Internet'e Bağlanırken Gerekenler: Firewall ve Proxy

Yazan:
Deniz Akkuş

Ocak 2003

Özet

Bir kaç saatlik bir çalışma ile ufak bir makina üzerine bir güvenlik duvarı kurabilirsiniz. Güvenlik duvarları salt dış saldırılara karşı sisteminizi korumakla kalmaz, performans arttırıcı ve izin politikası uygulayıcı amaçlar için de kullanılırlar.

Konu Başlıkları

Sürüm Bilgileri

v1.0

Yasal Uyarı

1. Giriş

Her şirkette bir gün İnternete bağlantı ve bunun getireceği yararlar gündeme getirilir. Fakat Internet bağlantısı, Bilgi İşlem açısından bazı sorunları beraberinde getirmektedir.

- Dışarıdan içeriye yapılacak saldırılar.
- İçeriden yetkisiz kişilerin dışarıya bilgi göndermesi.
- Internet'de "tehlikeli alanlarda" dolaşma sonucunda sisteme virüs bulaşması.
- Internet'de özellikle vakit kaybettirici bazı sitelere ulaşımın şirket içerisinde, çalışma saatlerinde yapılması.
- Yetkisiz kullanıcıların Internet'de gezinmesi.

Günün sonunda, %100 güvenlik ve kontrol yoktur. Fakat güvenlik ve kontrolü, kolaylıkla bertaraf edilebilir halden çıkarmak mümkündür.

1.1. Güvenlik Duvarı (Firewall) Nedir?

Bütününe güvenlik duvarı dediğimiz servisler aslında *bir kaç alt kavramdan oluşmaktadır* (sayfa: 3): *Tabya (Bastion Host)* (sayfa: 3), *Ağ Adres Çevrimi (NAT), Maskeleme* (sayfa: 4), *Paket Filtreleme* (sayfa: 5), *Vekil (Proxy)* (sayfa: 6). Bütün güvenlik duvarları (ticari olanlar ve olmayanlar), bu uygulamaların hepsini veya bir kısmını uygularlar.

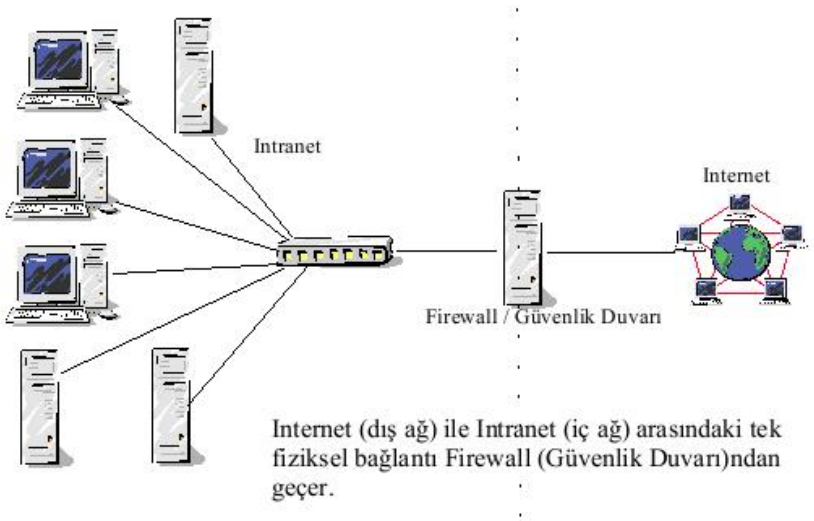
2. Güvenlik Duvarı Kavramları

2.1. Tabya (Bastion Host)

İdealde, ağınızdaki güvenlik, ağ seviyesinde ve ağdaki her bir makinada uygulanır. Pratikte ise, bu ya yapılamamakta ya da ihtiyaç duyulan kimi protokollerin güvenlikten yoksun olduğu bilinse dahi kullanılmaktadır. Böyle durumlarda güvenlik duvarı, içeride birbirlerine güvenen, az korumalı makinaların olduğu bir ağla, dış dünya arasına yerleştirilir ve aradaki fiziksel bağlantı yalnızca güvenlik duvarı tarafından sağlanır. Dolayısıyla içerideki ağa girmek isteyen her kötü niyetli dış saldırı, önce özel olarak korumalı tasarlanmış güvenlik duvarı makinasını bertaraf etmek zorundadır. Bu makinaya "kale", "nöbetçi kale" anlamına gelen ***tabya***[1] (bastion host) da denir. Tabyamız, fiziksel olarak iki farklı ağa bağlıdır: iç ağ (Intranet) ve dış ağ (Internet). Tabya iki özelliğe sahiptir:

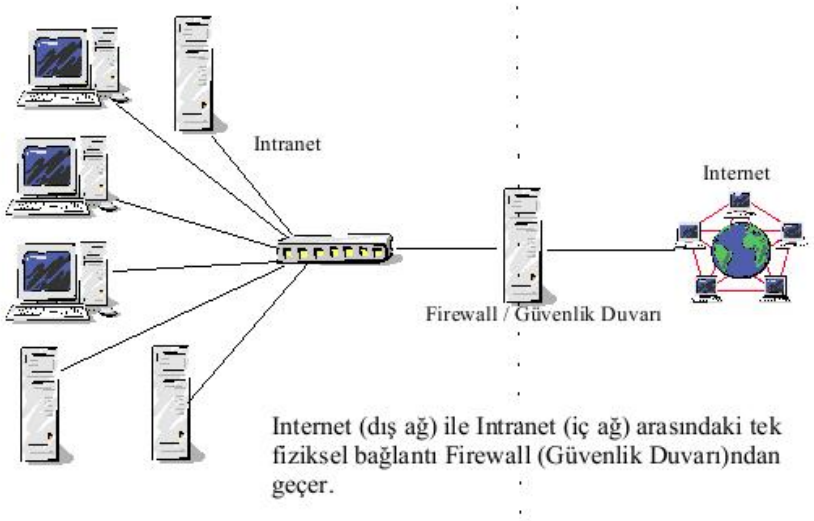
- Yüksek güvenliğe sahip olmalıdır — yani bu makinaya izinsiz erişim son derece zor hale getirilmelidir.
- İki (bazen üç) fiziksel ağ bağlantısına sahip olmalı ve bu farklı ağlar arasındaki iletişimin nasıl yapılacağına dair karar verebilmelidir.

Şekil 1. İç ağ ile dış ağ arasında güvenlik duvarı



- Yüksek güvenliğe sahip olmalıdır — yani bu makinaya izinsiz erişim son derece zor hale getirilmelidir.
- İki (bazen üç) fiziksel ağ bağlantısına sahip olmalı ve bu farklı ağlar arasındaki iletişimin nasıl yapılacağına dair karar verebilmelidir.

