



T.C.

GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

GÜVENLİK DUVARI TASARIMI

Servet Haydar KARABAŞ
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI

98296

TEZ DANIŞMANI
Yrd.Doç.Dr. İbrahim SOĞUKPINAR

GEBZE

2000

T.C. GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ
DOKÜMANİSTLİK BÖLÜMÜ



Servet Haydar KARABAŞ'ın tez çalışması, G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 18.07.2000 tarih ve 2000/1.1.6 sayılı kararıyla oluşturulan jüri tarafından **Bilgisayar Mühendisliği** Anabilim Dalı YÜKSEK LİSANS tezi olarak kabul edilmiştir.

JÜRİ

ÜYE : İbrahim SOĞUKPINAR, Yrd.Doç.Dr., GYTE
.....
(Tez Danışmanı)

ÜYE : E.Murat ESİN, Yrd.Doç.Dr., GYTE

ÜYE : Ertuğrul KARAÇUHA, Yrd.Doç.Dr., GYTE

ONAY

G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 14.08.2000 tarih ve 2000/1.7 sayılı kararı.

İmza/Mühür

ÖZET

Kısa bir zaman sürecinde elektronik ve bilgisayar teknolojilerinde baş döndürücü gelişmeler yaşandı. Bunun neticesinde bilgisayarlar ve bilgisayar ağları alanına yatırımlar arttı. İhtiyaçların karşılanması için hızla yeni projeler üretildi. Üretilen ağ çözümleri bir çok işleri kolaylaştırdı. Devlet politikası çapında da desteklenecek boyutlara erişmiş birçok ağ projesi oluşturuldu. Bu projelerden en önemlisi internettir.

İnternetin hızla gelişmesi ve yatırımcıların internet üzerinden çözüm sunmaları, kullanımını arttırmıştır. Günlük hayattaki işlemlerin internet üzerinden yapılabilir olması çalışmaları buraya odaklamıştır. Zamanla yapılan işlemlerin çokta güvenli olmadığı gerçeği ile karşılaşmıştır. Bazı insanların verilen hizmetleri aksatmak ya da bir kazanç sağlamak amacıyla saldırılarda bulunduğu haberleri duyulmaya başlanmıştır. Bundan sonra verilen hizmetlerin ve sistemlerin güvenliği üzerinde çalışmalar artmıştır. Sonuçta savunma teknolojileri geliştirilmiş ve iyi bir alt yapı oluşturulmuştur.

Çalışmalar sonucundaki ağ teknolojilerinin hayatı kolaylaştırdığı gerçektir. Alt yapının iyi tartışılıp oluşturulmadan hızlı çözümlerin üretilmiş olması, bir takım sorunları oluşturmuştur. Bir çok sorun sayılabilir, en önemlisi güvenlik sorunudur.

Bütün bunlar, bilgisayar ağlarındaki güvenlik konusunu çekici hale getirmiştir.

- 1-) İnternet ortamının nekadard güvenli olduğu
 - 2-) Neleri kimlerden nekadard korumak gerektiği
 - 3-) Saldırı yöntemleri ve savunma mekanizmaları
 - 4-) Güvenlik politikaları ve yazımı
 - 5-) İnternet üzerinden verilen servisler ve nekadard güvenli oldukları
- gibi konular üzerinde çalışmalar artmıştır.

Bu tez çalışmasında saldırı yöntemleri, korunma mekanizmaları, güvenlik sınıflamaları ve güvenlik duvarı yapısı incelenmiş olup, dağıtık sistem yapısına açık bir uygulama seviyesi güvenlik duvarı gerçekleştirilmiştir.

Anahtar Kelimeler: Güvenlik duvarı, vekil sunucu

SUMMARY

Tremendous developments have occurred in computer and electronics technologies, recently. Therefore, a lot of investments have been done on computers and networks. New projects have been developed quickly to satisfy the needs. The developed network solutions facilitated many works. Many large network projects are formed that are also supported by the government. The most important one of these projects is Internet.

The rapid growth of Internet and the provision of solutions over Internet by investors increased its use. The possibility of doing daily activities over Internet focused studies at this point. Some time later, the fact that the operations done over Internet are not much secure has been revealed. There has been news about people who committed attacks in order to gain personal benefits or disable the services provided by competitors. So, studies over the security of the systems and the provided services have started. As a result, defensive technologies have been developed and a firm infrastructure is formed.

It is a fact that the results of the studies in network technologies have made life easier. However, developing quick solutions before forming a good infrastructure caused some problems. Many problems can be counted of which, the most important one is security.

In this thesis study, attack methods, defense mechanisms, security classifications and firewall structures are examined and an application level firewall is realized.

Keywords: Firewall, proxy server

TEŞEKKÜR

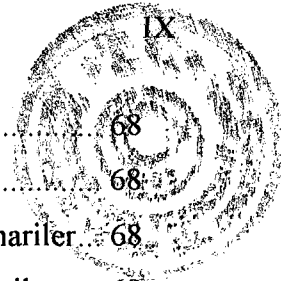
Tez için yapılan araştırma ve gerçekleştirmede desteğini hiçbir zaman esirgemeyen danışmanım Yrd. Doç. Dr. İbrahim SOĞUKPINAR' a sonsuz teşekkürler.



İÇİNDEKİLER DİZİNİ

	<u>Sayfa</u>
ÖZET	IV
SUMMARY	V
TEŞEKKÜR	VI
İÇİNDEKİLER DİZİNİ	VII
SİMGELER VE KISALTMALAR DİZİNİ	IX
ŞEKİLLER DİZİNİ	X
ÇİZELGELER DİZİNİ	XI
1. GİRİŞ.....	1
1.1. İnternetin Gelişimi.....	1
1.2. Güvenlik Problemleri	2
1.3. Güvenlik Çözümleri	3
1.4. Uygulama Seviyesi Güvenlik Duvarı Çözümü	4
2. GÜVENLİK GEREKLERİ VE KORUNACAK VARLIKLAR.....	6
2.1. Giriş.....	6
2.2. Korunacak Varlıklar	6
2.2.1. Veriler	6
2.2.2. Kaynaklar	7
2.2.3. Saygınlık	7
2.3. Güvenlik Seviyeleri.....	8
2.3.1. D Seviyesi	8
2.3.1.1. D1 Seviyesi.....	8
2.3.2. C Seviyesi.....	8
2.3.2.1. C1 Seviyesi.....	9
2.3.2.2. C2 Seviyesi.....	9
2.3.3. B Seviyesi.....	10
2.3.3.1. B1 Seviyesi.....	10
2.3.3.2. B2 Seviyesi.....	10
2.3.3.3. B3 Seviyesi.....	10
2.3.4. A Seviyesi	11
2.3.4.1. A1 Seviyesi.....	11
2.4. Kimlerden Korunmalı.....	12
2.5. Ağ Güvenliği.....	12
2.5.1. Güvenlik Politikasının Belirlenmesi.....	13
2.5.2. Fiziksel Güvenlik.....	14
2.5.3. Erişim Kontrol.....	14
2.5.4. Kimlik Onaylama.....	15
2.5.5. Şifreleme	15
2.5.6. Takip.....	15
2.6. Şifreleme.....	16
2.6.1. Gizli Anahtarlı (Simetrik) Şifreleme	16
2.6.2. Açık Anahtarlı (Asimetrik) Şifreleme	17

2.6.3.	Kimlik Doğrulama ve Sayısal İmzalar.....	18
2.6.4.	Elektronik Posta Güvenliği	19
2.6.5.	Ağ Yönetim Güvenliği.....	20
2.7.	Güvenlik Politikası	20
2.7.1.	Güvenlik Politikası Yazılması.....	20
2.7.1.1.	Risk Yönetimi.....	22
2.7.1.2.	Güvenlik Sorumluluğu.....	23
2.7.1.3.	İtaat	25
2.7.2.	Güvenlik Politikası Yönetimi.....	25
2.7.2.1.	Büyük Sistemlerin Güvenlik Yönetimi.....	25
2.7.2.2.	Tanımlanmış Politikaya Karşı Gerçek Performansın Ölçülmesi.....	25
2.7.2.3.	Politika Kategorileri.....	25
3.	SALDIRILAR VE GÜVENLİK YÖNTEMLERİ.....	27
3.1.	Standart Internet Servisleri	27
3.1.1.	www.....	27
3.1.2.	FTP	27
3.1.3.	SMTP.....	28
3.1.4.	Telnet, rlogin, rsh.....	28
3.1.5.	Usenet News.....	29
3.1.6.	DNS	29
3.1.7.	NFS	29
3.1.8.	Bilgi Tabanlı Diğer Servisler	29
3.2.	Bilgisayar Ağına Saldırı.....	30
3.2.1.	Saldırganlar	30
3.2.1.1.	Kötü Niyetli Saldırganlar (Malicious hacker).....	31
3.2.1.2.	Kötü Niyetli Olmayan Saldırganlar	31
3.2.2.	İşletim Sistemlerinin Güvenliği.....	32
3.2.3.	Saldırı Türleri	34
3.2.3.1.	Saldırıların Sınıflandırılması	34
3.2.3.2.	Saldırıların Gruplandırılması.....	36
3.2.3.3.	Bazı Saldırı Yöntemleri.....	40
3.3.	Güvenlik Duvarı.....	50
3.3.1.	Koruma Mekanizmaları	52
3.3.2.	Güvenlik Duvarı Araçları ve Teknolojileri	52
3.3.2.1.	Paket Filtreleyici	54
3.3.2.2.	Vekil Sunucular	56
3.3.2.3.	Güvenlik Duvarı Ürünleri Karşılaştırması	60
3.3.3.	Güvenlik Duvarı Mimarileri.....	61
3.3.3.1.	Çift Ağ Arayüzlü Güvenlik Duvarı (dual homed gateway).....	62
3.3.3.2.	Denetlenen Konak Güvenlik Duvarı (screened host firewall)	63
3.3.3.3.	Denetlenen altağ güvenlik duvarı (screened subnet firewall)	65
3.3.4.	Karışık mimarili güvenlik duvarı çözümü	67



3.3.4.1. Birden Fazla Korumalı Konaklı Mimariler	68
3.3.4.2. Tek Yönlendiricili Mimariler	68
3.3.4.3. Korumalı Konak ile Dış Yönlendiricinin Birleştirildiği Mimariler	68
3.3.4.4. Korumalı Konak ile İç Yönlendiricinin Birleştirildiği Mimariler	69
3.3.4.5. Çoklu İç Yönlendiricili Mimariler	69
4. VEKİL SUNUCULAR	70
4.1. Giriş	70
4.2. Web Vekil Sunucuların Tarihçesi	70
4.2.1. CERN Vekil Sunucusu	71
4.3. Vekil Sunucuların Genel Özellikleri	71
4.3.1. Saydamlık	71
4.3.2. İstemci Kontrollü Kullanım	72
4.3.3. Merkez Sunucunun Vekil Sunucudan Haberdar Olmaması	72
4.4. Vekil Sunucu Türleri	73
4.4.1. Genel Güvenlik Duvarı Vekil Vekil Sunucular	73
4.4.2. Vekil Zinciri	74
4.4.3. Alansal Vekil Sunucular	74
4.4.4. Kişisel Vekil Sunucular	75
4.4.5. Özelleştirilmiş Vekil Sunucular	75
4.5. Dinamik İçerik	76
5. GYTE-ProS VEKİL SUNUCU	77
5.1. Giriş	77
5.1.1. Kullanıcı Ayarları	79
5.1.2. Geliştirme Ortamı	80
5.1.3. Mimarisi	80
5.1.4. Kaynak Kodlar	82
5.1.5. Pencereleler	84
5.1.5.1. Ana Pencere	84
5.1.5.2. Filtreleme Penceresi	84
5.1.5.3. İzleme Penceresi	86
5.1.5.4. Hakkında Penceresi	87
6. SONUÇ	88
KAYNAKLAR	90
ÖZGEÇMİŞ	92
EKLER	
EK 1. GYTE-ProS KAYNAK KODLARI	

SİMGELER VE KISALTMALAR DİZİNİ

ACK	Acknowledgment
AFS	Andrew File System
CERT	Computer Emergency Responce Team
CGI	Common Gateway Interface
DARPA	Defence Advanced Research Prject Agency
DMZ	Demilitarized Zone
DNS	Domain Name Service
FNC	Federal Networking Control
FSP	File Service Protocol
FTP	File Transfer Protocol
GYTE	Gebze Yüksek Teknoloji Enstitüsü
GYTE-ProS	Gebze Yüksek Teknoloji Enstitüsü-Proxy Server
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
Ipv4	Internet Protocol version 4
MIT	Massachusettes Institute of Technology
MS-DOS	MicroSoft Disk Operatiing System
MTU	Maximum Transfer Unit
NFS	Network File System
NNTP	Network News Transfer Protocol
RFC	Request For Comments
RPC	Remote Procedure Call
SMTP	Simple Mail Transfer Protocol
SYN	Synchoronization
TCP/IP	Transport Control Protocol / Internet Protocol
TFTP	Trivial File Transfer Protocol
URL	Uniform Resource Locator
UUCP	Unix To Unix Copy
WAIS	Wide Area Information Service
WWW	World Wide Web

ŞEKİLLER DİZİNİ



<u>Sekil</u>	<u>Sayfa</u>
2.1 Gizli Anahtar Şifreleme Sistemi	16
2.2 Açık Anahtar Şifreleme Sistemi	17
2.3 Güvenlik Politikası Oluşturmak İçin İşlemler	21
2.4 Güvenlik-Korunmasızlık Maliyetinin Dengelenmesi	23
3.1 Süreçsel Sınıflandırma	35
3.2 İşlemsel Sınıflandırma.....	35
3.3 TCP El Sıkışma Mekanizması	42
3.4 Güvenlik Duvarı.....	51
3.5 OSI Güvenlik Duvarı Çözümleri	52
3.6 Uygulama Seviyesindeki Güvenlik Duvarı.	57
3.7 Çift Ağ Arayüzlü Güvenlik Duvarı.....	63
3.8 Denetlenen Konak Güvenlik Duvarı Mimarisi	64
3.9 Denetlenen Altağ Güvenlik Duvarı (Paket filtrelemeli).....	66
3.10 Ağ Adres Çevirimi.	67
4.1 Alansal Vekil Sunucu	75
5.1 Vekil Sunuculu Ağ.....	78
5.2 Kullanıcı İçin Vekil Sunucu Ayarları.....	80
5.3 İstemci, GYTE-ProS, Sunucu Mesaj Aktarımı.....	81
5.4 GYTE-ProS Servis Thread Bağlantı Yapısı	82
5.5 GYTE-ProS Nesne İlişkileri	83
5.6 Ana Pencere	84
5.7 Fitreleme Penceresi	85
5.8 İzleme Penceresi.....	87
5.9 Hakkında Penceresi	87

TABLÖLAR DİZİNİ

<u>Tablo</u>	<u>Sayfa</u>
2.1 DoD TCSEC’de Tanımlanmış Güvenlik Seviyeleri	11
3.1 Saldırganlar ve Amaçları	32
3.2 CERT Gruplandırması	36
3.3 TCP el sıkışma Mekanizması	45
3.4 Güvenlik Duvarı Mimarilerinin Karşılaştırılması	60



1. GİRİŞ

1.1. İnternetin Gelişimi

Haberleşme teknolojilerinin tarihine bakıldığında çok hızlı geliştikleri görülmüştür. Hızlı gelişimin başlıca sebebi, bireylerin ve toplumların etkileşime olan hayati ihtiyaçlarındandır. İnternette sunulan hizmetlerin çoğu aslında toplumsal gereksinimden ortaya çıkmıştır. İnternet artık sadece bir haberleşme aracı olmakla kalmamış aynı zamanda bilgi bankası, alış-veriş ortamı ve eğlence gibi günlük hayatımızda olan fonksiyonları yerine getiren bir teknoloji haline gelmiştir.

Çok kısa bir sürede internet teknolojisinde insanların hayal gücünü zorlayan gelişmeler yaşanmıştır. Bu gelişmeler, internetin etkileşime ne kadar açık olduğunun kanıtıdır. Haberleşme ve bilgilerin kolay paylaşımı sonucunda her bilim dalında olduğu gibi internette de büyük aşamalar kaydedilmiştir.

İnternetin gelişimi oldukça hızlı ve karmaşık olmuştur. Dolayısıyla kademe kademe gelişiminden bahsetmek zordur. Bu kadar hızlı gelişmesine çok sebep söylenebilir. Bunlardan en önemlisi;

1-) Toplumsal bilgilenme ve bilinçlenme gerekliliğinin duyulması, ağın insanların bilgi paylaşımına olanak sağlanması.

2-) Kuruluşların aynı ağ üzerinde birbirleriyle etkileşime geçme isteği.

gibi benzer sebepler sonucunda ağ genişlemiş ve hızla büyümüştür.

Bu kadar hızlı gelişmesinde resmi, ticari ve eğitim kuruluşlarının çok büyük etkisi vardır. İnternetteki gelişmeleri zorlayan etkilere en önemlisi de toplumsal gerekliliklerdir. İnternetin gelişiminden önce, insanların bulundukları yerden bilgiye kolayca ulaşma istekleri uzun süredir tartışılan bir konuydu. Bu konuyla ilgili kavramları ortaya koyan makaleler yayınlanmıştır (MIT, "Galactic Network", 1962).

1966 yılında bilgisayar ağıları konusunda çalışmalar başlamıştır. 1967 yılında Amerika Savunma Bakanlığının araştırma ve geliştirme bölümü olan DARPA (Defence Advanced Research Project Agency)'da bu konuda yapılan çalışmalara ARPANET adı verilmiştir. Bu proje internetin doğuşu olarak kabul edilmektedir. 1969 yılı sonunda paket anahtarlama yapacak cihazlar geliştirilmiş ve ARPANET projesi üzerinde optimizasyon yapıp değişik yerlerde bulunan dört bilgisayar birbirine bağlanmıştır. Daha sonraki yıllarda yeni bilgisayarlar da bu ağa bağlanmıştır. Bu fiziksel gelişmeye paralel olarakta ağ üzerinden verilecek hizmetler üzerinde de çalışılmaya başlanmış ve 1972 elektronik posta servisi tanıtılmıştır.

Internet, birbirinden farklı tasarımlara sahip, birbirinden bağımsız ağları birbirlerine bağlama fikrine dayanmaktadır. Ağlar arasındaki haberleşme protokolü olarak ta TCP/IP geliştirilmiştir.

Bütün bunların sonucunda internet bu günkü şeklini almıştır. 1995'te FNC tarafından yayınlanan bir bildiride internet için aşağıdaki tanımlar yapılmıştır;

- 1-) Birbirlerine IP protokolüne dayalı bir adres uzayı ile lojik olarak bağlı bilgisayarlardan oluşan bir bilgi sistemidir.
- 2-) TCP/IP ya da IP uyumlu protokoller kullanarak haberleşmeyi sağlar.
- 3-) Yüksek düzeydeki servislere ulaşılmasına olanak sağlar.

Şu anda gerçek zamanlı ses ve görüntü aktarımı üzerine çalışmalar vardır. Internet telefonu ve televizyonu gibi kavramlar çıkmıştır. Bu çalışmalar ileride başka çalışmaların önünü açacak ve hiç bitmeyecek gelişim süreci devam edecektir.

1.2. Güvenlik Problemleri

İnternetteki gelişmeler çok kısa zaman içinde olmuştur. Kullanımı 1985'ten sonra yaygınlaşmıştır. Gelişimin hızlı olması altyapı çalışmalarının iyi yapılmasını engellemiştir. Sonuçta standartlar oluşturulmadan çözümler üretildiği için problemler çıkmıştır. Bu problemlerin en önemlisi de güvenlik problemleridir.

Güvenlik her ağ uygulamasında düşünülmesi gereken unsurların başında gelmelidir. İnternet gibi herkesin kullandığı bir ağda çok fazla önem arz etmektedir. Artık kuruluşların büyük bir kısmı internet üzerinden ürünlerini tanıtmak, hizmet vermek durumundadır. Hatta şirketlerin farklı yerlerdeki bölümlerinin bağlantısı internet üzerinden yapılmaktadır. Ağ üzerinde yapılan tüm işler güvenlik açısından risk içermektedir. İnternet üzerinde bu risk maksimum seviyedir. Bütün risklere rağmen bir çok kuruluş internet üzerinden işlem yapmaktadır. Kurumların riske girmesinin sebebi ya verilerin gizliliğinin olması yada çeşitli güvenlik mekanizmalarını kullanmalarındandır.

İnternet üzerinden gerçekleştirilen veri transferinde oluşabilecek güvenlik sorunları basit olarak;

- 1-) Bilgilerin istenmeyen şahıslar tarafından dinlenilmesi
- 2-) Gönderilen verilerin başkaları tarafından değiştirilmesi
- 3-) Gönderilen verilerin hedefe ulaşmasının engellenmesi

Yukarıda sayılan güvenlik problemleri şirketlere ya da kurumlara zarar verecek türdendir. Aktarılan veriler ve haberleşen şahısların önemi arttıkça güvenlik konusu da önemde kazanmaktadır.[14]

1.3. Güvenlik Çözümleri

Güvenli bir ağ tasarımı sağlam bir güvenlik politikası belirlemekle başlar. Bu politika bilginin gizlilik derecesini , kullanıcı hakları ve uygulama sınırlamalarını hesaba katar. Politika güvenliğin temelini oluşturmaktadır. Kurumun önemli gördüğü korunması gereken bilgiler ile bu bilgileri tehdit eden hareketleri belirler ve kullanıcılara bildirir. [3]

Yukarıda anlatılan güvenlik problemleri için değişik çözümler geliştirilmiştir. İnternet denildiğinde güvenlik için düşünülen ilk çözüm güvenlik duvarıdır. Kurum ağını internetten yalıtmak, ağa giriş çıkışları kontrol altına almak, özellikle yerel ağ içerisindeki bilgisayarlara yönelik saldırıları karşılamak için güvenli bir yoldur.

