Kampüs Ağlarında Köprü-Güvenlik Duvarı (Bridge Firewall) ve Transparent Proxy

Volkan Sönmez
Süleyman Demirel Üniversitesi
Bilgi İşlem Daire Başkanlığı

1. Giriş

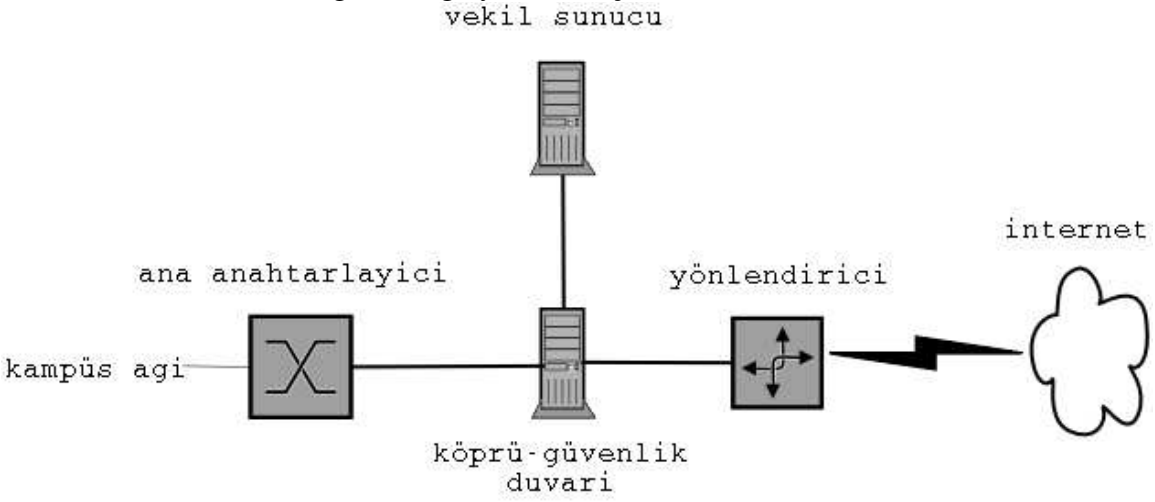
Bu bildiride kampüs ağlarında transparent proxy uygulamasının ve köprü-güvenlik duvarı uygulamasının linux sunucu kullanarak nasıl yapılacağı anlatılacak. Bilindiği gibi web-caching tekniği internet kullanımında %50 ye varan hız artışı kazandırıyor. Fakat kampüs ağları gibi orta ölçekli ağlarda bu uygulamanın gerçekleştirilmesinde çeşitli güçlükler karşımıza çıkmakta. Kampüs içerisinde internete bağlı her bilgisayarda vekil sunucu ayarı yapmak zaman alıcı, kontrolü zor bir iştir. Binin üzerinde bilgisayarda vekil sunucu ayarlarının yapılması zaman kaybına yol açacaktır. Ve insanları vekil sunucu kullanmaya ikna etmek gerekecektir.

Bir kampüs ağında transparent proxy çalıştırmak için,
1. Layer 4 bir anahtarlayıcı kullanarak 80 portuna yapılan web istemleri proxy sunucusuna yönlendirilebilir. Layer 4 anahtarlayıcıların maliyeti yüksektir. Ve ne kadar yük kaldıracakları da tartışma konusudur.
2. Tüm internet trafiği vekil sunucu üzerinden geçirilip, bu sunucuya yapılan web istemlerinin web-cache sunucuya yönlendirilmesi sağlanılabilir. Bu işlem vekil sunucunun çok yüklenmesini sağlayacaktır. Genel ağ trafiği de kötü yönde etkilenecektir. Bununla beraber iç ağdaki cihazlar özel ip blokları kullanılarak tekrar ayarlanmalı ya da subnetting yapılmalıdır ki bu şekilde bir miktar ip numarası boş yere kaybolacaktır.
3. Yönlendiricinin web istemlerini proxy sunucuya yönlendirmesi sağlanılabilinir. Bu işlem yönlendiriciye ek yük getirecektir. Ve bazı yönlendiriciler bu özelliğe sahip değildirler.
4. Bir linux sunucu kullanarak bu işlemi gerçekleştirilebilinir. Linux sunucu köprü-güvenlik duvarı (Bridge-Firewall) olarak ayarlanıp tüm trafiğin bu sunucu üzerinden geçmesi sağlanır. Vekil sunucu linux köprü sunucuya ek bir ağ kartı ile bağlanır. Web istemleri web-cache sunucusuna yönlendirilir.
Bu yöntem layer 4 bir anahtarlayıcı almaya göre daha ucuz bir çözüm sunmaktadır. Bununla beraber köprü-güvenlik duvarı sunucusu üzerinde bant genişliği kısıtlaması yapılabilmekte, istenildiği takdirde sakıncalı portlar kapatılabilmekte snort gibi uygulamalar yardımı ile iç ağa yapılan olası saldırılar tespit edilip bloklanabilmektedir. Gelecek bölümlerde linux kullanarak bu işlemin nasıl gerçekleştirilebilineceği anlatılacak.