Şekil 2. İç ağ ile dış ağ arasında güvenlik duvarı



2.2. Ağ Adres Çevrimi (NAT), Maskeleme

Günümüzde iç ağların hemen hepsi tahsisli olmayan IP numaraları (`10.0.0.0, 192.168.0.0` vs.) kullanmaktadır. Bu IP numaraları Internet üzerindeki yönlendiriciler (router) tarafından bilinmez. Dolayısıyla bu ağlardan Internet'deki herhangi bir makinaya bir erişim olduğu zaman Internet'deki makina bu ağa nasıl geri döneceğini bilmez ve pratikte iletişim yapılamaz. Güvenlik duvarı ise, dinamik veya statik olarak Internet'de bilinen ve kendisine yönlendirme yapılabilen bir IP numarasına sahiptir. İç ağdaki makinalara erişim sağlayabilmek için güvenlik duvarı, kendisine iç ağdan gelen her paketin kaynak adresini kendi adresi olarak değiştirir. Kendisine Internet'den gelen paketlerin de hedef adresini iç ağdaki ilgili makinanın adresi olarak değiştirir ve bu yolla

iç ağdaki makinaların Internet üzerindeki makinalarla haberleşmesini sağlar. Bu işleme IP ***IP Maskelemesi*** (Masquerade) veya ***Ağ Adres Çevrimi*** (NAT – Network Address Translation) denir.

NAT yapıldığı zaman, oluşan trafiğin Internet'den görüldüğü hali, Internet'de bulunan tek bir makinanın (tabyamız) bazı Internet alışverişleri yaptığıdır. Internet'e, bu makinanın arkasındaki ağın büyüklüğü, bu ağdaki makinaların cinsi, sayısı, ağın yapısı vs. hakkında herhangi bir bilgi gitmez. Dolayısıyla NAT, yalnızca tahsissiz ağlardan Internet'e erişimi sağlamakla kalmaz, ağınızdaki makinalar hakkında bilgi edinilmesini (ve dolayısıyla size karşı yapılabilecek saldırıları) zorlaştırır.

2.3. Paket Filtreleme

Yukarıda bahsedilen önlemler (güvenlik duvarının tek fiziksel bağlantı olması, NAT uygulanması) ağınıza belli bir miktar güvenlik sağlar, fakat esas güvenlik, paket filtreleme yöntemlerinden gelir. Bu yöntemler, güvenlik duvarından geçen her IP paketine bakılması ve ancak belli şartlara uyarsa geçişine izin verilmesi şeklinde uygulanır.

Örneğin:

- İç ağınızdan kimsenin Internet'de ICQ kullanmasını istemiyorsunuz.
- Dışarıdan içeriye hiç kimsenin telnet yapabilmesini istemiyorsunuz.

Bu hedefleri gerçekleştirmek için paket filtreleme yöntemleri kullanacaksınız. *Paket filtreleme*, güvenlik duvarının her fiziksel bağlantısı üzerinde ayrı ayrı ve yöne bağlı (dışarıya çıkışa izin ver, fakat içeriye girişe izin verme) olarak uygulanabilir.

Paket filtrelemede özellikle yapmanız gereken minimum, dışarıdan gelip de kaynağını içerisi gibi gösteren (IP spoofing – IP aldatmacası) paketleri ve devam etmekte olan bir trafiğin parçası imiş gibi gelen paketleri (IP fragments) filtrelemek ve bunların geçişine izin vermemektir. Çoğu saldırı, bu şekilde başlar.

Bu minimumu sağladıktan sonra, dışarıdan içeriye yapılmasına izin verdiğiniz erişimleri (telnet yapsınlar mı?, ping yapabilsinler mi?) ve içeriden dışarıya yapılmasına izin verdiğiniz erişimleri (kullanıcılarınız dışarıya telnet yapabilsin mi? Web'e erişsinler mi? ICQ yapabilsinler mi?) belirlemeniz ve güvenlik duvarı üzerindeki filtre protokollerinizi buna göre oluşturmanız gerekir.

2.4. Dinamik (Stateful) Filtreleme

Eskiden filtreleme yöntemleri ağırlıklı olarak statikti — yani genel olarak ağınıza ICQ paketlerinin girmesine izin verip vermeme kararı söz konusu idi. 2.4 Çekirdeği ve bizim aşağıda örneğini verdiğimiz **`iptables`** uygulaması[B7] ile birlikte dinamik filtreleme Linux üzerinde kullanılabilir hale geldi. Aradaki fark, paketin sırf protokolüne bakarak karar vermek yerine, güvenlik duvarının bir bağlantıyı hangi tarafın başlattığını takip etmesi ve çift yönlü paket geçişlerine buna göre karar vermesidir. Yani bir telnet bağlantısında her iki taraftan da paketler gelir ve gider. Fakat *dinamik filtreleme* ile, bir telnet bağlantısı iç ağınızdan başlatılmışsa izin verir, başlangıç istemi dış ağdan gelmişse reddedebilirsiniz. *Dinamik filtreleme* özelliği olmayan güvenlik duvarlarını kullanmanızı önermiyoruz. 2.4 çekirdeği ve **`iptables`** uygulaması olan her Linux üzerinde *dinamik filtreleme* yapabilirsiniz. **`iptables`** kullanımı hakkında daha ayrıntılı bilgiyi Iptables'ın Basit Kullanımı[B8] belgesinde bulabilirsiniz.

2.4.1. Bazı Internet Servislerinin İç Ağdan Verilmesi

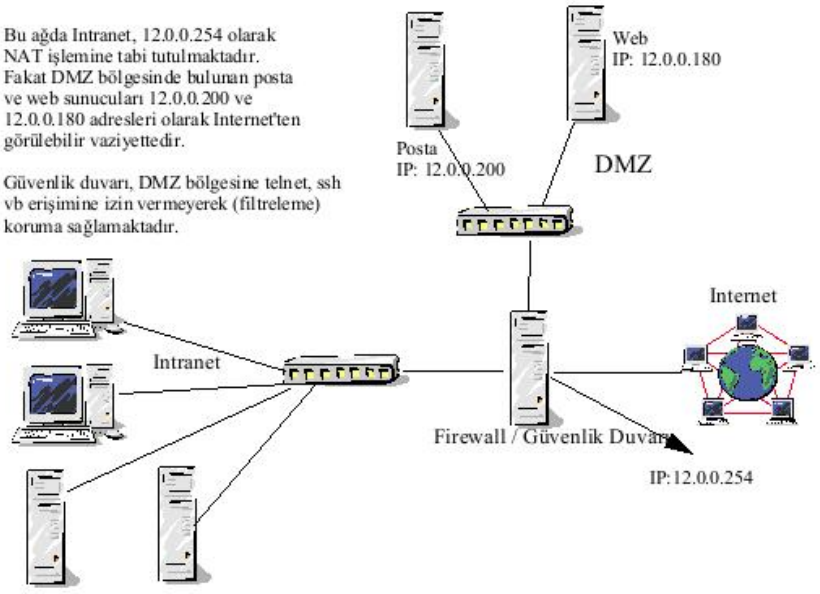
Ağınızda Internet'den erişimi olması gereken web, posta gibi sunucular bulunabilir. Bu sunuculara erişimi iki yoldan vermeniz mümkündür:

- Silahsızlandırılmış bölge uygulaması (DMZ – Demilitarized Zone)
- İç ağınızda bu servislere doğrudan filtreleme yaparak.

Silahsızlandırılmış bölge (DMZ – DeMilitarized Zone)

DMZ, güvenlik duvarı tarafından daha az korunan, daha fazla erişime izin verilen bir bölgedir. Güvenlik duvarına üçüncü bir ağ çıkışı eklenmesi ve Internet'e servis verecek olan makinaların buraya konulması ile oluşturulur. Örneğin DMZ'deki makinalara NAT uygulanmayabilir, tahsisli IP numaralarına sahip olabilirler. Güvenlik duvarı, telnet, ssh gibi kimi protokollerin buraya erişimini filtreleyerek DMZ bölgesindeki makinalara güvenlik sağlar. Dikkat edilecek nokta, DMZ'de bulunan makinaların daha fazla erişime (ve dolayısıyla saldırıya) açık olmasıdır. Buradaki makinalar dikkatli kurulmalı, güvenliğe aykırı protokoller vs. burada yer almamalıdır.

