YANLIŞ yapacak, dolayısıyla erteleme olmayacak.
greylist_acl:
  # Normal teslimatlar için griliste sınanacak.

  # Griliste sınanıp, acl_m8'e "accepted", "deferred" veya "unknown"
  # ve acl_m9'a kayıt numarası döndürülecek.

  warn set acl_m8 = ${lookup mysql{GREYLIST_TEST}{$value}{result=unknown}}
       # Burada acl_m8 = "result=x id=y"

       set acl_m9 = ${extract{id}{$acl_m8}{$value}{-1}}
       # Artık acl_m9 kayıt numarasını (veya -1) içerecek.

       set acl_m8 = ${extract{result}{$acl_m8}{$value}{unknown}}
       # acl_m8 unknown/deferred/accepted içerecek.

  # Bu üçlüyü bilmiyorsak, ileti ekleyeceğiz yoksa erteleyeceğiz
  accept
```

```
        # yukarıdaki sınama unknown (henüz kayıt yok) döndürmüşse
        condition = ${if eq{$acl_m8}{unknown}{1}}
        # ayrıca bir kayıt ekleyeceğiz
        condition = ${lookup mysql{GREYLIST_ADD}{yes}{no}}

  # Şimdi günlük kaydı yapacağız, sonucun önemi yok.
  # Üçlüyü bilmiyorsak bir günlük girdisine gerek yok çünkü
  # yukarıda oluşturma sırasında dolaylı olarak yapıldı.
  #
  .ifdef GREYLIST_LOG_ENABLED
  warn condition = ${lookup mysql{GREYLIST_LOG}}
  .endif

  # Üçlü hala engelleniyor mu bakalım
  accept
        # Yukarıdaki sınama deferred döndünmüşse ertele
        condition = ${if eq{$acl_m8}{deferred}{1}}
        # ve kayda geçir
        condition = ${lookup mysql{GREYLIST_DEFER_HIT}{yes}{yes}}

  # Bakılan kayıtları saymak için bir warn deyimi kullanıyoruz.
  warn condition = ${lookup mysql{GREYLIST_OK_COUNT}}

  # Özdevinimli kayıtlarda zaman aşımını belirlemek için bir
  # warn deyimi kullanıyoruz. Ancak, posta boş göndericili değilse
  # zamanaşımı uygulanacak, aksi takdirde zamanaşımı uygulanmayacak.
  #
  warn !senders =: postmaster@*
        condition = ${lookup mysql{GREYLIST_OK_NEWTIME}}
  warn senders =: postmaster@*
        condition = ${lookup mysql{GREYLIST_OK_BOUNCE}}

  deny
.endif
```

Gönderici adresi boş olmayan üçlüleri grilistelemek için bu ACL'yi *acl_rcpt_to* (sayfa: 62) ACL'nize yerleştirin. Böylece, gönderici varlık doğrulaması yapmanız mümkün olacak:

```
.ifdef GREYLIST_ENABLED
  defer !senders =: postmaster@*
        acl      = greylist_acl
        message  = greylisted - try again later
.endif
```

Onu ayrıca *acl_data* (sayfa: 36)'e de yerleştirin, fakat sadece gönderici adresinin boş olduğunu tespit ettikten sonraya. Bu, spamcıların grilistelemeyi gönderici adresini boş bırakarak aşmaya çalışmalarını önlemek içindir.

```
.ifdef GREYLIST_ENABLED
  defer senders  =: postmaster@*
        acl      = greylist_acl
        message  = greylisted - try again later
.endif
```

A.7. SPF Sınamalarının Eklenmesi

Burada Exim kullanarak *Gönderici Yetkilendirme Dizgesi (SPF)* (sayfa: 20) kayıtlarını sınamak için iki yöntemden bahsedeceğiz. Bu doğrudan sınama mekanizmalarından başka, yakın bir gelecekte SpamAssassin (2.70

sürümünde sanırım) çeşitli SPF sınamalarına derecelendirme uygulayan, biraz daha ince eleyip sık dokuyan SPF sınamaları ile gelecek.

Bu sınamayı en erken *acl_mail_from* (sayfa: 60) ACL'sinde yapabiliriz. Bu kararı almamızı sağlayan etken: SPF geleneksel eposta yönlendirmesi ile uyumlu değildir. Yönlendiren konak SRS[B261] uygulamıyorsa, yönlendirilen postayı reddetmek kaçınılmaz olur, çünkü postanın *Zarf Göndericisi* (sayfa: 74) adresindeki alanadının DNS kayıtlarındaki SPF kaydı böyle bir konağı posta göndermeye yetkili konaklardan biri olarak içermeyecektir.

Bunu yapmaktan kaçınmak için, kabul edilmesi gereken yönlendirilmiş postaları gönderen konakların kullanıcı tarafından belirlendiği listelere bakmamız gerekir (bu durum, *Yönlerdirilmiş Postaların Sınama Dışı Tutulması* (sayfa: 56) bölümünde açıklanmıştır). Bu da sadece alıcının kullanıcı adını bilebileceğimiz **RCPT TO:** komutundan sonra mümkün olur.

Böyle bir durumda, bu sınamayı *acl_rcpt_to* (sayfa: 62) içinde sonuncu `accept` deyiminden önce ve/veya varsa grilisteleme sınamasından önceye ekleyeceğiz.

A.7.1. Exiscan–ACL üzerinden SPF sınamaları

Tom Kistner'in `Exiscan-ACL` yamasının son sürümü (bkz. *Öngereksinimler* (sayfa: 30)) SPF için destek içermektedir[17]. Kullanımı çok basittir. Bir `spf` ACL kuralı eklenir ve `pass`, `fail`, `softfail`, `none`, `neutral`, `err_perm` veya `err_temp` anahtar sözcükleriyle karşılaştırma yapılır.

acl_rcpt_to (sayfa: 62) ACL'sine grilisteleme sınamalarının ve/veya sonuncu `accept` deyiminin öncesine aşağıdaki satırları yerleştirin:

```
  # Gönderici adresinin alanadı için varsa, SPF kayıtlarını sorgulayalım.
  # Gönderici konak bu alanadı için yetkilendirilmişse teslimatı kabul
  # yoksa red edeceğiz.
  #
  deny
    message      = [SPF] $sender_host_address is not allowed to send mail \
                   from $sender_address_domain
    log_message  = SPF check failed.
    spf          = fail


  # İleti başlığına bir SPF-Received: satırı ekleyelim.
  warn
    message      = $spf_received
```

Bu deyim, eğer gönderici adresinin alanadının sahibi, postayı teslim etmeye çalışan konağı teslimat için yetkilendirmemişse postayı reddedecektir. Bazılarına göre alan adı sahibi için bu kadar kontrol yetkisi fazladır, hatta bu yetkiyle istem dışı olarak rahatça kendilerini sabote edebilirler. SPF sınamalarını başka sınamalarla birleştirmek de önerilmektedir. Örneğin, Gönderici Varlık Sınamaları ile birlikte (fakat eğer posta sunucunuz göstermelik sunucu ise – yani, postaları bir dış sunucu üzerinden gönderiyorsa – bunu yapamazsınız).

```
  # Gönderici adresini varlık doğrulamaları ile doğrulatamazsak ve
  # gönderici adresin alanadı sahibi SPF kaydıyla teslimatı yapmaya
  # çalışan konağı yetkilendirmemişse postayı reddedeceğiz.
  #
  deny
    message      = The sender address does not seem to be valid, and SPF \
                   information does not grant $sender_host_address explicit \
                   authority to send mail from $sender_address_domain
    log_message  = SPF check failed.
    !verify      = sender/callout,random,postmaster
    !spf         = pass
```

```
# İleti başlığına bir SPF-Received: satırı ekleyelim.
warn
  message     = $spf_received
```

A.7.2. Mail::SPF::Query üzerinden SPF sınamaları

`Mail::SPF::Query` bir resmi SPF deneme paketidir ve `http://www.openspf.org/downloads.html` adresinden edinilebilir. Debian kullanıcıları,

```
# apt-get install libmail-spf-query-perl
```

ile kolayca kurabilir.

`Mail::SPF::Query` paketi gelen istekleri bir UNIX soketinden dinleyen bir artalan süreci (**`spfd`**) ile gelir. Ama, bu artalan sürecini başlatmak için bir başlatma betiği ile gelmez. Bu bakımdan, aşağıdaki örnekte, bizim SPF isteklerimiz için anlık bir uygulama olarak çalıştırılan **`spfquery`** aracını kullanacağız.

Aşağıdaki satırları *acl_rcpt_to* (sayfa: 33) içinde, yukarıdaki gibi sonuncu `accept` deyiminden önce ve/veya varsa grilisteleme sınamasından önceye yerleştirin:

```
  # Bu gönderici/konak için SPF durumunu öğrenmek için "spfquery"
  # kullanacağız. Eğer komuttan dönen kod 1 ise bu bir yetkisiz
  # göndericidir.
  #
  deny
    message     = [SPF] $sender_host_address is not allowed to send mail \
                  from $sender_address_domain.
    log_message = SPF check failed.
    set acl_m9  = -ipv4=$sender_host_address \
                  -sender=$sender_address \
                  -helo=$sender_helo_name
    set acl_m9  = ${run{/usr/bin/spfquery $acl_m9}}
    condition   = ${if eq {$runrc}{1}{true}{false}}
```

A.8. MIME ve Dosya türü Sınamalarının Eklenmesi

Bu sınamalar Tom Kistner'in `Exiscan-ACL` yamasındaki özelliklere bağımlıdır – ayrıntılar için *Öngereksinimler* (sayfa: 30) bölümüne bakınız.

Exiscan–ACL yaması MIME kodlamasının ve dosya ismi soneklerinin (Windows'çası uzantılarının) sınamalarını içerir. Bu sınamalar tek başlarına çoğu Windows virüsünü engelleyecektir – ama bunlar `.ZIP` arşivleri olarak geliyorsa ya da ileti içeriği zarar verici Outlook/MSIE HTML kodları içeriyorsa bunları engelleyemez – bkz. *Virüs Tarayıcıları* (sayfa: 23).

Bu sınamalar *acl_data* (sayfa: 36) içindeki sonuncu `accept` deyiminin öncesine yerleştirilmelidir:

```
  # Birtakım MIME hataları olan iletileri reddedeceğiz.
  #
  deny
    message     = Serious MIME defect detected ($demime_reason)
    demime      = *
    condition   = ${if >{$demime_errorlevel}{2}{1}{0}}


  # MIME taşıyıcıyı aç ve kurtlar tarafından kullanılan dosya uzantıları
```

```
  # varsa reddet. Bu çağrılar tekrar demime uygulayacaktır, ama sonuçlar
  # arabellekli olarak dönecektir.
  # Uzantı listesinin eksik olabileceğini unutmayın.
  #
  deny
    message     = We do not accept ".$found_extension" attachments here.
    demime      = bat:btm:cmd:com:cpl:dll:exe:lnk:msi:pif:prf:reg:scr:vbs:url
```

Yukarıdaki örnekte, `demime` koşulunun iki defa çağrıldığına dikkat edin. Bununla birlikte, sonuçlar arabelleğe alındığından ileti aslında iki defa baştan değerlendirilmeyecektir.

A.9. AntiVirüs Yazılımlarının Eklenmesi

Exiscan–ACL yaması bazı virüs tarayıcılarının doğrudan eklemlenebilmesine, bir kısmının da `cmdline` arkayüzü vasıtasıyla komut satırından çalıştırılmasına imkan tanır.

Bu özelliği kullanabilmek için Exim yapılandırma dosyanızın *ana bölümünde* (sayfa: 31) hangi virüs tarayıcısını kullanacağınızı, tarayıcıya aktarılacak seçeneklerle birlikte belirtmelisiniz. Bununla ilgili sözdizimi şöyledir:

`av_scanner =` *tarayıcı–türü*`:`*seçenek1*`:`*seçenek*`:...`

Örneğin:

```
av_scanner = sophie:/var/run/sophie
av_scanner = kavdaemon:/opt/AVP/AvpCtl
av_scanner = clamd:127.0.0.1 1234
av_scanner = clamd:/opt/clamd/socket
av_scanner = cmdline:/path/to/sweep -all -rec -archive %s:found:'(.+)'
...
```

DATA ACL'de asıl taramayı gerçekleştirmek için `malware` koşulunu kullanabilirsiniz:

```
  deny
    message  = This message contains a virus ($malware_name)
    demime   = *
    malware  = */defer_ok
```

Paketle gelen `exiscan-acl-spec.txt` dosyasında ayrıntılı kullanım bilgilerini bulabilirsiniz.

A.10. SpamAssassin'in Eklenmesi

SMTP sırasında SpamAssassin çağrısı Exim'de genelde şu iki yoldan biri ile yapılır:

- `Exiscan-ACL` yamasının içerdiği `spam` kuralı üzerinden. Bu, bizim burada kullanacağımız mekanizma olacak.
- `SA-Exim` üzerinden. Marc Merlins <marc (at) merlins.org> tarafından özellikle Exim'in SMTP sırasında SpamAssassin'i çalıştırması için yazılmıştır. Bu uygulama Exim'in `local_scan()` arayüzünden işlem yapar. Ya doğrudan Exim'e bir yama olarak uygulanarak ya da Marc'ın kendi `dlopen()` eklentisi üzerinden (Debian'ın `exim4-daemon-light` ve `exim4-daemon-heavy` paketleri ikinci yolu kullanır).

  `SA-Exim` başka özellikler de içerir: *grilisteleme* ve *katran çukuru*. Bununla birlikte, tarama işlemi ileti verisi alındıktan sonra yapıldığından, bu iki özelliğin yararlı olabilmeleri için SMTP aktarımının başlarında uygulanmaları gerektiğinden, faydalı olmayabilir.

  `SA-Exim` `http://marc.merlins.org/linux/exim/sa.html` adresinden edinilebilir.

A.10.1. SpamAssassin'in Exiscan üzerinden çağrılması

Exiscan-ACL'nin "spam" kuralı hem SpamAssassin hem de Brightmail'i kullanabilir ve ileti bir döküntüyse bu kural bunu belirtecek şekilde tetiklenir. Exim öntanımlı olarak, localhost üzerinde çalışan bir SpamAssassin artalan sürecine (spamd) bağlanır. Ancak konak adresi ve port, Exim yapılandırma dosyasının *ana* bölümüne bir spamd_address ataması ile belirtilerek başka bir konaktaki SpamAssassin'in kullanılması sağlanabilir. Daha ayrıntılı bilgi için bu yamayla birlikte gelen exiscan-acl-spect.txt dosyasına bakınız.

Bizim gerçeklenimimizde, spam olarak tasnif edilmiş iletileri reddedeceğiz. Bununla birlikte, bu tür iletilerin bir kopyasını, kullanıcı arasıra *Hatalı Olumlama* (sayfa: 71)lar için bu dizini tarayabilsin diye, bir süreliğine ayrı bir posta dizininde tutacağız.

Exim, kabul edilen bir iletiye freeze adı verilen bazı denetimler uygulayabilir. Exiscan–ACL yaması bu denetimlere, fakereject ismiyle başka denetimler ekler. Bu şöyle bir SMTP yanıtına sebep olur:

```
550-FAKEREJECT id=ileti–kimliği
550-İletiniz reddedildi ancak değerlendirilmek üzere tutuluyor.
550 Eğer meşru bir iletiyse, hala alıcılarına teslim edilebilir.
```

Bu özelliği kendi gerçeklenimimize aşağıdaki satırları *acl_data* (sayfa: 36) içinde sonuncu accept deyiminden önceye yerleştirerek kullanacağız:

```
  # $spam_score ve $spam_report'a veri sağlamak için SpamAssassin'i
  # çağıracağız. Tasnife bağlı olarak, $acl_m9 "ham" veya "spam"
  # değerini alacak.
  #
  # İleti spam olarak tasnif edilmişse, reddetmiş gibi yapacağız.
  #
  warn
    set acl_m9  = ham
    spam        = mail
    set acl_m9  = spam
    control     = fakereject
    logwrite    = :reject: Rejected spam (score $spam_score): $spam_report

  # İletinin başlığına bir X-Spam-Status: satırı ekleyelim.
  #
  warn
    message     = X-Spam-Status: \
                  ${if eq {$acl_m9}{spam}{Yes}{No}} (score $spam_score)\
                  ${if def:spam_report {: $spam_report}}
    logwrite    = :main: Classified as $acl_m9 (score $spam_score)
```

Bu örnekte, $acl_m9 değişkeni "ham" değeriyle ilklendirildi. SpamAssassin mail kullanıcısı olarak çağrıldı. Eğer ileti spam olarak tasnif edilmişse, $acl_m9'a "spam" değeri atanıp, yukarıdaki FAKEREJECT yanıtı verildi. Bunu yapmada ana fikir, *Posta Teslimatçısı* (sayfa: 73)nın veya alıcının *Posta İstemcisi* (sayfa: 73)nin bu başlığı kullanarak döküntü postayı ayrı bir dizinde toplayabilmesine imkan sağlamaktır.

A.10.2. SpamAssassin yapılandırması

Öntanımlı olarak, SpamAssassin raporunu ayrıntılı olarak tablo benzeri bir biçimde ya ileti gövdesine yazar ya da bir eklenti olarak iletiye ekler. Biz ise, yukarıdaki örnekte olduğu gibi X-Spam-Status: başlığına uygun kısa ve özlü bir rapor istiyoruz. Bunun olması için, aşağıdaki satırları SpamAssassin'in yapılandırma dosyasına ekleyeceğiz (/etc/spamassassin/local.cf, /etc/mail/spamassassin/local.cf, vb.):

```
### Rapor şablonu
```

```
clear_report_template
report "_TESTSSCORES(, )_"
```

Ayrıca, bir *Bayes* (sayfa: 70) derecelendirme özelliği yerleşik olarak vardır ve öntanımlı olarak etkindir. Bunu normal olarak biz kapatacağız, çünkü kullanıcıya özel eğitilmesi gerekir, dolayısıyla SMTP sırasındaki filtreleme için kullanıma uygun değildir:

```
### Bayes derecelendirmesi kapalı
use_bayes 0
```

Bu değişikliklerin etkin olabilmesi için SpamAssassin artalan süreci olan **spamd**'yi yeniden başlatmalıyız.

A.10.3. Kullanıcı verileri ve ayarları

Kullanıcılarınızın bazıları kişisel SpamAssassin tercihlerini belirtebilmek isteyebilirler; örneğin, spam eşiği, posta kabul ettikleri diller ve karakter kümeleri, kara ve ak listeli kullanıcılar, vs. Hatta, SpamAssassin'in yerleşik Bayes derecelendirmesini kullanmanın bile mümkün olmasını isteyebilirler (bunun anlamlı olacağını düşünmesem de[18]).

Bu belgede evvelce *Kullanıcı Verileri ve Ayarları* (sayfa: 27) bölümünde açıklandığı gibi, bunu yapmanın bir yolu vardır. Gelen her teslimattaki alıcı sayısını bir ile sınırlandırmamız gerekir. İlk **RCPT TO:** komutunu kabul ettikten sonra diğerlerini bir **451** SMTP yanıtı ile erteleriz. *Grilisteleme Desteğinin Eklenmesi* (sayfa: 40)nde olduğu gibi, eğer bağlanan posta aktarımcısı işini iyi bilen bir yazılımsa, bu yanıtın nasıl yorumlanacağını bilecek ve teslimatı yineleyecektir.

A.10.3.1. Exim'e "her teslimatı sadece bir alıcı için kabul et" demek istersek

acl_rcpt_to (sayfa: 62) ACL'sinde, alıcı adresi doğrulandıktan sonra ve uzak konaklardan yerel kullanıcılara kimlik kanıtlamasız gelen bağlantılar için bir `accept` deyiminin öncesine (yani, grilistelemeyi ve zarf gönderici imlerini sınamadan önceye) aşağıdaki deyimi yerleştireceğiz:

```
  # Kullanıcı verilerini ve ayarlarını (Spamassasin gibi) destekleyebilmek
  # için gelen her iletinin alıcı sayısını bir ile sınırlayalım.
  #
  # BİLGİ: Çok sayıda kullanıcınıza gönderilmiş bir postanın yerine
  #        ulaşması her alıcı için 30 dakika veya daha fazla olmak
  #        üzere katlanarak gecikecektir. Bu, özellikle zamanın kritik
  #        önemde olduğu durumlarda sorunlara yol açacaktır.
  #
  defer
    message      = We only accept one recipient at a time - please try later.
    condition    = $recipients_count
```

A.10.3.2. SpamAssassin'e alıcının kullanıcı isminin aktarılması

acl_data (sayfa: 66) ACL'sinde, evvelce bahsettiğimiz `spam` kuralını değiştirerek, alıcı adresindeki yerel kısımda belirtilmiş kullanıcı ismini SpamAssassin'e aktaracağız.

```
  # $spam_score ve $spam_report'a veri sağlamak için SpamAssassin'i
  # çağıracağız. Tasnife bağlı olarak, $acl_m9 "ham" veya "spam"
  # değerini alacak.
  #
  # Alıcı adresinin kullanıcı adını SpamAssassin'e aktaralım.
  # Bunun için adresin '=' veya '@' karakterinden önceki kısmını
  # küçük harfe dönüştüreceğiz. Evvelce bir defadaki alıcı sayısını
```

```
  # önceden bir ile sınırladığımızdan çok sayıda alıcı olmayacak.
  #
  # İleti spam olarak tasnif edilmişse, reddetmiş gibi yapacağız.
  #
  warn
    set acl_m9  = ham
    spam        = ${lc:${extract{1}{=@}{$recipients}{$value}{mail}}}
    set acl_m9  = spam
    control     = fakereject
    logwrite    =:reject: Rejected spam (score $spam_score): $spam_report
```

Dikkat ederseniz, Exim'in `${local_part:...}` işlevini kullanmak yerine "@" veya "=" karakterinden önceki kısmı kendimiz ayırdık. Bunun sebebi, ileride *zarf gönderici imlemesi* (sayfa: 51) için "=" karakterini kullanacak olmamızdır.

A.10.3.3. SpamAssassin'de kullanıcı verilerinin ve ayarlarının etkinleştirilmesi

SpamAssassin'e tekrar bakalım. Herşeyden önce, yapılandırma dosyasına evvelce yerleştirdiğimiz `use_bayes 0` atamasını silebilirsiniz. Bu durumda, her kullanıcı kendi ayarlarını belirtebilme ayrıcalığına kavuşacaktır.

Eğer sisteminizdeki posta kutularının isimleri yerel UNIX hesaplarına göre açılmışsa bu mümkün olur. Öntanımlı olarak SpamAssassin artalan süreci, kendisine aktarılan kullanıcı ismine önce bir `setuid()` uygular ve kullanıcının verilerini ve ayarlarını kullanıcının ev dizinine kaydeder.

Eğer yapınız bu işleme uygun değilse (örneğin, posta hesaplarınız Cyrus SASL veya başka bir sunucu tarafından yönetiliyordur), SpamAssassin'e kullanıcı tercihlerini ve verilerini içeren dosyaları nerede bulacağını belirtmeniz gerekir. Ayrıca, **spamd**'nin mevcut olmayan bir kullanıcıya `setuid()` yapmasını önlemek için onun belli bir yerel kullanıcı adıyla çalışmasını sağlamanız gerekir.

Biz bu seçenekleri **spamd**'yi başlatırken belirteceğiz:

- Debian'da `/etc/default/spamassassin` dosyasının `OPTIONS=` satırını düzenleyerek.
- Red Hat'ta `/etc/sysconfig/spamassassin` dosyasının `SPAMDOPTIONS=` satırını düzenleyerek.
- Diğerlerini siz bulun.

Gereken seçenekler:

- `-u` *kullanıcı* – **spamd**'nin hangi kullanıcının (örn. `mail`) aidiyetinde çalışacağı belirtilir.
- `-x` – kullanıcıların ev dizinlerindeki yapılandırma dosyalarına bakılmaz.
- `--virtual-config-dir=/var/lib/spamassassin/%u` – kullanıcı verilerinin ve ayarlarının yeri belirtilir. "%u" SpamAssassin tarafından kullanıcı ismi ile değiştirilerek kullanılır. **spamd** bu dizini oluşturmaya veya bu dizinde değişiklik yapmaya yetkili olmalıdır:

  ```
  # mkdir /var/lib/spamassassin
  # chown -R mail:mail /var/lib/spamassassin
  ```

Bu kadar, bu değişiklikleri yaptıktan sonra **spamd**'yi yeniden başlatmanız yetecektir.

A.11. Zarf Gönderici İmlerinin Eklenmesi

Burada, giden posta için *Zarf Gönderici İmleri* (sayfa: 25) uygulayacak ve zarf göndericisi boş olarak gelen postalarda bu imlerin varlığına bakacağız.

Makinemizden giden postaların zarf göndericisi adresini şu şekilde değiştireceğiz:

gönderici=alıcı=alıcı.alanadı=birdeğer@gönderici.alanadı

Ancak, bu şema istenmeyen sonuçlar doğuracağından (posta listelerinin sunucularından gelen postalar gibi), bu şemanın kullanımını kullanıcıların tercihine bırakacağız. Eğer kullanıcının ev dizininde ".return–path–sign" isminde bir dosya varsa, giden postanın zarf göndericisini imlemeyi sadece bu kullanıcılar için ve sadece bu dosyada belirtilen alan adları için yapacağız. Bu dosyanın içi boş bırakılmışsa, tüm alan adları için bu sınamanın yapılacağını anlayacağız.

Bu yolla, sadece zarf gönderici adresi boş bırakılmış olarak gelen postalardan, sadece ev dizininde böyle bir dosya bulunan kullanıcılara gelenlerde alıcı adresi imlemesinin varlığına bakacağız. Kullanıcılar bazı konakları *Yönlerdirilmiş Postaların Sınama Dışı Tutulması* (sayfa: 56) bölümünde açıklandığı gibi kullanıcıya özel aklistelere kaydederek bu sınamaların dışında bırakabilirler.

Bu şema, Exim'in yapılandırma dosyasındaki ACL'lerden başka yönlendiriciler ve aktarıcılar bölümlerinde de değişiklik yapmayı gerektirdiğinden bu sınamayı *Tamamlanmış ACL'ler* (sayfa: 58)e doğrudan dahil etmeyeceğiz. Bu bölümdeki açıklamaları okuyarak isterseniz, burada açıklanan ACL bölümünü kendiniz ekleyebilirsiniz.

A.11.1. Gönderici adresini imlemek için bir Transport oluşturmak

Önce giden postalarda kullanmak üzere gönderici adresini imleyen bir Exim *transport*' u oluşturacağız.