Güvenlik duvarları bilginin yerel ağ çıkışından sonraki güvenlik problemlerine çözüm getirme konusunda yetersiz kalmaktadır. Özellikle farklı alanlardaki yerel ağlarını internet üzerinden birbirine bağlamak isteyen kurumlar için başka güvenlik mekanizmalarından da yararlanmak gerekir. Bunların başında veri şifreleme mekanizması gelmektedir. Transfer edilen bilgilerinin değiştirilip değiştirilmediğinin kontrolü için parity kontrol mekanizması da gerekmektedir.

Verinin transfer edildiği hedef bilgisayarın kimliğinden emin olunmasını sağlayan mekanizmalar geliştirilmiştir. Kimlik belirleme diyebileceğimiz bu yöntem vasıtasıyla taraflar karşılıklı olarak bilgi göndermeye başlamadan önce, birbirlerinin kimliklerinden emin olmak için kimlik belirleme mekanizmasının belirlediği işlemleri yerine getirirler. Böylece önemli bilgilerin saldırgan tarafından başka bir bilgisayara yönlendirilmesi engellenmiş olmaktadır.

Burada anlatılan güvenlik çözümleri daha çok alt düzey güvenliktir. Bunun yanında, uygulama seviyesinde geliştirilmiş birçok güvenlik çözümü vardır.[10]

1.4. Uygulama Seviyesi Güvenlik Duvarı Çözümü

Güvenlik duvarı kullanılarak yapılan güvenlik çözümlerinde, yazılımlar güvenli konak (bastion host) adı verilen makinalarda çalışır. Bu makinalarda OSI modelindeki alt katmanlarda çalışan paket süzme, saldırı tahmini gibi mekanizmalar bulunmaktadır. Güvenliğin tam olarak sağlanabilmesi için OSI modelindeki bütün katmanlara korunma mekanizmasının uygulanması gerekir. OSI modelinde oturum sunuş ve uygulama katmanlarında çalışan güvenlik sistemlerine vekil sunucu (proxy server) adı verilir.

Vekil sunucuların içerik, konak adı ve erişim kontrol gibi güvenliğin sağlanmasına yönelik işlevleri vardır. Ayrıca performansın artırılmasına yönelik olarak, http bağlantısı sonucunda sunucudan alınan bilgilerin daha sonraki bağlantılarda vekil sunucudan doğrudan alınabilmesi için kayıtlama (caching) yaparlar. Yine vekil sunucuların internet bağlantılarında bir yerel ağı tek bir gerçek IP numarası üzerinden internete bağladıkları bilinmektedir. [1]

Bu tez çalışmasında içerik, konak adı ve url adres kontrolü yapabilen bir vekil sunucu tasarımı ve gerçekleştirmesi yaptık. Vekil sunucunun adı GYTE-ProS' dur. Bu vekil sunucu java dilinde gerçekleştirildiğinden işletim sistemi (platform) bağımsızdır. İlerideki çalışmalarda vekil sunucuların diğer özellikleri de eklenebilecek şekilde esnek bir yapısı vardır. Çoklu makinaların GYTE-ProS üzerinden internete çıkılacağı düşünülerek GYTE-ProS'un dağıtık mimari tasarımı da yapılmıştır.



2. GÜVENLİK GEREKLERİ VE KORUNACAK VARLIKLAR

2.1. Giriş

İnternet, insanların bilgiye kolay ulaşım dolayısıyla çalışmalarındaki verimin artması sağlayan büyük bir bilgi ağıdır. Bilgiye kolay ulaşım için sunulan hizmetler (servisler, http, ftp,) aynı zamanda zarar verilebilme riski de taşımaktadır. İnternetin sunduğu imkanlardan faydalanmak fakat gelebilecek zararları en aza indirmek gerekir. Fakat bu tedbir birtakım şeylerden ödün vermemizi gerektirir. Güvenliği ön plana almak, hızı da aynı oranda azaltmak anlamına gelmektedir.

Alınabilecek güvenlik önlemlerini tartışmadan önce güvenlik konusunun neden gerekli olduğunun, nelerin korunması gerektiğinin anlaşılması daha faydalı olacaktır.

2.2. Korunacak Varlıklar

Bir ağda güvenlik ile ilgili bir çalışma yapılmaya başlandığında ilk karar verilmesi gereken nelerin korunması gerektiğidir. Korunması gereken varlıklar üç ayrı ana başlıkta toplanabilir.

1-) Veriler

2-) Kaynaklar

3-) Saygınlık

Bu varlıklar ayrı ayrı incelenecektir.

2.2.1. Veriler

Veriler, güvenlikle ilgili olarak üç özelliğe sahip olmalıdır;

Gizlilik: Verilerin , başkaları tarafından öğrenilmesi istenmeyebilir.

Bütünlük : Sahip olunan verilerin başkaları tarafından değiştirilmesi istenmeyebilir.

Kullanıma hazırlık : Verilerin istendiği zaman ulaşılabilir olup kullanıma hazır olması istenir.

Daha çok gizlilikle ilgili güvenlik üzerinde durulur. Gerçekten de bu konuda risk çoktur. Bir çok kişi ya da kuruluş için gizli bilgiler bilgisayar üzerinde tutulur. Bu bilgisayarların güvenliği de internet bağlantısı kopartılarak sağlanmaktadır. Bu şekilde bilginin gizliliği sağlanmış olabilir ama kolay ulaşılabilirlik ortadan kalkmış olur. Yani bir şekilde ağa bağlanılmalıdır. Bu durumda güvenlik politikaları belirlenerek, bilgilerin güvenliğinin sağlanması gerekmektedir.

2.2.2. Kaynaklar

İnternete bağlanmakla riske atılacak ikinci şey, bilgisayar kaynaklarıdır. Başka insanların bir kuruluşa ait bilgisayardaki sabit diskte yer alan boş alanları kendi amacı için kullanmak istemesi her ne kadar mevcut verilere zarar vermeyecek bir şey olsa da istenecek bir durum değildir. Bunun gibi diğer kaynakların da (işlemci,bellek, ...) başkaları tarafından kullanılması, kabul edilebilir bir şey olamaz

2.2.3. Saygınlık

Her kişi ya da kurumun saygınlığının internet üzerinde de korunması önemlidir. Meydana gelebilecek güvenlik problemleri kişi ve kurumların doğrudan aleyhine olup kötü reklamdır. İnternet üzerinde işlemler yapan bir kişinin, başka bir kişinin adını kullandığı düşünülürse, zarar verme durumunda direk muhatap alınacak kişi saygınlığını kaybetme durumuyla karşı karşıya kalacaktır.

Genelde başka birinin hesabından girip sahte elektronik postalar atarak zarar verilir. Bunun sahte olduğunun kanıtlanması neredeyse imkansızdır. Böyle durumlarda, sahteciliği yapan kişinin kullandığı hesaba sahip kişi kadar kurum da zarar görür.

İnternete açılmayı düşünen kurumların eğitim ya da güvenlik politikası içinde saygınlığın korunması için kişilere düşen güvenlik tedbirlerinin anlatılması gerekir. Ayrıca periyodik olarak takibinin yapılması şarttır.



2.3. Güvenlik Seviyeleri

Güvenlik konusunda olabilecek yaklaşımları gördükten sonraki konu, sistemlerin sahip oldukları güvenlik seviyelerini belirlemek ve yükseltmek olabilir. Sistemlerin içerdikleri donanım ve yazılımlara göre güvenlik seviyeleri özellikleri belirlenmiş ve standartları oluşturulmuştur. Güvenlik seviyelerinde çeşitli fiziksel korumalar, işletim sistemini güvenli hale getirme gibi işlemleri içerir.

1985 yılında DoD tarafından yayınlanan TCSEC yayınında, dört güvenlik seviyesi ve alt sınıfları belirtilmiştir.

2.3.1. D Seviyesi

Sadece bir güvenlik seviyesi içerir. En düşük düzeyde güvenlik seviyesidir.

2.3.1.1. D1 Seviyesi

Mevcut en düşük güvenlik olanaklarını sunar. Bu seviyede bir güvenliğe sahip sistem, bütün olarak güvensizdir. Donanım elemanları için herhangi bir koruma mekanizması yoktur. İşletim sistemi kolaylıkla geçilebilir ve istenen amaçlara uygun şekilde kullanılabilir. Sistem kaynaklarına yetkili kişilerin ulaşmasını denetleyecek bir erişim kontrol sistemi yoktur. MS-DOS, MS-Windows 3.1/95/98 ve Apple Macintosh bu sınıftandır.

2.3.2. C Seviyesi

C1 ve C2 olmak üzere iki alt güvenlik seviyesine ayrılmıştır. Bu seviye güvenlikte kullanıcı için hesap tutulur(account) ve izleme(audit) yapılır.

2.3.2.1. C1 Seviyesi

Sınırlı bir güvenlik koruması vardır. Daha çok kullanıcı hatalarından sistemi korumak gerekli tanımlar bulunur. Dışardan gelecek saldırılara karşı koruma mekanizmaları yoktur.

Unix işletim sisteminin sunduğu güvenlik gereklerini içerir. Ayrıca donanım elemanları için bazı güvenlik mekanizmalarını da içerir. Donanım elemanlarının istenmeyen kişiler tarafından ulaşılması zorlaştırılmıştır. Sistem kaynaklarına ulaşmak isteyen kullanıcıların erişim kontrolü yapılmaktadır. Erişim kontrolü, kullanıcı adı ve parolasına göre yapılmakta ve sonuçta kullanıcının sisteme erişim hakkı varsa sisteme alınmaktadır. Kullanıcı sisteme girdikten sonra kendisine verilmiş haklar ve sınırlamaların dışına çıkamaz.

Kullanıcı adı ve şifresinin belirlediği erişme hakkı, sistemdeki dosya ve dizinlere ilişkin izinlerdir. Dosya ve dizinlere erişim hakkı o dizin sahibi ya da sistem yöneticisi tarafından verilebilir. Böylece istenmeyen kişilerden korunma sağlanır. Sistem yöneticisi için bir sınırlama olamaz.

UNIX ve IBM MVS bu sınıfa örnektir.

2.3.2.2. C2 Seviyesi

C1 seviyesine göre daha güvenli hale getirilmiştir. C2 seviyesinde kaynaklara kontrollü erişim sağlanabilmektedir. Bunun anlamı, bir kullanıcının bir dosya veya dizine ulaşması sırasında sadece haklarına bakılarak izin verilip verilmemesine karar verilmez. Bununla birlikte bir yetkilendirme mekanizması da geliştirilerek, kullanıcının belirtilen dizine ulaşmaya yetkili olup olmadığına veya bir komut koşturmak için gerekli yetkiye sahip olup olmadığına bakılarak, istediği işlemi yapması sağlanır ya da reddedilir. Yetkilendirmenin dışında bu seviye güvenlikte yapılan işlemlerin kontrol edilmesi gerekir. Bunun için de sistemde yapılan her iş ile ilgili bir kayıt tutulur.

C1 güvenlik seviyesindekilere ek olarak yapılan işlemlerin kontrol ve kaydedilmesi, C1 seviyesinde yaşanan güvenlik problemlerini ortadan kaldıracaktır.



Yapılan fazladan kontrol ve kayıt işlemleri, sistemin işlemci zamanını ve diskten alan alacaktır.

C2 güvenlik seviyesine örnek sistemler Windows NT 4.0 ve Digital Equipment VAX/VMS 4.x gösterilebilir.

Güvenlik arttıkça kaynaklara erişimdeki hız düşmektedir.

2.3.3. B Seviyesi

Üç alt güvenlik seviyesine (B1,B2,B3) ayrılır. Zorunlu erişim denetimi kullanılır. Sistemdeki her nesnenin güvenlik seviyeleri tanımlanır.

2.3.3.1. B1 Seviyesi

Çok katmanlı güvenlik yapısı kurulmasını sağlar (gizli,en gizli vb..). Sistemde, güvenliği sağlanacak nesnelerin diğerlerinden kesinlikle ayrılması gerekmektedir. Bu nesneler diskette ya da diskte saklanacak türdendir.

Bu sisteme örnek olarak OSF/1, AT&T V/MLS, IBM MVS/ESA sistemleri verilebilir.

2.3.3.2. B2 Seviyesi

Bu seviyedeki güvenlik için sistemdeki bütün nesnelerin (birimlerin) etkilenmesi gerekmektedir. Diskle, teypler veya terminaller, bir veya daha fazla olabilecek güvenlik seviyesi ile ilişkilendirilebilir. Güvenlik düzeyi yüksek olan bir cihaz ile güvenlik düzeyi düşük cihazın haberleşmesinde problemler çıkacaktır, bunlara dikkat edilmesi ve çözülmesi gerekir.

Bu güvenlik seviyesine örnek olarak Honeywell Information Systems'in Multics Sistemi, Trusted XENIX verilebilir,

2.3.3.3. B3 Seviyesi

Güvenliği, donanımların uygun kurulumlarıyla sağlamaya çalışan yöntemi içerir. B2 seviyesine göre daha sağlam, ciddi bir sistem tasarımı vardır. Güvenlik

yönetimi, güvenli kurtarma ve saldırıların ya da oluşan zararların sistem yöneticisine hemen bildirilmesi gibi özellikleri içerir. Bu seviyeye örnek olarak Honeywell XTS-200 verilebilir.

2.3.4. A Seviyesi

Tek sınıf içermektedir. En üst güvenliği sunan güvenlik seviyesidir. Donanım ve yazılım açısından dizayn, kontrol ve doğrulama işlemlerini içerir. Daha önce bahsedilen güvenlik seviyelerindeki bileşenleri içermektedir. Bir sistemin dizayn, geliştirme ve gerçekleştirme aşamalarında güvenlik isteklerinin sağlanması istenir. Her aşamayla ilgili isteklerin dökümana uygun olarak yerine getirilmesi gerekmektedir.

2.3.4.1. A1 Seviyesi

Dizaynın sağlamlığının matematiksel olarak incelenip, test edilmesi ve doğrulanması gerekmektedir. B3 sınıfına ek olarak güvenli dağıtım (trusted distribution) özelliği eklenmiştir. Güvenli dağıtım ilkesi uyarınca, sisteme ilişkin yazılım ve donanım bileşenleri üzerinde, güvenlik sistemini etkileyecek değişikliklerin, sistemlerin aktarılması aşamasında tekrar güvenliğin sağlanması gerekir. Güvenlik seviyeleri tablo 2.1'de özet olarak verilmiştir.

Tablo 2.1. DoD TCSEC'de Tanımlanmış Güvenlik Seviyeleri

Güvenlik Seviyesi	Alt Seviye	Özet Bilgi
D	D1	En düşük düzeyde güvenlik
C	C1	İsteğe(Kullanıcıya) bağlı güvenlik
	C2	Kontrollü erişim
B	B1	Etiketli güvenlik
	B2	Yapısal güvenlik
	B3	Güvenlik Alanlı koruma
A	A1	En yüksek düzeyde güvenlik

2.4. Kimlerden Korunmalı

Ağın yapısına bağlı olarak saldırganlar değişebilir. Saldırganlarının büyük çoğunluğu iç ağdan gelmektedir. Ağa açılan bilgisayarların verdiği hizmetlere göre, ne tür saldırılara uğrayacağı ve saldırgan türleri de ortaya çıkabilir.

En çok görülen saldırı tiplerinden biri, sisteme davetsiz misafirlerin girmesidir. Bunlar sisteme girdikten sonra çeşitli saldırılar yaparlar. Genelde yapılan, verilerin servislerin kilitlenmesine yönelik saldırılardır.

2.5. Ağ Güvenliği

Sistemlerin büyümesi ve sistem içerisindeki birimlerin farklı özelliklere sahip olması durumunda her bir düğümün (makine, bilgisayar) güvenliğinin sağlanması oldukça zor olacaktır. Bu da ağın güvenliğinin sağlanması çalışmalarına yönelmeye zorlamaktadır. Bu yaklaşımdan yola çıkılarak, ağda hizmet veren makinalara ve ağa erişimlerin tamamının kontrol edilmesi gerekir. Her ağa, güvenlik delikleri sayesinde içeriden ya da dışarıdan izinsiz erişimler olabilmektedir. Ağın önemli ve hassas bilgiler barındırması sebebiyle içerdeki verilerin ve hizmetlerin korunması önemlidir. Önemli verilerin sadece iç ağdaki kullanıcılara değil aynı zamanda dışarıdan girebilecek kişilere karşı da korunmuş olması gerekir.

En iyi güvenlik çözümü ağ bağlantısını kesmektir. Ancak bu tür bir çözüm bilgisayar haberleşmesi teknolojisini kullanan organizasyonların zararına olacaktır. Dıştaki güvenilmez ağ ile içteki güvenilir özel ağ arasında koruma sağlayan güvenlik duvarları, ağ bağlantısını kesmek yerine kullanılabilir.

Güvenlik duvarlarının kullanılması, komple güvenlik çözümünü oluşturan esaslar olan;

- 1-) Güvenlik politikasının belirlenmesi
- 2-) Fiziksel güvenlik
- 3-) Erişim kontrol

4-) Kimlik onaylama

5-) Şifreleme

6-) Takip

fonksiyonlarının sadece bir parçasını oluşturacaktır. Güvenlik duvarı kullanarak çok büyük yerel alan ağları korunabilir.

2.5.1. Güvenlik Politikasının Belirlenmesi

İdeal güvenlik, ağ tasarımını, bilginin gizlilik derecesini ve kullanıcı hakları ile uygulama sınırlamalarını hesaba katan sağlam bir güvenlik politikası ile başlar. Politika güvenliğin temelini oluşturur. Kurumun önemli gördüğü ve korunması gereken bilgiler ile bu bilgileri tehdit eden hareketleri belirler. Güvenlik politikasının oluşturulmasında sorulacak anahtar sorular şunlardır.

1-) Kime, nereye, ne zaman, nereden ve hangi yetkiyle izin verilebilir ?

2-) Hangi aktiviteler tehdit olarak görülür ve güvenlik riski yaratırlar ?

3-) Ne tip gözden kaçmalar ve dikkatsizlikler olabilir?

4-) Güvenlik politikasının gerçekleştirilmesinde kim, ne yetki ve sorumluluğa sahiptir?

Güvenlik riskleri içeriden veya dışarıdan gelebilecek saldırılardan kaynaklanır. Güvenliğin sağlam olması için yapılması gerekenler şunlardır;

1-) Dışarıdan içerideki özel ağa izinsiz erişimi engellemek için Dış Güvenlik (Internet ya da modem bağlantıları),

2-) Hassas bilgilerin içeriden izinsiz erişimini engellemek için İç Güvenlik.

Dış ve İç güvenlik için problem yaratan ağ sistemi zayıflıkları üç ayrı kategoride özetlenebilir.

1-) Protokol, bilgisayarlar ve ağlar arasındaki iletişimi yöneten ana kurallar. Bir organizasyon ağı içindeki ağ için bir ya da birden fazla protokol, ve internet ile haberleşmek için başka bir protokol kullanabilir. Yapıları gereği, bazı protokoller saldırganların kullanabileceği güvenlik deliklerine sahiptir. Bir güvenlik duvarı içindeki ağ bir çok protokol sorunlarından koruyabilir.

2-) Yazılım, işletim sistemleri ile bilgisayar ve ağların kendi aralarındaki iletişimi sağlayan ağ iletişimi uygulamaları. Yazılım zayıflıkları genelde güvenli olmayan sürümlerin kullanılmasından kaynaklanır.

3-) Konfigürasyon, ağdaki donanım ve yazılımın kullanıcı tarafından parametrelerinin düzenleme kuralları. Konfigürasyon zayıflıkları donanım ve yazılımın güvenli olarak nasıl kurulacağını bilinmemesinden ve etkili güvenlik politikasının olmamasından kaynaklanabilir.

2.5.2. Fiziksel Güvenlik

Ağ güvenliği fiziksel güvenlikle bağlantılıdır. Ağ makinesinin boyutu ve şeklinin yanı sıra ağın ihtiyaçtan dolayı ve karşılıklı güven ilişkilerine dayalı olarak bir binayı, kampüsü, ülkeyi ya da dünyayı sarabilme ihtimali vardır. Fiziksel güvenlik politikasının, ağ güvenliği politikası oluşturulurken yenilenmesi veya dikkate alınması gerekebilir.

2.5.3. Erişim Kontrol

Erişim kontrol, gelen her ağ paketinin içeriye alınıp alınmayacağına ve pakete karşı yapılacak davranışa karar verir. Bir güvenlik duvarı, paketin veya oturumun tanımlanmış güvenlik politikasına uygunluğunu belirler. İyi tasarlanmış güvenlik duvarı detaylı güvenlik politikalarını gerçekleştirebilir. Ayrıca güvenlik duvarları özel bir ağı uzaktan erişim zayıflıklarına karşı koruyan en iyi çözüm olarak kabul görmüşlerdir.

Ağ protokolleri, yazılım ve konfigürasyondaki yanlışlıklar ve problemlerden kaynaklanan uzaktan erişim zayıflıkları, saldırganların yetki alarak dışarıdan sisteme girişlerini kolaylaştırmaktadır.