Şekil 3. Silahsızlandırılmış bölge (DMZ)



Doğrudan Filtreleme

DMZ oluşturmak için ek ekipman ve IP numarası gerekir. Güvenlik duvarında üçüncü bir ağ birimi, ayrı bir switch, daha fazla adette tahsisli IP numarası ve iç ağınızda başka herhangi bir görev görmeyecek olan sunucu makinalar gerekir. Eldeki imkanlar buna yetişmeyebilir. Böyle durumlarda, güvenlik duvarınızdaki filtreleme politikasını değiştirerek iç ağınızdaki kimi makinalara dışarıdan sınırlı erişim imkanı verebilirsiniz. Örneğin güvenlik duvarınız ağınızın genelinde dışarıdan gelen SMTP (posta) protokolünü filtrelerken, sadece posta sunucunuza dışarıdan SMTP protokolü erişimini verebilir. NAT ile birleştirileceğinden, bu dışarıdan bakıldığı zaman sanki güvenlik duvarınız posta sunuculuğu yapıyormuş izlenimini verir.

2.5. Vekil (Proxy)

Proxy'nin kelime anlamı vekil'dir. Yukarıdaki yöntemlerin hepsi, belli kurallara bağlı olarak Internet'deki bir makina ile iç ağdaki bir makina arasında doğrudan alışverişe izin verir. Vekil uygulamaları ise, bu doğrudan alışverişin arasına girer. Dolayısıyla protokol bazlı herhangi bir saldırı, vekil sunucuya yönelik gerçekleşir, iç ağdaki makinayı etkilemez. Örneğin bir http (web) vekili, iç ağdan dışarıya giden bütün web isteklerini toplar. Bu istekleri kendisi yapar, gelen sonuçları iç ağa dağıtır. Örneğin eğer web protokolü yolu ile istemci makinanın bazı bilgilerinin alınması veya bir saldırı yapılması söz konusu olur ise, bundan etkilenen sadece web vekili makina olur, iç ağda web erişiminde bulunan her makina değil.

Güvenlik amacı ile proxy kullanımı, uygulama temelli güvenlik duvarı (application level firewall) olarak adlandırılır.

2.5.1. Vekillerin Başka Kullanımları

- **Güvenlik amaçlı** – yukarıda bahsedilmiştir.
- **İzin amaçlı** – İç ağınızdan bazı servislere kimin erişebileceğini belirlemekte, izin politikası uygulamakta kullanılırlar.
- **Performans amaçlı** – Pek çok istemci aynı istekte bulunuyorsa, bunların bir defaya indirgenmesini sağlayarak hem sunucu makinanın üzerindeki yükü, hem de kullanılan bağlantı yükünü hafifletirler.

Vekil sunucular, en fazla kullanılan örneği olan *Web Vekili (Squid)* (sayfa: 15) üzerinde daha ayrıntılı olarak aşağıda anlatılmıştır.

3. Güvenlik Duvarı – Satın Almak, Kendiniz Yapmak?

Neyi, nasıl güvenlik altına aldığınızı bilmeden, pahalı bir ticari güvenlik duvarı satın almak size güvenlik sağlamaz. Dünyanın en pahalı ve gelişkin güvenlik duvarı, eğer çeşitli protokolleri açmış, fiziksel bağlantının tekliği kavramına uymamış, her tür erişime izin vermiş iseniz, size bir fayda sağlamaz. Elinizdeki Cisco Router'unuzu paket filtrelemek için programlamaktan tutun, ticari ve pahalı bir güvenlik duvarı satın almaya kadar uygulayacağınız her tür yöntem, neyi, ne için yaptığınızı biliyorsanız faydalıdır. Güvenlik duvarları, sizin ağ altyapınız ve sizin erişim ihtiyaçlarınız ile alakalıdır. Dolayısıyla ticari bir güvenlik duvarı satın almak niyetinde olsanız dahi, güvenlik duvarlarının *ne yaptığını* (sayfa: 3) öğrenmek ve erişim ihtiyaçlarınızı belirlemek zorundasınız.

Güvenlik duvarınızı kendiniz, Linux temelli bir makina üzerinde oluşturabilirsiniz veya kendiniz oluşturmak istemezseniz Linux temelli hazır bir *güvenlik duvarını* (sayfa: 8) uygulayabilirsiniz. Ticari olarak satılan güvenlik duvarlarının yapıp, doğru oluşturulmuş bir Linux sisteminin yapamadığı hiç bir şey yoktur.

3.1. Satın Almak

Linux Temelli Hazır Güvenlik Duvarları

Her ne kadar genel bir dağıtım (örneğin Red Hat) ile başlayıp kendiniz güvenlik duvarını oluşturabilseniz dahi bazı sebeplerden dolayı bunu yapmak istemeyebilirsiniz:

- Güvenlik duvarı olarak kullanacağınız makinayı doğru kurmanız gereklidir. Bunun üzerindeki gerekmeyen servisleri kaldırmanız, makinayı güvenli çalışabilecek şekilde kurmanız gereklidir. Bunları yapmakta kendinize güvenmiyorsanız, aşağıda bahsedilen hazır Linux güvenlik duvarlarından birini kurmak isteyebilirsiniz.
- Güvenlik duvarı bir kez kurulup ondan sonra hiç güncellenmeyecek bir sistem değildir. Ticari güvenlik duvarları da sürekli olarak yeni bulunan eksiklikleri kapatmak için güncellenirler. Genel bir dağıtım kullanarak bir güvenlik duvarı oluşturduktan sonra, sürekli olarak yeni çıkan güncellemeleri takip etmek zorundasınız. Eğer bu takibi yapmaya zaman ayıramayacaksanız, aşağıdaki hazır Linux güvenlik duvarlarından birini kullanın. Yalnızca bu paketlere gelen güncellemeleri takip eder ve genel bir dağıtıma yapılan güncellemelerin sizin açınızdan gerekli olup olmadığına karar vermek yükünden kurtulursunuz.
- Güvenlik duvarı üzerinde aşağıda bahsedilen yöntemleri doğru uygulamanız gereklidir. Eğer bunları doğru uygulayacak sistem bilgisine sahip değilseniz ve öğrenmek istemiyorsanız, hazır bir güvenlik duvarı sizin için en iyi yöntem olabilir.
- Güvenlik duvarını gün be gün yönetecek kişi ile güvenlik duvarını kuracak kişi aynı olmayabilir. Aşağıda bahsedilen Linux temelli güvenlik duvarları gayet profesyonel görüntülü, grafik arayüzlerine sahiptir. Bunları kullanmak ve yönetmek kendi oluşturacağınız bir makinayı kullanmak ve yönetmekten daha kolay olacaktır.
- Güvenlik duvarı için gereken bütün servisleri bir araya getirmek azımsanmayacak bir sistem entegrasyonudur. Bu işi yapmak yerine hazır, Linux temelli bir güvenlik duvarı kurmayı tercih edebilirsiniz.