```
remote_smtp_signed:
  debug_print    = "T: remote_smtp_signed for $local_part@$domain"
  driver         = smtp
  max_rcpt       = 1
  return_path    = $sender_address_local_part=$local_part=$domain=\
                   ${hash_8:${hmac{md5}{SECRET}{${lc:\
                     $sender_address_local_part=$local_part=$domain}}}}\
                   @$sender_address_domain
```

Bu deyime göre, gönderici adresinin "yerel kısmı" birbirlerinden eşit işaretleri ile ayrılmış şu parçalardan oluşacak:

- Göndericinin kullanıcı ismi, yani adresin yerel kısmı,
- alıcı adresinin yerel kısmı,
- alıcı adresinin alanadı kısmı,
- Gönderici/alıcı/sunucu üçlüsüne özel bir dizge. Şöyle üretilir:
  - Gönderici adresinin yeniden yazılan ilk üç elemanı ile yapılandırma dosyasının *ana bölümünde tanımladığımız* `SECRET` *dizgesi* (sayfa: 31) Exim'in `${hmac{md5}...}` işlevi ile şifrelenir[19].
  - Sonuç, 8 küçük harf üretecek şekilde Exim'in `${hash...}` işlevi ile çırpılanır.

Eğer sunucunuz, başka bir göstermelik konağın kimlik kanıtlamalı olarak postalarını gönderiyorsa, uygun bir `hosts_try_auth` satırını da buraya ekleyin. (Onu mevcut "smarthost transport"'undan alabilirsiniz.)

A.11.2. Giden teslimatlar için yeni bir yönlendirici oluşturmak

Giden postalarınızı işleme sokmakta olan mevcut yönlendiricilerinizin (*router*) önüne yeni bir yönlendirici ekleyeceğiz. Bu yönlendirici uzak teslimatlar için yukarıdaki aktarımı (*transport*) kullanacak, fakat sadece kullanıcının ev dizininde bir ".return–path–sign" dosyası varsa ve alıcının alanadı bu dosyada mevcutsa. Örneğin, postanızı doğrudan internet üzerinden son hedefine gönderiyorsanız:

```
# Kullanıcının ev dizininde bir ".return-path-sign" dosyası
# varsa ve alıcının alanadı bu dosyada mevcutsa, uzak konaklara posta
# teslimatı yaparken zarf göndericisi adresini imleyeceğiz. Eğer dosya
# var ama içi boşsa, zarf göndericisi adresini daima imleyeceğiz.
#
dnslookup_signed:
  debug_print    = "R: dnslookup_signed for $local_part@$domain"
  driver         = dnslookup
  transport      = remote_smtp_signed
  senders        =! : *
  domains        =! +local_domains : !+relay_to_domains : \
      ${if exists {/home/$sender_address_local_part/.return-path-sign}\
                  {/home/$sender_address_local_part/.return-path-sign}\
                  {!*}}
  no_more
```

Veya, bir göstermelik sunucu (smarthost) kullanıyorsanız:

```
# Kullanıcının ev dizininde bir ".return-path-sign" dosyası
# varsa ve alıcının alanadı bu dosyada mevcutsa, uzak konaklara posta
# teslimatı yaparken zarf göndericisi adresini imleyeceğiz. Eğer dosya
# var ama içi boşsa, zarf göndericisi adresini daima imleyeceğiz.
#
smarthost_signed:
  debug_print    = "R: smarthost_signed for $local_part@$domain"
  driver         = manualroute
  transport      = remote_smtp_signed
  senders        =! : *
  route_list     = * göstermelik.sunucu.adresi
  host_find_failed = defer
  domains        =! +local_domains : !+relay_to_domains : \
      ${if exists {/home/$sender_address_local_part/.return-path-sign}\
                  {/home/$sender_address_local_part/.return-path-sign}\
                        {!*}}
  no_more
```

Sizce olması gereken diğer seçenekleri de (`same_domain_copy_routing = yes` gibi) ekleyin, tabii mevcut yönlendiricilerinizi tamamen oluşturduktan sonra.

Dikkat ederseniz, bu yönlendiriciyi zarf göndericisi adresi boş olan postalar için kullanmıyoruz – bunları birbirine karıştırmayalım![20]

A.11.3. Gelen teslimatlar için `redirect` yönlendiricisi oluşturmak

Bundan sonra yapacağınız iş, Exim'e yukarıdaki biçimde adreslenmiş olarak gelen teslimatların alıcı adresine ait posta kutusunun ilk eşit işaretinden önceki kısım olduğunu belirtmektir. Bunu gerçekleştirmek için yapılandırma dosyanızın `routers` bölümünün başlarında bir yere bir `redirect` yönlendirici yerleştirmeniz gerekir – yerel teslimatlarla (örn, *system alias* yönlendiricisi) ilgili yönlendiricilerin öncesine):

```
hashed_local:
  debug_print       = "R: hashed_local for $local_part@$domain"
  driver            = redirect
  domains           = +local_domains
  local_part_suffix = =*
  data              = $local_part@$domain
```

Eşit işareti içeren alıcı adreslerinin yerel kısmı eşit işaretinden arındırılarak yeniden yazılır ve tüm yönlendiricilerde tekrar işleme sokulur.

A.11.4. İmleme Sınama ACL'si

Bu şemanın son parçası, Exim'e imli olarak geçerli alıcı adreslere gelen teslimatların daima kabul edileceğini ve boş gönderici adresli diğer teslimatların ise eğer alıcı bu şemayı seçmişse reddedileceğini belirtmektir. Böyle durumlarda grilisteleme yapılmamalıdır.

Aşağıdaki satırları *acl_rcpt_to* (sayfa: 62) ACL'sinde olası bir SPF, grilisteleme ve/veya sonuncu `accept` deyiminin öncesine yerleştirin:

```
  # Kendine özgü imlemesini içeriyorsa, alıcı adresini kabul ediyoruz.
  # Bu, teslimatın, daha önce bizden gönderilmiş bir postanın teslimat
  # durum bildirimi olduğunu gösterir.
  #
  accept
    domains     = +local_domains
    condition   = ${if and {{match{${lc:$local_part}}{^(.*)=(.*)}}\
                            {eq{${hash_8:${hmac{md5}{SECRET}{$1}}}}{$2}}}\
                           {true}{false}}

  # Aksi takdirde, posta boş gönderici adresli ise ama alıcı, imlemeli zarf
  # gönderici adresi şemasını seçenlerden biri ise postayı reddediyoruz.
  #
  deny
    message     = This address does not match a valid, signed \
                  return path from here.\n\
                  You are responding to a forged sender address.
    log_message = bogus bounce.
    senders     =: postmaster@*
    domains     = +local_domains
    set acl_m9  = /home/${extract{1}{=}{${lc:$local_part}}}/.return-path-sign
    condition   = ${if exists {$acl_m9}{true}}
```

Postayı gönderirken iletinin başlığındaki adreslere (örneğin, gönderdiğiniz postanın `From:` alanındaki adrese) varlık doğrulaması yapan konakların varlığını bu noktada dikkate almak gerekir. Buradaki `deny` deyimi normal olarak böyle bir doğrulatma çabasına olumsuz yanıt verecektir.

Bunun olmaması için, `deny` deyimini `warn` deyimi haline getirmek, red iletisini `$acl_m0` değişkeninde saklayıp asıl reddi **`DATA`** komutundan sonra yapmak isteyebilirsiniz:

```
  # Aksi takdirde, posta boş gönderici adresli ise ama alıcı, imlemeli zarf
  # gönderici adresi şemasını seçenlerden biri ise, red iletisini $acl_m0
  # ve günlük iletisini $acl_m1 değişkenine kaydedip, bunları daha sonra
  # postayı reddederken kullanacağız. Red sırasında göndericinin oyalanıp
  # oyalanmayacağına bunların varlığına bakarak karar vereceğiz.
  #
  warn
    senders     =: postmaster@*
    domains     = +local_domains
    set acl_m9  = /home/${extract{1}{=}{${lc:$local_part}}}/.return-path-sign
    condition   = ${if exists {$acl_m9}{true}}
    set acl_m0  = The recipient address <$local_part@$domain> does not \
                  match a valid, signed return path from here.\n\
                  You are responding to a forged sender address.
```

```
    set acl_m1  = bogus bounce for <$local_part@$domain>.
```

Ayrıca, alıcı giden postasında imlemeli zarf gönderici adresi kullanmayı seçmiş bile olsa, bazı konaklardan gelen postaların zarf gönderici adresi boş olsa bile sınama dışı tutulmasını isteyebilir. Bilhassa eposta listelerinin sunucuları buna en iyi örnektir, bu konuda daha ayrıntılı bilgi için *Zarf Gönderici İmleri* (sayfa: 25) bölümüne bakınız.

A.12. Göndericisi Olmayan Postaların sadece Gerçek Kullanıcılar için Kabul Edilmesi

Göndericisi olmayan postaları sadece gerçek kullanıcılar için kabul edin (sayfa: 26) bölümünde açıklandığı gibi, `postmaster` gibi sistem kullanıcılarına ve rumuzlarına gönderilmiş hatalı *teslimat durum bildirimlerini* (sayfa: 73) yakalamamızı önleyen bir durum vardır. Burada, göndericisi olmayan postaları sadece gerçekten posta göndericisi olan kullanıcılar için kabul/red ettiğimizden emin olmamızı sağlayacak iki yöntem üzerinde duracağız.

A.12.1. Alıcı posta kutuluranın sınanması

İlk yöntem için *acl_rcpt_to* (sayfa: 62) ACL'sini kullanacağız. Burada, yerel bir posta kutusu olan bir alıcı adresinin varlığını sınayacağız:

```
  # Eğer gönderici adresi boşsa, bir posta kutusu olmayan kullanıcılara
  # (örn, postmaster, webmaster, v.s.) gelen postayı reddediyoruz.
  # Bu kullanıcılar posta göndermezler, dolayısıyla onlara bir posta
  # (teslimat durum bildirimi) dönemez.
  #
  deny
    message      = This address never sends outgoing mail. \
                   You are responding to a forged sender address.
    log_message  = bogus bounce for system user <$local_part@$domain>
    senders      =: postmaster@*
    domains      = +local_domains
    !posta kutusu sınaması
```

Talihsizliğe bakın ki, postanızı nasıl teslim ettiğinize bağlı olarak *posta kutusu sınaması* için yapacağımız işlem farklı olacak (örn, *imlemeli zarf göndericisi adresleri* (sayfa: 51) için alıcı adresinin eşit işaretinden önceki kısmını ayırmak gibi):

- Eğer posta kutusu isimleri olarak sunucunuzdaki kullanıcıların hesapları kullanılmışsa, alıcı isimleri ile normal kullanıcıların kullanıcı kimlikleri (500 ile 60000 arasında) karşılaştırılabilir:

  ```
    set acl_m9 = ${extract{1}{=}{${lc:$local_part}}}
    set acl_m9 = ${extract{2}{:}{${lookup passwd {$acl_m9}{$value}}}{0}}
    condition  = ${if and {{>={$acl_m9}{500}} {<${acl_m9}{60000}}} {true}}
  ```

- Posta teslimatlarınızı Cyrus[B294] IMAP yapıyorsa, posta kutularının varlığına bakmak için **`mbpath`** komut satırı aracını kullanabilirsiniz. Bunun için, Exim'in posta kutularını sınama yetkisine sahip olmasını sağlamanız gerekir (örn, onu `cyrus` grubuna ekleyebilirsiniz: **`# adduser exim4 cyrus`**).

  ```
    set acl_m9 = ${extract{1}{=}{${lc:$local_part}}}
    condition  = ${run {/usr/sbin/mbpath -q -s user.$acl_m9} {true}}
  ```

- Tüm postaları teslim etmesi için bir dış makineye yolluyorsanız, bu makinenin postayı kabul edip etmeyeceğine karar verebilmek için bir *Alıcı Varlık Sınaması* (sayfa: 17) uygulamanız gerekebilir. Varlık sınaması için özgün zarf göndericisi adresini aynen kullanmanız gerekir:

```
  verify = recipient/callout=use_sender
```

Postanın yerel olarak teslimatı durumunda, bu posta kutusu sınamaları yönlendiricilerde (routers) uygulananların birer tekrarı olacağından ve posta teslimat mekanizması bizim siteye özel olacağından, bu işlem bizim gibi mükemmelliyetçiler için biraz zorlu bir süreç olur. Bu bakımdan, şimdi başka bir yönteme bakacağız.

A.12.2. Boş göndericilerin `system_aliases` yönlendiricisinde sınanması

`postmaster` ve `mailer-demon` gibi sistem rumuzlarına gelen postaları asıl alıcısına yönlendiren `system_aliases` veya benzer isimli bir yönlendiriciniz herhalde vardır. Normalde bu rumuzlar giden postalarda gönderici olarak kullanılmazlar. Dolayısıyla, bunlara hiç gelmemesi gereken *teslimat durum bildirimlerini* (sayfa: 73) yönlerdiriciye bir kural ekleyerek yakalayabilirsiniz:

```
!senders =: postmaster@*
```

Örnek bir rumuz yönlendiricisi şöyle görünürdü:

```
system_aliases:
  driver          = redirect
  domains         = +local_domains
  !senders       =: postmaster@*
  allow_fail
  allow_defer
  data            = ${lookup{$local_part}lsearch{/etc/aliases}}
  user            = mail
  group           = mail
  file_transport  = address_file
  pipe_transport  = address_pipe
```

Böylece bazı sistem rumuzlarına gelen göndericisiz postaları engelleyebilmemize rağmen mevcut sistem kullanıcılarınının rumuzlarına ("root", "daemon", v.s) gelenleri henüz engelleyemedik. Yerel posta teslimatı için `accept` sürücüsünü ve alıcı adresleri doğrulamak için `check_local_user` kullanıyorsanız, kendinizi bu sistem hesaplarına posta yönlendirirken bulabilirsiniz.

Bu sorunu gidermek için, yerel postanızı elde etmekte kullandığınız yönlendiricide (örn, *local_user*), alıcının sadece mevcudiyetine değil, "gerçek" bir kullanıcı olup olmadığına da bakabilirsiniz. Örneğin, önceki bölümdeki gibi 500 ile 60000 arasındaki kullanıcı kimlikleriyle eşleşen kullanıcıları seçebilirsiniz:

```
  condition  = ${if and {{>={$local_user_uid}{500}}\
                        {<{$local_user_uid}{60000}}}\
                    {true}}
```

Yerel teslimatlar için örnek bir yönlendirici şöyle görünürdü:

```
local_user:
  driver           = accept
  domains          = +local_domains
  check_local_user
  condition        = ${if and {{>={$local_user_uid}{500}}\
                              {<{$local_user_uid}{60000}}}\
                             {true}}
  transport        = transport
```

Bu yöntemi kullanırken dikkatli olun, sistem kullanıcılarına ve rumuzlarına gönderilmiş göndericisiz postalar için red yanıtı bilinmeyen alıcı şeklinde (burada **`550 Unknown User`** olarak) olacaktır.

A.13. Yönlerdirilmiş Postaların Sınama Dışı Tutulması

SMTP aktarımına bu sınamaları ekledikten sonra, kendimizi posta listelerinden veya diğer sitelerin posta hesaplarından yönlendirilmiş postaları reddederek dolaylı spam yapar bir halde bulabiliriz (*Yönlendirilen Postalar* (sayfa: 27) bölümüne bakınız). En azından bizim spam ve/veya virüs filtrelemelerimiz sonucunda gönderdikleri postaların reddedilmemesii için bu konakları aklisteye alabilir ve sınamalarımızın dışında tutabiliriz.

Bu örnekte, her **RCPT TO:** komutuna yanıt verirken iki dosyaya bakıyoruz:

- Yedek posta alıcılarını ve diğer aklisteli göndericileri içeren genel amaçlı bir ak liste: `/etc/mail/whitelist-hosts`
- Yönlendirimiş posta alıcısı olan kullanıcıların postalarını yönlendiren konakları belirttikleri `/home/kullanıcı/.forwarders` dosyaları.

Eğer posta kullanıcılarınızın yerel hesapları ve ev dizinleri yoksa, bu dosyaların bulunacağı yolları değiştirebilir ve/veya sisteminize uygun bazı arama mekanizmaların kullanılmasını sağlayabilirsiniz (örn, veritabanı veya LDAP sorguları).

Eğer gönderici konak bu aklistelerden birindeyse, *Seçimlik Gecikmeler* (sayfa: 38) bölümünde anlatıldığı gibi "accept" sözcüğünü `$acl_m0` değişkenine kaydedip *$acl_m1* değişkeninin içini boşaltacağız. Böylece bu postanın daha sonraki sınamalarda reddedilmesini önlemiş olacağız.

acl_rcpt_to (sayfa: 62) ACL'sinde alıcı adresini doğruladıktan sonraya ve uzak konaklardan yerel kullanıcılara kimlik kanıtlamasız teslimatlarla ilgili `accept` deyimlerinin öncesine (varsa grilisteleme ve zarf gönderici imlemesi sınamalarının öncesine) aşağıdaki satırları yerleştireceğiz:

```
  # Gönderici konak genel akliste içindeyse postayı kabul edeceğiz.
  # Geçici olarak $acl_m9 değişkenine bu dosyayı atayacağız.
  # Konak listedeyse, $acl_m0'a bir değer yerleştirip $acl_m1'i
  # temizleyeceğiz, böylece daha sonra bu postayı reddetmeyeceğiz.
  #
  accept
    set acl_m9  = /etc/mail/whitelist-hosts
    hosts       = ${if exists {$acl_m9}{$acl_m9}}
    set acl_m0  = accept
    set acl_m1  =


  # Gönderici konak alıcının ev dizinindeki ".forwarders" dosyasındaysa
  # postayı kabul edeceğiz. Geçici olarak $acl_m9 değişkenine bu dosyayı
  # atayacağız. Konak listedeyse, $acl_m0'a bir değer yerleştirip $acl_m1'i
  # temizleyeceğiz, böylece daha sonra bu postayı reddetmeyeceğiz.
  #
  accept
    domains     = +local_domains
    set acl_m9  = /home/${extract{1}{=}{${lc:$local_part}}}/.forwarders
    hosts       = ${if exists {$acl_m9}{$acl_m9}}
    set acl_m0  = accept
    set acl_m1  =
```

acl_data (sayfa: 66) ACL'sinin çeşitli deyimlerinde *$acl_m0*'ın değerine bakarak eğer yukarıdaki gibi boş bırakılmışsa postayı reddetmeyeceğiz. Örneğin, aklisteli konaklardan gelen bir postanın RFC 2822[B301] başlığının bulunmayışı sebebiyle reddedilmesini önlemek istersek:

```
  deny
    message     = Your message does not conform to RFC2822 standard
    log_message = missing header lines
   !hosts      = +relay_from_hosts
```

```
   !senders     =: postmaster@*
    condition   = ${if !eq {$acl_m0}{accept}{true}}
    condition   = ${if or {{!def:h_Message-ID:}\
                           {!def:h_Date:}\
                           {!def:h_Subject:}} {true}{false}}
```

Bu ve ilgili diğer sınamalar sonraki *Tamamlanmış ACL'ler* (sayfa: 58) bölümünde bulunabilir.

A.14. Tamamlanmış ACL'ler

Tamam, artık canlanalım! Çok uzun bir okumadan sonra buraya kadar gelebildiğinize göre bir tebriği hakettiniz!

Bu bölümdeki ACL'ler, bu belgede bu gerçeklenim için şimdiye dek bahsettiğimiz sınamaların tamamını içermektedir. Ancak bazıları iptal edilmiştir (açıklama haline getirilerek). Bunların bazı sebepleri var:

- *Grilisteleme* (sayfa: 40). Bunun sağlanması ya ek bir yazılımın kurulu olmasını ya da Exim yapılandırma dosyasına ek ACL'ler ve tanımlamalar eklemek yoluyla oldukça karmaşık bir yapılandırma gerektirir. Yine de şiddetle tavsiye ederim.
- *Virüs tarama* (sayfa: 48). Spamı tanımlamakta kullanılan SpamAssassin gibi herkesin kullandığı *her yerde hazır ve nazır* bir tarayıcı olmadığından. Diğer yandan, `Exiscan-ACL` ile gelen belgede bu konuda epey bilgi bulabilirsiniz.
- *SpamAssassin'in kullanıcıya özel ayarları* (sayfa: 50). İletinin ilk alıcısı dışında kalan tüm alıcılarının postalarının ertelenmesi prensibiyle çalıştığından herkesin yararına değildir.
- *Zarf Göndericisi İmleri* (sayfa: 51). Dolaşımdaki kullanıcılar gibi bazıları için bazı olumsuzlukları vardır. Ayrıca diğer ACL'lerde olduğu kadar yönlendiricilerin ve aktarımcıların yapılardırılmasında da değişiklikler yapılmasını gerektirir. Daha ayrıntılı bilgi için *Zarf Gönderici İmlerinin Eklenmesi* (sayfa: 51) bölümüne bakınız.
- *Göndericisi Olmayan Postaların sadece Gerçek Kullanıcılar için Kabul Edilmesi* (sayfa: 55). Bunu yapmanın çeşitli yolları vardır ve gerçek kullanıcıların nasıl saptanacağı posta teslimatının nasıl yapıldığıyla çok ilgilidir.

Telaşa gerek yok, hepimizin beklediği sınamaları düzgün sıralanmış ACL'lerin hepsi burada.

A.14.1. `acl_connect`

```
# Bu erişim denetim listesi gelen bağlantının başlangıcında
# kullanılır. Bu sınamalar bağlantı kabul ya da red edilinceye
# kadar sırayla yapılır.

acl_connect:
  # Gecikme uygularken başlangıç olarak kullanmak üzere o anki zaman
  # bilgisini kaydediyoruz.
  warn
    set acl_m2  = $tod_epoch


  # Posta yerel SMTP üzerinden alınmışsa (yani, TCP/IP bağlantısı
  # ile gelmiyorsa), kabul ediyoruz. Bunu boş bir konak alanını
  # sınayarak yapacağız. Ayrıca, postalarını rölelediğimiz
  # konaklardan ve yerel arabirimlerden gelen postaları da kabul edeceğiz.
  accept
    hosts       =: +relay_from_hosts
```

```
  # Eğer bağlanan konak DNS karalistelerinde kayıtlıysa, $acl_c1'e
  # bir uyarı iletisi kaydedeceğiz. Bu iletiyi daha sonra posta
  # başlığına ekleyeceğiz. Varlığı bize geciktirme uygulayacağımızı
  # belirtecek.
  #

  warn
    !hosts      = ${if exists {/etc/mail/whitelist-hosts} \
                              {/etc/mail/whitelist-hosts}}
    dnslists    = list.dsbl.org : \
                  dnsbl.sorbs.net : \
                  dnsbl.njabl.org : \
                  bl.spamcop.net : \
                  dsn.rfc-ignorant.org : \
                  sbl-xbl.spamhaus.org : \
                  l1.spews.dnsbl.sorbs.net
    set acl_c1  = X-DNSbl-Warning: \
                  $sender_host_address is listed in $dnslist_domain\
                  ${if def:dnslist_text { ($dnslist_text)}}


  # Benzer şekilde, gönderici konağın DNS sorgusu başarısız olursa
  # (örn, rDNS kaydı yoksa veya belirtilen isim bağlantı kudulan IP
  # ile eşleşmiyorsa), $acl_c1'e bir uyarı iletisi kaydedeceğiz. Bu
  # iletiyi daha sonra posta başlığına ekleyeceğiz.
  warn
    condition   = ${if !def:acl_c1 {true}{false}}
    !verify     = reverse_host_lookup
    set acl_m9  = Reverse DNS lookup failed for host $sender_host_address
    set acl_c1  = X-DNS-Warning: $acl_m9


  # Bağlantıyı kabul ediyoruz, fakat $acl_c1'de evvelce kaydedilmiş bir
  # ileti varsa, göndericiyi 20 saniye dolana kadar bekletiyoruz.
  accept
    set acl_m2  = ${if def:acl_c1 {${eval:20 + $acl_m2 - $tod_epoch}}{0}}
    delay       = ${if >{$acl_m2}{0}{$acl_m2}{0}}s
```

A.14.2. acl_helo

```
# Bu erişim denetim listesi gelen bir SMTP aktarımında HELO veya EHLO
# komutları için kullanılır. Bu sınamalar selamlaşma kabul ya da
# red edilinceye kadar sırayla yapılır.


acl_helo:
  # Gecikme uygularken başlangıç olarak kullanmak üzere o anki zaman
  # bilgisini kaydediyoruz.
  warn
    set acl_m2  = $tod_epoch


  # Posta yerel SMTP üzerinden alınmışsa (yani, TCP/IP bağlantısı
  # ile gelmiyorsa), kabul ediyoruz. Bunu boş bir konak alanını
  # sınayarak yapacağız. Ayrıca, postalarını rölelediğimiz
```

```
  # konaklardan gelen postaları da kabul edeceğiz.
  #
  accept
    hosts       =: +relay_from_hosts


  # Bağlanan konak selamlaşma sırasında bir IP adresi belirtmişse, $acl_c0'a
  # bir red iletisi ve $acl_c1'e bir günlük iletisi kaydedeceğiz. Bunları
  # sonra bir "deny" deyiminde kullanacağız ve bu sırada göndericiyi 20 saniye
  # bekleteceğiz.
  #
  warn
    condition   = ${if isip {$sender_helo_name}{true}{false}}
    set acl_c0  = Message was delivered by ratware
    set acl_c1  = remote host used IP address in HELO/EHLO greeting


  # Bağlanan konak selamlaşma sırasında bizim isimlerimizden birini
  # belirtmişse aynı işleme tabi tutuyoruz.
  #
  warn
    condition   = ${if match_domain{$sender_helo_name}\
                    {$primary_hostname:+local_domains:+relay_to_domains}\
                    {true}{false}}
    set acl_c0  = Message was delivered by ratware
    set acl_c1  = remote host used our name in HELO/EHLO greeting.


  # HELO doğrulaması başarısız olmuşsa, acl_c1'e bir uyarı iletisi
  # kaydediyoruz. Bu iletiyi daha sonra posta başlığına ekleyeceğiz.
  # Varlığı bize geciktirme uygulayacağımızı belirtecek.
  #
  warn
    condition   = ${if !def:acl_c1 {true}{false}}
   !verify      = helo
    set acl_c1  = X-HELO-Warning: Remote host $sender_host_address \
                  ${if def:sender_host_name {($sender_host_name) }}\
                  incorrectly presented itself as $sender_helo_name
    log_message = remote host presented unverifiable HELO/EHLO greeting.