2.5.4. Kimlik Onaylama

Kimlik onaylama yetkili personel ve bölümlerin serbestçe haberleşmelerini sağlarken izinsiz erişimi engellemektedir. Kullanılan onaylama yöntemi kullanıcının nereden ve nasıl onaylandığına bağlıdır. İnternet ve öteki uygulamalar için en popüler onaylama yöntemleri, “Neredeler”, “Neleri var” ve “Neler Biliyorlar” dır. Ancak IP adres onaylaması veya “Neredeler” yöntemi, IP adres sahtekarlığı saldırı yöntemi kullanılarak geçilebilir.

2.5.5. Şifreleme

Şifreleme veri bütünlüğünü garantileyebilir ve güvenli hatlardan yollanan bilgiyi koruyabilir. Önemli şirket bilgilerine uzaktan erişimde veya organizasyon intranetine erişimde korumanın sağlanabilmesi için şifreleme kullanılabilir.

Ancak şifrelemede önemli olan anahtarların hangi yoldan gönderileceği ve nasıl yönetileceği konularıdır. Anahtarlar verinin şifrlenmesi ve açılmasında kullanılır. Otomatik anahtar yönetimi bir çok konağı bulunan ağ için şarttır. PKI bu alandaki çalışmaları içermektedir.

2.5.6. Takip

Güvenlik politikası uygulanmaya başladıktan sonra, bütün sistem parçaları ve personelin güvenlik politikasına uygunluğunun periyodik olarak kontrol edilmesi gerekmektedir. Yeterli denetimin olmaması durumunda, bir güvenlik ihlali sonrası takip için yeterli delil olmaması durumunda, bir güvenlik ihlali sonrası takip için yeterli bilgi olmayabilir. Denetimin yapılması, problemleri önceden tespit ederek güvenlik boşluklarına dönüşmelerini engelleyebilir. Günlük kayıt ve anında uyarı mesajının yollanması, kısa sürede önlem alınmasını ve atak kaynağının tespitini kolaylaştırır.

2.6. Şifreleme

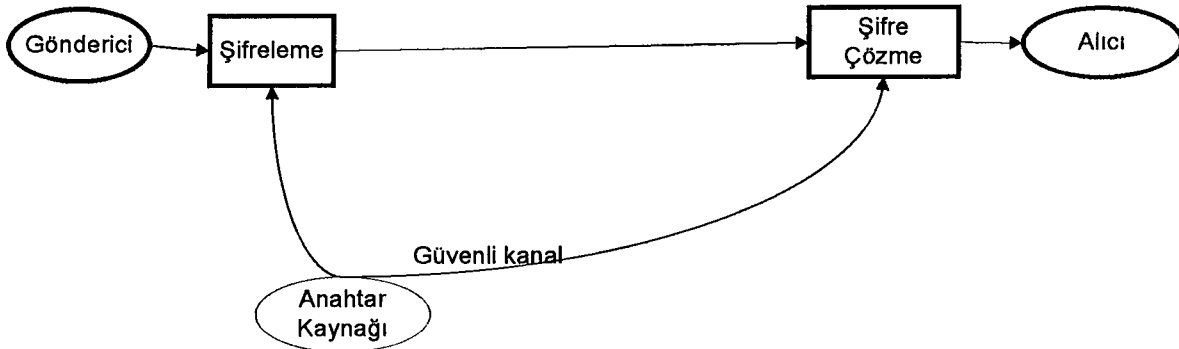
Bilgisayar ağlarının ve haberleşme sistemlerinin güvenliğinin sağlanması için kullanılan en önemli işlem, verilerin şifrlenerek anlamsız hale getirilip hedefe gönderilmesi ve hedefte tersi işlem yapılarak tekrar eski hale getirilmesidir.

Bir şifreli haberleşme için;

- 1-) Şifreleme algoritması (E)
- 2-) Deşifreleme algoritması (D)
- 3-) Bir anahtar bilgisine (K) ihtiyaç vardır.

2.6.1. Gizli Anahtarlı (Simetrik) Şifreleme

Gizli anahtarlı şifreleme sistemleri uzun süredir kullanılmaktadır. Bu sistemlerde şifreleme algoritması ve deşifreleme algoritması birbirinin tersi şeklindedir. Öncelikle haberleşecek iki grup aralarında gizli bir anahtar tespit ederler. Bu iki grup birbirlerine yakın yerlerde yer almıyorlarsa güvenli bir haberleşme kanalı veya güvenilir bir kurye yoluyla anahtarları birbirlerine ulaştırabilirler. Bir taraf şifreleme algoritmasında girdi olarak açık metin (P) ve anahtarı (K) uygular, ardından şifreli metin (C) yi elde eder ve mesajın alıcısına gönderir. Mesaj alıcısı ise deşifreleme algoritmasının girdileri olarak şifreli metin (C)'yi ve aynı (K) anahtarını kullanır ve ardından çıktı olarak açık metin (P)'yi elde eder.



Şekil 2.1. Gizli Anahtar Şifreleme Sistemi

Gizli anahtarlı kriptu sistemler, verileri iki şekilde şifreleyebilirler;

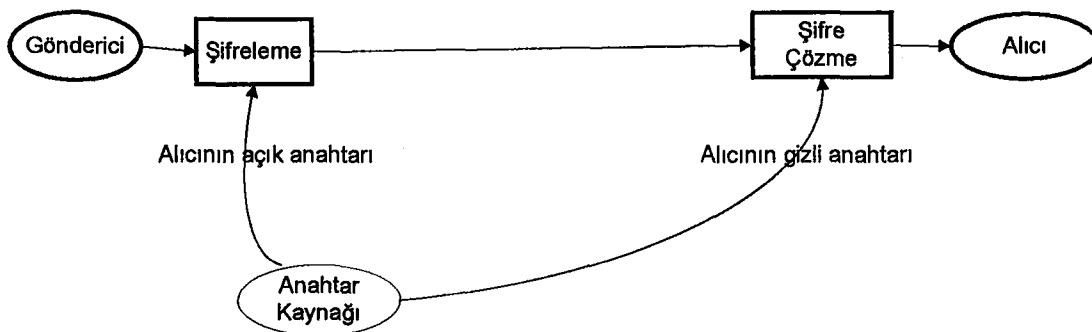
Blok Şifreleme: Şifrelenecek veriler belli uzunluktaki parçalara bölünüp her parça ayrı ayrı şifrelenir.

Dizi Şifreleme: Veriler bir anahtar üreticinden çıkan kayar anahtar dizisiyle işleme tabi tutulur. Dizi şifreleme blok şifrelemeye göre daha hızlı şifreleme ve deşifreleme yapabilmektedir.

Gizli anahtarlı kriptu sistemler ağ üzerinde uçtan uca şifreleme işleminde kullanılabildiği gibi ağa giriş çıkışların kontrol edildiği parola koruma sistemlerinde girilen parolaların şifrenip veri tabanında saklanması da kullanılabilir. En sık kullanılan blok şifreleme algoritmaları DES,3DES,IDEA,FEAL dır. Dizi şifrelemeye RC4 algoritması örnek olarak verilebilir.[9]

2.6.2. Açık Anahtarlı (Asimetrik) Şifreleme

Açık anahtarlı şifreleme sistemi henüz yeni kullanıma girmiştir. Açık anahtarlı kriptu sistemlerinin amacı belli bir anahtar üzerinde anlaşmanın ve karşı tarafa u anahtar güvenli olarak ulaştırabilmenin zorluğunu ortadan kaldırmaktır. Burada tek yönlü mesajlaşma söz konusudur. Mesaj alıcısı sadece kendisinin bileceği (Gizli) ve diğer kişilere dağıtabileceği bir (Açık) anahtardan oluşan çifti belirler. Mesaj göndericisi alıcıya ait herkesçe bilinen açık anahtar kullanarak açık anahtarlı kriptu algoritmasıyla, göndereceği mesajı kapatır ve alıcıya gönderir. Mesajın alıcısı ise yalnız kendisinin bildiği gizli anahtar ile deşifreleme algoritmasını kullanarak mesajı açabilir. Açık anahtar sadece alıcı tarafında olduğu için başka birinin bu mesajı açması mümkün olmayacaktır.



Şekil 2.2. Açık Anahtar Şifreleme Sistemi

En yaygın kullanılan açık anahtarlı algoritmalar 1978'de Rivest, Shamir, Adleman tarafından sunulan RSA algoritması ve 1985 yılında ElGamal tarafından sunulan ElGamal açık anahtarlı kriptosistem algoritması sayılabilir. Ayrıca eliptik eğriler üzerine kurulu algoritmalarda kullanım hızlarının yüksek olamamaları ve daha az anahtar uzunluğu gerektirmeleri sebebiyle yaygın olarak kullanılmaktadır. [9]

2.6.3. Kimlik Doğrulama ve Sayısal İmzalar

Ağ üzerinde hareket eden verilerin geldikleri adresten tam olarak gönderildiği şekliyle gidecekleri yere ulaşmaları amacıyla veriler ve paketler değişik şekillerde özetlenirler. Bu şekilde kullanılan özetleme fonksiyonları (Hash Functions), verileri tek yönlü bir matematiksel fonksiyona tabi tutup özet değeri oluştururlar. Bu özet değer paket içerisinde yollar. En çok kullanılan özetleme fonksiyonları SHA (Secure Hash Algorithm) ve MD5 algoritmalarıdır. Başka bir yolda veri paketinin sonuna her paket için bir tane üretilen CRC (Cyclic Redundancy Check) kodu veya toplam kontrol bilgisinin (CS) eklenmesidir.

Verilerin ağ üzerinde doğrulanmasının dışında ayrıca gerçekten üzerinde yazılı olan adresten gelip gelmediğinin kontrolü ise sayısal imza algoritmaları ile sağlanmaktadır. Sayısal imza algoritmalarında, veriyi gönderen adresin kendisine ait gizli bir anahtarla verinin kendisi, imzalama algoritmasına girer ve çıkan bilgi bize o pakete ait imzayı vermektedir. Paketin alıcısı ise, imzanın doğrulanması aşaması için, imzanın doğruluğunu kontrol eder. Sayısal imza algoritmaları olarak, aynı zamanda açık anahtarlı kriptosistem algoritması amacıyla kullanılan RSA ve ElGamal algoritmaları kullanılabilir. Bunlara ek olarak DSA (Digital Signature Algorithm) imza algoritması ve pek çok özel tür algoritmada kullanılabilir. [10]

Ağ ve internet güvenliğinde bugün yaygın olarak kullanılan kimlik doğrulama uygulamaları arasında Kerberos protokolünü sayabiliriz. Kerberos daha çok gizli anahtarlı şifreleme üzerine kurulu bir sistem olmakla beraber istemci sunucu arasındaki diyalogların doğrulanmış bir şekilde yapılması işlemini yönetir.

X.509 dizin doğrulama servisi ise X.500 dizin servisinin kullanımı ile yaygınlaşan bir kimlik doğrulama standardıdır. X.509 dizin servisi daha çok açık

anahtarlı kript sistemler üzerine kuruludur. Bu serviste hiyerarşik bir düzende yer alan kullanıcılar arasındaki haberleşmelerde servisin kullanıcılara sağladığı sertifikalar söz konusudur. Her sertifika verinin göndereceği adresin kimlik numarası ve ona ait açık anahtarı içermektedir. Sertifikaların içeriğinin değiştirilmesinin engellenmesi amacıyla sertifikalar belli bir sertifikasyon otoritesi sunucu tarafından imzalanır. Dolayısıyla veri gönderecek adresteki birim alıcıya ait sertifika, sertifika otoritesinden (CA - Certification Authority) elde eder, altındaki imzayı kontrol eder. Sertifikanın doğruluğundan emin olduktan sonra içeriğinde yer alan açık anahtarı kullanarak alıcıya mesajı şifreleyerek gönderir.

2.6.4. Elektronik Posta Güvenliği

Ağ üzerindeki elektronik postaların güvenli bir şekilde gönderici ve alıcı arasında yol alması amacıyla değişik uygulamalar değişik protokolleri, bu protokoller de farklı şifreleme ve imzalama algoritmalarını kullanmaktadırlar. Bunlardan biri olan PGP protokolünde X.509'a benzer bir sertifikasyon yapısı vardır. PGP protokolünde Her kullanıcı aynı zamanda bir sertifikasyon otoritesidir(CA). Bunun anlamı ise her kullanıcı kendine kendine ait bir gizli anahtar seçip bu anahtara uygun bir açık anahtar oluşturacak ve bu anahtarı kimlik bilgisiyle ağ üzerinde ortak kullanılan bir sunucuya aktaracaktır. Kendisine ait gizli anahtarı uygun bir yerde saklayan kullanıcı haberleşeceği adresin açık anahtarını ise Public-Key ring adı verilen bir soyada toplayacaktır. PGP, açık anahtarlı algoritma olarak RSA, gizli anahtarlı algoritma olarak IDEA ve özetleme fonksiyonu olarak MD5 algoritmasını kullanmaktadır. Kendi simetrik algoritmasına ait üretilen gizli anahtarla mesajı şifreleyen kullanıcı, gizli anahtarla alıcı kişinin açık anahtarlı algoritmasına ait açık anahtarla şifreleyecek ve bütün bu şifreli kısımları da kendine ait gizli anahtarla imzalayacak ve alıcıya gönderecektir. Alıcı ise kendi açık anahtarlı algoritmasının gizli anahtarıyla, şifrelenmiş anahtarı açacak ve buradan elde edilen anahtarla da şifrelenmiş mesajı açacaktır. Mesajın imzasını ise gönderenin açık anahtarıyla kontrol edecektir.

2.6.5. Ağ Yönetim Güvenliği

Birbirine bağlı ağların kullanımının artmasıyla birlikte ağların yönetim sistemlerindeki güvenlik probleminin çözülmesi gerektiği de önem kazanmıştır. Ağ yönetim protokollerinden SNMPv1 sadece ulaşım kontrolü vardı. Güvenlikle ilgili eklemeler yapıldı ve ikinci sürümü çıkartıldı. SNMPv2 ulaşım kontrolünün yanında kimlik doğrulama ve gizlilik fonksiyonlarını da içermektedir.

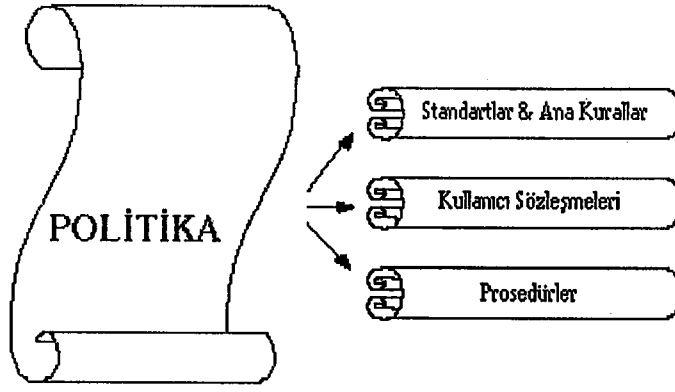
2.7. Güvenlik Politikası

Politika, farklı tiplerdeki bilgilere erişim için kişilere yetki sağlar. Ayrıca ne tür ve ne kadar güvenlik zorunluluğu ve ölçülerine uyulması gerektiğine bakarak kuralları ve standartları belirler. Prosedürler, politikayı sağlayan standart ve kuralların(guidelines) uygulanması için metotları sağlar. Neredeyse bütün kurumların bir bilgi güvenliği politikası vardır. Fakat bazı kurumlarda politika açıkça yazılmamıştır.

Politika yazılmazsa, organizasyon çalışanların yanlış anlamalarından dolayı risk altındadır demektir. Ayrıca bir güvenlik problemi olması durumunda cezanın boyutu da belirsiz olacaktır. Bir politika yazmak, organizasyonun kararlılığının başlangıcıdır. Ayrıca ana kuralların belirlenmesi ve onların uygulamaya konulması açısından da önemlidir.

2.7.1. Güvenlik Politikası Yazılması

Bir bilgi güvenliği politikası bir vakum içinde yazılmaz. O kurumun ihtiyaçlarıyla direk olarak ilişkilidir. Bütün kurumlara tam olarak uyan genel bir güvenlik politikası yoktur.



Şekil 2.3. Güvenlik Politikası Oluşturmak İçin İşlemler

Bir organizasyonun güvenlik politikası oldukça kısa olmalıdır. Yaklaşık olarak beş sayfa olabilir. Platform bağımlı terimler olmamalı, genel olmalıdır. Yapı ve teknolojiye göre esnek olarak hazırlanmalıdır. Birkaç yıllık bir ömrü olmalıdır.

Güvenlik politikası dokümanı aşağıdaki bilgileri içeren dokümanlarla ilişkili olmak zorundadır.

- 1-) Standartlar ve ana kurallar
- 2-) Kullanıcı sözleşmeleri
- 3-) Prosedürler

Bunlar bilgisayar platformu, teknolojisi, uygulamaları, kullanıcı güvenirliliği ve organizasyon yapısıyla ilgili özel bilgileri içeren dokümanlardır. Politika, güvenliğin uygulanması için beyanname ve ilgili dokümanlar uygulama metotlarını sağlar.

Politika Esasları

Üç temel güvenlik amacına dayanır.

- 1-) Gizlilik : Hassas bilgilerin sadece yetkili kişiler tarafından okunduğundan emin olma ve yetkili olmayan kişilere bilgilerin ifşa edilmemesi.
- 2-) Bütünlük : Yazılım ve verilerin bozulmamasının sağlanması.

3-) Kullanılabilirlik : Sistem, ağ, uygulama ve verilere yetkili kullanıcılar istediklerinde erişip işlem yapabilmelerinin sağlanması.

Gizlilikte seviye isteyen farklı uygulama tipleri için, bütünlük ve kullanılabilirlik de değişecektir. Örneğin nükleer silah araştırma sistemi için yüksek seviyeli gizlilik ister fakat, birkaç saatlik sistemin kapalı tutulmasına dayanabilir. Genel bilgi verilen sistemlerde çok düşük seviyeli gizlilik yeterlidir, fakat yüksek seviyede bütünlük ister. Telefon anahtarlama sistemlerinde yüksek seviyeli kullanılabilirlik istenmektedir.

Ayrıca yukarıdaki amaçların tam olarak sağlanması için politika esaslarının aşağıdaki özellikleride içermesi gerekmektedir.

1-) Sorumluluk : Kimin neler yaptığını anlayabilme yeteneğidir.

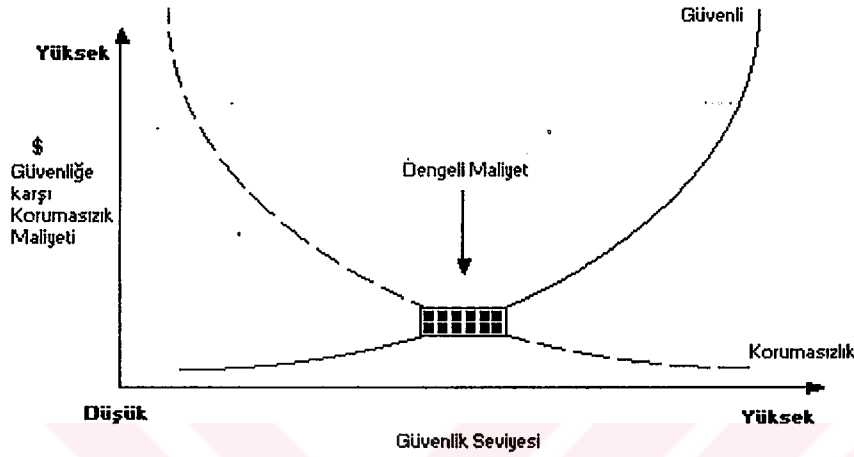
2-) Kaynak Kontrol : Bilgisayar ekipmanlarını kazalardan, doğal ve kasıtlı olabilecek zararlardan korumak ve yetkisiz kişilerin bu birimlere erişimini sınırlamak gerekir.

2.7.1.1. Risk Yönetimi

Politika tanımlarken, bilgi sistemlerinizin risk ve tehditlerini bilmeniz gerekir. Tehditler potansiyel güvenlik problemlerinin kaynağıdır. Tehditler insan ve doğal kaynaklıdır. Doğal kaynaklı olanlar yangın,su,deprem, insan kaynaklı olanlar ise kazara ya da kasıtlı olarak yapılan tehditlerdir. İnsan kaynaklı tehditlerin bazıları terör ve savaş durumları gibi toplumsaldır. Diğerleri de içerideki kullanıcılar ya da dışarıdaki saldırganlardan kaynaklanır. Son incelemelere göre içerideki kullanıcıların sistemlere saldırganlara (dışarıdan gelen) göre daha fazla zarar verdikleri görülmüştür. Buradan içerideki kullanıcıların büyük sistemlere erişimi sevdiklerini ve çok hassas aynı zamanda önemli bilgilerin nerelerde bulabileceklerini bildikleri söylenebilir.

Risk değerlendirmesi için bir metot “en kötü durum” senaryosuna göre bir maliyet saptanmaktadır.

Risk yönetimi, risk & korunmasızlık maliyetine karşı koruma maliyetini dengeleyen süreçtir. Şekil 2.4'te temel teori gösterilmiştir. Koruma ve korunmasızlık maliyetinin neredeyse aynı olduğu nokta gölgeli olarak gösterilmiştir. Bu alan dengelenmiş ve mantıklı güvenlik ölçülerini göstermektedir. Yoksa, güvenlik için gereğinden fazla harcama yapabilirsiniz ya da daha az harcama yaparak sisteminizi korumasız bırakarak riske atmış olursunuz.



Şekil 2.4. Güvenlik-Korunmasızlık Maliyetinin Dengelenmesi

Risklere karşı üç temel seçim vardır.