Genel olarak, eğer Internet erişiminiz, kullanacağınız makina, uygulayacağınız protokoller Linux temelli hazır güvenlik duvarlarının birisi tarafından karşılanıyorsa, bunu kurmayı tercih edin. Eğer daha özel bazı istekleriniz varsa, kullandığınız donanım bu sistemler tarafından desteklenmiyorsa, o zaman aşağıda Red Hat dağıtımından yola çıkılarak *sıfırdan güvenlik duvarının nasıl oluşturulduğu* (sayfa: 8) anlatılmıştır.

Astaro[B13]

Astaro gerçekten profesyonel bir güvenlik duvarı. Internet'den indirebilirsiniz. Ticari kuruluşların lisans alması isteniyor. Güvenlik duvarı üzerinde, Internet'den gelen postalar virüs kontrolünden geçiriliyor ve alınan lisans ağırlıklı bununla ilgili. Modem bağlantısı desteklemiyor — fakat eğer sabit bir bağlantı ile Internet'e erişiyorsanız, bu sistemi kurun. Piyasadaki bütün ticari, kapalı güvenlik duvarları ile rahatlıkla yarışabilecek bir ürün. Lisanslandığı zaman otomatik olarak virüs dosyalarını ve kendisini güncelleyebiliyor. Sahiden güzel bir web tabanlı grafik arayüzü var.

Smoothwall[B14]

Daha ziyade ev kullanıcıları için tasarlanmış, modem bağlantısı da destekleyen bir güvenlik duvarı. Sevenleri çok, fakat 2.2 çekirdeği kullandığından dolayı dinamik filtreleme yapamıyor. Geliştiricileri pek geçimli değil ve son zamanlarda ürünlerini giderek daha fazla kapalı hale getirme çabası içerisindeler. Salt bu sebeplerden dolayı ben kullanmazdım.

3.2. Kendiniz Yapmak

Sıfırdan, Genel Bir Linux Dağıtımı (Red Hat) Kullanarak Güvenlik Duvarı Oluşturmak

3.2.1. Neden Kendi Güvenlik Duvarımızı Kurduk?

Bizim Firma A.Ş.'de henüz hızlı bir Internet bağlantısı bulunmamaktadır. Sabit hattın kurulması beklenirken Internet erişimini başlatmak istedik. Bunu yaparken elde olan Equinox çoklu modem kartını kullanmak ve karttaki modemlerden birisini Internet'e tahsis etmek istedik. Karttaki diğer modemler başka işler için kullanılacak. Bir modem ise gerektiği zaman çevir/bağlan yöntemi (dial–on–demand) ile Internet'e bağlantı sağlayacak.

Bir modem ile sürekli olmayan bir bağlantı üzerinden ne kadar dış saldırı gelebileceği tartışılabilir. Fakat bu uygulamayı gelecek olan sürekli bağlantı için bir ön çalışma olarak kullandık. Üstelik, bir modemin bağlantısını paylaştırabilmek için NAT yapmamız, bu kadar düşük hızlı bir bağlantıda biraz daha iyi performans sağlamak için web vekili (squid) kullanmamız ve ICQ vs. gibi iş için gerekli olmayan erişimleri engellememiz gerekmekte idi. Bu da zaten genel bir güvenlik duvarı oluşturmakta kullanılan bütün kavramları devreye soktu.

Özel bir modem kartı kullanmamız, bu karttaki başka modemler üzerinde farklı hizmetler vermek istememiz ve Internet bağlantımız modem üzerinden olduğu için Astaro[B15]'yu kullanamadık.

3.2.2. Kullanılan Donanım

- Pentium III 800 Mhz CPU, on–board (Trident Generic) grafik kartı.
- 128 MB RAM
- 16 GB IDE Hard Disk
- Intel Pro Dual 100 Mbit Ethernet kartı (çift portlu, ileride biri sabit bağlantı için kullanılacak)
- Equinox SST 8 Çoklu Modem Kartı (8 modemin birisi Internet bağlantısı için kullanılıyor)

3.2.3. İşletim Sistemi ve Kurulum

Makina üzerine Redhat 7.2 kurduk. Custom server (özel sunucu) seçeneğini kullandık. Disk bölümünü Disk Druid ile elle yaptık, 96 MB `/boot`, 1 GB takas alanı (ileride RAM'i arttırırsak ek iş çıkarmamak için gereğinden 4 kat fazla), geri kalanı da `/` olarak bölümledik. `/boot` ve `/` üzerinde ext3 günlüklü dosya sistemini kullandık.

Makina üzerindeki iki ethernet kartının birini devre dışı bıraktık. İleride sabit bağlantı olduğu zaman bu kart üzerinden dışarıya bağlanacağız. İçeriye bağladığımız ethernet kartına elle 10.254.254.254 IP numarasını verdik. Ağın genelinde bu IP numarası ağgeçidi (gateway) adresi olarak kullanılacak. DNS sunucu numarası 127.0.0.1 (makina üzerinde yerel bir DNS sunucusu çalıştırıyoruz, iç ağdaki DNS sunucusundan bağımsız).

Güvenlik duvarını kendimiz kuracağımızdan dolayı "no firewall" seçeneğini seçtik. Kurulumda yaptığımız paket seçimleri:

- Classic X
- X Windows
- KDE
- Network Support – Ağ desteği
- Dialup Support – Çevirmeli ağ desteği
- DNS Server – Alan adı sistemi sunucusu
- Web Server – Web sunucusu
- Messaging/Web Tools – İletişim/Web araçları
- Router/Firewall – Yönlendirici/Güvenlik duvarı
- Authoring/Publishing – Yazarlık/Yayıncılık
- Emacs
- Utilities – Yardımcı uygulamalar
- Software Development Yazılım geliştirme
- Kernel Development – Çekirdek geliştirme

Makinayı metin bazlı açma seçeneğini etkinleştirdik. Kurulum esnasında kurduğumuz paketler gerekenden bir hayli fazla. Bunların bir bölümünü daha sonra makinadan kaldırdık. Güvenlikli bir sistem kurmanın püf noktası makinada gerekmeyen hiç bir şeyin bulunmamasından geçmekte. Dolayısıyla daha sonra makinadan pek çok paketi kaldırdık.

3.2.4. Servislerin Kapatılması

Red Hat kurulduktan sonra ilk işimiz gerekmeyen servisleri kapatmak oldu. Teker teker bütün servisleri gözden geçirip gerekip gerekmediğini düşündük.

```
# cd /etc/rc.d/rc3.d
# ls *
# mv S09isdn K09isdn
# mv S28autofs K28autofs
# mv S80sendmail K80sendmail
# mv S13portmap K13portmap
# mv S14nfslock K14nfslock
# mv S60lpd K60lpd
# mv S56rawdevices K56rawdevices
# mv S25netfs K25netfs
# mv S55sshd K55sshd
#
```

Bundan sonra `/etc/xinetd.d` dizininde yer alan, **`xinetd`** vasıtası ile başlatılan servislere baktık (**`telnet`**, **`ping`** vs.). Bunların hepsinin kapalı olduğundan emin olduk.

3.2.5. Gereksiz Paketlerin Kaldırılması

Sistemden kaldırılan paketler biraz sizin tercihinize bağlı. Biz bütün paketlere bakarak gerekmediğini düşündüğümüz herşeyi kaldırdık. Genel teamüle aykırı olarak geliştirme (development) ile ilgili paketleri ve istediğimiz zaman grafik arayüz kullanmak için kullanılan paketleri sistemde bıraktık. Bunun ne kadar doğru olduğu tartışılabilir, fakat güvenlik duvarı makinası üzerinde yapılacak işlemleri bizim açımızdan kolaylaştırmakta.