  # Selamlaşmayı kabul ediyoruz, fakat $acl_c1'de evvelce kaydedilmiş
  # bir ileti varsa, göndericiyi 20 saniye dolana kadar bekletiyoruz.
  accept
    set acl_m2  = ${if def:acl_c1 {${eval:20 + $acl_m2 - $tod_epoch}}{0}}
    delay       = ${if >{$acl_m2}{0}{$acl_m2}{0}}s
```

A.14.3. acl_mail_from

```
# Bu erişim denetim listesi gelen bir SMTP aktarımında MAIL FROM:
# komutu için kullanılır. Bu sınamalar gönderici adresi kabul
# ya da red edilinceye kadar sırayla yapılır.
#

acl_mail_from:
  # Gecikme uygularken başlangıç olarak kullanmak üzere o anki zaman
```

```
  # bilgisini kaydediyoruz.
  warn
    set acl_m2  = $tod_epoch


  # Posta yerel SMTP üzerinden alınmışsa (yani, TCP/IP bağlantısı
  # ile gelmiyorsa), kabul ediyoruz. Bunu boş bir konak alanını
  # sınayarak yapacağız. Ayrıca, postalarını rölelediğimiz
  # konaklardan gelen postaları da kabul edeceğiz.
  #
  # Alıcı doğrulamasını burada atlıyoruz, çünkü çoğu durumda
  # istemciler kullanıcıların posta istemcileridir ve SMTP
  # hata iletileri ile ne yapacaklarını bilmezler.
  #
  accept
    hosts     =: +relay_from_hosts


  # İleti, kimlik kanıtlaması yapılan bir bağlantı üzerinden
  # geliyorsa kabul ediyoruz. Yine, bu iletiler kullanıcıların posta
  # istemcilerinden geldiklerinden alıcı doğrulaması yapmıyoruz.
  #
  accept
    authenticated = *


  # Bu SMPT aktarımı sırasında yapılacak her teslimata uygulanacak red
  # ve/veya uyarı iletileri varsa, bunlar $acl_c0 ve $acl_c1 değişkenlerinde
  # kayıtlıdır. Bunları iletiye özel değişkenler olan $acl_m{0,1}'a
  # kopyalayıp, $acl_m1'deki uyarı iletisini ileti başlığına ekleyeceğiz.
  # (Bir red durumunda, $acl_m1 artık bir günlük kaydı içeriyor olacak,
  # ama iletiyi bununla değil başlığındaki kayıtla reddedeceğiz.)
  #
  warn
    set acl_m0  = $acl_c0
    set acl_m1  = $acl_c1
    message     = $acl_c1


  # Gönderici bir HELO/EHLO selamlaşması yapmamışsa, $acl_m0'e bir red,
  # ve $acl_m1'e bir günlük iletisi kaydedeceğiz. Bunları sonra bir
  # "deny" deyiminde kullanacağız ve bu sırada göndericiyi 20 saniye
  # bekleteceğiz.
  #
  warn
    condition   = ${if def:sender_helo_name {0}{1}}
    set acl_m0  = Message was delivered by ratware
    set acl_m1  = remote host did not present HELO/EHLO greeting.


  # Gönderici adresi doğrulanamazsa, $acl_m1'e bir uyarı iletisi
  # kaydedeceğiz ve bunu ileti başlığına ekleyeceğiz.
  # Varlığı bize geciktirme uygulayacağımızı belirtecek.
  #
  # "callout" seçeneğini isterseniz silebilirsiniz. Özellikle, postanızı
  # doğrudan değil de göstermelik sunucu (smarthost) olarak
  # gönderiyorsanız, bu seçenek anlamsız olacaktır.
```

```
  #
  warn
    condition   = ${if !def:acl_m1 {true}{false}}
    !verify     = sender/callout
    set acl_m1  = Invalid sender <$sender_address>
    message     = X-Sender-Verify-Failed: $acl_m1
    log_message = $acl_m1


  # Göndericiyi kabul ediyoruz, fakat $acl_c1'de evvelce kaydedilmiş
  # bir ileti varsa, göndericiyi 20 saniye dolana kadar bekletiyoruz.
  accept
    set acl_m2  = ${if def:acl_c1 {${eval:20 + $acl_m2 - $tod_epoch}}{0}}
    delay       = ${if >{$acl_m2}{0}{$acl_m2}{0}}s
```

A.14.4. acl_rcpt_to

```
# Bu erişim denetim listesi gelen bir SMTP aktarımında RCPT
# komutu için kullanılır. Bu sınamalar alıcı adresi kabul
# ya da red edilinceye kadar sırayla yapılır.

acl_rcpt_to:

  # Posta yerel SMTP üzerinden alınmışsa (yani, TCP/IP bağlantısı
  # ile gelmiyorsa) kabul et. Bunu boş bir konak alanını
  # sınayarak yapacağız. Ayrıca, postalarını rölelediğimiz
  # konaklardan gelen postaları da kabul edeceğiz.
  #
  # Alıcı doğrulamasını burada atlıyoruz, çünkü çoğu durumda
  # istemciler kullanıcıların posta istemcileridir ve SMTP
  # hata iletileri ile ne yapacaklarını bilmezler.
  #
  accept
    hosts       =: +relay_from_hosts


  # İleti, kimlik kanıtlaması yapılan bir bağlantı üzerinden
  # geliyorsa kabul ediyoruz. Yine, bu iletiler kullanıcıların posta
  # istemcilerinden geldiklerinden alıcı doğrulaması yapmıyoruz.
  #
  accept
    authenticated = *


  # Yerel kısım @  %  / |! karakterlerinden birini içeriyorsa,
  # iletiyi reddediyoruz. Bunlar normal yerel kısımlarda çok nadir
  # görülür, çoğunlukla röleleme sınırlamalarını aşmaya çalışanlarca
  # kullanılır.
  #
  # Ayrıca, yerel kısım bir nokta ile başlıyorsa da reddediyoruz.
  # Boş bileşenler RFC 2822'de kuraldışıdır, fakat Exim bu yaygın
  # olduğundan bunlara izin verir. Buna rağmen, bir nokta ile
  # başlayan bir yerel kısım bir dosya ismi olarak kullanılmışsa
  # (örneğin, bir posta listesi), sorunlara yol açabilir.
  #
  deny
```

```
    local_parts = ^.*[@%!/|]: ^\\.


  # Eğer $acl_m0'da kayıtlı bir sebep varsa, göndericiyi 20 saniye
  # beklettikten sonra reddediyoruz.
  #
  deny
    message     = $acl_m0
    log_message = $acl_m1
    condition   = ${if and {{def:acl_m0}{def:acl_m1}} {true}}
    delay       = 20s


  # Alıcı adres bizim postalarını kabul ettiğimiz alanlardan birine
  # ait değilse, göndericiyi 20s beklettikten sonra reddediyoruz.
  #
  deny
    message     = relay not permitted
   !domains    = +local_domains : +relay_to_domains
    delay       = 20s


  # Alıcı adres bizim postalarını kabul ettiğimiz alanlardan birine
  # ait fakat geçersizse, göndericiyi beklettikten sonra reddediyoruz.
  #
  deny
    message     = unknown user
   !verify     = recipient/callout=20s,defer_ok,use_sender
    delay       = ${if def:sender_address {1m}{0s}}



  # Zarf göndericisi adresi boş fakat postanın birden fazla alıcısı
  # varsa, bağlantıyı kesiyoruz. Meşru teslimat durum bildirimleri
  # asla bir defada birden fazla alıcıya gönderilmez.
  #
  drop
    message      = Legitimate bounces are never sent to more than one \
                   recipient.
    senders      = : postmaster@*
    condition    = $recipients_count
    delay        = 5m


  # -----------------------------------------------------------------------
  # Kullanıcı verilerini ve ayarlarını (Spamassasin gibi) destekleyebilmek
  # için gelen her iletinin alıcı sayısını bir ile sınırlayalım.
  #
  # BİLGİ: Çok sayıda kullanıcınıza gönderilmiş bir postanın yerine
  #        ulaşması her alıcı için 30 dakika veya daha fazla olmak
  #        üzere katlanarak gecikecektir. Bu, özellikle zamanın kritik
  #        önemde olduğu durumlarda sorunlara yol açacaktır.
  #
  #defer
  #  message      = We only accept one recipient at a time - please try later.
  #  condition    = $recipients_count
  # -----------------------------------------------------------------------
```

```
# Gönderici konak alıcının ev dizinindeki ".forwarders" dosyasındaysa
# postayı kabul edeceğiz. Geçici olarak $acl_m9 değişkenine bu dosyayı
# atayacağız. Konak listedeyse, $acl_m0'a bir değer yerleştirip $acl_m1'i
# temizleyeceğiz, böylece daha sonra bu postayı reddetmeyeceğiz.
#
accept
  domains     = +local_domains
  set acl_m9  = /home/${extract{1}{=}{${lc:$local_part}}}/.forwarders
  hosts       = ${if exists {$acl_m9}{$acl_m9}}
  set acl_m0  = accept
  set acl_m1  =


# Gönderici konak genel akliste içindeyse postayı kabul edeceğiz.
# Geçici olarak $acl_m9 değişkenine bu dosyayı atayacağız.
# Konak listedeyse, $acl_m0'a bir değer yerleştirip $acl_m1'i
# temizleyeceğiz, böylece daha sonra bu postayı reddetmeyeceğiz.
#
accept
  set acl_m9  = /etc/mail/whitelist-hosts
  hosts       = ${if exists {$acl_m9}{$acl_m9}}
  set acl_m0  = accept
  set acl_m1  =


# ----------------------------------------------------------------------
# Zarf Gönderici İmlemesinin Sınanması.
# Bu kısım öntanımlı olarak iptal edilmiştir, çünkü 'transports' ve
# 'routers' bölümlerindeki yapılandırmanın da değiştirilmesi gerekir.
#
# Kendine özgü imlemesini içeriyorsa, alıcı adresini kabul ediyoruz.
# Bu, teslimatın, daha önce bizden gönderilmiş bir postanın teslimat
# durum bildirimi olduğunu gösterir.
#
#accept
#  domains     = +local_domains
#  condition   = ${if and {{match{${lc:$local_part}}{^(.*)=(.*)}}\
#                           {eq{${hash_8:${hmac{md5}{SECRET}{$1}}}}{$2}}}\
#                         {true}{false}}
#
# Aksi takdirde, posta boş gönderici adresli ise ama alıcı, imlemeli zarf
# gönderici adresi şemasını seçenlerden biri ise postayı reddediyoruz.
#
#deny
#  message     = This address does not match a valid, signed \
#                return path from here.\n\
#                You are responding to a forged sender address.
#  log_message = bogus bounce.
#  senders     =: postmaster@*
#  domains     = +local_domains
#  set acl_m9  = /home/${extract{1}{=}{${lc:$local_part}}}/.return-path-sign
#  condition   = ${if exists {$acl_m9}{true}}
# ----------------------------------------------------------------------
```

```
# ------------------------------------------------------------------
# Eğer gönderici adresi boşsa, bir posta kutusu olmayan kullanıcılara
# (örn, postmaster, webmaster, v.s.) gelen postayı reddediyoruz.
# Bu kullanıcılar posta göndermezler, dolayısıyla onlara bir posta
# (teslimat durum bildirimi) dönemez.
#
# BİLGİ: Bu kısım öntanımlı olarak iptal edilmiştir, çünkü uygulanacak
#        kural yerel postanın teslimat biçimine özeldir. Bu sınamayı
#        etkinleştirmek isterseniz, aşağıdaki kurallardan size uygun
#        olan birini (ama sadece birini) devreye sokun.
#
#deny
#  message     = This address never sends outgoing mail. \
#                You are responding to a forged sender address.
#  log_message = bogus bounce for system user <$local_part@$domain>
#  senders     =: postmaster@*
#  domains     = +local_domains
#  set acl_m9  = ${extract{1}{=}{${lc:$local_part}}}
#
# --- Alıcıların yerel hesapları varsa, bu iki satırı devreye alın:
#  set acl_m9  = ${extract{2}{:}{${lookup passwd {$acl_m9}{$value}}}{0}}
# !condition  = ${if and {{>={$acl_m9}{500}} {<${acl_m9}{60000}}} {true}}
#
# --- Posta teslimatını Cyrus yapıyorsa bu satırı devreye alın:
#  condition  = ${run {/usr/sbin/mbpath -q -s user.$acl_m9} {true}}
# ------------------------------------------------------------------



# Gönderici adresinin alanadı için varsa, SPF kayıtlarını sorgulayalım.
# Gönderici konak bu alanadı için yetkilendirilmişse teslimatı kabul
# yoksa red edeceğiz.
#
deny
  message     = [SPF] $sender_host_address is not allowed to send mail \
                from $sender_address_domain
  log_message = SPF check failed.
  spf         = fail


# İleti başlığına bir SPF-Received: satırı ekleyelim.
warn
  message     = $spf_received


# ------------------------------------------------------------------
# Belli bir sunucu/gönderici/alıcı üçlüsü için grilisteleme durumunu
# sınayacağız. Bu satırları devreye almadan önce "greylistd"'yi
# kurmuş olmanız gerekir.
# Bkz:  http://packages.debian.org/unstable/main/greylistd
#
# Grilisteleme iletilerini bir boş gönderici için yapmıyoruz, çünkü
# boş gönderici adresli varlık doğrulaması işimize yaramaz (gerçek
# göndericinin varlığını sınamak için bir konağa posta gönderemeyiz).
#
#defer
#  message     = $sender_host_address is not yet authorized to deliver mail \
```

```
  #                from <$sender_address> to <$local_part@$domain>. \
  #                Please try later.
  #  log_message = greylisted.
  #  domains     = +local_domains : +relay_to_domains
  # !senders    =: postmaster@*
  #  set acl_m9  = $sender_host_address $sender_address $local_part@$domain
  #  set acl_m9  = ${readsocket{/var/run/greylistd/socket}{$acl_m9}{5s}{}{}}
  #  condition   = ${if eq {$acl_m9}{grey}{true}{false}}
  #  delay       = 20s
  # -----------------------------------------------------------------------

  # Alıcıyı kabul ediyoruz.
  accept
```

A.14.5. acl_data

```
# Bu erişim denetim listesi gelen bir SMTP aktarımında ileti verisi
# tamamen alındıktan sonra kullanılır. Bu sınamalar alıcı adresi
# kabul ya da red edilinceye kadar sırayla yapılır.