2. Teori

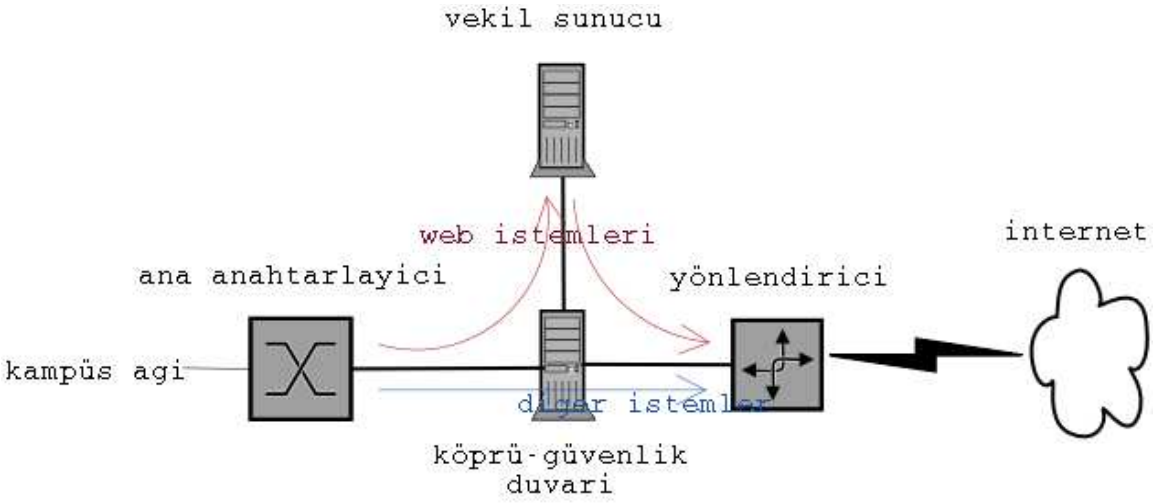
Transparent proxy işlemi için iki tane sunucuya ihtiyacımız olacak. Bir sunucunun

üzerinde web-cache uygulaması çalışacak. Diğer sunucu köprü-güvenlik duvarı işlevini yerine getirecek. Bu sunucuda üç adet ağ kartı takılı olcak ve sunucunun bir bacağı yönlendiriciye bağlı olacak diğer bacağı da kampüs ağına bağlı olacak. Üçüncü bacağı ise web-cache sunucusuna bağlantı sağlayacaktır. (Şekil 2.1)



Şekil 2.1. Genel yapı.

Köprü-güvenlik duvarı (KGD) sunucusunun bir ip numarasına sahip olması gerekmemektedir. Çünki anahtarlayıcı görevi görecektir. Kampüs ağından yapılan web istemleri KGD sunucusu üzerinden geçecek ve web istemleri vekil sunucuya yönlendirilecektir. Diğer istemler (e-posta, DNS, ftp vb.) normal akışına devam edecek, yönlendirici üzerinden internete çıkacaktır. (Şekil 2.2)



Şekil 2.2. Trafik akışı.

3. Kurulum

Web istemlerini KGD sunucusundan vekil sunucuya yönlendirmek için linux

çekirdeğinde bridge-netfilter desteği olması gerekmemektedir. Birçok dağıtımla gelen çekirdeklerde bu destek bulunmamaktadır. Bu yüzden gerekli çekirdek yamaları nı ve çekirdek kaynak kodunu indirip gerekli yamaları yapmamız gerekecektir. bridge-netfilter yaması http://bridge.sf.net adresinden indirilebilinir.Bu belgenin yazıldığı sıralarda en son çekirdek yaması bridge-nf-0.0.7-against-2.4.19.diff idi. Bununla beraber *bridge-utils* ve *iptables* paketlerine ihtiyacımız olacak. Bu iki paket birçok dağıtımla beraber gelmektedir. *Bridge-utils* paketi http://bridge.sf.net adresinden elde edilebilir.