3.2.6. Güncellemeleri Uyguladık

Red Hat[B16] web sitesinden sistemimizi ilgilendiren bütün güncellemeleri uyguladık. Genel prensip olarak, eğer sistemde kurulu ise ve güncellenmesi çıkmış ise, bunu indirip kurduk. Zaman zaman da bu güncellemeyi yapmaya devam edeceğiz. Bu makinanın güncellenmesi, iç ağda yer alan herhangi bir makinanın güncellenmesinden çok daha önemli. Saldırılar genelde bilinen ve yeni sürümlerde düzeltilmiş olan eksikleri kullanarak güncellenmemiş sistemlere yönelik oluyor.

3.2.7. Modem Kartımızı Sisteme Tanıttık

Kullandığımız modem kartının sürücülerini Internet'ten indirmek zorundayız. Equinox[B17] web sitesinden `eqnx-4.01-1.i386.rpm` paketini indirdik ve kurduk. Bu paket **`rpm -Uvh eqnx*`** komutu ile sürücüsünü derliyor ve sistem açılırken gerekli olan ayarlarını `/etc/rc.d/rc.local` dosyasına yazdırıyor. Bu işlemden sonra Internet için kullanacağımız modem `/dev/ttyQ1a1` aygıtı olarak sistem tarafından görüldü. Eğer seri port üzerinden haricî modem kullansa idik, aygıtımız `/dev/ttyS01` olarak görülecekti ve bu işlemi yapmak zorunda olmayacaktık.

3.2.8. PPP Ayarlarının Yapılması

Bağlantının Sağlanması

Grafik arayüzleri (**`kppp`** vs.) normal masaüstü kullanımı için bu işlemi çok kolay hale getirmiş durumda, fakat bizim yaptığımız gibi sunucu şeklinde otomatik aranacak bir sistemde kendi bağlanma betiklerimizi hazırlamak zorundayız. Bu işlem biraz deneme yanılma gerektiriyor ve bir ISP'de çalışan betik bir diğerinde çalışmayabiliyor.

Deneme yanılma yöntemi ile bulduğumuz, NetOne ve AttGlobal için çalışan arama betikleri (telefon yazan yere telefon numarasını, örneğin 08225551212, yazın):

call–netone

```
TIMEOUT         5
ABORT           '\nBUSY\r'
ABORT           '\nNO ANSWER\r'
ABORT           '\nRINGING\r\n\r\nRINGING\r'
''              \rAT
'OK-+++\c-OK'   ATH0
TIMEOUT         30
OK              ATZ0
OK              ATM0L0
OK              ATDTtelefon
'\r'            ''
CONNECT         ''
```

call–attglobal

```
TIMEOUT         5
ABORT           '\nBUSY\r'
```

```
ABORT            '\nNO ANSWER\r'
ABORT            '\nRINGING\r\n\r\nRINGING\r'
''               \rAT
'OK-+++\c-OK'    ATH0
TIMEOUT          40
OK               ATZ0
OK               ATM0L0
OK               ATDTtelefon
CONNECT
'\n'             ''
''               ''
```

Bu betikleri `/etc/ppp` dizinine kaydedin:

```
# cp call-netone /etc/ppp
# cp call-attglobal /etc/ppp
```

Başka ISP'ler için çalışan bağlanma betikleriniz varsa, lütfen bana gönderin. <deniz (at) arayan.com> .

Bu betiklerin dosya izinlerinin gerektiği kadar olduğundan emin olun:

```
# chmod 600 /etc/ppp/call*
```

ISP'deki parola ve kullanıcı isminizi `chap-secrets` ve `pap-secrets` dosyasına kaydedin.

`chap-secrets` dosyası:

```
# Secrets for authentication using CHAP
# client        server  secret                  IP addresses
kullanici-ismi  netone  parola
```

`pap-secrets` dosyası:

```
# Secrets for authentication using PAP
# client        server  secret                  IP addresses
kullanici-ismi netone       parola
kullanici-ismi attglobal    parola
```

AttGlobal Chap desteklemiyor, dolayısıyla onu yalnızca `pap-secrets` dosyasına koyduk. Bu dosyalarda `kullanici-ismi` yazan yere ISP'deki kullanıcı isminizi (örneğin: mehmet), `parola` yazan yere parolanızı (örneğin: c2fj80d90) yazın. Bu dosyaları `/etc/ppp` dizinine kaydedin (orada boş dosyalar olduğundan soru sorabilir):

```
# cp chap-secrets /etc/ppp
# cp pap-secrets /etc/ppp
```

Bu dosyaların dosya izinlerinin gerektiği kadar olduğundan emin olun:

```
# chmod 600 /etc/ppp/*secrets
```

PPPD Seçeneklerinin Yapılandırılması

PPPD çok amaçlı bir servis. Bütün seçeneklerini ayrıntılı anlatmamıza imkan yok. Önemli bir nokta, eğer `/etc/ppp/options.ttyXYZ` diye bir dosya bulur ise, o aygıt için o dosyayı otomatik uygulayacağı. Bizim aygıtımız `/dev/ttyQ1a1` olduğu için, `/etc/ppp/options.ttyQ1a1` dosyası aşağıda:

```
#Bu seçenek bağlantı olduğu zaman aradaki ayar paketlerinin sayısını
#belirliyor. Bazen öntanımlı 10 paket yetmeyebiliyor. 30 olarak kullandık.
lcp-max-configure 30
#Bağlantı başlayınca modemi kilitliyor
lock
```

```
#10 dakika bir iletişim olmadığı zaman bağlantıyı kapatıyor
idle 600
#Dışarıdaki bağlantı ppp'nin iki ucundaki dinamik IP adreslerini belirler
ipcp-accept-remote
ipcp-accept-local
#Biz parola sormuyoruz, onlar bize soruyor
noauth
#Hangi kullanıcı olarak bağlandığımız. secrets dosyalarındaki parolayı
#bulmak için kullanılıyor.
user kullanici-ismi
#Birden fazla hesap secrets dosyasında yer alıyor. Bunlardan hangisi
#kullanılacak.
remotename attglobal
#Modem hızımız
57600 crtscts
#İlk başlatıldığı zaman ppp'nin iki ucundaki IP numaralarını belirlemek
#zorundayız. Bağlantıdan sonra bunlar değişecek ve gerçek (ve o bağlantıya
#has) IP numaraları olacak. Ama şimdi bir şeyler vermek zorundayız.
139.92.80.128:152.158.100.30
#Bağlantı betiğimiz.
connect '/usr/sbin/chat -v -f /etc/ppp/call-attglobal'
#Dial-on-demand, yani dışarıya biri bir paket gönderdiği zaman
#bağlanacağız, hemen değil.
demand
#ppp bağlantısı bu makina için öntanımlı ağgeçidi olacak.
defaultroute
```

Bu dosyayı birebir değil, sizin modem aygıtınızın ismi ile kaydedeceksiniz. Yani modeminiz `/dev/ttyS01` ise,

```
# cp options.ttyQ1a1 /etc/ppp/options.ttyS01
```

Bu dosyanın dosya izinlerinin gerektiği kadar olduğundan emin olun:

```
# chmod 600 /etc/ppp/options*
```

PPP Servisinin Başlatılması

Bizim modemimiz ancak `/etc/rc.d/rc.local` dosyasında ilgili satırlar işlendikten sonra sistem tarafından görülebilir. Dolayısıyla biz **pppd** başlatma komutumuzu `rc.local`'e koyduk. Bu bütün sistem ayağa kalktıktan sonra en son işlendiği için sizin için de çalışacaktır. `/etc/rc.d/rc.local` dosyasına

```
/usr/sbin/pppd ttyQ1a1
```

ilave ettik. Sizin modeminiz `/dev/ttyS01` ise, **/usr/sbin/pppd ttyS01** yazacaksınız. Bundan sonra makinanız her yeniden başlatıldığında, "dial–on–demand" yntemi ile Internet'e bağlanmaya hazırdır. Henüz bu Internet bağlantısını başkalarına paylaştırmak konusunda bir bilgisi yok, fakat kendisi Internet'e otomatik olarak ihtiyaç gördükçe bağlanır, ve belli bir süre trafik olmazsa bağlantıyı kapatır.