acl_data:
  # Bazı başlık satırlarını günlüğe kaydedelim.
  warn
    logwrite    = Subject: $h_Subject:


  # İleti kendi konaklarımızdan alınmış ve Message-ID başlığını
  # içermiyorsa, onu biz ekleyeceğiz.
  warn
    condition   = ${if !def:h_Message-ID: {1}}
    hosts       = +relay_from_hosts
    message     = Message-ID: <E$message_id@$primary_hostname>


  # Posta yerel SMTP üzerinden alınmışsa (yani, TCP/IP bağlantısı
  # ile gelmiyorsa), kabul ediyoruz. Bunu boş bir konak alanını
  # sınayarak yapacağız. Ayrıca, postalarını rölelediğimiz
  # konaklardan gelen postaları da kabul edeceğiz.
  #
  accept
    hosts       =: +relay_from_hosts

  # İleti, kimlik kanıtlaması yapılan bir bağlantı üzerinden
  # geliyorsa kabul ediyoruz.
  #
  accept
    authenticated = *


  # Eğer $acl_m0'da kayıtlı bir sebep varsa, göndericiyi 20 saniye
  # beklettikten sonra reddediyoruz.
  #
  deny
    message     = $acl_m0
    log_message = $acl_m1
    condition   = ${if and {{def:acl_m0}{def:acl_m1}} {true}{false}}
    delay       = 20s
```

```
# İleti uzunluğu sınırlamasını devreye sokalım.
#
deny
  message     = Message size $message_size is larger than limit of \
                MESSAGE_SIZE_LIMIT
  condition   = ${if >{$message_size}{MESSAGE_SIZE_LIMIT}{yes}{no}}


# Başlıktaki adreslerin sözdizimi hatalıysa reddediyoruz.
#
deny
  message     = Your message does not conform to RFC2822 standard
  log_message = message header fail syntax check
 !verify      = header_syntax


# Message-ID:, Date: veya Subject: başlıklarından biri olmayan bir
# ileti dışardan gelmişse reddetmek için aşağıdaki satırları devreye alın.
#
# Bazı özelleştirilmiş posta aktarımcılarının, örneğin posta listesi
# sunucularının boş gönderici adresi ile gönderdikleri postalara
# kendiliklerinden bir Message-ID üretmedikleri bilinmektedir;
# böyle durumlar için boş bir gönderici adresin varlığına da bakacağız.
#
#deny
#  message     = Your message does not conform to RFC2822 standard
#  log_message = missing header lines
# !hosts      = +relay_from_hosts
# !senders    =: postmaster@*
#  condition   = ${if !eq {$acl_m0}{accept}{true}}
#  condition   = ${if or {{!def:h_Message-ID:}\
#                         {!def:h_Date:}\
#                         {!def:h_Subject:}} {true}{false}}


# "Sender:", "Reply-To:" veya "From:" satırlarından en azından birindeki
# gönderici adres doğrulanabilir değilse, bir uyarı veriyoruz.
#
warn
  message     = X-Sender-Verify-Failed: No valid sender in message header
  log_message = No valid sender in message header
 !verify      = header_sender



# ----------------------------------------------------------------------
# Burada, zarf gönderici adresi olmayan iletilere grilisteleme
# uygulayacağız. Bunları RCPT TO:'dan sonra grilistelemeye konu
# etmeyeceğiz, çünkü gönderici varlık doğrulamaları yaparken
# karşı konaklarla olumsuz etkileşime girilebilir.
#
# Bu deyimi devreye almadan önce "greylistd" kurmuş olmalısınız.
# Bkz:  http://packages.debian.org/unstable/main/greylistd
#
#defer
```

```
#  message     = $sender_host_address is not yet authorized to send \
#                delivery status reports to <$recipients>. \
#                Please try later.
#  log_message = greylisted.
#  senders     =: postmaster@*
#  condition   = ${if !eq {$acl_m0}{accept}{true}}
#  set acl_m9  = $sender_host_address $recipients
#  set acl_m9  = ${readsocket{/var/run/greylistd/socket}{$acl_m9}{5s}{}{}}
#  condition   = ${if eq {$acl_m9}{grey}{true}{false}}
#  delay       = 20s
# -----------------------------------------------------------------



# --- EXISCAN yapılandırmasının BAŞLANGICI ---

# Birtakım MIME hataları olan iletileri reddedeceğiz.
#
deny
  message     = Serious MIME defect detected ($demime_reason)
  demime      = *
  condition   = ${if >{$demime_errorlevel}{2}{1}{0}}


# MIME taşıyıcıyı aç ve kurtlar tarafından kullanılan dosya uzantıları
# varsa reddet. Bu çağrılar tekrar demime uygulayacaktır, ama sonuçlar
# arabellekli olarak dönecektir. Uzantı listesinin eksik olabileceğini
# unutmayın.
#
deny
  message     = We do not accept ".$found_extension" attachments here.
  demime      = bat:btm:cmd:com:cpl:dll:exe:lnk:msi:pif:prf:reg:scr:vbs:url


# İletinin boyutu MESSAGE_SIZE_SPAM_MAX'dan büyükse spam veya virüs
# taraması yapmaksızın kabul ediyoruz.
accept
  condition   = ${if >{$message_size}{MESSAGE_SIZE_SPAM_MAX} {true}}
  logwrite    =:main: Not classified \
                (message size larger than MESSAGE_SIZE_SPAM_MAX)


# -----------------------------------------------------------------
# Antivirüs taraması
# Ana bölümde bir 'av_scanner' tanımı yapmış olmanız gerekir.
#
#deny
#  message  = This message contains a virus ($malware_name)
#  demime   = *
#  malware_name  = */defer_ok
# -----------------------------------------------------------------



# $spam_score ve $spam_report'a veri sağlamak için SpamAssassin'i
# çağıracağız. Tasnife bağlı olarak, $acl_m9 "ham" veya "spam"
# değerini alacak.
```

```
  #
  # İleti spam olarak tasnif edilmişse ve evvelce $acl_m0'ı iletiyi ne
  # olursa olsun kabul edeceğimizi belirtecek şekilde ayarlamamışsak,
  # iletiyi reddetmiş gibi yapacağız.
  #
  warn
    set acl_m9  = ham
    # ------------------------------------------------------------------
    # SpamAssassin için kullanıcı bazında ayarların kullanımını mümkün
    # kılmak için aşağıdaki satırı devreye alıp "spam = mail" satırını
    # iptal ediniz.
    # Alıcı adresinin kullanıcı adını SpamAssassin'e aktaralım.
    # Bunun için adresin '=' veya '@' karakterinden önceki kısmını
    # küçük harfe dönüştüreceğiz. Evvelce bir defadaki alıcı sayısını
    # önceden bir ile sınırladığımızdan çok sayıda alıcı olmayacak.
    #
    # spam        = ${lc:${extract{1}{=@}{$recipients}{$value}{mail}}}
    # ------------------------------------------------------------------
    spam        = mail
    set acl_m9  = spam
    condition   = ${if !eq {$acl_m0}{accept}{true}}
    control     = fakereject
    logwrite    =:reject: Rejected spam (score $spam_score): $spam_report



  # İletinin başlığına bir X-Spam-Status: satırı ekleyelim.
  #
  warn
    message     = X-Spam-Status: \
                  ${if eq {$acl_m9}{spam}{Yes}{No}} (score $spam_score)\
                  ${if def:spam_report {: $spam_report}}
    logwrite    =:main: Classified as $acl_m9 (score $spam_score)