- 1-) Kabul : Eğer korunmasızlık çok küçük boyutlarda ve onu koruma altına almak büyük maliyet getiriyorsa, politikanız riski kabul edebilir.
- 2-) Devretme : Bazı durumlarda riske karşı direk olarak koruma almaktansa onun için birini görevlendirmek daha az maliyet getirebilir.
- 3-) Kaçınma :Güvenlik olaylarının neredeyse hiç olmayacağı yerleri korumaya almak maliyet getireceğinden bundan kaçınılmalıdır.

2.7.1.2. Güvenlik Sorumluluğu

Politika, güvenlik için kimlerin sorumlu olduğunu belirtmek zorundadır. Basit anlamda, bütün kullanıcılar ilk koruyucu aynı zamanda da en büyük tehdit olduklarından, “herkes” sorumludur. Politika bütün kullanıcılardan sahip oldukları

sistem ve veriler için, güvenlik bilgi ve sorumluluklarının anlaşılmasını ve sözleşme yapılarak imzalanmasını ister.

Sözleşmeyle ilgili konular;

- Kurumun sistem ve verileri vardır
- Kullanıcılar yetkilerinin olmadığı veri ve yazılım kopyalarına erişmemeyi kabul etmelidirler
- Kullanıcılar parolalarını uzun olarak seçmeli ve gizli yerlerde saklamayı kabul etmelidirler
- Kullanıcılar güvenlik için kurum haklarını koruduklarını bilmelidirler

Politika yayınlanmalı ve bütün kullanıcılara açık olmalıdır. Güvenlik politikasına sadakat çalışanın performans değerlendirmesinin bir parçası olmalıdır.

Bireyler güvenlik sorumluluğu ile ilgili olarak direk olarak görevlendirilirken sadece fonksiyon ve ünvanlarına bakılmalıdır. Genel olarak görevler parçalanabilir.

1-) Güvenlik yöneticileri güvenliği izlemekle sorumlu olmalıdırlar.

2-) Sistem yöneticileri ve BT müdürleri güvenliğin uygulanmasına yardımcı olmalıdırlar.

3-) Bilgi sistemleri hesap tutucuları peryodik olarak bilgi sistemlerinin güvenliğini gözlemlemelidirler. Böylece yönetim sağlanmış ve politika izlenmiş olur.

Organizasyonun yapısına ve büyüklüğüne bağlı olarak, güvenlik işleri değişebilir. Fakat politika bunu açıkça belirtmelidir. Kesinlikle aynı kişi güvenliği izleme, denetim ve uygulama işlerini yapmamalıdır. Böyle bir politikada kontrol ve denge olmaz.



2.7.1.3. İtaat

Politika kurallara uyumdan emin olmak için bir prosedüre belirtmelidir. Bu peryodik olarak izleme ya da otomatik olarak politika hatalarının sistem tarafından bildirilmesi demektir. Bir hata bulunduğu zaman, aşağıdaki aksiyonlar oluşabilir.

- 1-) Yetkili uyarılır
- 2-) Problem düzeltilebilir
- 3-) Olay kaydedilebilir
- 4-) Dikkate alınmayabilir

2.7.2. Güvenlik Politikası Yönetimi

2.7.2.1. Büyük Sistemlerin Güvenlik Yönetimi

Enterprise ağın gelişiyile birlikte güvenlik politikalarının tanımlanması ve yönetimi birbirlerine bağlı UNIX,PC ler, PC LANlar, Windows NT ve mainframe sistemler için yapılması zorunludur. Güvenlik prosedür ve kurallarının bilgisayar platformuna bakılmaksızın sağlanması gerekir.

2.7.2.2. Tanımlanmış Politikaya Karşı Gerçek Performansın Ölçülmesi

Bir yer için politika tanımlandıktan sonra onun izlenip izlenmediğinin belirlenmesi gerekir. Bir güvenlik problemi oluştuğunda tekrarlanabileceği için yetenekleri değerlendirilmelidir. Bir organizasyon sıkı bir güvenlik ile dünyanın en büyük saldırganlarına karşı korunabilir. Fakat çok pahalıya malolur.

Devamlı olarak, gerçek güvenlik performansının ölçülmesi ve daha önceden tanımlanmış hedef kriterlerle karşılaştırılması gerekir.

2.7.2.3. Politika Kategorileri

Bilgi güvenliği politikaları birkaç güvenlik kategorisi içinde gruplanabilir. Bu kategorilerin herbiri kullanıcı hesap ve yetkileri, ağ ve sunucu ayarları, dosya sistemi

ve izinler için bir grup güvenlik kontrolleri sunar. Bu kategoriler politika problemlerini belirlemenizi sağlar ve zayıflıkları, potansiyel güvenlik tehditlerini gösterir. Kategoriler;

1-) Kullanıcı Hesapları ve Yetkileri

Hesap doğruluğu, giriş parametreleri, parola yapısı, kullanıcı dosyaları.

2-) Ağ ve Sunucu Ayarları

Sistem denetçisi, sistem posta yöneticisi, ağ üniteleri, sistem kuyrukları, başlangıç dosyaları.

3-) Dosya Sistemleri Ve Klasörler

4-) Dosya erişimi, dosya özellikleri



3. SALDIRILAR VE GÜVENLİK YÖNTEMLERİ

3.1. Standart Internet Servisleri

İnternette standart hale gelmiş bazı servisler vardır. Bunların dışında, firmaların kendilerine özel servisleride olabilir. Bu bölümde çokça karşılaşılan standart servisler anlatılacaktır. Bu servislerin her birinin kendilerine özgün güvenlik delikleri vardır.

3.1.1. www

World Wide Web (www) tamamen internet için geliştirilmiş bir servistir. SMTP, FTP, telnet, usenet gibi standart servisleri de kullanarak HHTTP servisi üzerine inşa edilmiştir.

www, HHTTP sunucularının bir araya gelmesiyle oluşmaktadır. HTTP, kullanıcıların www üzerinde yer alan dosyalara ulaşmasını sağlamaktadır. HHTTP protokolü detaylı olarak ele alınacaktır.

www çok fazla servisi içermesi esnek olarak tasarlanmış olmasından kaynaklanmaktadır. Çok fazla servisi sunabilmesi aynı zamanda, bu servislerin de güvenlik problemlerini taşıması anlamına gelmektedir.

3.1.2. FTP

İnternet üzerinden dosya transferi sağlayan bir servistir. File Transfer Protocol (FTP), bu amaçla kullanılan standart bir protokoldür.

Dosya transferinde güvenliğin sağlanabilmesi için, dışarıdaki bir FTP sunucusuna bağlanan kullanıcının yüklediği dosyanın içeriğinin kontrol edilmesi ve yerel FTP sunucuya bağlanan kullanıcıların, sistemin diğer kısımlarına ulaşmalarının engellenmesi gerekmektedir.

Dosya transferi ile ilgili olarak kullanılan diğer protokoller;

1-) TFTP (Trivial File Transfer Protocol)

2-) UUCP (Unix To Unix Copy)

3-) FSP (File Service Protocol)

4-) rpc



3.1.3. SMTP

Simple Mail Transfer Protokol (SMTP), elektronik mektup göndermek ve almak için kullanılan standart bir protokoldür.

Elektronik posta en popüler servislerdendir. Bu servisle ilgili güvenlik problemleri bulunmaktadır. Çok karmaşık olmasa da posta yoluyla virüs gelebilmektedir.

3.1.4. Telnet, rlogin, rsh

Bu servisler uzak bir sisteme bağlanmayı sağlar. Sisteme doğrudan bağlı bir terminal gibi çalışmanızı sağlayan bir servistir.

Telnet, uzaktaki sistemlere bağlanma servisi olarak, standart hale gelmiş çok kullanılan bir protokoldür. Kullanıcıların sisteme bağlanabilmeleri için kullanıcı adı ve parolası istediği için güvenli sayılabilir. Fakat, sunucu ile istemci arasındaki iletişim açık olarak yapılmaktadır. Hattı dinleyen birisi kullanıcı adı ve parolası başta olmak üzere ne tür işlemler gerçekleştirildiğini gözleyebilir. Bu güvenlik açısından büyük bir problemdir.

rlogin ve rsh uzak sistemlere erişim ve komut yürütme işlemlerini sağlayacak diğer servislerdir. Güvenlik önemli olduğunda internet üzerinden bu servislerin kullanımı oldukça zordur.

3.1.5. Usenet News

SMTP bire bir ya da birden çoğa haberleşme sunarken, usenet news çokça çok bir haberleşme ortamını sağlayan bir servistir. Haber gruplarından yararlanmak için kullanılır. Haber grupları, sayıları çok fazla olan kullanıcıların, her kullanıcının yazdığı mektubu birlikte okuyabilmeyi sağlayan bir servistir.

Bu servisle ilgili olarak karşılaşılabilecek güvenlik sorunları, elektronik posta için geçerli olanlar burada da geçerlidir.

3.1.6. DNS

Domain Name Server (DNS), bilgisayar isimlerini IP adresine dönüştüren servistir. Kullanıcılar doğudan kullanmazlar. SMTP, FTP ve Telnet gibi servisler tarafından kullanılır.

3.1.7. NFS

Bazı protokoller vasıtasıyla başka bilgisayarlar üzerindeki sabit sürücüdeki dosyalara ulaşmak mümkündür. Network File System(NFS), bu tür hizmet verilebilecek bir servistir. Bu servis, kendisine ulaşan bilgisayarlara güvendiğinden herhangi bir kullanıcı tanıma mekanizması kullanmaz. Bu nedenle de güvenlik konusunda dikkatli olmak gerekir. Özellikle internet üzerinden bu servis verilecekse, değişik güvenlik sistemleriyle birlikte kullanılması gerekmektedir.

Aynı işlevi yerine getiren başka bir serviste Andrew File System (AFS) dir. Bu servisin kullanıcı tanıma mekanizması ve eğer isteniyorsa bilgilerin aktarımında şifreleme işlemine imkan vermesi, NFS'e göre avantajlı duruma sokmaktadır.

3.1.8. Bilgi Tabanlı Diğer Servisler

Internet üzerinde bilgi temelli olan www dışındaki diğer servisler arasında şunlar da sayılabilir. Gopher, WAIS, Archie .

Gopher, karakter tabanlı ve menüler şeklinde dizayn edilmiş, internet üzerinde bilgi araştırmasına yardım eden bir servistir.

Wide Area Information Service (WAIS), kullanıcının vereceği anahtar kelimeleri kullanarak internet üzerinde sorgulama yapan ve bu anahtar kelimelerin bulunduğu sayfaları kullanıcıya bir liste olarak sunan servistir.

Archi, verilen bir dosya veya dizin adını, genel kullanıma açık ftp sunucularında arayan bir servistir. Özel istemci Archie programlarının yanında bu servisten yararlanmak için Telnet veya elektronik posta da kullanılabilir.

3.2. Bilgisayar Ağına Saldırı

Internet'in doğuşu ve gelişimi arasında aslında çok kısa bir zaman aralığı vardır. Bu kadar hızlı bir büyümenin olabileceği internetin doğuş yıllarında beklenmiyordu. Özellikle 1985 yılından sonra büyük yatırımlar yapıldı ve hızla yaygınlaştı. Ama bu hızlı gelişim birtakım konuların standartlarının tam olarak oluşturulmadan kullanıma geçirilmesinden dolayı bazı sorunları beraberinde getirir. Özellikle de güvenlik sorunlarını ortaya çıkardı.

Güvenlik hemen her bilgisayar ağında öncelikli olarak düşünülmesi gereken bir konudur. Internet ortamında ise çok daha önemlidir. Birçok ticari firma ya da başka kuruluşlar ürünlerini ve hizmetlerini internet ortamına aktarmak istemektedirler. Ancak bu birtakım risklerin de alınması gerektiği anlamına da gelir. Değişik güvenlik mekanizmalarının bir arada kullanılmasıyla bu riski azaltmak mümkündür.

3.2.1. Saldırganlar

Saldırgan (Hacker), ağ üzerindeki genelde bazı servisler veren makinalara hiçbir hakkı olmadan erişip zarar veren kişidir. Bilgi hırsızı olarak da tarif edilebilir. Fakat ev ya da banka soyguncularından çok farklıdır. İyi görünümlü sistemler hakkında çok bilgisi olan insanlardır. Genellikle sistemin bilinen açıklıklarından ve sistem yöneticisinin bilgisizliğinden faydalanırlar.

İstatistiki raporlara göre saldırıların çoğunun firma içerisinde yapıldığı tespit edilmiştir. İçeriden gelen saldırı, sistem sadece dışarıya karşı korunmalı durumda ise çok zarar verebilir.

Saldırganların büyük firmaların ağına girdikleri ve büyük ölçekte sistemlere zarar verdikleri bilinmektedir. Bunu genellikle eğlence, kendini göstermek ya da sisteme zarar vermek için yaparlar.

Saldırganları iki genel türde toplanabilir:

1-) Kötü niyetli saldırganlar

2-) Kötü niyetli olmayan saldırganlar

3.2.1.1. Kötü Niyetli Saldırganlar (Malicious hacker)

Sisteme gerçekten zarar vermek amacıyla girerler. Açığını buldukları sistemlere verebilecekleri en büyük zararı vermeyi amaçlarlar. Bu tür saldırganlar genelde ekip halinde çalışırlar. Kredi kartları kullanan sitelerden kart numaralarını ve parolalarını alıp kişi ve şirketlere büyük zararlar verebilirler.

Tablo 3.1’de belirtilen bilgisayar korsanları, casuslar, teröristler ve profesyonel suçlular bu gruba girmektedir.

3.2.1.2. Kötü Niyetli Olmayan Saldırganlar

Sistemlere genellikle eğlenmek için saldırıda bulunurlar. Çok fazla zararlı olmayan tiplerdir. Hatta sistem yöneticisi eksikliklerini ve sistemin zayıf noktalarını bu sayede görebilir. Bu tür saldırganlar, bir saldırgan grubuna üye değildir. Genellikle yaptıkları, kendilerine daha sonra kullanmak için hesap açmak ve sistemin zayıf gördüğü yerleri belirten notlar koymaktır. Bu işi zevk için yapan insanları bu kategoriye sokabiliriz. Bu gruba girenler tablo 3.1’de meraklılar olarak belirtilmiştir.

Saldırgan, kullandığı araçlar, sisteme erişim yolları ve amaçlarının ne olabilecekleri tablo 3.1’de özetlenmiştir.

Tablo 3.1. Saldırganlar Ve Amaçları



Saldırganlar	Araçlar	Erişim	Sonuç	Amaç
Bilgisayar korsanları	Kullanıcı komutları	Gerçekleme zayıflıkları	Bilgi bozma	Politik kazanç
Casuslar	Komut dosyası veya Program	Tasarım zayıflıkları	Bilgi çalma ya da açığa bilgi çıkartma	Finansal kazanç
Teröristler	Araç takımı	Yapılandırma zayıflıkları	Hizmet çalma	Zevk için
Meraklılar	Dağıtık araçlar	İzinsiz erişim	Hizmet önleme	Sosyal statüye meydan okuma
Profesyonel suçlular	Veri dinleyici sistemler			

3.2.2. İşletim Sistemlerinin Güvenliği

İşletim sistemi denildiğinde akla ilk gelenler UNIX ve Windows türevi sistemlerdir. İşletim sistemi seçilirken;

- 1-) Kurulum kolaylığı
- 2-) Donanım gereksinimleri, sürücü edinebilme
- 3-) Kullanım ve yönetim
- 4-) Güvenilirlik
- 5-) Güvenlik
- 6-) Uyumluluk
- 7-) Fiyat
- 8-) Destek

gibi özelliklere bakılarak seçilir. Bu özelliklerden güvenlik eğer sistem ağı açılacaksa çok büyük önem kazanacaktır. İşletim sistemlerinin güvenilirliği sürekli olarak tartışılmakta ve çıkan her yeni sürümde güvenlik delikleri kapatılmaktadır. Burada Windows NT ile UNIX işletim sistemlerinin güvenlik üzerine sundukları teknolojilerden kısaca bahsedilecektir.

Windows NT'nin Güvenlik Bileşenleri;

Giriş süreçleri (Logon Process): Kullanıcıların giriş isteklerini kabul eder. Kullanıcının ismi ve parolası kontrol edildikten sonra tanımlanan haklara göre hareket etmesini sağlayan sistemdir. Yakından ya da ağ üzerinden giriş yapılabilir.

Yerel Güvenlik makamı (Local Security Authority): kullanıcının sisteme erişim iznini denetler. Bu bileşen güvenlik alt sisteminin çekirdeğidir. Erişim jetonlarını üretir, yerel güvenlik prensiplerini yönetir ve etkileşimli kullanıcı onaylama hizmeti hizmetlerini sağlar. Yerel güvenlik makamı aynı zamanda kayıt denetimi prensiplerini de denetler ve güvenlik kayıt mesajlarını kaydeder.

Güvenlik hesap yöneticisi (SAM): Kullanıcı hesapları veritabanına bakarak kullanıcı yönetimini sağlar. Veritabanında tüm kullanıcı ve grupların hesapları vardır. SAM yerel güvenlik makamı tarafından kullanılan kullanıcı geçerli kılma hizmetlerini sağlar.

Güvenlik Başvuru İzleyicisi (Security Reference Monitor) : Kullanıcının bir nesneye erişim izninin olup olmadığının kontrolü ve yaptığı işlemleri denetler. Bu bileşen SAM tarafından tanımlanmış olan erişim iznini geçerli kılma ve kayıt hesabı üretim prensiplerini gerçekleştirir. Hem çekirdek hem de kullanıcı kiplerine, bir nesneye erişmek isteyen kullanıcı ve süreçlerin gerekli izinlerinin olduğunu denetleyen hizmetler sunar. Bu bileşen gerektiğinde kayıt hesabı mesajları da üretir.

Unix'te NT gibi sistem yöneticisine dayalı olarak kullanıcı hesap prensipleri belirleyebilmesi için birtakım özellikler sağlar. UNIX'te kullanıcı adları hangi gruba ait oldukları /etc/passwd dosyasında tutulur. NT'de ise kullanıcı bilgileri kayıt dosyalarında tutulur. Kayıt dosyaları(registry), sadece öncelikli çekirdek rutinleri tarafından erişilebilen, korunan ve şifrelenmiş veritabanıdır.

3.2.3. Saldırı Türleri

Bu bölümde son yıllarda internette kullanılan saldırı yöntemlerine değinilecek ve sınıflandırılmaya çalışılacaktır. Saldırıların tanımları yapılacak ve sisteme verebileceği zararlar üzerinde durulacaktır. Saldırıya karşı alınabilecek önlemler güvenlik duvarı olmaksızın anlatılacak ve güvenlik duvarı düzeyinde yapılabilecekler açıklanacaktır.

Saldırganlar sisteme ağ üzerinden ulaşabilecekleri için, ağa bağlı cihazlar her zaman saldırıya açık durumdadır. Burada yapacakları, hedef makinaya ulaşmak, yazılım ve donanıma zarar vermek şeklinde olabilir. Şirkete ait veritabanına ulaşip verilere erişebilir, değiştirebilir ya da silebilirler. Burada asıl olan, saldırganın ne yapmak istediğidir. İşine yarayan kayıtları, dosyaları alabilir ve sisteme (yazılım, donanım) zarar verebilir. Verilen hizmetleri servis dışı bırakabilir. Sadece internet bağlantısına zarar verebilir. Truva atı türünde programları bir şekilde hedef makinaya yükleyerek kullanıcıyı takip edebilir.

Güvenlik duvarı olarak linux işletim sistemi çekirdeği içerisinde bulunan ip_fw (ip_fw.c) programını ve mimari olarak da çift arayüzlü güvenlik duvarı göz önünde bulundurularak çözümler anlatılacaktır. Bu güvenlik duvarının kullanıcı düzeyinde konfigürasyonunun yapılabilmesi ve yönetilmesi için ipfwadm programı vardır.

3.2.3.1. Saldırıların Sınıflandırılması

Bilgisayar ve ağ saldırıları için değişik sınıflandırmalar yapılmıştır. Aşağıda süreçsel ve işlemsel sınıflandırmalar anlatılacaktır.

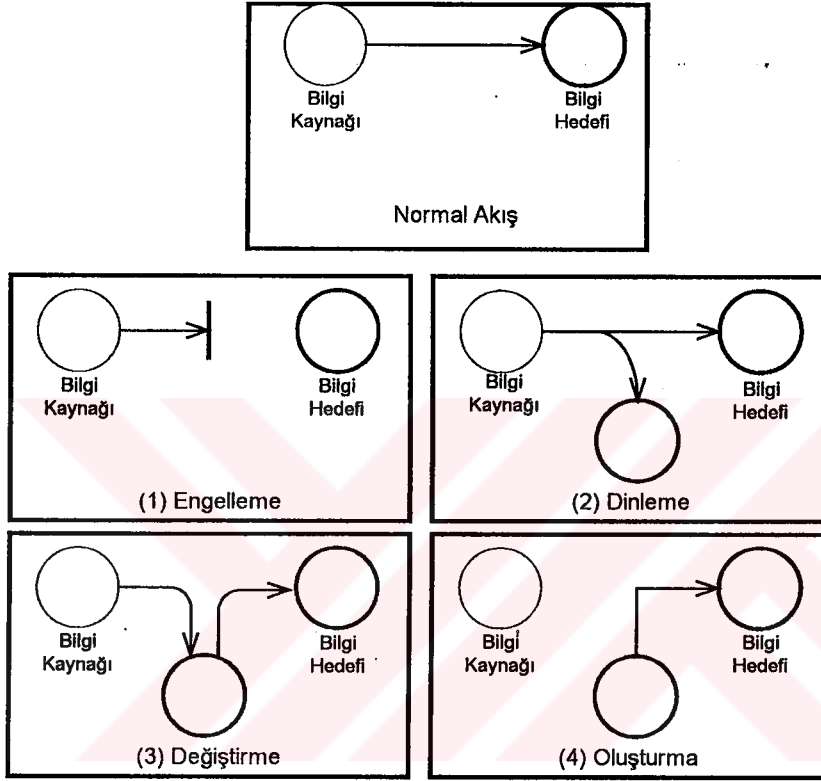
3.2.3.1.1. Süreçsel Sınıflandırma

Internette gerçekleştirilen veri transferiyle ilgili güvenlik sorunları dört kategoriye sokulabilir.