3.2.9. Güvenlik Duvarını Oluşturma Yazılımı: Fwbuilder

Bundan sonra ilgili NAT ve filtreleme kurallarını oluşturmamız gerekiyor. Bu işlemi elle de yapabilirdik. Fakat Internet'ten indirebileceğiniz **fwbuilder** yazılımı bu işlemi bir hayli kolaylaştırıyor.

Ön Gereklilikler

fwbuilder'ı derleyebilmemiz için bize `libxml2-devel`, `libxslt-devel`, `libsigc++` ve `libsigc++-devel` paketleri gerekti. Bunları Redhat CD'sinden yükledik. Ayrıca, Internet'ten `Gtkmm`[B19] paketini indirdik. Bu paketi

```
# rpm -tb gtkmm-1.2.8.tar.gz
```

komutu ile derledik ve oluşan RPM'leri sisteme kurduk:

```
# rpm -Uvh /usr/src/redhat/RPMS/i386/gtkmm*
```

Paketlerin derlenmesi

Fwbuilder web sitesinden[B20] `fwbuilder-1.0.0-1rh72.src.rpm` ve `libfwbuilder-0.10.4-1rh72.src.rpm` paketleri indirildi.

```
# rpm --rebuild libfwbuilder*
```

komutu ile `libfwbuilder` paketi derlendi,

```
# rpm -Uvh /usr/src/redhat/i386/libfwbuilder*
```

komutu ile sisteme kuruldu. Daha sonra

```
# rpm --rebuild fwbuilder*
```

komutu ile `fwbuilder` derlendi,

```
# rpm -Uvh /usr/src/redhat/RPMS/i386/fwbuilder*
```

komutu ile sisteme kuruldu.

NAT ve Filtreleme Politikalarının Yazılması

`fwbuilder` komutu ile yazılım açıldı. `Iptables` seçeneği seçildi. Yerel ağ (intranet) ve güvenlik duvarı makinası (perde adında) tanımlandı. "Help me build firewall policy" seçeneğinin yardımı ile ilk temel kurallar oluşturuldu, üzerine aşağıdaki kurallar ilave edildi.

Ağ Birimlerine Ait Politikalar – ppp0

Kaynak	Hedef	Servis	İşlem	Yön	Açıklama
Herkes	Herkes	ip_fragments	Paketi yok et (deny)	içeri gelen (inbound)	Modem kartına gelen (içeriden veya dışarıdan) her tür IP paket parçacığı reddedildi.
intranet, perde	Herkes	Bütün servisler	Paketi yok et (deny)	içeri gelen (inbound)	Modem kartına dışarıdan içeriye gelip de kaynağını iç ağ olarak gösteren bir paket bir ip taklidi (ip spoofing) saldırısı olabilir. Reddedildi.
XIntranet, Xperde	Herkes	Bütün servisler	Paketi yok et (deny)	dışarı çıkan (outbound)	Modem kartına içeriden gelip de kaynağını dışarısı olarak gösteren bir paket olmaması gerekir, fakat başka bir sorunun işareti olabilir. Reddedildi.

Ağ Birimlerine Ait Politikalar – lo0

Kaynak	Hedef	Servis	İşlem	Yön	Açıklama
Herkes	Perde	Hepsi	Kabul et	İçeri gelen (inbound)	Loopback, makinanın kendi içinde çalışması gerekli olan bir arabirim. Herşey buna açık.
Perde	Herkes	Hepsi	Kabul et	Dışarı çıkan (outbound)	Loopback, makinanın kendi içinde çalışması gerekli olan bir arabirim. Herşey buna açık.

NAT Politikaları

İlk Kaynak	İlk Hedef	İlk Servis	Çevrilen Kaynak	Çevrilen Hedef	Çevrilen Servis	Açıklama
Intranet	Herşey	http	Değişmedi	Perde	squid	Bir web vekili (squid) kullanıyoruz. Dolayısıyla iç ağdan web (http) kullanarak dışarı çıkmak isteyen bütün paketleri güvenlik duvarı üzerinde squid portuna gönderiyoruz. Bunu kullanabilmek için güvenlik duvarı üzerinde squid vekil sunucusu çalıştıracağız.
Intranet	Herşey	Herşey	Perde	Değişmedi	Değişmedi	Intranet'ten dışarıya çıkmak isteyen herşeyi sanki paket güvenlik duvarından kaynaklanmış gibi yeniden yazıyoruz.

Genel Politikalar

Kaynak	Hedef	Servis	İşlem	Açıklama
Herkes	Herkes	ip parçacıkları	Paketi yok et (deny)	Bütün olmayan IP paketlerini yok ediyoruz
Herkes	Intranet, perde	Faydalı ICMP	Kabul et	Bu icmp servisleri bazı servislerin doğru çalışması için faydalı.
Perde	Intranet	Zaman aşımı	Kabul et	Traceroute için bu gerekiyor
Intranet	Perde	Herşey	Kabul et	İçeriden güvenlik duvarına erişim var
XIntranet	Perde	ssh, telnet, http	Paketi yok et (deny)	İçeriden gelmeyen ssh, telnet, http isteklerine cevap vermiyoruz. Zaten güvenlik duvarı üzerinde ssh ve telnet çalıştırmıyoruz ve aşağıdaki genel kural bunları da engellerdi. Fakat gene de emin olalım dedik.
perde	Herşey	herşey	Kabul et	Güvenlik duvarı herkese erişebilir
Intranet	Herşey	http, https, dns_tcp, dns, ntp, traceroute, bütün icmp, telnet, imap, imaps, pop3, smtp, smtps, ssh, ftp, ftp data	Kabul et	İç ağdan bu servislerle dışarıya erişime izin var
Intranet	Herşey	Herşey	Reddet	Yukarıda kabul edilmeyen servisleri iç ağ için hemen reddediyoruz (bekleme olmuyor)
Herşey	Herşey	Herşey	Paketi yok et (deny)	Yukarıdaki kuralların dışında kalan bütün durumları reddediyoruz.

fwbuilder, verilerini bir XML dosyası halinde saklıyor. `/usr/local/firewall` adında bir dizin oluşturduk ve burada `perde.xml` adında bir dosyada tanım dosyamızı tutuyoruz. Bizim kullandığımız tanım dosyasını proxy–fw–files.tar.bz2[B21] paketinde bulabilir ve bunu değiştirerek kendi kurallarınıza uygun hale getirebilirsiniz. Kural kümemizi oluşturduktan sonra derleme (compile) seçeneği ile kuralları derliyoruz. Derlenen kurallar `/usr/local/firewall/Perde.fw` adında bir dosyaya konuyor.