  # --- EXISCAN yapılandırmasının SONU ---


  # İletiyi kabul ediyoruz.
  #
  accept
```

B. Terimler Sözlüğü

Burada, belge içinde kullanılan bazı terimlerin tanımlarına yer verilmiştir.

Açık Röle

(İng.: Open Relay) Her yerden açıkça posta kabul eden ve bu postaları heryere gönderen bir *Röle* (sayfa: 73) çeşidi.

1980'lerde sanal olarak her SMTP sunucusu birer açık röle idi. İletileri çoğunlukla tüm makineler kabul eder ve yerlerine gönderirdi. Şimdilerde ise, meşru postalar özellikle doğrudan gönderici uçtaki bir *Posta Aktarımcısı* (sayfa: 72) tarafından gönderilmekte ve alıcının alan adı için tahsis edilmiş *Posta Alıcıları* (sayfa: 72) tarafından kabul edilmektedir.

Hala internette röle işlemine açık sunucular bulunmakta ve bunlar şans eseri DNS karalistelerine girene kadar,

özellikle kimliklerini gizlemek isteyen spamcılar tarafından istismar edilmekte, çoğunlukla da milyonlarca iletiyi gönderirken yükü dağıtmak amacıyla kullanılmaktadırlar.

Ayrıca *Açık Röleye meydan vermemek* (sayfa: 16) bölümüne de bakınız.

Açık Vekil

(İng.: Open Proxy)

Her yerden açıkça TCP/IP bağlantısı kabul eden ve bunları her yere yönlendiren bir *Vekil* (sayfa: 73) çeşidi.

Bunlar, IP adreslerine gizli tutmak isteyen ve/veya yükü çeşitli konaklara ve ağlara daha verimli olarak dağıtmak isteyen spamcılar ve virüsler tarafından istismar edilirler.

Ayrıca bakınız: *Zombi Konak* (sayfa: 74)

Açıklama İsteği

(İng.: Request for Comments – *RFC*)

`http://www.rfc-«editor.org/`'daki tanımı:

> Açıklama İsteği (RFC) belgeleri internet hakkında teknik ve organizasyonel bilgiler içeren belgelerdir [...]. RFC serisindeki açıklamalar, bilgisayar ağları ile ilgili protokoller, yordamlar, programlar ve kavramlar hakkında bilgiler yanında bu konularda yapılmış toplantılardan elde edilmiş notlar, fikirler ile biraz da mizahi unsurlar içerir.

Bu belgeler protokollerin veri biçimlerinin açıklamalarını yaparak internet yönetimi ile ilgili kuralları oluşturur. Özellikle posta teslimatçıları ile ilgili olanlar:

- RFC 2821[B316], "Simple Mail transfer Protocol" (Basit Posta Aktarım Protokolü),
- RFC 2822[B317], "Internet Message Format" (Genel Ağ İleti Biçimi).

Bayes Filtreleri

(İng.: Bayesian Filters)

İleti içinde geçen sözcüklere ve sözcüklerin dizilişlerine bakarak iletinin spam olma olasılığıyla ilgili bir derecelendirme yapan bir filtre.

Meşru (ham) ve gayrimeşru (spam) iletileri belirterek bu filtreyi eğitebilirsiniz. Bu iletilerde (ham ve spam) geçen sözcüklerin bulunma sıklığına bağlı olarak her sözcüğe veya deyime bir puan verilir. Bu sözcükler ve puanları *Bayes indeksinde* saklanır.

Bu tür filtreler, yazılımcılar tarafından el yordamıyla oluşturulan anahtar sözcük tabanlı filtrelerden kaçanları yakalayabilmektedir. Çünkü bu işlemi özdevimli hale getirmektedirler.

Bayes sözcük indeksleri çoğunlukla eğitildikleri dile özeldirler. Hatta kullanıcıya özeldirler. Bu bakımdan sistem çapında, SMTP sırasında yapılan filtrelemeden ziyade kişisel içerik filtrelemesine uygundurlar (bkz. *Posta İstemcisi* (sayfa: 73)).

Ancak, spamcılar iletilerine kısa hikayeler ve sözlükten rasgele seçilmiş sözcükler ekleyerek basit bayes filtrelerini etkisiz kılacak teknikler geliştirdiler. Bu, bayes filtrelerinin atadığı spam puanını düşürerek uzun vadede bayes indeksinin kalitesini düşürmektedir.

Ayrıca bakınız: `http://www.everything2.com/index.pl?node=Bayesian`.

Alan Adı Sistemi

(İng.: Domain Name System – *DNS*)

İnternet alan adları hakkında bilgi sağlayan fiili standart. Bu bilgilere örnek olarak, sunucuların IP adresleri (*A kayıtları*), posta alıcılarının adları (*MX kayıtları*), genel sunucu bilgileri (*SRV kayıtları*) ve muhtelif dizgesel bilgiler (*TXT kayıtları*) verilebilir.

DNS, hiyerarşik ve dağıtık bir sistemdir; Her alan adı içerdiği her alt alan adı ile ilgili bilgileri içeren bir veya daha fazla sayıda DNS sunucusu ile ilişkilendirilir.

Örneğin, üst seviye alan adı olarak "org" The Public Interest Registry tarafından işleme sokulur; onun DNS sunucuları "tldp.org" alan adı sorguları için Linux Belgelendirme Projesine özel isim sunucularını görevlendirir. Sonuçta, TLDP isim sunucuları (aslında UNC sunucuları) "www.tldp.org" için ya bu alana ait bilgileri döndürür ya da bu alan adıyla ilgili bir alt seviye isim sunucusunu görevlendirir.

DNS sorguları genelde, sorgular bir internet servis sağlayıcının isim sunucularına yönlendirilerek (örn. DHCP üzerinden) gerçekleştirilir.

Bal Çanağı

(İng.: Honeypot[21] – *Sugarcane*)

Bir bal çanağı bilgi sistemlerini yetkisiz kullanmaya çalışanları saptamak ya da caydırmak amacıyla kurulan bir tuzaktır. Genelde saldırgan açısından değerli olabilecek bir bilgi içeriyormuş gibi görünen bir bilgisayardan, bir veri parçasından ya da bir ağ parçasından oluşur ve bir ağın parçasıymış gibi görünmesine rağmen ağdan yalıtılmış ve korunmuştur. Bunların açık vekil gibi davrananlarına şekerkamışı ismi verilir.

Daha ayrıntılı bilgi için `http://spamlinks.net/track-«trace-«honeypot.htm` adresine bakınız.

Çevresel Bozunma

(İng.: Collateral Damage)

Bir meşru gönderici konağın bir DNS karalistesindeki bir girdiden dolayı engellenmesi.

Bazı karalisteler (SPEWS gibi) şikayetler karşısında duyarsız davranan servis sağlayıcılarını, *tüm* müşterilerini de kapsayacak şekilde, bir IP bloğu halinde veritabanlarına kaydederler.

Ayrıca bakınız: *Hatalı Olumlama* (sayfa: 71)

Dolaylı Spam

(İng.: Collateral Spam)

Gönderici adresi taklit edilerek bir özgün iletiye yanıt gibi gönderilen otomatikleştirilmiş iletiler. Dolaylı spama örnek olarak virüs tarama raporları ("Virüs bulundu") veya *teslimat durum bildirimleri* (sayfa: 73) verilebilir.

Gönderici ücretlendirme şemaları

(İng.: Micropayment Schemes — *sender pay* schemes)

Bir iletinin her alıcısı için bir sanal *posta damgası* oluşturmak adına bazı makine kaynaklarını kullanma karşılığı olarak göndericinin ücret ödemesi şeklinde çalışan bir sistem. Bu posta damgası büyük miktarda bellek okuma/yazma işlemleri gerektiren bir matematiksel kimlik kanıtlama işleminin çözümlemesinin sonucudur. Bu damga iletinin başlığına eklenir ve alıcı bu damgayı daha basit bir kod çözme işlemi ile doğrulatır.

Ana fikir, iletinin her alıcısı için bir posta damgası gerektirmesi nedeniyle böyle bir sistemde yüzlerce binlerce kişiye spam göndermenin oldukça pahalıya malolacak olmasıdır.

Böyle iki sistem vardır:

- Camram[B323]
- Microsoft's Penny Black Project[B324]

Hatalı Olumlama

(İng.: False Positive)

Yanlışlıkla gayrimeşru posta olarak sınıflandırılmış (dolayısıyla engellenmiş) meşru posta.

Ayrıca bakınız: *Çevresel Bozunma* (sayfa: 71).

Hatalı Olumsuzlama
(İng.: False Negative)

Yanlışlıkla meşru posta olarak sınıflandırılmış (dolayısıyla filtrelenmemiş) gayrimeşru posta (spam, virüs, kötücül yazılım).

Joe İşi
Çoğunlukla üçüncü şahıslar nezdinde adres sahibine zarar vermek, hakkında yanlış kanaatler uyandırmak için onun geçerli adresinden geliyormuş gibi görünmek üzere tasarlanmış bir spam türü.

Ayrıca bakınız: http://www.everything2.com/index.pl?node=Joe%20Job

Kalleş Yazılım
(İng.: Ratware)

Çok kısa bir sürede büyük miktarlarda postayı teslim etmek üzere tasarlanmış spamcılar tarafından kullanılan eposta yazılımı ve postalama virüsleri.

Çoğu kalleş yazılım gerçeklenimi en iyi senaryo altında mümkün olduğunca sadece posta teslimatı için gerekli olan SMTP istemci koduyla işbirliğine girer. Alıcı konakla yaptıkları SMTP diyaloğunda yanlış veya belli–belirsiz bilgi verirler. Komutları göndermek için alıcının yanıtını beklemezler ve eğer alıcı taraftan birkaç saniye içinde bir yanıt alamazlarsa, bağlantıyı keserler. Geçici hataların oluşması durumunda işlem yineleme mekanizmasını kullanmazlar.

Nitelikli Alan Adı
(İng: Fully Qualified Domain Name – "FQDN")

DNS alanını da içeren, küresel olarak eşsiz internet ismi. Örneğin: "www.belgeler.org".

Nitelikli bir alan adı her zaman tek bir makineyi göstermez. Örneğin, yük dengelemesi amacıyla bazı hizmetler sunuculara dağıtılır, dolayısiyle "www" gibi hizmet isimleri çok sayıda IP adresini gösterebilir. Yine de, belli bir makinenin birincil konak ismi o makineye özel olmalıdır; örneğin: "p16.www.scd.yahoo.com".

Nitelikli bir alan adı daima bir nokta (".") içerir. İlk noktadan önceki parça küresel olarak eşsiz olmayan *niteliksiz isimdir*.

Posta Aktarımcısı
(İng.: Mail Transport Agent – *MTA*)

Bir posta sunucusunda çalışan, internet alanının posta alıcısı olarak davranabilen diğer konaklarla posta alışverişi yapabilen bir yazılım. Sendmail, Postfix, Exim ve Smail tanınmış posta aktarımcıları arasında sayılabilir.

Posta Alıcısı
(İng.: Mail Exchanger – *MX*)

Özellikle, bir internet alanına gelen postaları almaya (bazan göndermeye de) adanmış bir makine.

Bir internet alanının DNS bilgileri normalde, bu alan için gelen postaları alacak makinelerin bir listesini içerir. Bu listeye "MX kaydı" denir ve öncelik sırasını belirten bir numarayla imlenmiş makine isimleri içerir. Listedeki en küçük numaralı makine postaları almada en yüksek önceliğe sahip makine kabul edilir ve bu makineye "birincil posta alıcısı" denir.

Posta Döngüsü

(İng.: Mail Loop — *Ringing*)

Bir özdevinimli iletinin bir diğerini tetiklemesiyle dolaylı veya doğrudan ilk iletinin tekrar tetiklenmesiyle süregiden bir durum.

Üyelerinden birinin adresinin o eposta listesinin adresi olduğu durum buna bir örnek olarak verilebilir. Bu gibi durumların üstesinden, liste sunucusu tarafından iletinin başlığına bir "X–Loop:" satırı eklenerek gelinir; bu satırı içeren bir posta tekrar işleme sokulmaz.

Posta İstemcisi

(İng.: Mail User Agent – *MUA* veya *Mail Reader*)

Posta alma, gönderme, indirme, erişim gibi yetenekleri olan kullanıcı yazılımı. Örneğin, Microsoft Outlook/Outlook Express, Apple Mail, Mozilla Thunderbird, Ximian, Evolution, KMail.

Posta Teslimatçısı

(İng.: Mail Delivery Agent – *MDA*)

Kullanıcıların posta kutularının bulunduğu makinede çalışan, görevi postaları bu eposta kutularına teslim etmek olan yazılım. Bu teslimat çoğunlukla *Posta Aktarımcısı* (sayfa: 72) tarafından ikincil bir görev olarak yerine getirilir. Sadece posta teslimatçısı olan yazılımlara, Deliver, Procmail, Cyrmaster ve/veya Cyrdeliver (Cyrus IMAP ailesinden) örnek verilebilir.

Röle

Bir epostayı internetten alıp internete gönderen bir makine. Röleye örnek olarak bir servis sağlayıcının müşterilerine epostalarını gönderebilmeleri için tahsis ettiği konak ("smarthost") gösterilebilir.

Ayrıca bakınız: *Açık Röle* (sayfa: 69), *Vekil* (sayfa: 73)

Spam Tuzağı

(İng.: Spam Trap)

Kamuya açık alanlardan eposta adresi toplayan robotları *yemlemekte* kullanılan bir eposta adresi. Tuzağa yakalananlar *DNS Karalisteleri* (sayfa: 13) ve *Döküntü Posta İmza Depoları* (sayfa: 22) gibi dost araçları beslemekte kullanılır.

Bu adreslere gönderilmiş postalar normal olarak ya spamdır ya da kötücül yazılımdır. Ancak bazıları dolaylı spam olacaktır – gönderici adreslerinin taklit edilmesi durumu (*Teslimat Durum Bildirimi* (sayfa: 73)). Bu tür (dolaylı spam) iletilerin spam olarak kayda geçirilmemesi için gerekli önlemler alınmış olmalıdır, aksi taktirde spam tuzağı güvenilir olmayacaktır.

Teslimat Durum Bildirimi

(İng.: Delivery Status Notification, *DSN*)

Bir özgün iletinin göndericisini durum hakkında bilgilendirmek için bir *Posta Aktarımcısı* (sayfa: 72) veya *Posta Teslimatçısı* (sayfa: 73) tarafından özdevinimli olarak oluşturulan bir ileti. Teslimat Durum Bildirimleri genelde, bir iletinin geçici veya kalıcı bir sorundan dolayı teslim edilemediği ve/veya bir süre daha bu teslimatın gerçekleştirilmesinin denenip denenmeyeceğini hakkında özgün iletinin göndericisine bilgi vermek için gönderilir.

Teslimat Durum Bildirimleri bir boş *Zarf Göndericisi* (sayfa: 74) adresiyle gönderilir.

Vekil

Başkalarının yararına çalışan bir makine. TCP/IP bağlantıları ya da HTTP istekleri için genellikle internet adresleriyle ilgili yönlendirme yapar. Örneğin, şirketlerin – bazan bir ülkenin tamamının – dahili ağlarından yapılan HTTP isteklerini filtrelemek için sıkça kullanılır. Bundan, son kullanıcının haberi olabileceği gibi olmayabilir de.

Ayrıcı bakınız: *Açık Vekil* (sayfa: 70), *Röle* (sayfa: 73).

Zarf Alıcısı
(İng.: Envelope Recipient)

İletinin gönderildiği e–posta adres(ler)i. Bunlar SMTP aktarımı sırasında `RCPT TO` komutuyla kullanılır. Bu adres(ler) iletinin "To:" ve "Cc:" başlıklarında belirtilenden farklı adres(ler) olabilir.

Ayrıca bakınız: *SMTP Aktarımı* (sayfa: 10)

Zarf Göndericisi
(İng.: Envelope Sender)

Bir iletinin SMTP aktarımı sırasında, `MAIL FROM:` komutunda gönderici olarak belirtilen e–posta adresi. Bu adres iletinin "From:" başlığında belirtilenden farklı bir adres olabilir.

Tek özel durum *Teslimat Durum Bildirimi* (sayfa: 73) (gönderici adresi olmayan [bounced] ileti, return receipt, vacation message..) durumudur. Böyle postalar için *Zarf Göndericisi* (sayfa: 74) boştur. Bu genellikle, *Posta Döngüsü* (sayfa: 72)nden kaçınmak ve bunları "normal" postalardan ayırmayı mümkün kılmak için yapılır.

Ayrıca bakınız: *SMTP Aktarımı* (sayfa: 10)

Zombi Konak
(İng.: Zombie Host)

Postalama virüsleri veya kurtları bulaşmış internete bağlı bir makine. Bu makineler değişmez bir şekilde Microsoft® Windows® ailesinden bir işletim sistemi kullanan makinelerdir ve hemen hemen daima mahalli IP adres bloklarındadırlar. Bunların sahiplerinin makinelerine virüs bulaştığından ya haberleri yoktur ya da önemsemiyorlardır ve çoğunlukla da bunların servis sağlayıcıları bunlara hizmet vermemek gibi bir önleme başvurmaz.

Bu sebeplerden, böyle "mahalli" adres bloklarını veritabanlarına ekleyen "dul.dnsbl.sorbs.net" gibi bazı DNS karalisteleri vardır. Böyle servis sağlayıcılardan hizmet alanlar meşru postalarını göndermek için normalde servis sağlayıcının posta sunucusunu kullandıklarından mahalli adreslerden gelen postaları reddetmek için bu karalisteleri kullanmalısınız.

Notlar

Belge içinde dipnotlar ve dış bağlantılar varsa, bunlarla ilgili bilgiler bulundukları sayfanın sonunda dipnot olarak verilmeyip, hepsi toplu olarak burada listelenmiş olacaktır.

(*B1*) `../howto/gpl.pdf`

(*B6*) `http://www.tldp.org/HOWTO/html_single/Spam-«Filtering-«for-«MX/#history`

(*B7*) `http://www.tldp.org/HOWTO/html_single/Spam-«Filtering-«for-«MX/#history`

(*B10*) `http://www.openspf.org/`

(*B11*) `http://www.danisch.de/work/security/antispam.html`

(*B13*) `http://rhyolite.com/anti-«spam/dcc/`

(*B14*) `http://razor.sf.net/`

(*B15*) `http://pyzor.sf.net/`

(*B16*) `http://www.spamcop.net/`

(*B17*) `http://www.spamhaus.org/`

(B18) http://www.sorbs.net/

(B19) http://cbl.abuseat.org/

(B20) http://moensted.dk/spam/

(B21) http://www.spamassassin.org/

(B22) http://www.spamassassin.org/full/3.0.x/dist/CREDITS

(B28) http://www.spamassassin.org/

(B31) http://slett.net/gallery/2003-«05/IMG_1655

(1) Eğer postayı reddederseniz, güvenilmez üçüncül sunucular hala dolaylı spam üretebilir. Yine de, bu sunucu bir *Açık Vekil* (sayfa: 70) veya *Açık Röle* (sayfa: 69) olmadıkça, büyük ihtimalle postaları meşru göndericilerin geçerli adreslerine teslim edecektir. Postayı reddederek postanın sizin gelen posta kuyruğunuzda donması yerine onların giden posta kuyruklarında donmasını sağlamak daha iyidir. Böylelikle, bu sunucuların sahiplerine de bir ipucu verilmiş olur.

(B59) http://tmda.net/

(2) Kişisel olarak, hiçbir durumda bu tekniklerin iyi bir fikir olmadığını düşünüyorum. Bunlar, *Dolaylı Spam* (sayfa: 71) üretirler. Bunlar örneğin aylık banka talimatları gibi özdevinimli hale getirilmiş kaynaklardan posta gönderimi sözkonusu olduğunda özel bir dikkat gerektirirler ve kişilerin birbiriyle iletişim kurabilmesi için önlerine aşmaları gereken engeller konulmuş olduğundan epostanın kullanım rahatlığını bozarlar. Çoğu zaman, meşru posta göndericileri doğrulama isteği karşısında neler yapılması gerektiğini bilmediklerinden ya da bununla uğraşmak istemediklerinden posta kaybolur.

(3) SPEWS (http://www.spews.org/) *karalistesi* (sayfa: 13) benzeri listeleri ortaya koyan "spam filtreylemcileri"nden çok kesin bir şekilde farklı bir yaklaşım ortaya koymaya çalıştım. Örnek olarak SPEWS listesinin benimsediği yaklaşım, *yaratılacak ikincil hasar* (sayfa: 71) üzerinden İSS'ler üzerinde baskı oluşturmak ve kullanıcılardan gelecek şikayetlere İSS'lerin yanıt vermesini sağlamaktır. Kullanıcıların ikincil hasar yaratanlara yaptıkları şikayetlere aldıkları yanıt tipik olarak "sizin İSS'niz bundan sorumlu, gidin onlara başvurun" veya "İSS'nizi değiştirin" şeklinde olmaktadır.

Mamafi bunlar uygulanabilir seçenekler değildir. Gelişmekte olan ülkeleri ele alırsak kullanıcıların daha iyi hizmet alabilecekleri başka bir İSS olmayabilir. Gelişmiş ülkelerdeki geniş bant sağlayıcılarını ele aldığımızda ise çoğu yerde bunların birer tekel olduğunu görmekteyiz. Dolayısıyla SPEWS gibi listelere güvenmekle karşılacağımız sorunlar aşikardır.

Daha basit bir şekilde ifade edersek, döküntü postayı ayıklamak için başka çok daha iyi ve doğruluğu yüksek yöntemler mevcuttur.

(B69) ../rfc/rfc2821.pdf

(B70) http://www.brandenburg.com/specifications/draft-«crocker-«mail-«arch-«00.htm

(B71) ../rfc/rfc2821.pdf

(B75) ../rfc/rfc2822.pdf

(B78) ../rfc/rfc2821.pdf

(B81) ../rfc/rfc2821.pdf

(4) Gelen bir SMTP bağlantısını kapının önünde bekletirken dikkatlı olmalısınız, çünkü kendi sunucunuzun TCP soketini de meşgul ediyor olacağınız gibi bu işleme ayrılmış bellek ve diğer sunucu kaynakları da fazladan harcanıyor olacaktır. Eğer sunucunuz genel olarak meşgul durumdaysa servis reddi (DoS) saldırılarına da açık hale gelecektir. Daha kabul edilebilir bir seçenek, göndericinin bir kalleş yazılım olduğunun kesin kanıtını elde ettikten sonra bağlantıyı kesmek olabilir.

(5) Karaliste denince "mail–abuse.org" anlaşıldığından bunlara DNS karalisteleri – DNSbl – ya da daha tanımlayıcı olarak "Gerçek zamanlı Karadelik Listeleri" denir.

Farklı amaçlar için kullanılan benzer listeler de vardır. Örneğin, "bondedsender.org" bir DNS aklistesidir (DNSwl); bunlarda kayıtlı olan "güvenilir" IP adreslerinin sahipleri kendilerinden bir spam kaynaklandığı takdirde bir ceza ödemeyi (bir senetle) taahhüt etmişlerdir. Diğer listeler, ülkelere ve İSS'lere özel IP adresleri içeren listelerdir.

(6) Örneğin, dünyanın en büyük İSS'sinin, comcast.net'in posta aktarımcıları bu belgenin yazımı sırasında SPEWS'in 1. seviye listesine kaydedilmişti. Comcast'ın kendi müşterilerini kurallara uymaya daha etkin zorlaması gerektiğinden bu tamamen haksız bir uygulama olmasa da, bunun sonuçlarından benim gibi bunda hiç suçu olmayanlarla birlikte, ABD internet kullanıcılarının %30'u etkileniyor.

Duruma açıklık kazandırmak için SPEWS SSS (http://spews.org/faq.html)'inde yayınlanmış olan açıklamaya bakalım:*1. seviye listesi ekseriyetle, birkaç meşru müşterisi varmış gibi görünen ama genelde spamcılar veya spam işlemine destek verenlerin sahibi oldukları IP bloklarından oluşur.* Teknik olarak, bu bilgi doğrudur, ancak (a) Comcast'ın bir "spam destekçisi" olduğunu, (b) "diğerleri"nin azınlık olduğunu varsayarsanız. Duruma bakınca bu bilginin açıkça yanlış olduğu görülür.

(B104) ../rfc/rfc2821.pdf

(B107) ../rfc/rfc2821.pdf

(B109) ../rfc/rfc2821.pdf

(7) Döküntü postayı normal postadan ayırmak açısından bu sınama normalde yeterliymiş gibi görünse de, listserv (http://www.lsoft.com/products/default.asp?item=listserv) kurulumlarının liste sunucusunun çıplak IP adresiyle selamlaşmayı başlattığı şeklinde L–Soft'un hata raporları vardır.

(8) *Teslimat Durum Bildirimi* (sayfa: 73) ve özdevinimli üretilmiş diğer yanıtlarda kullanılan **MAIL FROM:** <> gibi boş zarf göndericili komutlar *özel* bir durumdur.

(B125) http://www.livinginternet.com/e/ew_addr.htm

(B131) ../man/man5/man5-«passwd.pdf

(B132) http://www.openssh.org/

(B133) ../man/man1/man1-«scp.pdf

(B134) ../man/man8/man8-«cron.pdf

(9) Nadiren, groups.yahoo.com gibi bazı "meşru" büyük hacimli posta göndericileri geçici başarısızlığa uğramış teslimatları gerçekleştirmeye çalışmazlar. Evan Harris böyle göndericileri aklisteler için yararlı olabileceğini düşünerek liste halinde derlemiş: http://cvs.puremagic.com/viewcvs/greylisting/schema/whitelist_ip.txt?view=markup.

(10) Büyük siteler giden postalar için çoğunlukla birden fazla sunucu kullanırlar. Örneğin, anlık teslimatlar için bir sunucu veya bir sunucu havuzu kullanılırken, ilk teslimat başarısız olduğunda posta, büyük kuyrukları idare edecek şekilde yapılandırılmış son çare sunucularına devredilir. Bu nedenle, böyle sitelerden gelen ilk iki teslimat başarısız olacaktır (IP adresinin değişmesi nedeniyle).

(11) Ç.N.: Alan adı sahibi olmak hiçte zor olmadığından bu iddia sönük kalmakta, ayrıca İSS gibi kurumların müşterilerinin dinamik adreslerden gönderdikleri postaların genel bir kabul olarak spam olarak değerlendirildiği de gözönüne alındığında bu iddia iyice mesnetsiz kalmaktadır.

(B148) http://www.openspf.org/srs.html

(B151) ../rfc/rfc2822.pdf

(B157) ../rfc/rfc2822.pdf

(12) Posta listelerinin sunucuları gibi bazı özel amaçlı posta aktarımcıları "boş adresli" (bkz. *Teslimat Durum Bildirimi* (sayfa: 73)) iletiler için kendiliklerinden bir Message-ID: başlığını üretmezler. Bu tür iletiler boş bir *Zarf Göndericisi* (sayfa: 74)nin varlığıyla kendilerini belli ederler.

(B165) http://razor.sf.net/

(B166) http://pyzor.sf.net/

(B167) http://rhyolite.com/anti-«spam/dcc/

(B168) http://asg.web.cmu.edu/cyrus/

(13) IMAP protokolü posta istemcilerine boş karakterlerin aktarılmasına izin vermez. Bu bakımdan, Cyrus geliştiricileri bu karakteri içeren postalardan kurtulmanın en kolay yolu olarak onları reddetme kararı aldılar.

(B175) http://www.vanja.com/tools/sophie/

(B176) http://www.kapersky.com/

(B177) http://clamav.elektrapro.com/

(B178) http://www.sald.com/

(B181) http://www.spamassassin.org/

(B182) http://www.brightmail.com/

(14) Virüs tarama yazılımı üreticilerinin virüs içeren epostalardaki gönderici adreslerine neden güvendiklerini açıklamak ancak psikoanalitik bir çalışmanın konusu olabilir.

(B222) http://duncanthrax.net/exiscan-«acl/

(15) Dağıtımın öntanımlı posta aktarımcısının Exim olması Debian GNU/Linux (http://www.debian.org/) kullanıcılarının özelinde, Exim'i en çok tercih edilen posta aktarımcısı haline getirmektedir. Eğer siz de bir Debian ("Sarge" veya üstü) kullanıcısı iseniz, Exim+Exiscan–ACL'ye exim4-daemon-heavy paketini kurarak kavuşabilirsiniz:

```
# apt-get install exim4-daemon-heavy
```

(B224) http://www.spamassassin.org/

(B225) http://packages.debian.org/unstable/mail/greylistd

(16) *Debian kullanıcılarının dikkatine:* exim4-config paketi Exim yapılandırmasını tek bir yapılandırma dosyası halinde mi tutacağınızı yoksa /etc/exim4/conf.d altındaki çeşitli dizinlere ve dosyalara bölünmüş olarak mı tutacağınızı seçme imkanı tanır.

Eğer, çok dosyalı yapılandırmayı seçerseniz (ben bunu öneriyorum!), exim4-config ile sağlanan dosyalara dokunmadan (içeriklerini değiştirmeden) bu alt dizinlere kendi oluşturduğunuz dosyaları ekleyerek yapılandırmayı özelleştirebilirsiniz. Örneğin, **RCPT TO:** komutu için kendi ACL'nizi belirtmek istersseniz, /etc/exim4/conf.d/acl/80_local-config_rcpt_to isimli bir dosya oluşturabilirsiniz (bkz. *acl_rcpt_to* (sayfa: 33)).

Exim başlatma betiği (/etc/init.d/exim4) her çalıştırılışında bu dosyalardan büyükçe ve tek bir çalışma anı yapılandırma dosyası oluşturur.

(B227) http://www.exim.org/exim-«html-«4.60/doc/html/spec.html/ch14.html#id2571138

(B228) http://www.exim.org/exim-«html-«4.60/doc/html/spec.html/ch39.html

(B261) http://www.openspf.org/srs.html

(17) Debian Kullanıcılarının dikkatine: 14 Temmuz 2004 itibariyle Exiscan–ACL'nin SPF desteği içeren sürümü exim4-daemon-heavy paketine dahil edilmemişti. (Ç.N.: Bu çeviri yapılırken Debian'ın kararlı dağıtımıyla gelen exim4-daemon-heavy paketi hala bu desteği içermiyordu – Aralık 2005. Ancak diğer Debian dağıtımlarında bu destek var.) Şimdilik, başka bir SPF gerçeklenimi kurabilirsiniz:

```
# apt-get install libmail-spf-query-perl
```

(18) Her ne kadar Bayes eğiticisi işleyiş olarak kullanıcıya özel ise de SpamAssassin'in Bayes eğiticisi naçizane fikrime göre herhalükarda o kadar da parlak değildir. Örnek olarak spam göndericiler sözlükten rasgele seçilmiş kelimeler ve öykülerle bu tür sistemleri tohumlayarak alt edebilmektedir.

(19) Eğer bu işlemin bu derece karmaşık olması gerekmediğini düşünüyorsanız, çok yüzeysel bakıyorsunuz demektir. Bu belgenin eski sürümlerinde imin son elemanını üretmek için basitçe ${hash_8:SECRET=....} kullanmıştım. Ancak, Exim'in ${hash...} işlevini biraz kavrayınca ve farklı alıcılara gönderilen örnekleri biraz inceleyince, teknik olarak imlemenin taklit edilmesinin mümkün olacağını görürsünüz. Matthew Byng–Maddic <mbm (at) colondot.net> bu konuda şöyle diyor:

> Normal şartlar altında ürettiğiniz belgeyi çok sayıda insan kullansın diye hazırlarsınız. Ama aynı zamanda Kirşof kanunu da işlemeye başlar, tüm gizliliği sağlayan şey kullandığınız anahtardır. Spam göndericileri için bir kaç tane geri dönüş yolunu kullanarak

> bu anahtarı geri kazanmak imkansız değildir ve bir kez bunu yaptıklarında aynı alan adından tekrar geçerli geri dönüş yolu içeren spam iletilerini yaymaya başlarlar, siz de başladığınız yere geri dönmüş olursunuz. [...] Bana göre daha iyisi, işi baştan sıkı tutmaktır.

(20) Yukarıdaki örneklerde, `/home//.return-path-sign` dosyası mevcut olmayabileceğinden `senders` kuralı aslında gereksizdir. Ancak, ongunluk açısından kuralı açıkça kullanıyoruz.

(B294) http://asg.web.cmu.edu/cyrus/

(B301) ../rfc/rfc2822.pdf

(B316) ../rfc/rfc2821.pdf

(B317) ../rfc/rfc2822.pdf

(21) "Bal Çanağı" özgün belgede yoktur. Çevirmen tarafından eklenmiştir.

(B323) http://www.camram.org/

(B324) http://research.microsoft.com/research/sv/PennyBlack/

Bu dosya (spam–filtering.pdf), belgenin XML biçiminin TEXLive ve belgeler-xsl paketlerindeki araçlar kullanılarak PDF biçimine dönüştürülmesiyle elde edilmiştir.

26 Ocak 2007