1. Engelleme : Sistemin bir kaynağı yok edilir veya kullanılamaz hale getirilir.
2. Dinleme : İzin verilmemiş bir taraf bir kaynağa erişim elde eder.

3. Değiştirme : İzin verilmemiş bir taraf bir kaynağa erişmenin yanı sıra üzerinde değişiklik de yapar.
4. Oluşturma : İzin verilmemiş bir taraf, sisteme yeni nesneler ekler.

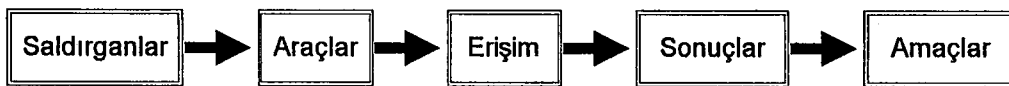
Burada dinleme pasif bir saldırı türüdür. Engelleme, değiştirme ve oluşturma ise etkin bir saldırı türü olarak görülmektedir.



Şekil 3.1. Süreçsel Sınıflandırma

3.2.3.1.2. İşlemsel Sınıflandırma

Genel anlamda bir saldırı yöntemler, kullanılan yollar ve sonuçları açısından düşünülebilir. Bilgisayar ya da bilgisayar ağına saldıran kişi, istediği sonuçlara çeşitli adımlardan geçerek ulaşmak zorundadır. Bu adımlar şekil 3.2’de de görüldüğü gibi araçlar, erişim ve sonuçlar şeklindedir.



Şekil 3.2. İşlemsel Sınıflandırma

3.2.3.2. Saldırıların Gruplandırılması

Saldırıları değişik açılardan değerlendirerek, değişik gruplamalar yapmak mümkündür. Saldırıda kullanılan yöntem, saldırının kullandığı yöntem, saldırının amaçladığı uygulamalar ve saldırı sonucunda oluşan zararlar gruplama için kullanılabilecek yöntemlerdir.

3.2.3.2.1. CERT Gruplandırması

(Computer Emergency Response Team) tarafından yapılmış olan saldırı türleri ve açıklamaları tablo 3.2’de listelenmiştir.

Tablo 3.2. CERT Gruplandırması

Probe,Scan,Scam	Bir sistemdeki açık ve kullanılan portların taranması ve bu portlardan hizmetlere yönelik saldırıları türüdür.
Prank	Kullanıcı hesaplarının yanlış oluşturulması sonucu oluşan açıklardan yapılan saldırı türleridir.
Email spoofing	Başka bir kullanıcı adına e-posta gönderilmesi...
Email bombardment	Bir e-posta adresine genelde farklı adresten çok sayıda e-posta gönderilmesi...
Sendmail attack	Smtip portuna yönelik saldırılardır...
Break-in	Verilen hizmetlerin devre dışı bırakılmasına yönelik saldırı türüdür.
Intruder gained root access	Saldırganın normal kullanıcı olarak girdiği sistemde süper kullanıcı yetkisini kazanması.
Intruder installed trojan horse program	Saldırganın girdiği sisteme genelde daha sonra tekrar rahat girebilmesi ya da uzaktan yönetim için ajan program yerleştirmesi.

Intruder installed packed sniffer	Saldırgan tarafından hedef makinaya yönelik paket dinleyici yerleştirilerek yapılan saldırı türüdür. Bu şekilde bir yerel ağ korumasız bir konak üzerinden saldırılara açık hale gelebilir.
NIS attack	Ağ kullanıcı yönetim sistemine yönelik saldırı türüdür.
NFS attack	Ağ dosya yapısına yönelik saldırı türüdür. Genellikle ağ erişimini devre dışı bırakmada kullanılır.
Telnet attack	Uzaktan erişim protokolünün açıklarından faydalanılarak yapılan saldırı türüdür.
Rlogin or rsh attack	Uzaktan erişimde kullanılan servislerin açıklarına yönelik yapılan saldırı türüdür.
Cracked password	Kolay tahmin edilebilir parolaların tahmini ya da şifreli hallerine göre sözlük saldırısı yapma türüdür.
Anonymous FTP abuse	Anonim erişim izni verilen dosya aktarım sunucularına yönelik saldırılardır.
IP spoofing	IP adres yanıltmasıyla yapılan saldırı türüdür.
Configuration error	Çok kullanılan programdaki kullanıcılardan kaynaklanan konfigürasyon hatalarından doğan açıklıklardır.
Misuse of hosts resources	Konak kaynaklarının yanlış kullanımı sonucu ortaya çıkan açıklıklar.
Worm, Virus	Konaklarda kullanıcılardan habersiz çalışan zararlı programlar.
...	

Yukarıda da belirtildiği gibi değişik gruplamalara gitmek mümkündür. Bazı saldırıları sadece bir gruba koymak mümkün değildir.

www.cert.org adresinde bu gruplandırılmış saldırıların ne kadar süredir var olduğu, ne tür zararlara yol açtığı ve yaygınlığı gibi bilgiler bulunmaktadır.

3.2.3.2.2. İletişim Protokollerini Kullanan Saldırıları

IP sahteciliği (IP spoofing)

TCP dizi numarası saldırısı (TCP sequence number attack)

ICMP atakları

Ölümöl ping

TCP SYN seli atağı (TCP SYN Flood Attack)

IP parçalama saldırısı (IP Fragmentation Attack)

İnternet yönlendirme saldırısı (Internet Routing Attacks)

UDP sahteciliği ve Dinleme (UDP Spoofing and Sniffing)

UDP portunu servis dışı bırakma saldırısı (UDP Port Denial-of-Service Attack)

Rasgele port taraması (Random port scanning)

ARP saldırıları (ARP Attacks)

Ortakdaki adam saldırıları

3.2.3.2.3. IP saldırıları

IPv4'te bulunan güvenlik eksikliklerinden faydanılarak yapılan atak türleridir. Bazı örnekleri şunlardır ; Out of Band Nuke, Land, Teardrop, Boink, Nestea, Brkill, ICMP Nuke, Jolt/Ssping, Smurf, Suffer3

3.2.3.2.4. İşletim Sistemine Özel Saldırıları

Exploit olarak isimlendirilen bu saldırılar, sistem tabanlı olarak çalışırlar. Yani Unix için yapılan bir exploit MS Windows için çalışmaz.

Windows Null Session Exploit

PHF Exploit

ASP Exploit

Sendmail Exploit

3.2.3.2.5. Uygulama Katmanı Saldırıları

DNS,SMTP,MIME,NFS,NTP saldırıları

Uzaktan giriş ile saldırılar (Hacking and Remote Login)

Bilgi sızdıran saldırılar

URL sahteciliği (URL Spoofing)

CGI saldırıları

X-Windows sistem saldırıları

Kötü niyetli java ve AktiveX uygulamaları

Sistem log seli

Program ve ağ üstündeki virüsler

Sistem log seli güvenlik duvarına çok sayıda giriş yapılarak log dosyasının dolmasına ve sonuç olarak sistemin kapanmasına neden olan saldırı türüdür.

3.2.3.3. Bazı Saldırı Yöntemleri

Artık klasik hale gelmiş ve genelde iletişim protokollerinin deliklerinden yararlanılarak yapılan saldırılar anlatılacaktır.

3.2.3.3.1. IP Aldatması (IP Spoofing)

Aldatma saldırısının genelde yapılan türü IP aldatmasıdır. Sahtecilik olarak çevirmek mümkündür. IP paketlerinin kaynak IP' sini değiştirmekle sağlanmaktadır. Böylece paketi alan hostun, paketin geldiği kaynak adresini bilmesini engellenmiş olur. Host gelen paketin saldırgandan değil de kullanıcıdan geldiğini sanır.

IP adreslerine göre çalışan servisler üzerinde oldukça etkilidir. Sahte paketler üretme (fabrication) bu şekilde yapılabilmektedir.

İnternete bağlanan bilgisayarlar birbirleriyle IP adreslerini kullanarak haberleşmektedir. IP paketleri üretildiği bilgisayardan hedef bilgisayara IP adreslerini kullanarak giderler. İletişimde bulunan bilgisayarın IP numarasının bilinmesi güvenlik açısından oldukça önemlidir. İnternette servis veren bazı bilgisayarlar sadece belirli IP adreslerine hizmet verirler. Güvenlik duvarlarının kullandığı paket filtreleme mekanizması genelde bu adresler üzerinde yapılır. Paket filtreleme örneğinde de anlatıldığı gibi bazı IP adreslerine ait paketlerinin geçişine izin verilir ya da reddedilir.

Ayrıca güvenlik duvarı kullanılarak, bazı IP adreslerine verilecek hizmetlerin sınırlandırması da sağlanabilir. Gelen isteğin kaynak IP adresine bakılarak böyle bir işlem yapılır. Bunun anlamı kullanıcıya IP adresine göre yetki vermektir. IP adresine bakılarak yetki verilmesi çalışmaları rahatlatacaktır; fakat güvenlik açısından bazı riskler taşımaktadır.

Kaynak IP adresine göre yüksek yetki verilen bir kullanıcıya ait adres bilinmesi durumunda risk ortaya çıkmış olur. Saldırgan kaynak IP adresini değiştirmek suretiyle hakkı olmayan bir yetkiyle sisteme girebilir.

Saldırganın sahte IP paketleri kullanarak sisteme girmesi durumunda verebileceği zarar, IP adresine verilen yetkiyle orantılıdır. Bu şekilde yetkili

kullanıcı olarak sisteme girebilen bir saldırgan sisteme çok kolay ve oldukça fazla zarar verebilir.

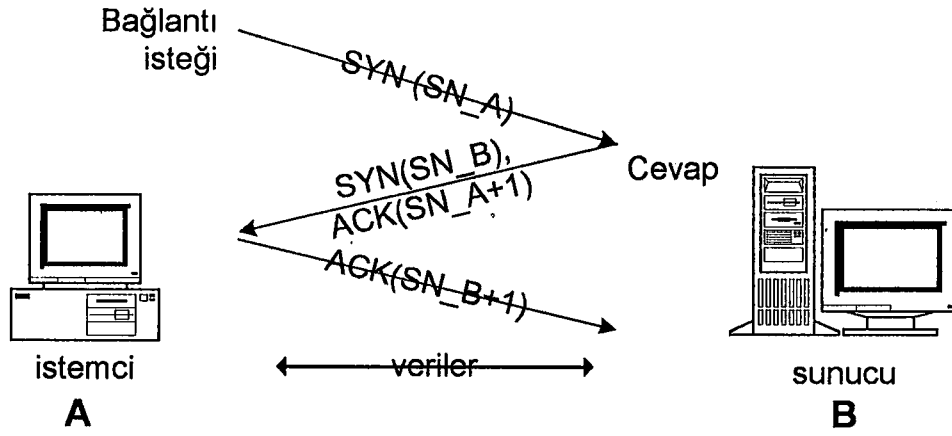
Böyle bir saldırıya karşı alınabilecek önlem yoktur. Ancak Kaynak IP yanında Hedef IP ve MAC adresine de bakılarak saldırı olduğuna karar verilebilir. Bunu iç ağın internete açıldığı noktaya güvenlik duvarı kurmakla sağlamak mümkündür. [6]

3.2.3.3.2. TCP SYN paketi akışı saldırıları

IP adresi kullanılarak gerçekleştirilen bir saldırı türüdür. Genelde TCP/IP servislerinin devre dışı bırakmak için kullanılır.

TCP bağlantı temelli bir protokoldür. Birbiriyle iletişim kuran iki bilgisayar, paketlerini kurulu bir hat üzerinden aktarırlar. Bunun için de iletişim başlangıç aşamasında bu hattın kurulması gerekir. Bu da bazı el sıkışma kurallarıyla mümkün olmaktadır. TCP protokolünün el sıkışma sistemi üç yollu el sıkışma (three way handshake) mekanizmasıdır. Üç yol denmesi, bu işlemin üç aşamada gerçekleşmesinden dolayıdır.

İlk olarak, bir sunucu bilgisayardan servis isteyecek diğer bilgisayar (istemci), bağlantı kurmak istediğini göstermek için SYN bayrağı kalkık(set) paketini, sunucu bilgisayara gönderir. Bu paketi alan sunucu SYN paketi aldığını ve bağlantı isteğini onayladığını yine SYN bayrağı kaldırılmış(set) paketi (SYN-ACK paketi) istemci bilgisayara gönderir. Üçüncü aşamada ise sunucu bilgisayarın gönderdiği SYN-ACK paketini alan istemci bilgisayar sunucuya bu paketi aldığını bildiren bir paket gönderir(ACK).



Şekil 3.3. TCP El Sıkışma Mekanizması

Bu mekanizmada saldırıya açık nokta, bir bilgisayarın istemci olarak bir bilgisayara el sıkışma mekanizmasını başlatacak SYN paketini göndermesi, buna karşılık sunucu bilgisayardan SYN-ACK paketini aldıktan sonra son paketi (ACK) göndermemesi ile oluşur. Böyle bir durumda sunucu tarafında açılmış fakat tamamlanmamış bir bağlantı isteği oluşacaktır. Bu bağlantı isteği uzun bir süre dolumuna kadar açık tutulacaktır. Bu tür bağlantılara yarı açık bağlantılar denir. Bu mekanizmanın kötüye kullanılabilecek noktası, arka arkaya belirtilen bu işlemlerin yapılmasıyla ortaya çıkar. Böyle bir durumda yarı açık birçok bağlantı oluşacaktır.

Herhangi bir hizmet bir port üzerinden verilmektedir. Buraya gelen istekler bir kuyruğa alınır ve istemciden üç yollu bağlantıdaki son paket gelene kadar kuyrukta tutulur. Dolayısıyla kuyruktaki tamamlanmayan bağlantı isteklerinin artması kuyruğun dolmasına sebep olacaktır. Kuyruk dolduğunda ise yeni gelen bağlantı isteklerine cevap veremeyecektir. Dolayısıyla servis kilitlenecektir. Bazı sistemlerde bellek taşmasına sebep olacağından bu durum sunucu bilgisayarı devre dışı da bırakabilir.

Bu saldırıyı önlemek için TCP protokolü üzerinde açıklıklarını kapatacak değişiklikler yapılmaktadır. Bu değişiklikler paket filtreleme düzeyinin üstünde TCP protokolü düzeyinde yapılmaktadır.

3.2.3.3.3. TCP sıra numarası tahmini yoluyla gerçekleştirilen saldırılar

Daha önce IP adresi yoluyla yanıltma yöntemi anlatılırken, IP adresine göre yetkilendirme yapan servislere karşı bu yöntemin kullanılması durumunda sistemin saldırıya karşı olduğundan bahsedilmişti. IP adresi yoluyla yanıltma tekniğinin birlikte kullanıldığı yöntemlerden biri TCP sıra numarası tahmini yoluyla yapılan saldırıdır. Bu TCP/IP protokolünün açıklıklarından olarak kabul edilir. IP adresine göre yetkilendirme yapan programlara saldırılarda bu yöntem kullanılır.

Bu saldırıyı anlamak için TCP protokolündeki üç yollu el sıkışma mekanizmasını detaylı incelemek gerekmektedir. TCP protokolü daha öncede bahsedildiği gibi bağlantı temelli bir protokoldür. Bunun anlamı bir oturuma ilişkin IP paketlerinin aktarılması sırasında verilen güvendir. Hedef uca uygun sırada ulaşmamaları durumunda, düzenlenmesi ve aradan paketlerin gelmemesi durumunda da tekrar o paketin istenmesi TCP katmanında yapılmaktadır. Bunları gerçekleştirmek için bazı mekanizmalar kullanılır. TCP sıra numarası (TCP sequence number), gönderilen paketlerin oturum içindeki sırasını gösterir. Bu değer kullanarak doğru sırada alınmamış paketlerin sıraya konulması ve en son alınan pakete ilişkin bilgilendirme karşı tarafa yapılmaktadır. Bu sayede paketin hedefe ulaştığından emin olunması sağlanmış olur; aksi durumda paketin tekrar gönderilmesi işlemi yapılır.

Saldırıya açık kısmı ise, TCP üç yollu bağlantı şekli üzerinden açıklayalım. İstemci bilgisayar sunucu bilgisayara istekte bulunacağını bir SYN paketi göndererek ifade etmektedir. Bu pakette J ile gösterilen, istemci bilgisayarın gönderdiği bilgilere verdiği sıra numarası bilgisidir. Bu paketi alan sunucu bilgisayar cevap olarak SYN-ACK paketi göndermektedir. Bu pakette de, K ile gösterilen, sunucu bilgisayarın gönderdiği pakete verdiği sıra numarası vardır. SYN(K) ile gösterebileceğimiz bu bilgiyle birlikte, haberleşmenin senkronize bir şekilde devam etmesini sağlamak üzere istemci tarafından gönderilen ve J sıra numarasını taşıyan paketin alındığına ilişkin bilgilendirmenin de yapılması gerekir. Bu amaçla da ACK(J+1) ile gösterebileceğimiz ve en son alınmış olan pakete ilişkin sıra numarası bilgisi de ACK olarak istemci bilgisayara gönderilir. Bu paketi alan istemci bilgisayar da, sunucunun göndermiş olduğu son paketi aldığını göstermek üzere ACK(K+1)

paketini sunucuya göndererek, bağlantının kurulması işlemini tamamlar. Bağlantının kurulabilmesi ve daha sonra da devam edebilmesi için bu bilgilendirme ACK paketlerinin karşılıklı olarak aktarılması şarttır.

Saldırı yöntemi aşağıdaki örnekle açıklanmıştır

A,B ve C bilgisayarlarından oluşan bir sistem olsun. A bilgisayarının IP adresi yoluyla aldatma yöntemini kullanmak suretiyle kendisini C bilgisayarındaymış gibi göstererek, sunucu olan B bilgisayarına bağlanmaya çalıştığını düşünelim ve üç yollu el sıkışma mekanizmasında gerçekleşecekleri inceleyelim;

A bilgisayarı ilk SYN paketini gönderme konusunda bir zorluk yaşamayacaktır. Bu pakete kendi sıra numarasını (SN_A) ekleyerek ve C bilgisayarından geliyormuş gibi göstermek üzere 'kaynak adres' kısmını değiştirerek B bilgisayarına gönderecektir. Bu paketi alan B sunucu bilgisayarı, SYN(SN_B) ACK (SN_A+1) bilgilerini taşıyan paketi, kendisiyle haberleşmek istediğini düşündüğü C bilgisayarına gönderecektir. Üç yollu el sıkışma mekanizmasının tamamlanabilmesi için son paket olan ACK(SN_B+1) paketinin B bilgisayarına gönderilmesi gerekir. Oysa SN_B paket numarası bilgisini içeren paket, üç yollu el sıkışma mekanizmasını başlatan ve kendisini C bilgisayarı gibi gösteren A bilgisayarına değil de C bilgisayarına gönderilmiştir. Dolayısıyla A bilgisayarının kendisi kendisini C bilgisayarı gibi gösteren B sunucusuna bağlantıyı tamamlayabilmesi için, SN_B değerini belirlemesi ve ACK (SN_B+1) paketini yine C bilgisayarından geliyormuş gibi göstererek B bilgisayarına göndermesi gerekmektedir. Bunu başarabildiği anda üç yollu el sıkışma mekanizması tamamlanacak ve B sunucu bilgisayarı C bilgisayarı ile haberleştiğini düşünecektir.

Eğer bu saldırı, IP adresi ile yetkilendirme yapan bir servise yapılıyorsa, B sunucusunun C'ye güvendiği ve ona verdiği yetki oranında, A bilgisayarı B üzerinde işlemler yapacaktır.

A bilgisayarının B sunucusuna C bilgisayarımış gibi bağlanma senaryosu tablo 3.3'te özetlenmiştir;

Tablo 3.3. TCP El Sıkışma Mekanizması

Paket Üretici	Kaynak Adres	Hedef Adres	Paket
A	C	B	SYN(SN_A)
B	B	C	SYN(SN_B), ACK(SN_A+1)
A	C	B	ACK(SN_B+1)

Bundan sonra karşılıklı bilgi akışı başlar.

Görüldüğü gibi saldırının en kritik noktası, A bilgisayarının SN_B' yi belirlemesidir. Eğer A bu değeri doğru olarak belirleyebilirse, saldırıya başlangıç için büyük oranda başarılı olmuş demektir. Sıra numarası bilgisinin ne şekilde verildiğini incelememiz, bu konuya da açıklık getirecektir. Bir bağlantıda kullanılacak sıra numarası bilgisi için bir başlangıç değeri belirlenir; daha sonradan da her veri aktarımında bu değer, aktarılan veri miktarı kadar artırılır. Başlangıç değeri belirlenirken de, bu işlem için kullanılan, 32 bitlik sayıcıdaki değer kullanılır. Bu değer belli aralıklar ve durumlarda artırılarak düzenli bir çalışmanın yapılması garanti edilmeye çalışılır. Her saniye için belli bir sayı ve her gerçekleşen bağlantı içinde bir başka sayı sayıcıya eklenir.