NAT ve Filtrelemenin Devreye Alınması

Filtreleme kurallarımız `/usr/local/firewall/Perde.fw` dosyası altında oluştu. Şimdi bunu devreye almamız gerekli. Red Hat'in `iptables` betiğini bu iş için değiştirerek[B22] kullandık. Burada dikkat edilecek nokta, filtreleme işinin modem devreye alınıp **pppd** çalıştırıldıktan sonra yapılması gerektiği. Dolayısıyla Red Hat'in öntanımlı **iptables** çalıştırma sırası da değişmek zorunda. Bu betiği `rc.local`'dan çalıştıracağız.

Önce bir hata olmaması için `iptables` betiğini normal yerinden sildik:

```
# rm -f /etc/rc.d/rc3.d/*iptables
```

Değiştirilmiş `iptables` betiğini[B23] `/etc/rc.d/init.d` altına kopyaladık. Dosya izinlerinin doğru olduğundan emin olduk:

```
# chmod 755 /etc/rc.d/init.d/iptables
```

`/etc/rc.d/rc.local` betiğinde, **pppd** satırından sonra:

```
# /etc/rc.d/init.d/iptables start
```

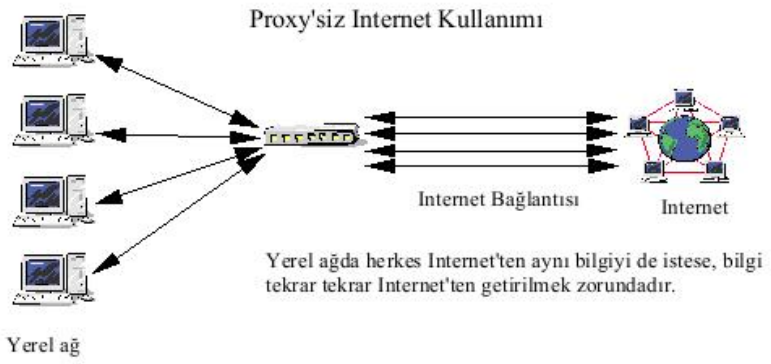
satırını ekledik. Bundan sonra makina yeniden başlatıldığı zaman otomatik olarak NAT ve filtreleme başlayacak. Yerel ağdaki makinalara ağgeçidi adresi olarak güvenlik duvarımızın IP adresi olan 10.254.254.254'ü verip, iç ağdan herhangi bir şekilde Internet'e ulaşmak istediğimiz zaman güvenlik duvarımız Internet'e bağlanacak ve oluşturduğumuz kurallar dahilinde erişimi sağlayacak.

4. Web Vekili (Squid)

Eğer bir vekil kullanmayacak olsak, güvenlik duvarımız kullanmaya hazır idi. Fakat vekil sunucular, özellikle http için performans kazandırıcı bir unsurdur.

Özellikle Web (Internet'te sörf) kullanımında, her bir sayfa, her bir tarayıcı istemi için tekrar tekrar ana sunucudan getirilmektedir.

Şekil 4. Vekilsiz Internet kullanımı



Ağ üzerinde yüz istemcinin Hürriyet gazetesini okumak istemesi, aynı sayfanın tam 100 defa ağa getirilmesi demektir. Araya vekil konduğu zaman ise, ilgili sayfa yalnızca bir defa getirilir ve daha sonra isteyen bütün istemcilere vekilin kendi deposundan sunulur. Dolayısıyla Internet trafiği azaltıldığı gibi, ilk istemden sonraki bütün istemcilere yerel ağdan sunum yapıldığı için kullanıcılar sayfaya çok daha hızlı (yerel ağ hızlarında) erişirler. Ülkemizde Internet bağlantılarının pahalılığı ve yavaşlığı göze alınırsa, şirketlerin hiç bir izin mekanizması uygulamak niyetleri olmasa dahi, salt performans arttırmak için vekil uygulamaları tavsiye edilir.

Şekil 5. Vekille Internet kullanımı – Performans

Vekilde izin mekanizması uygulamak ve dolayısı ile çeşitli hizmetleri herkese yasaklamak, çeşitli kullanıcılara bazı hizmetleri yasaklamak gibi izinlendirme politikaları uygulamak mümkündür.

Şekil 6. Vekille Internet kullanımı – İzinlendirme

Burada Linux üzerinde squid[B24] ve squidGuard[B25] vasıtası ile:

- Performans kazandırıcı [depolama (caching)]
- Şeffaf (transparent) — kullanmak için kullanıcı tarafında bir ayar gerektirmeyen
- Web sitesi bazında izin mekanizmalı

bir web vekil uygulaması konu alınmıştır.

4.1. Squid Kurulumu

Redhat 7.2 güncelleme paketi `squid-2.4.STABLE1-6.i386.rpm` indirilmiş ve

```
# rpm -Uvh squid-2.4.STABLE1-6.i386.rpm
```

komutu ile kurulmuştur. Sistem açılışında **`squid`** servisinin başlatılması için

```
# mv /etc/rc.d/rc3.d/K25squid /etc/rc.d/rc3.d/S25squid
```

komutu uygulanmıştır.

4.2. Ayarlar

- Squid'in Internet web deposu (cache) için 1 GB yer ayrılmıştır.

- İzinlendirme **squidGuard** paketi ile yapılmaktadır.
- Yasaklanan siteler: Kimi sitelere erişim engellenmektedir. Bu siteler **squidGuard**'un yayınladığı Internet karalistesi kullanılarak belirlenmektedir.

Ayar dosyası

`/etc/squid/squid.conf` dosyası bir hayli uzun olduğundan yalnızca değiştirilen ayarlar buraya alınmıştır. Dosyanın tamamını proxy–fw–files.tar.bz2[B26] paketinde bulabilirsiniz.

```
#VEKIL
#Öntanımlı portun yanısıra 8080 üzerinden de proxy servisi veriyoruz
http_port 3128 8080

#VEKIL
#Başka squid'lerle bilgi paylaşmıyoruz
icp_port 0

#VEKIL
#Yerel ağımızı burada tanımlıyoruz
acl intranet src 10.0.0.0/255.0.0.0

#VEKIL
#erişim engellemesi squidGuard tarafından yapılacağı için
#burada yer almıyor
http_access allow intranet

#VEKIL
#Eger proxy yolu ile ftp yaparsak, anonymous isteklerde bu adresi
#verecek
ftp_user Squid@bizimfirma.com.tr

#VEKIL
#1 Gig'lik (1000) büyüklüğünde bir cache dizini kullanıyoruz.
#Ayrıca diske bloksuz yazma yöntemini kullanıyoruz
cache_dir aufs /var/spool/squid 1000 16 256

#VEKIL
#squidGuard kullanıyoruz
redirect_program /usr/local/squidGuard/bin/squidGuard

#VEKIL
#20 adet squidGuard başlatıyoruz. Çok fazla sayıda olursa kaynak israfı
#çok az sayıda olursa beklemeye yol açabilir.
redirect_children 20


#VEKIL
#ICP sorgularına izin vermiyoruz
icp_access deny all


#VEKIL
#Hata iletilerimizi Türkçe verelim
error_directory /usr/lib/squid/errors/Turkish

#VEKIL
#Squid'i şeffaf (yani kullanıcıların herhangi bir ayar yapmasına gerek
```

```
#kalmaksızın şeffaf olarak kullanmak için bu ayarlara ihtiyacımız var.
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
httpd_accel_single_host off
```

4.3. Squid Kurulumunda ikinci adım: SquidGuard

Neden SquidGuard?