3.1 Çekirdek Kurulumu

Bu kurulumda 2.4.19 çekirdeğini kullanılacağız. Çekirdek kayna kodu /usr/src dizinine açıldıktan sonra

patch -p1 < bridge-nf-0.0.7-against-2.4.19.diff

komutu verilerek çekirdek yaması yapılır. Daha sonra çekidek konfigurasyonunu yapacağız. Burada önemli olan seçenekler;

802.1d Ethernet bridging modülü ve ona bağlı olarak
- "**netfilter (firewalling) support**" modülü. Bununla beraber,
- "**IP: Netfilter configuration"** menüsünden gerekli iptables modullerini işaretlememiz gerekecektir.

Ve çekideğimizi derleyip yeni çekirdekle sunucumuzu başlatacağız.

3.2 Diğer Paketler

Eğer sistemimizde kurulu değilse iptables ve bridge-utils paketlerini de kurmamız gerekecektir. Bunu için;

bridge-utils paketini /usr/src dizinine açıyoruz *tar -zxf bridge-utils-x.y.z.tar.gz*
derleyecegiz : *make*
ve kuracağız *cp brctl/brctl /usr/sbin*

iptables paketini /usr/src dizinine açıyoruz *bzip2 -d iptables-x.y.z.tar.bz2; tar -xf iptables-x.y.z.tar*
derleyecegiz : *make KERNEL_DIR=/usr/src/linux; make install KERNEL_DIR=/usr/src/linux*
ve kuracağız *cp iptables /sbin* , *cp iptables-restore /sbin* , *cp iptables-save /sbin*

KGD sunucusunun kurulumu bu kadar. Vekil sunucumuza squid paketini kuracağız.

Squid paketini www.squid-cache.org adresinden indiriyoruz. Ve paketi /usr/src dizinine açıyoruz. Sonra squid dizininde sırayla

./configure –enable-linux-netfilter' , make ve make install komutları nı çalı ştı rı p squid paketini kuruyoruz.

4. Ayarlar

4.1KGD Sunucusu

Örnegimizde KGD sunucusunda eth0 arayüzü kampüs ağı na bakı yor. Eth1 vekil sunucuya ve eth2 internet ağı na bakı yor. KGD sunucusuna 192.168.1.254 ip numarası nı vereceğiz. Vekil sunucunun ip numarası ise 192.168.1.253 olacak.
KGD sunucumuzda köprü servisinin aktif olması için sı rayla şu komutları yazı yoruz;

```
brctl addbr br0
brctl addif br0 eth0
brctl addif br0 eth1
brctl addif br0 eth2
ifconfig eth0 0.0.0.0 up
ifconfig eth1 0.0.0.0 up
ifconfig eth2 0.0.0.0 up
```

KGD sunucumuzun aktif olması için bu komutları yazdı ktan sonra biraz zaman geçmesi gerekiyor. Sunucuya uzaktan erişmek için ip numarası verbiliyoruz. (Not : ip numarası olmadan da KGD sunucumuz çalı şabilir. Ip numarası nı sunucuya uzaktan erişim için veriyoruz).

```
ifconfig br0 192.168.1.254 up
```

ve son adı m. Web istemlerini vekil sunucuya yönlendireceğiz,

```
iptables -A PREROUTING -i <kampüs ağı na bakan ağ arayüzü> -p tcp -m tcp --dport 80
-j DNAT --to-destination <vekil sunucu ip numarası >:3128
```

ya da örneğimizde olduğu gibi,

```
iptables -A PREROUTING -i eth0 -p tcp -m tcp --dport 80 -j DNAT --to-destination
192.168.1.253:3128
```

4.2Vekil Sunucu Ayarları

Vekil sunucuda squid servisinin ayarları nı yapacağı z. Squid servisini transparent proxy olarak ayarlayacağı z. Bunun için squid.conf dosyası nda birkaç değişiklik yapmak gerekecek. Dosyada,

```
httpd_accel_host virtual
```

```
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
```

satı rları nı bulup önlerindeki “#” işaretlerini kaldı racağı z. Ve squid servisini tekrar başlatacağı z. Vekil sunucumuz transparent olarak ayarlanmı ş oluyor. Artı k web istemleri vekil sunucu üzerinden gidiyor.

Bu ayarlara ek olarak KGD sunucusu üzerinde istediğiniz portları kapatı labilirsiniz. 'tc' aracı nı kullanarak bant genişliği kı sı tlaması yapabilirsiniz. Linux birçok avantajları ndan yararlanabilirsiniz.