Acaba böyle düzenlenen bir sayıcıdaki değer, dolayısıyla gerçekleşecek yeni bir bağlantıya verilecek sıra numarasının (senaryoda SN_B idi), başka bilgisayarlarca (örnekte A) belirlenebilmesi, sayıcının düzenlenmesindeki algoritmanın ne kadar iyi olduğuna bağlıdır. Bu algoritmayla sıra numarasının tahmini kolaylaşmaktadır. Bunu yapacak saldırgan, hedef bilgisayara önce normal bir bağlantı gerçekleştirecek, hedef bilgisayardaki sıra numarası sayıcısının o andaki değerini belirleyecektir. Sonrasında da, sayıcının artış algoritmasını, hedef bilgisayarla aradaki mesafeyi ve diğer bazı unsurları göz önüne alarak, bir süre sonra kullanılacak sıra numarasını tahmin edebilecektir.

Bu tahminler yapılabildiği takdirde saldırı için gerekli zemin hazırlanmış demektir. Hedef bilgisayara güvenilir bir bilgisayar gibi ulaşılmış demektir.



Burada dikkat edilmesi gereken bir nokta daha vardır. Bağlantı kurulumu aşamasında B sunucu bilgisayarının, C bilgisayarına SYN (SN_B) ACK(SN_A+1) paketlerini göndermesi durumunda nelerin olacağının da göz önüne alınması gerekir. Böyle bir durumda, paketi C bilgisayarı algılasa bağlantı isteğinde bulunmadığını gösteren RST paketini B bilgisayarına gönderecektir. Bu durumda B sunucu bilgisayarı üç yollu bağlantı işlemini kesecektir. Bu nedenle A bilgisayarındaki saldırgan ya C bilgisayarının devrede olmadığı bir anı gözleyecek ya da B bilgisayarının göndermiş olduğu paketin, C bilgisayarı üzerinden gideceği servise ilişkin portun SYN kuyruğunu doldurarak, bu paketin C bilgisayarı tarafından alınamamasını sağlayacaktır. Dolayısıyla C bilgisayarı cevap vermeyecektir.

Saldırıya ilişkin en kritik nokta, sıra numarasının tahmin edilebilir olmasıdır. Bu sayının tahmin edilemez hale getirilmesi durumunda saldırılar önlenecektir. Bunun için de sıra numarasını belirleyen sayıcının algoritmanın güçlendirilmesi gerekir.

3.2.3.3.4. IP Servis Durdurma Saldırısı (smurf)

Yanlış kaynak adresi bilgisiyle oluşturulmuş ICMP ‘echo request’ paketleri kullanılarak gerçekleştirilen, çoğu durumda hedef bilgisayarın kilitlenmesine sebep olan, ayrıca saldırıda hedef olarak kullanılan ağlarda önemli derecede performans sorunları yaratabilen bir saldırdır. “smurf” adının verilmiş olması, saldırganların bu işlevleri yapan programlardan birisinin smurf olmasından dolayıdır.

Saldırı iki işlemten oluşmaktadır. Hedef alınan bilgisayarın IP adresinin “kaynak adres” olarak kullandığı sahte ICMP “echo request” paketlerinin hazırlanması ve bu paketlerin herkese yayınlanacak (broadcast) şekilde tüm bilgisayarlara yönlendirmesinin sağlanmasıdır.

ICMP (Internet Control Mesaj Protocol), ağ üzerindeki hataların belirlenmesi, bazı kontrol bilgilerinin karşılıklı değiştirilmesi, ağın belli açılardan gözetlenmesi imkanlarını sağlayan hizmetler verir. Bunlardan biri de, bir cihazın o anda açık olup

olmadığını, belli fonksiyonları yerine getirip getirmediğini belirlemede kullanılan hizmettir. ICMP bu işlemi gerçekleştirmek için, denetlenmek istenen cihaza “echo request” paketi gönderir. Bunu alan cihaz ise, “echo reply” paketi ile, ağ üzerinde faal bir şekilde bulunduğunu bildirir. Bu sayede cihazların ulaşılabilir olup olmadığını, bu yolla da eğer bir sorun yaşıyorsa sorunun nereden kaynaklandığını belirlemek mümkün olabilmektedir. Bu mekanizma çoğu işletim sistemi tarafından, genellikle de “ping” adı altında gerçekleştirilmektedir.

Saldırının ikinci unsurunu ICMP paketlerinin, bir ağın herkese yayın (broadcast) IP adresine yönlendirilmesi sonucu oluşturur. Bilindiği gibi çoğu durumda hazırlanan IP paketlerin “hedef adres” kısmında, paketin ulaşması gereken bilgisayarın IP adresi bulunur. Eğer hedef IP adres kısmında herkese yayın adresi varsa, bu paket tüm bilgisayarlara yöneltilecek demektir. Herkese yayın adresleri, IP adresinin konak kısımlarına ilişkin bitlerin tamamının ‘1’ olduğu adrestir. Örneğin C sınıfı bir IP adresi ağı için ağ IP adresi : 193.140.76.0 ise herkese yayın adresi : 193.140.76.255 olacaktır.

Saldırgan kendisine hedef olarak seçtiği cihazın IP adresini ‘kaynak adres’ kısmına yerleştirdiği ICMP ‘echo request’ paketlerini, yine ara hedef olarak seçtiği ve saldırısında basamak olarak kullanacağı ağın herkese yayın adresine gönderir. Böyle bir saldırıda saldırı, ara hedef ve hedef olmak üzere üç nokta vardır. Şimdi belirtilen özellikteki paketlerin saldırı tarafından üretilerek gönderilmesi durumunda olabilecek sonuçları inceleyelim.

Bilindiği üzere bu paketlerin ‘hedef adres’ kısmında, ara hedef ağın herkese yayın adresi bulunmaktadır. Dolayısıyla, eğer yol üzerinde veya ara hedef ağın internete açılan yüzünde bulunan yönlendiricide bu paketlerin filtrelenmesi yönünde bir çalışma yapılmamışsa, saldırı tarafından üretilen paket, ara hedef olan ağa ulaşacaktır. Paket herkese yayın adresini taşıdığından, ağ üzerinde yer alan ve açık olan bütün bilgisayarlar tarafından alınacaktır. ICMP ‘echo request’ paketi olduğundan, paketin kaynak adres kısmında yer alan bilgisayara (hedef bilgisayar) ara hedef ağ üzerinde yer alan her bilgisayar ICMP ‘echo reply’ paketi gönderecektir. Bu paketler de ara hedef ağ trafiği üzerinde etkili olacak, performansı

kötüleştirecektir. Saldırganın kullandığı paket boyu, gönderilme süresi ve ağda bulunan aktif bilgisayar sayısı arttıkça performans düşüşü daha fazla olacaktır.

Aslında asıl amaç ağa değil de hedef bilgisayara zarar vermektir. Bu da ağda yer alan bütün bilgisayarların ICMP 'echo reply' paketlerini hedef bilgisayara yöneltmekle gerçekleşmiş olur. Yoğun paket bombardımanına tutulan bilgisayar kilitlenir hatta yerel ağda da problemler çıkar. Saldırganlar bu yüzden birden fazla ara ağ kullanmaya çalışırlar.

Bu saldırı biçimi herkese yayın IP trafiğinin filtrelenmesi ve bilgisayarların herkese yayın IP kaynak adresli ICMP paketlerine cevap vermesinin engellenmesi yöntemleriyle engellenebilir.

3.2.3.3.5. UDP Portlarından Saldırı

Smurf saldırılarının benzeri olan, ICMP paketleri yerine UDP paketlerinin kullanıldığı saldırı türüdür. Bir bilgisayar üzerinde veya birkaç bilgisayar arasında, gerçek UDP portlarına yöneltilecek yoğun paket akışıyla gerçekleştirilen bu saldırılar, tek bir bilgisayar üzerinde gerçekleştiriliyorken bu bilgisayarın performansının düşmesine, birden fazla bilgisayar arasında gerçekleştiriliyorken ise, ağın performansının düşmesine sebep olacaktır.

Birbiriyle haberleşmekte olan iki UDP servisinden birisi veya her ikisi, üreteceği yoğun paket akışıyla, karşısındaki bilgisayarın servisini kilitlemeyi, bilgisayarın performansını kötüleştirmeyi başarabilir. UDP servisleri bağlantı temelli olmadıklarından, herhangi bir el sıkışma mekanizması ya da bazı kontrol bilgilerinin karşılıklı değiştirilmesi gerekmediğinden, bu tür saldırılara açıktır. Örneklerle açıklayalım;

7 numaralı portu kullanan UDP echo servisi, karşısındaki bilgisayardan (istemci) aldığı bilgileri olduğu gibi geri gönderir. 19 numaralı port üzerinden servis veren UDP 'chargen' servisi ise, istemci bilgisayardan her paket alışında, rasgele sayıdaki verilerden oluşan paketi geri gönderir. Bu iki servise ilişkin UDP portlarının aynı bilgisayar üzerinde veya değişik bilgisayarlar arasında birbirine bağlanması,

sonsuz bir trafiğin oluşmasına sebep olacaktır. Bu hem servisi veren bilgisayarı hem de trafiğin aktığı ağı etkileyecektir.

Böyle bir saldırı sonucunda doğabilecek sonuçları şunlardır:

Saldırının yöneltildiği servisler kilitlenebilir. Bu servisleri veren bilgisayarların performansı düşebilir ve servisleri veren bilgisayarların bulunduğu ağın trafiğini artırır.

Bu saldırı tipinden korunmak için alınabilecek önlemlerin başında saldırıda kullanılan servislerin bilgisayarların üzerinden kaldırmak gelir. Bu servislere ilişkin paketlerin güvenlik duvarı üzerinden filtrelenmesini sağlamaktır. Tabii bu yaklaşım kullanılıyorken, iptal edilecek servislerin ne kadar gerekli olduğu da önemlidir. Bazı servislerden vazgeçilmesi zordur (Örneğin UDP kullan DNS servisleri). Böyle bir durumda iyi hazırlanmış paket filtreleme kuralları ile, sadece belli bilgisayarların bu servislerden yararlanması sağlanıp diğerlerinin ulaşmaları reddedilebilir. Bu tür saldırılarda en çok kullanılan UDP servisleri chargen ve echo servisleridir. Bu servisler aslında neredeyse hiç kullanılmazlar. Dolayısıyla bu servislerin iptal edilmesi ya da güvenlik duvarı üzerinden filtrelenmesi, normal çalışmayı etkilemeyecektir.

Bir bilgisayar üzerinde çalışan UDP echo ve chargen servislerini iptal etmek için yapılacaklar her sistem için farklıdır. Unix sistemler için yapılması gereken, inetd.conf dosyası içerisinde, bu servislere ilişkin tanımlamaların yapıldığı satırların başına '#' koymak ve o satırın dikkate alınmaması sağlanır. Aktif olması için sistemin tekrar başlatılması gerekir. Aynı işlemler diğer UDP servisleri içinde uygulanabilir. Böylece bu servislerden gelebilecek saldırılar önlenmiş olacaktır.

Saldırıların daha çok hangi servislere yapıldığının tespiti için ağa saldırıları kontrol edip raporlayan (intrusion detection) programların kullanılması faydalı olacaktır.

3.2.3.3.6. NFS'e yönelik saldırılar

NFS (Network File System), ağ üzerindeki bilgisayarların dosyalarını birbirleriyle paylaşmalarını sağlayan bir protokoldür. Ancak bu protokolün açıklarının kullanılmasıyla, sistemi büyük zararlar verilebilecek bir saldırıya açmış olursunuz.

NFS uzun zamandır üzerinde çalışılan bir protokoldür ve saldırı programları internette yaygın olarak bulunmaktadır.

NFS'e yönelik saldırılar değişik sonuçlar doğurabilir. Saldırgan hedef bilgisayar üzerinde süper kullanıcı yetkisiyle işlemler yapabilecek duruma gelebilir.

Alınabilecek önlemler;

Güvenlik duvarından NFS servislerine ait paketlerinin geçişi engellenebilir. Aynı şekilde internet üzerinden bu servislere ulaşım yasaklanabilir. Bu önlemler NFS'e dışardan gelebilecek saldırılar içindir. İç ağdan gelebilecek saldırılara hala açıktır.

Güvenlik duvarı üzerinden alınabilecek önlemlerin yanı sıra /etc/exports dosyası üzerinde yapılacak bazı düzenlemelerle alınabilecek önlemlerde vardır. NFS ile ilgili yamaların takip edilip programa eklenmesi de saldırıların etkisini azaltacaktır.

3.3. Güvenlik Duvarı

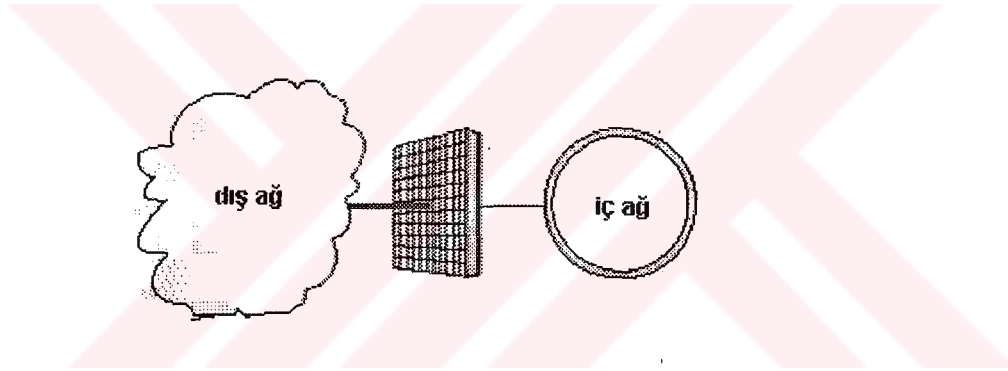
İnternetin bütün dünyada hızla büyümesi, bununla birlikte verimliliğinin artması, şirketlerin internette kendi web sunucularını kurmaları konusunda cesaretlendirmiştir. İnternette bu varoluş ve ortaya çıkma işi birtakım riskleri de yanında getirmiştir. Saldırganlar (Hacker) şirket ya da kurum ağına izinsiz olarak girebilir ve sistemlere zarar verebilirler. İç ağları bu tür davetsiz misafirlerden korumak için güvenlik duvarı(firewall)kullanmak zorundasınız.

İşletmelerin intranet ve extranet uygulamaları dolayısıyla dış dünyayla olan bağlantıları artmıştır. Böyle sistem yapılarının saldırılardan korunması daha zordur.

Terim olarak güvenlik duvarı (firewall) özel bir donanım ya da yazılım parçası değildir. Sadece iç ağı dışarıdan gelebilecek saldırılardan, davetsiz misafirlerden koruyabilmek için kullanılan donanım ve/veya yazılım çözümüdür.

Güvenlik duvarı (firewall), bir korumalı ağa giren ve çıkan haberleşme trafiğine, belirli (tanımlanmış) bir güvenlik politikası (security policy) uygulayan, donanım ve yazılım mekanizmalarından oluşan bir teknolojidir. Güvenlik politikası belli bir ağı korumak için geliştirilen bir plandır. Bir güvenlik duvarı, Şekil 3.4' te görüldüğü gibi, donanım, yazılım veya her ikisi ile beraber bir ağı korur ve izole eder.

Bu araçların öncelikli görevi güvenilir ağ (iç ağ) ile güvenilmez ağ (internet) arasında bir güvenlik duvarı oluşturmaktır. Bir güvenlik duvarı bir iç ağı (*private*, örneğin bir intranet) bir dış ağdan (güvenilmeyen, örneğin internet) korur ve izole eder.



Şekil 3.4. Güvenlik duvarı

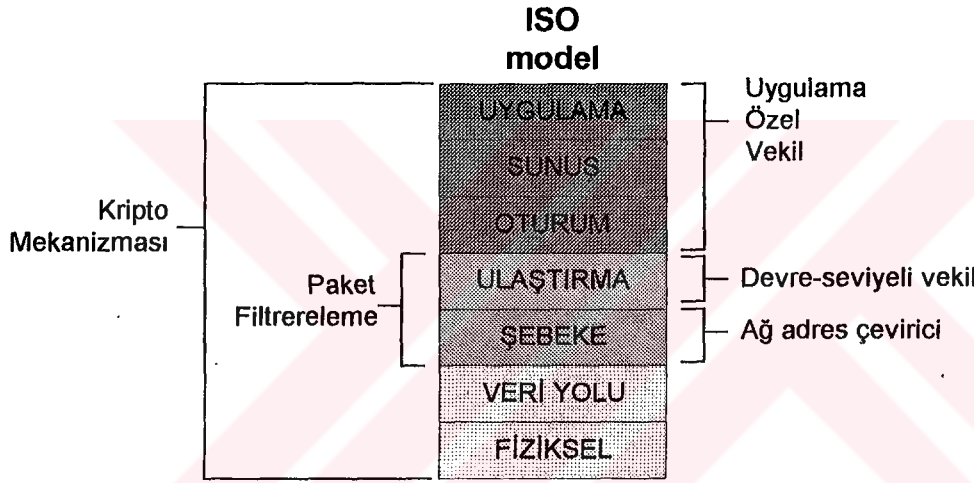
Güvenlik duvarları, günümüzde yaygın olarak TCP/IP tabanlı ağlarda (Internet ve özel Intranet ağlarında) kullanılmakla birlikte, temel kavramları diğer ağlar için de uygulanabilir. Bu teknoloji halen gelişen bir teknoloji olarak tanımlanmasına karşın, bir güvenlik politikası sağlayarak bir ağı güvenli kılan günümüzde mevcut en iyi yollardan biridir.

NT ve Unix ortamları için en güvenli korunma uygulama geçidi (Application gateway) ve adaptif vekil (Adaptive Proxy) sayesinde geliştirilen durumsal paket filtrelenmesi yöntemidir. adaptif vekil (Adaptive Proxy) teknolojisi güvenlik duvarı

teknolojilerinde önemli dönemeçlerden birisini temsil etmektedir. Çünkü kullanıcılara hem durumsal paket filtreleme denetimini hem de uygulama geçit güvenlik duvarı (Application gateway firewall) teknolojisini bir arada sunmaktadır.

3.3.1. Koruma Mekanizmaları

Bir güvenlik duvarının kullanacağı güvenlik mekanizmaları, Şekil 3.5' te görülen, OSI ağ modelinde kullanılacak katmanlara göre değişir. Örneğin, paket-filtreleme (packet-filtering) ve ulaştırma (transport) katmanlarında çalışmasına karşın, ağ adres çevirimi (network address translation) şebeke (network) katmanında çalışır. Uygulamaya bağlı çalışan vekil (proxy) koruyucular ise en üst üç katmanda çalışır. Şifreleme mekanizmaları yedi katmanda da uygulanabilir.



Şekil 3.5. OSI Güvenlik Duvarı Çözümleri

3.3.2. Güvenlik Duvarı Araçları ve Teknolojileri

Güvenlik duvarı standart bazı araçlardan (firewall building blocks) oluşur. Bu araçlar mimariye göre farklı yerlere konabilir. Bunlar;

- 1-) Onaylama Sunucusu (authentication server)
- 2-) Denetlemeli Yönlendirici (screening router)
- 3-) Korumalı konak (bastion host)

4-) Vekil sunucular (Application proxy servers)

Bir çok ticari üründe sunulan çözümlerde yukarıdaki araçların bir kısmı ya da karışımı tek bir ürün olarak sunulmaktadır.

Koruma için kullanılan ürünlerde aşağıdaki teknolojiler kullanılmaktadır.

- 1-) Paket filtreleyici
- 2-) Uygulama filtreleyici ağ geçidi
- 3-) Vekil sunucu ağ geçidi
- 4-) Durumsal paket denetleyiciler.

Her güvenlik duvarı teknolojisinin kendisine göre avantaj ve dezavantajları vardır. Bunlar aşağıda özetlenmiştir.

Paket filtreleyici (Yönlendirici) güvenlik duvarı hızlıdır ve ağ bağlantısı ile ilgili bir takım yaygın hataları ve saldırıları önler. Yönlendirici üzerinde çalışır.

Uygulama filtreleyici ağ geçidi güvenlik duvarı, uygulamaların kapsamlı kontrolünü uygulama seviyesindeki davranışları denetler.

Vekil sunucu güvenlik duvarları, istemci ve sunucu bilgisayarlar arasındaki uygulama ve oturumların kontrolü ile adres çevirim fonksiyonlarını sağlar.

Durumsal denetleyici güvenlik duvarları, ağa sızmaya yönelik gelişmiş saldırıları önler. Herhangi bir vekil sunucu ya da uygulama ağ geçidi güvenlik duvarlarına göre daha kapsamlı ve gelişmiş filtreleme özelliklerine sahiptir.

Bir organizasyon için ideal güvenlik çözümü, kullanılan uygulamaların yapısına da bağlı olarak bütün filtreleme metodlarını gerçekleştirebilen ürünlerdir. Ağ ortamında, hiçbir güvenlik duvarı mimarisi tek başına bütün güvenlik problemlerini çözemez.[8]

3.3.2.1. Paket Filtreleyici

Paket filtrelemede ana rol oynayan bir yönlendirici (router) belli kuralları sağlamayan ağ trafiğine izin vermez. Örneğin, TPC/IP paket filtreleme güvenlik duvarında, yönlendirici datagram paketlerinin başlıklarındaki kaynak ve hedef adresler ile port numarasına bakar.

İki tür paket filtreleme yöntemi vardır;

1-) Durumsuz paket filtreleme (stateless packet filtering) : Eğer filtreleme işlemindeki kurallar, daha önceki veri paketlerinin durumlarından bağımsız olarak uygulanıyorsa, bu yönteme durumsuz paket filtreleme (stateless packet filtering) denir.