SquidGuard[B27], **squid** ile birlikte çalışan ve erişim izinlerini belirleyen bir yazılımdır. **squid**'in kendisi de erişim izinlerini belirleyebildiğine göre **SquidGuard**'a ihtiyaç olmadığı düşünülebilir. Fakat, **SquidGuard** kullanımının bazı avantajları vardır:

- Erişim izinlerini `squid.conf`[B28]'un içine yazmak yerine, diskte ayrı dosyalar halinde tutabilirsiniz. Dolayısıyla bir ek izin/kısıtlama ilave ederken yanlışlıkla **squid** ayarlarını bozma ihtimaliniz ortadan kalkar.
- **SquidGuard**, Internet'te bir arama yazılımı yardımı ile çeşitli kriterlerde karalisteler hazırlamakta ve bunları düzenli olarak güncellemektedir. **SquidGuard** kullanmazsanız, buna alternatifiniz trafiğinizi kontrol etmek ve kullanıcıların gittiği siteleri gözden geçirip kendi karalistenizi oluşturmaktır. Bunun yerine **SquidGuard**'un karalistesini kullanabilirsiniz.

SquidGuard Kurulumu

SquidGuard[B29] web sitesinden `squidguard-1.2.0.tar.gz` ve `blacklists.tar.gz` paketleri indirildi ve `/usr/local/src` dizinine kaydedildi.

```
# tar -xzvf squidguard-1.2.0.tar.gz
#
```

komutu ile paket açıldı.

```
# cd /usr/local/src/squidGuard-1.2.0
#
```

Paketin ayarları için `configure.squidGuard` betiği:

```
./configure --prefix=/usr/local/squidGuard \
            --with-db-lib=/usr/lib \
            --with-db-inc=/usr/include/db2 \
            --with-sg-config=/usr/local/squidGuard/configs/filter.conf \
            --with-sg-logdir=/usr/local/squidGuard/logs \
            --with-sg-dbhome=/usr/local/squidGuard/db
```

aynı dizine kaydedildi:

```
# cp configure.squidGuard /usr/local/src/squidGuard-1.2.0
# cd /usr/local/src/squidGuard-1.2.0
# chmod 755 configure.squidGuard
# ./configure.squidGuard
# make
# make install
#
```

komutları ile paket sisteme kuruldu.

Karalistenin Oluşturulması

İndirilmiş olan `blacklists.tar.gz` paketi `/usr/local/squidGuard/db/blacklists` altına açıldı. `/usr/local/squidGuard/configs/filter.conf`[B30] dosyasında erişim izinleri belirlendi. Esas olarak kara listedeki tüm adresler engellendi. Karaliste **SquidGuard** tarafından haftada 3 kez güncelleniyor. Biz de, zaman zaman bu güncellemeyi yapacağız.

Hata İletisi

SquidGuard'ın bir kötü tarafı kendi içerisinde bir hata iletisi oluşturmayıp, bir web sunucusuna gereksinim duyması. Sırf bu sebeple güvenlik duvarı üzerinde bir **apache** web sunucusu kurmak zorunda kaldık. Aslında bunun için içeride herhangi bir web sunucusu da kullanılabilir, fakat mümkün olduğunca güvenlik duvarını kendi başına çalışabilecek halde kurmak istedik.

SquidGuard paketinin içinden çıkan `squidGuard.cgi` betiğine Türkçe dilini ilave ettik ve dışarıdan bir siteye verdiği gif bağını kaldırıp yerel bir bağ haline getirdik. Betik öntanımlı olarak tarayıcıdaki dil seçeneğine göre dil seçimini yapmakta idi, fakat biz salt Türkçe olmasını istedik. Değiştirilmiş `squidGuard.cgi` buradan temin edilebilir[B31]. Bu betiği `cgi-bin` dizinine kaydettik:

```
# cp squidGuard.cgi /var/www/cgi-bin
#
```

Erişim izinlerinin doğru olduğundan emin olduk:

```
# cd /var/www/cgi-bin
# chown nobody.nobody squidGuard.cgi
# chmod 755 squidGuard.cgi
#
```

Yasak işareti veren `forbidden.gif`[B32] dosyasını ilgili dizine kaydettik:

```
# cp forbidden.gif /var/www/html
#
```

En sonunda web sunucusunun sistem açıldığı zaman açılmasını sağladık:

```
# mv /etc/rc.d/rc3.d/*httpd /etc/rc.d/rc3.d/S15httpd
#
```

5. DNS Sunucusu

Sistem kurulurken bir DNS sunucusu kurmuştuk. Bunun sistem açıldığı zaman başlatılmasını sağladık:

```
# mv /etc/rc.d/rc3.d/*named /etc/rc.d/rc3.d/S45named
#
```

6. Sistemi Yeniden Başlattık

İlgili servisleri kapatıp açmak ve sistemi yeniden başlatmamak mümkün, ama en kolayı güvenlik duvarını yeniden başlatmak. İç ağınızda ağgeçidi adreslerini 10.254.254.254 olarak değiştirdiğiniz zaman Internet erişiminiz ve güvenlik duvarınız hazırdır.

7. Sonuç

Bir kaç saatlik bir çalışma ile ufak bir makina üzerine bir güvenlik duvarı kurabilirsiniz. Güvenlik duvarları salt dış saldırılara karşı sisteminizi korumakla kalmaz, performans arttırıcı ve izin politikası uygulayıcı amaçlar için de kullanılırlar.

Notlar

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

(1) Tabya, sözlük anlamı olarak, "ayrı olarak yapılmış ve silahlarla güçlendirilmiş istihkam"dır. Güvenlik duvarının kurulacağı makina için bu terimin uygun olacağını düşündüm. Bu tabya şüphesiz, caydırıcı silahlarla teçhiz edilmiş olacak :–)

(B7) ../howto/iptables-«usage.pdf

(B8) ../howto/iptables-«usage.pdf

(B13) http://www.astaro.com/

(B14) http://www.smoothwall.org/

(B15) http://www.astaro.com/

(B16) http://www.redhat.com/

(B17) http://www.equinox.com/

(B19) http://gtkmm.sourceforge.net/

(B20) http://www.fwbuilder.org/

(B21) ../indirin/proxy-«fw-«files.tar.bz2

(B22) ../indirin/proxy-«fw-«files.tar.bz2

(B23) ../indirin/proxy-«fw-«files.tar.bz2

(B24) http://www.squid-«cache.org/

(B25) http://www.squidguard.org/

(B26) ../indirin/proxy-«fw-«files.tar.bz2

(B27) http://www.squidguard.org/

(B28) ../indirin/proxy-«fw-«files.tar.bz2

(B29) http://www.squidguard.org/

(B30) ../indirin/proxy-«fw-«files.tar.bz2

(B31) ../indirin/proxy-«fw-«files.tar.bz2

(B32) ../indirin/proxy-«fw-«files.tar.bz2

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

1 Şubat 2007