2-) Durumlu paket filtreleme (stateful packet filtering : Aynı bağlantıya ait paketlerin durum bilgileri saklanıp bir mesajın içeriğine göre filtreleme kararı veriliyorsa durumlu paket filtreleme (stateful packet filtering) yapılıyor demektir. Paket filtreleme, bir güvenlik duvarından geçen veri paketlerini incelemeye ve kontrolde kullanılabilecek bir yöntem olmasına karşın tek başına güvenlik politikaları uygulamak için yeterli değildir.

3.3.2.1.1. Uygulama

Paket Filtreleme Örneği;

C sınıfından 195.55.55.0 adresine sahip bir şirket olduğu varsayalım. Bu şirket bir ISS (internet servis sağlayıcı) aracılığı ile Internet'e bağlanmıştır. Şirketin politikası herkesin giden paketlerine izin vermek fakat bütün gelen bağlantıları "mail host" a yönlendirilmesidir. Gelecek olan servisler sadece Mail ve DNS olacaktır.

- İzin , Çıkan bütün TCP bağlantıları
- İzin, Gelen SMTP ve DNS isteklerini mailhost makinasına
- TCP portu 1024 ten büyük olan , gelecek FTP veri bağlantılarına izin ver

- Yüksek port numaraları ile çalışan servisleri korumaya çalış

Bu konfigürasyona göre sadece internet üzerinden gelen paketler kontrol edilecektir. Kurallar, uygunluk bulunduğu zaman durdurulur ya da yönlendirilir. Erişim listesinin (Access List) sonunda kesin bir kural vardır ki kabul şartlarına uymayan bütün paketler kesinlikle reddedilecektir. Aşağıdaki örnekte IP erişim listesi Cisco IOS v.10.3 ve sonrası için geçerlidir.

1. no ip source-route

2. !

3. interface ethernet 0

4. ip address 195.55.55.1

5. !

6. interface serial 0

7. ip access-group 101 in

8. !

9. access-list 101 deny ip 195.55.55.0 0.0.0.255

10. access-list 101 permit tcp any any established

11. !

12. access-list 101 permit tcp any host 195.55.55.10 eq smtp

13. access-list 101 permit tcp any host 195.55.55.10 eq dns

14. access-list 101 permit udp any host 192.55.55.10 eq dns

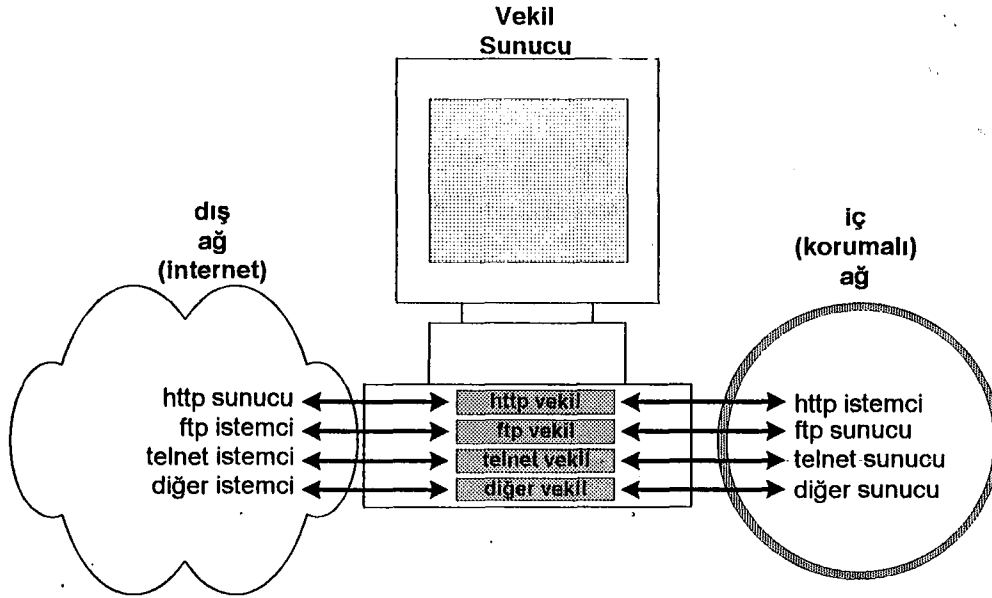
15. !



16. access-101 deny tcp any any range 6000 6003
17. access-list 101 deny tcp any any range 2000 2003
18. access-list 101 deny tcp any any eq 2049
19. access-list 101 deny udp any any eq 204
20. !
21. access-list 101 permit tcp any 20 any gt 1024
22. !
- 23.:
24. access-list 101 permit icmp any any
25. !
26. snmp-server community FOOBAR RO 2
27. line vty 0 4
28. access-class 2 in
29. access-list 2 permit 195.55.55.0 255.255.255.0

3.3.2.2. Vekil Sunucular

Uygulama Seviyesindeki Güvenlik Duvarıdır. Her uygulama için özel bir vekil (proxy) aracı programı ile ağ haberleşmesi sağlanır. Bu vekil program belirlenen güvenlik politikasına göre kontrol yapar. Şekil 3.6' da görüldüğü gibi her uygulama için ayrı bir vekil yazılımı gerekir.



Şekil 3.6. Uygulama Seviyesindeki Güvenlik Duvarı.

Vekil sunucular genelde çift arayüzlü konaklar üzerinde çalışırlar (dual-homed). Bu konaklar çift ağ arayüzüne sahip olup sunucu bilgisayarlardır. Her ağ arayüzü ayrı bir IP numarasına sahiptir. Bu ağ adaptörlerinden biriyle iç ağa diğeriyle de internete bağlanır. Konak üzerinde çalışan vekil sunucu yazılımı bir ağdan gelen trafiği diğer ağa geçirir ya da iptal eder.

Çift ağ arayüzüne sahip bulunan konak üzerinde çalışan vekil sunucu basit fonksiyonlu yönlendirici gibi çalışmaktadır. Farklı olarak daha akıllı ve zengin özelliklere sahiptir:

1-) Gizliliği azaltma ve bant genişliğini koruma : Uygulama seviyesi vekiller verileri depolayabilir (cache) ve kendi kayıtlarından hizmet verebilirler. Bu şekilde, hızlı verilmesi gereken cevaplara karşı disk alanı oluşturulmuş ve bant genişliği korunmuş olur.

2-) Gelişmiş erişim kontrolü : Uygulama seviyesi vekil yazılımları doğrulamayı (authentication) sağlamaktadırlar. Doğrulama bilgilerini erişim izni için kullanır.

3-) Gelişmiş filtreleme : Uygulama seviyesi vekil yazılımlarının çok gelişmiş filtreleme özellikleri özellikleri vardır. Protokollerin içeriğini anlayabilir ve sadece TCP/IP paketlerinin başlığıyla sınırlı kalmaz.

4-) Kayıt tutma ve denetim : Vekil sunucu üzerinde olan işlemlerin kaydı tutulur. Bu şekilde web trafiğinin denetimi sağlanır.

Genelde yönlendirici ve vekil sunucular daha güvenli güvenlik duvarı çözümü sağladığı için birlikte kullanılır. Ayrıca vekil sunucular çift ağ arayüzlü bilgisayarlar üzerinde çalıştırılmak zorunda değildir. Ağ seviyesindeki trafik kontrolü yönlendirici tarafından yapılır.

Şekil 3.8' da vekil sunucunun yönlendirici ile güvenlik duvarı konfigürasyonu görülmektedir. Yönlendirici internetten gelen, izin verilen port dışındaki (SMTP) bağlantı isteklerini iç ağa geçirmeyecektir. Buradaki yönlendiricinin tipik fonksiyonları şunlardır;

1-) İç ağdan vekil sunucu bilgisayara olacak bağlantılara izin ver.

2-) İç ağdan internete direk bağlantıları reddeder. Bu kural dış ağa çıkış için vekil sunucu kullanımına zorlar. Vekil sunucular böylece dış ağa çıkmak için tek bir noktayı kullanmak zorunda kalır. Bu şekilde internet ile iç ağ arasındaki olaylar ile ilgili kayıtlar sağlıklı olarak tutulmuş olur.

3-) Vekil sunucudan internete olan bağlantılarda iyi bilinen port bağlantılarına izin verir (HTTPS,HTTP gibi).

4-) Internetten vekil sunucuya olan bütün bağlantı isteklerini reddeder.

Daha güvenli bir güvenlik duvarı mimarisi aşağıdaki şekil 3.9' da görülmektedir. Vekil sunucu konağı (Firewall bastion ya da bastion host olarak da isimlendirilir.) iki tarafı da yönlendiriciler tarafından çevrilmiştir. Bu üç katmanlı güvenlik sağlar. Bu mimaride korumalı konak (bastion host) üzerinde çalışan vekil sunucunun görevleri şunlardır;

1-) Web protokollerinin (HTTP, FTP,Gopher) dışarıya çıkışına izin ver.

2-) SSL(Secure Socket Layer) protokolünün bilinen portlarla bağlantısına izin verir.(HTTPS için 443, SNEWS için 563).

3.3.2.2.1. Devre Seviyeli Vekil Sunucular

Devre seviyeli vekil sunucular (Circuit-Level Proxy Servers) bir yazılım olup bağlantı seviyesinde çalışır. Uygulamaların bağlantılarını sağlar; sonra uygulama seviyesi protokolüne bakmaksızın iki yönlü veri aktarımına başlar.

Devre seviyeli vekil sunucular kavramsal olarak yönlendiriciler ile uygulama seviyesi vekil sunucular arasında yer alır. Yönlendirici paket seviyesinin üstünde ve uygulama seviyesi vekil sunucu protokol yapısı altında olduğundan bağlantı işlevini gerçekleştirir. Devre seviyeli vekil sunucu protokolü SOCKS olarak adlandırılmıştır.

Uygulama seviyesi vekil sunucularının eksiklikleri vardır. Birkaç tanımlı protokol için sadece bir bağlantının kontrolüne imkan tanınır. Uygulama seviyesi vekil sunucu yazılımının sürümü güncellemeden, yeni ya da bilinmeyen protokoller çalışmaz. Devre seviyeli vekil sunucu bu noktada devreye girmektedir. Devre seviyeli vekil sunucular genellik oluşturabilmektedir. Böylece protokol bağlantıları bunlar üzerinden yapılabilir.

3.3.2.2.2. Saydam Vekillik

Buradaki saydamlık iki farklı anlamda kullanılır. Genel anlamda kullanıcının direk sunucuya bağlanmakla, vekil sunucu üzerinden bağlanması arasında bir fark görmemesidir. Bu anlamda bütün vekil sunucular saydamdır.

Diğer anlamda saydam vekillik son zamanlarda bildirilmiştir. İstemci yazılım da vekil sunucunun farkında değildir. Genellikle bilinen, istemci yazılımın hangi vekil sunucuyla konuştuğunu bildiğidir. İstemci yazılımın direk ya da vekil sunucu üzerinden bağlantı kuracağını farkedebileceği bilinir.

Saydam vekillik uygulamasında, yönlendirici direk olarak vekil sunucudan istek alacak şekilde programlanır. Bütün HTTP bağlantıları kesilip vekil sunucu tarafından tekrar internete yönlendirilir. Bu bağlantılar saydam vekil sunucu tarafından kontrol edilir. Vekil sunucu tarafından filtreleme ve erişim kontrol

kuralları uygulanır. Diğer istekler ya reddedilip başka vekil sunuculara gönderilir ya da vekil sunucu kayıtlarından (cache) hizmet verilir.

Saydam vekillikte, istemcinin kesinlikle arada bir vekil sunucu olduğundan haberi yoktur. Fakat, HHTP/1.1 protokolü arada bu tür bir vekil sunucu sağlamaz.

3.3.2.3. Güvenlik Duvarı Ürünleri Karşılaştırması

Yaygın olan ve yukarıda özetlen üç farklı güvenlik duvarının temel özellikleri tablo 3.4' te özetlenmektedir.

Tablo 3.4. Güvenlik Duvarı Mimarilerinin Karşılaştırılması

Güvenlik Duvarı	Avantajları	Dezavantajları
Paket Filtreleyici Güvenlik Duvarı	Ucuz Yüksek performans Kullanıcılara göre saydam	Sadece ağ protokollerini anlayabilirler. İçerideki ağ yapısını dışarıdan gizleyemezler. Daha yüksek katmanlardaki uygulamalara yönelik atakları engelleyemezler. Denetim kabiliyeti sınırlıdır. Kimlik onaylama, zamana ve saate göre kontrol, kayıt gibi özelliklere sahip değildir.
Vekil Sunucu Güvenlik Duvarı	Uygulama katmanındaki protokolleri anlayabilirler. En sağlam güvenlik çözümü sağlarlar. Ayarlanması ve	Düşük performans. Kullanıcılara göre saydam değildir. Her protokol veya uygulama için yeni bir vekil geliştirilmesi gerekir.

	programlanması kolaydır. İçerideki ağ yapısını gizleyebilirler. Denetim ve kullanıcı kayıt araçlarına sahiptir. Kimlik onaylaması ile zaman ve saate göre kontrol gibi özelliklere sahiptirler.	İşletim sistemini dış ataklara açarlar. OSI modelinin sadece 5 ile 7 arasındaki katmanları incelenir. UDP ve RPC gibi bağlantısız protokoller vekiller tarafından desteklenemez.
Durumsal Denetleyici Güvenlik Duvarı	Bütün OSI modeli kapsanır. Gelişmiş ataklar engellenir. Çok yüksek performans sağlanır. Kullanıcılara göre saydamdır. Bağlantısız protokollerin takibi için sanal oturum bilgisi oluşturulur.	Uygulama seviyesinde denetleme yapmak için veri akışı uygulama katmanına yükseltilmelidir. Gelişmiş kullanıcı kaydı yapmak zordur. İçerik güvenliği ve karmaşık uygulama filtrelemesini yapmak zordur.

3.3.3. Güvenlik Duvarı Mimarileri

Çeşitli güvenlik duvarı mimarileri vardır bir kısmı burada incelenmiştir. Bunlar;

- 1-) Çift ağ arayüzlü güvenlik duvarı (dual-homed gateway)
- 2-) Denetlenen konak güvenlik duvarı (screened host firewall)
- 3-) Denetlenen altağ güvenlik duvarı (screened subnet firewall)

4-) Karışık mimarili güvenlik duvarı çözümü

3.3.3.1. Çift Ağ Arayüzlü Güvenlik Duvarı (dual homed gateway)

Çift ara yüzlü güvenlik duvarı mimarisi, en az iki ağa bağlantı sağlayan arayüze sahip bir makine üzerine kurulmuştur. Şekil 3.7’de de görüldüğü gibi çift ağ arayüzüne sahip bilgisayarın yönlendirici olarak da çalışması mümkündür. IP yönlendirme yeteneğine sahip olacak; fakat bir ağdan gelen paket diğerine doğrudan aktarılmayacaktır.

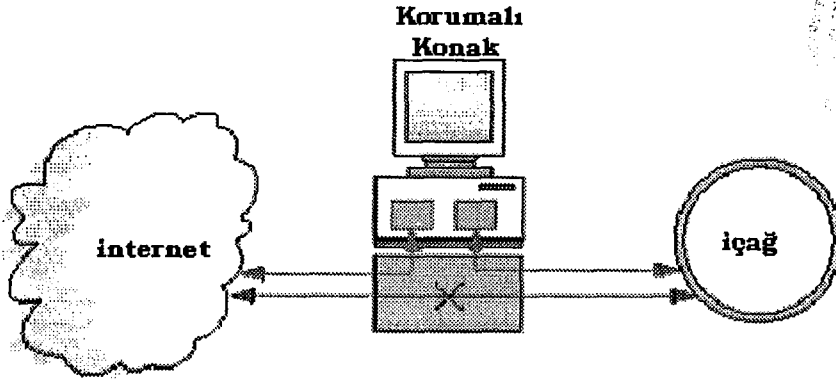
Bu mimaride yüksek seviyeli bir trafik kontrolü gerekmektedir. Böyle bir mimari kullanıldığında, belli bir servis için istek geldiğinde, bu isteğe ilişkin veri tiplerinin uygun olmamaları durumunda isteğin geri çevrilmesi mümkündür. Bu da çok büyük avantajları da beraberinde getirecektir. Servis isteklerinin, her servisin isteklerine ilişkin şartları yerine getirip getirmediğine bakılarak, bağlantıya kurması ya da reddetmesine olanak sağlayacaktır.

Çift ağ arayüzlü güvenlik duvarı, iki şekilde hizmet sunarak iç ağ ile dış ağ arasındaki bağlantıları kurar.

1-) Vekil sunucu (proxy server) ya da,

2-) Kendisinden hizmet almak isteyen bütün kullanıcıların bir kullanıcı adı ve parolasıyla onaylanması suretiyle.

Kullanıcı adı ve şifresi kullanılarak gerçekleştirilecek bir bağlantı kontrolü işlemi, güvenlik açısından bazı sorunlar çıkartacaktır. Kullanıcı adı ve şifrelerinin öğrenilmesi sistemi güvensiz hale getirir. Bu durumda vekil sunucular kullanılarak servis verilmesi işlemi daha güvenlidir. Ancak burada da bazı zorluklar vardır. Verilmek istenen servislerin tamamı için vekil sunucu bulmak zordur.



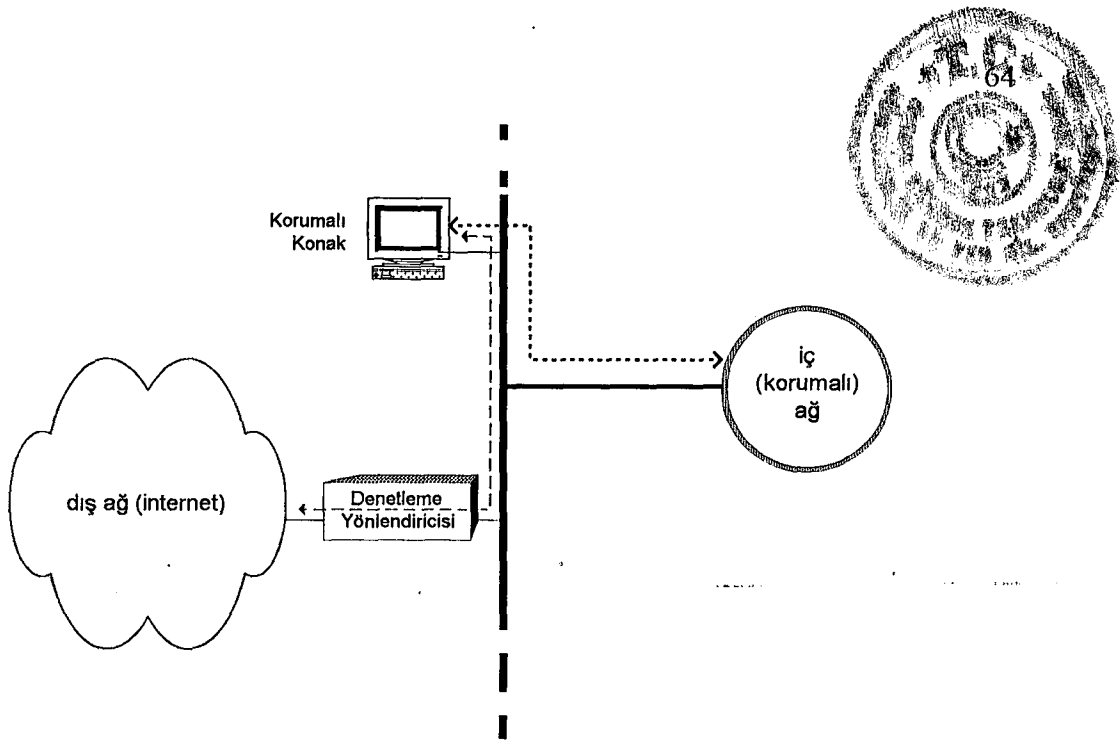
Şekil 3.7. Çift Ağ Arayüzlü Güvenlik Duvarı

3.3.3.2. Denetlenen Konak Güvenlik Duvarı (screened host firewall)

Paket filtreleme güvenlik duvarları mimarisine örnektir. Bir yönlendirici tek bir makineye erişimleri kontrol eder.

Denetlenen konak (screened-host) güvenlik duvarında, ağ katmanındaki bir yönlendirici, bütün korumalı ağ trafiğinin üzerinden geçmek zorunda olduğu ve korumalı konak (bastion host) olarak adlandırılan tek bir makineye erişimi kontrol eder. Korumalı ağa doğrudan erişim engellenir ve bastion host paketleri göndermez. Bu makine iyi korumalı ve bütün saldırılara karşı koyması beklenen güçlü bir noktadır.

Bu mimaride servisler sadece yerel ağa bir arayüzü bulunan bir konak(host) üzerinden verilir. Bu konak şekil 3.8’de de görüldüğü gibi bir yönlendiriciye bağlıdır ve bu yönlendirici üzerinden dış dünyaya açılmaktadır.



Şekil 3.8. Denetlenen Konak Güvenlik Duvarı Mimarisi

Bu mimaride güvenlik için paket filtreleme mekanizması önemlidir.

Korumalı konak yerel ağda yer alır. Yönlendirici üzerinde yer alan paket filtreleme kuralları, internette yer alan bilgisayarların yalnızca korumalı konak makinasına bağlantı sağlamaya izin verecek şekilde gerçekleştirilmiştir. Ayrıca sadece izin verilen servislere bağlanılabilesine müsaade edecek şekilde düzenlenmiştir. Dış ağdan (internet) gelecek tüm servis istekleri korumalı konağ yönlendirileceğinden, bu konağın üst düzeyde güvenliğinin sağlanması gerekir. Aynı zamanda korumalı konak, dış ağda yer alan sunuculara bağlanarak istekte bulunma işlemini de yerine getirir.

Yönlendirici üzerinde çalışan paket filtreleme iki şekilde davranabilir.

- 1-) Yerel ağda yer alan bilgisayarların, belirlenmiş servisler için dış ağa bağlanabilmelerine izin verir.
- 2-) Yerel ağda bulunan bütün bilgisayarların dışarıya erişimlerini yasaklayarak, bu tür isteklerin tamamının korumalı konak üzerinde bulunan vekil sunucular (proxy servers) üzerinden yapılmasını sağlar.

Bu yöntemler ayrı ayrı kullanılabileceği gibi birlikte de kullanılabilir. Bazı servisler için, sadece paket filtreleme mekanizması kullanılarak doğrudan dışarıya ulaşma izin verilirken bazıları içinse sadece vekil sunucu üzerinden dışarıya çıkmasına izin verilir.

Çift ağ arayüzlü güvenlik duvarı mimarisinde dış ağdan gelen iptal edilecek paketler yerel ağa aktarılmamaktadır. Oysa bu mimaride geçebilmektedir. Dolayısıyla bu mimari güvenlik açısından daha riskli gözükmemektedir. Buna karşılık çift ağ arayüzlü güvenlik duvarı mimarisinde güvenlik duvarı olarak kullanılan bilgisayar üzerinde olabilecek bir sorun sonucu dışarıya olan bağlantı kesilecektir.

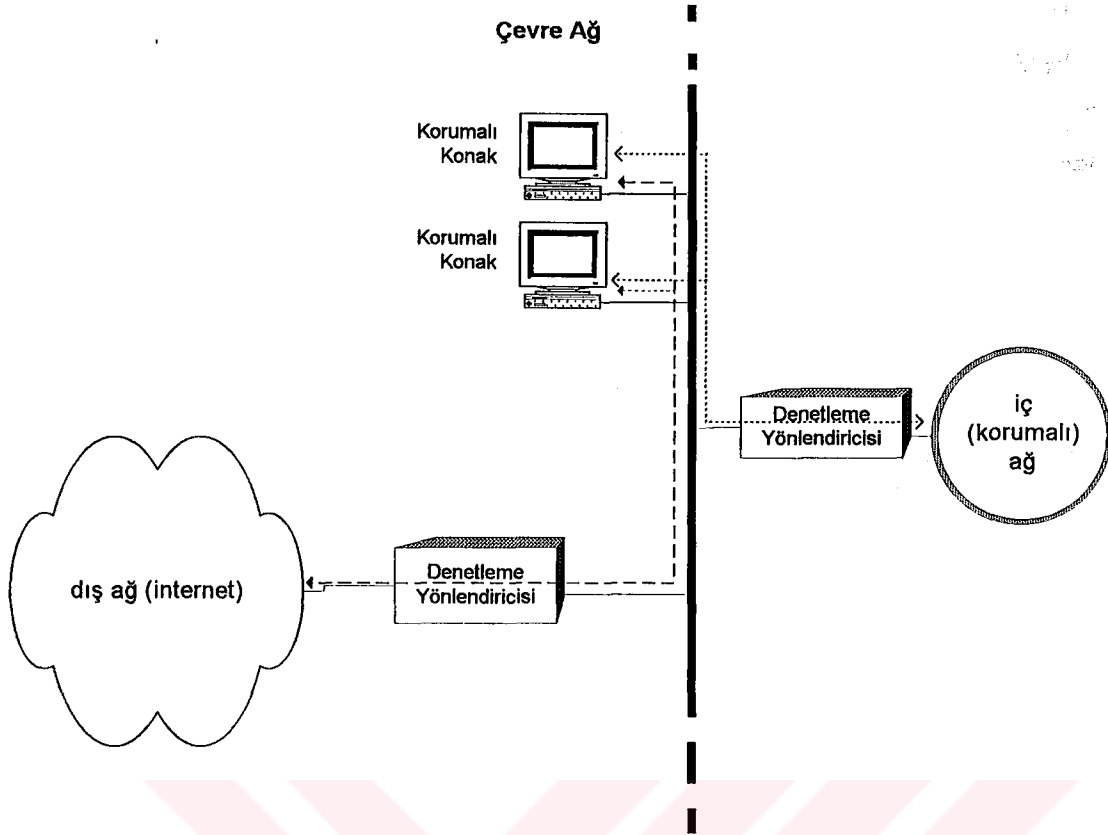
Bir yönlendiriciyi saldırılara karşı korumak, bir bilgisayarı korumaktan daha kolay olacaktır. Çünkü yönlendiricinin yapacağı işlemler sınırlıdır ve bunlarla ilgili olarak güvenliği sağlamak da, bir konağa ilişkin güvenliği sağlamaya göre daha kolay olacaktır. Bu yüzden bu mimarinin çift ağ arayüzlü güvenlik duvarı mimarisine göre daha güvenli ve kullanışlı olacağı açıktır.

Bunlara karşın, bu mimarinin dezavantajları vardır. Saldırganın korumalı konağa (bastion host) ulaşması durumunda, ağdaki diğer makinalara ulaşmak için önünde bir engel kalmamış olacaktır. Denetleme yönlendiricisinde meydana gelecek herhangi bir sorun durumunda da saldırırganın tüm ağa ulaşması için önünde bir engel kalmayacaktır.

3.3.3.3. Denetlenen altağ güvenlik duvarı (screened subnet firewall)

Paket filtreleme güvenlik duvarları mimarisine örnektir. Yönlendirici çifti kullanılarak “DeMilitarized Zone — DMZ” olarak adlandırılan bir alanla ağ koruması yapılır.

Şekil 3.9’da da verilen güvenlik duvarı mimarisinde (screened-subnet), bir çift yönlendirici yardımıyla DMZ oluşturularak ağ koruması yapılır.



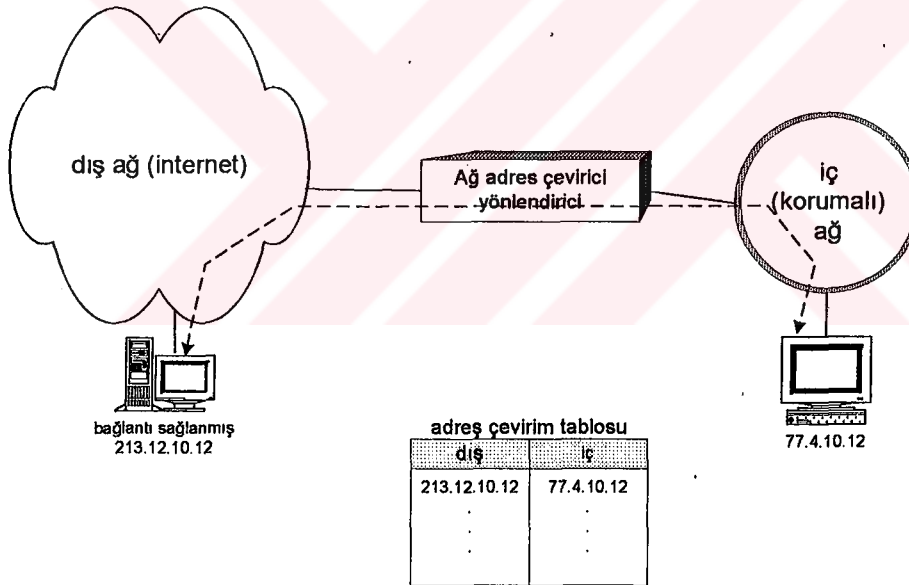
Şekil 3.9. Denetlenen Altağ Güvenlik Duvarı (Paket filtrelemeli).

Böyle bir mimarinin geliştirilmiş olmasının en önemli sebebi *denetlenen konak güvenlik duvarı* mimarisindeki tehlikenin önüne geçmektir. Denetlenen Konak Mimarisinde korumalı konaklar açık birer hedef durumundadır. Saldırıları çoğunlukla buralara yapılacaktır; çünkü bunlar dış dünyaya açılan makinalardır. Saldırganların korumalı konağı aşması durumunda, iç ağa ulaşmaları için bir engel kalmamaktadır. Bu mimaride ise korumalı konağı aşan bir saldırganın, yerel ağda bulunan konaklara ulaşabilmesi için bir engeli daha aşması gerekmektedir.

En basit yapıdaki denetlenen alt ağ mimarisinde iki adet denetleme yönlendiricisi vardır. Bunların her ikisi de çevre ağa bağlanmıştır. Biri çevre ağ ile yerel ağ arasında yer alırken diğeri de çevre ağ ile dış ağ arasında yer alır. Yerel ağda yer alan konaklara saldırmak isteyen kişinin her iki denetleme yönlendiricisini de geçmesi gerekir.

Bazı güvenlik duvarı çözümlerin de bu mimariden yola çıkılarak birden fazla çevre ağ kullanmak suretiyle çok katmanlı bir güvenlik mekanizması kullanılmaktadır. Böylece saldırganların yerel ağa ve burada yer alan sunuculara ulaşması zorlaştırılmaktadır. Yalnız böyle bir çalışmada dikkat edilmesi gereken bir nokta vardır. Her katmanın güvenlik özellikleri farklı olmalıdır. Aksi durumda, bir katmanın güvenlik mekanizmasını aşan bir saldırgan, aynı yöntemle diğer katmanları da aşacaktır.

Bir yönlendirici kullanılarak yapılan diğer bir güvenlik yöntemi Ağ Adres Çevirimidir (Network Address Translation—NAT). Bu yöntemde bir iç ağdaki adresler dış dünyadan saklanır (Şekil 3.10). Bir NAT yönlendiricisi iç ve dış adresleri tutan bir tablo içerir. Dışarıdan gelen mesajdaki adres NAT yönlendiricisi tarafından gizlenen iç adrese çevrilir. Dış erişimlerde de benzeri şekilde adres çevirimi yapılır. Bu tablonun bazen değişmesiyle iç adreslerin dışarıdan kolay bir şekilde bulunması engellenir.



Şekil 3.10. Ağ Adres Çevirimi.

3.3.4. Karışık mimarili güvenlik duvarı çözümü

Güvenlik duvarı mimarilerinin en çok kullanıldıkları şekilleri ile girebilecekleri sınıflar yukarıda anlatılmıştır. Fakat bazı durumlarda, gerek güvenlik politikası, gerek bütçe, gerekse de ağın sahip olduğu mimari, bu mimarilerden sadece birinin,

istenen güvenlięi saęlamaması sonucunu doęurabilir. Byle bir durumda, bu mimarilerin deęişik zellik ve bileşenlerinin birlikte kullanıldığı yeni mimariler ve zmler ortaya ıkacaktır.

3.3.4.1. Birden Fazla Korumalı Konaklı Mimariler

Denetlenen alt aę mimarisinin birden fazla korumalı konak kullanılarak oluşturulmuş zm olarak dşnlebilir. Byle bir zme sistemin srekli alıřabilirlięinin (high availability)saęlanması iin gidilmiş olabilir. Ayrıca performansı da arttıracaktır. HİZMETLER KORUMALI KONAKLAR ARSINDA PAYLAŞTIRILARAK performansta artış saęlanabilir ama bu yedekleme yapılmaması anlamı taşıyacağından risklidir.

3.3.4.2. Tek Ynlendiricili Mimariler

Bu mimaride i ve dıř denetleme ynlendirici yerine tek ynlendirici kullanılır. Burada ynlendiricinin bu yapıya uygun olması ve konfigrasyonunun dzgn yapılması gerekir. İ ve dıř ynlendiriciler, ierden dıřarıya ve dıřardan ieriye gerekleştirilen baęlantıları denetlemektedir. Bunların yerine konan ynlendiricinin de bu yeteneklerinin olması gerekir.

Byle bir mimari, gvenlik ve paket filtreleme konusunda yetenekli ynlendirici kullanılması durumunda iyi bir zmdr. Bu ynlendiricinin korumalı konaęa, i aęa ve dıř aęa aılan  aę arayzne sahip olması gerekir.

3.3.4.3. Korumalı Konak ile Dıř Ynlendiricinin Birleřtirildięi Mimariler

ift aę arayzne sahip bir korumalı konak makinasına ynlendirici fonksiyonlarının da eklenip kullanıldığı zmlerdir. ift aę arayzl konaęın aynı zamanda bir ynlendirici olarak alıřması performans aısından sistemi olumsuz olarak etkileyecektir. Ayrıca korumalı konaęın direk olarak internete baęlantısının yapılmış olması gvenlięini daha da nemli hale getirir.

3.3.4.4. Korumalı Konak ile İç Yönlendiricinin Birleştirildiği Mimariler

Korumalı konak ile dış yönlendiricinin birleştirildiği mimaride çok fazla güvenlik problemi olmamasına karşı bu çözümde vardır. Güvenlik açısından tehlikeli bir çözümdür. Çünkü iç yönlendiriciler hem korumalı konak hem de dış yönlendirici için güvenlik katmanı olarak çalışmaktadır.

3.3.4.5. Çoklu İç Yönlendiricili Mimariler

Çevre ağı yerel ağı değişik noktalardan bağlamak üzere kullanılabilecek birden fazla yönlendirici ile sağlanan bir çözümdür. Güvenlik açısından bir takım problemler oluşturacağından tavsiye edilen bir mimari değildir.

Performansın artırılması için kullanılması zorunlu hale gelen birden fazla iç yönlendiricili mimarilerde güvenliğin artırılması için ayrıca çalışılması gerekir. Aslında iç yönlendirici performans problemi oluşturmaz. Eğer iç yönlendirici dış yönlendiriciye göre çok hızlıysa ve trafik genelde iç ağ ile internet arasında akıyorsa, çoklu iç yönlendiricili mimariye geçilebilir. Bu sorun güçlü bir yönlendirici alınarak da çözülebilir.

Ayrıca organizasyondan kaynaklanan sebeplerden dolayı da böyle bir çözüme gidilmiş olabilir. Bu durumda da uygulanabilecek çözümler vardır. İç yönlendiricinin ağ arayüzü artırılarak bu çözüm sağlanabilir.

4. VEKİL SUNUCULAR

4.1. Giriş

Bu bölümde vekil sunucuların tarihçesi, web vekil sunucunun gelişimi, vekil sunucu içindeki çeşitli karakteristikler üzerinde incelemeler yapılacaktır. Vekil sunucuların kullanılmak istenilen amaçlara göre birkaç farklı tipi vardır. Bunlar;

Genel güvenlik duvarı vekil sunucular, şirket içinde depolama (caching) ve yüksek performanslı verim için şirket güvenlik duvarı üzerine dizayn edilirler.

Bölümsel vekiller (departmantel proxy) fonksiyonları güvenlik duvarı vekil sunucularıyla aynı olmasına karşın daha küçük ölçektedirler.

Proxy server karakteristikleri farklı filtreleri, izleme seçeneklerini, performans ile ilgili fonksiyonları içerir.[1]

4.2. Web Vekil Sunucuların Tarihçesi

Web'in gelişim tarihi sayılabilecek 1990'lı yılların başında, vekil sunuculardan geçit (gateway) olarak söz edilirdi. İlk genel WWW geçidi CERN'de WWW takımı tarafından yazıldı.

Geçit paket yönlendiren veya bazen protokoller arası dönüşüm yapan cihazlarla ilişkili bir terim olarak kullanılırdı. 1993 yılı içinde Web geçidi terimi için Web vekil sunucuları (daha önce Web geçidi), intranetlerin güvenliği için en iyi yolu belirleyenlere internet/güvenlik duvarı geçidi (daha önce vekiller) ve Web üzerindeki üçüncü parti bilgi sistemlerine bilgi geçidi (daha önce geçit) denilmesine karar verildi.

Internet/güvenlik duvarı geçitlerine aslında yarı istemci olarak çalışmasına rağmen vekil sunucu ismi verildi. Bilgi geçitleri ise yarı sunucu olarak çalışırlar. Bazen tersinir vekil olarak da bilinir. Günümüzde, bilgi geçidinin çoğunluğu, bir

amaç için yazılmış tek başına çalışan sunucu yerine CGI ve diğer Web sunucu API uygulamalarıyla yapılmaktadır.

4.2.1. CERN Vekil Sunucusu

İlk genel internet/güvenlik duvarı vekil sunucusu CERN vekil sunucusudur (CERN httpd). Aslında bu ürün Web sunucu ile vekil sunucunun karışımı bir çalışmaydı. CERN 'in şu an bu konudaki ürünü W3C httpd'dir.

4.3. Vekil Sunucuların Genel Özellikleri

Vekil sunucuların genel özellikleri;

- 1-) Saydamlık; vekiller üzerindeki filtreleme yapmasını bir yana bırakırsak, sonuçtan kimse etkilenmeyecektir. Kullanıcı, bağlantısı vekil sunucu ya da direk olsa da, yine aynı yanıtları alacaktır.
- 2-) İstemci vekil kullanıp kullanmayacağını belirler.
- 3-) Sunucu arada bulunan vekil sunucu tarafından etki yapılmaz. Bütünüyle onlardan habersizdir.

Yukarıda verilmiş anahtar özellikler örneklerle daha detaylı anlatılacaktır. Özel uygulamalar için yukarıda belirtilmiş özellikler önemlidir. Ama filtreleme saydamlığı etkilemektedir.

4.3.1. Saydamlık

Saydamlık vekil sunucuların genel özelliği olup, kullanıcının vekil sunucunun varlığından haberdar olmasının gerekmemesi anlamını vermektedir. Web erişimi vekil sunucu üzerinden olmaktadır. Genelde, Kullanıcının, isteğinin vekil sunucu üzerinden gittiğine dair bir bilgisi yoktur. Sonuçta kullanıcı vekil sunucu ve merkez sunucunun (origin server) davranışlarını görür. Eğer vekil her hangi bir şekilde transferi engellemezse, vekilsiz çalışmakla aynı anlamda olacaktır. Genellikle depolama (caching) performansı artırır ve vekil kullanıldığında yanıtlar daha hızlı olacaktır. Başka bir ifadeyle, vekil erişim kontrolü ve filtreleme sağlıyorsa kesinlikle

saydam olmayan bir etki gösterecektir ve kullanıcılar vekillerin karışık olduğunu düşünmelerine sebep olacaktır. Ayrıca bir problem oluştuğunda, hata mesajları problemin nereden kaynaklandığını gösterir.

Eskiden saydam olmayan vekil sunucular vardı. İstemci programlar kullanılarak direk olarak erişim sağlanırdı. Asıl URL adresi vekil sunucu URL adresinin arkasına eklenerek bağlantı isteği yapılırdı. Vekil sunucu dokümanı geri aldığı zaman ve bütün referans adreslerini değiştirir ve istemciye gönderirdi.

Örnek olarak; Lagoon saydam olmayan bir vekil sunucudur. Hollanda'da Eindhoven Üniversitesinde Reinier Post tarafından yazılmıştır. Daha sonra saydam hale dönüştürülmüştür. Burada bahsedilen saydamlık sadece URL adresleriyle ilgili saydamlıktır.

Saydam olmayan vekil sunucular, vekil teknolojisindeki ilk adımdır. Bu vekiller, istemci yazılımında hiçbir değişiklik istemeden varolan istemcilerle beraber çalışabiliyorlardı. Daha sonra, istemciler vekilleri fark edecek şekilde yapıldı ve saydam vekil sunuculuk sağlanmış oldu.

4.3.2. İstemci Kontrollü Kullanım

Vekillerin kullanıcılara saydam olmasına rağmen, istemci yazılımları doğrudan merkez sunucu değil de vekiller ile konuştuklarının farkındadırlar. Bu davranışlar istemcinin vekil konfigürasyonu tarafından kontrol edilir.

4.3.3. Merkez Sunucunun Vekil Sunucudan Haberdar Olmaması

Bağlanılmak istenilen web sunucular için (merkez sunucu) isteğin direk olarak istemci ya da vekil sunuculardan geldiği önemli değildir. HTTP protokolü arada bir vekil sunucu olduğu bilgisini sağlar ama genellikle bu bilgi merkez sunucu tarafından dikkate alınmaz.

4.4. Vekil Sunucu Türleri

Birkaç farklı tipte vekil sunucu vardır. Bazıları web erişiminin düzgün olmasını ve depolama (caching) özelliğini sağlayan genel (generic) vekillerdir. Diğerleri, özel uygulamalar için yeni bir takım özellikler eklenmiş sunuculardır. Farklı tipte uygulama seviyesi vekil sunucular;

1-) Genel güvenlik duvarı vekiller (Generic firewall)

2-) Alansal vekiller (Departmental)

3-) Kişisel vekiller (personel)

4-) Özelleştirilmiş vekiller

İstemci ile diğer vekiller arasındaki vekiller

Format çevirimi yapan vekiller

Hızlandırıcılar

5-) Tersinir vekiller (reverse)

4.4.1. Genel Güvenlik Duvarı Vekil Vekil Sunucular

En yaygın olan vekil sunucu tipidir. Web protokolleri trafiğini kontrol ederler. Bunlar HHTP,FTP ve Gopher protokollerinin yanı sıra SSL, HTTPS ve SNEWS gibi güvenli protokolleri kotarırlar.

Genel vekil sunucuların zengin özellikleri vardır. Erişim kontrol, filtreleme, loglama (logging) ve depolama (caching) fonksiyonlarını sağlamaktadır. Bu özelliklerin içerikleri daha sonraki bölümlerde detaylı olarak anlatılacaktır.

Güvenlik duvarı vekil sunucular bir güvenlik duvarı üzerinde çalışırlar. Güvenlik duvarı içeriden gelen istekleri internete yönlendirir, sonuçlarını da tekrar istemciye iletir. Depolama (caching) bu tür vekil sunucularda yaygın olarak

kullanılır. Böylece bazı istekler asıl sunucu (origin server) yerine, kayıtlardan hizmet verilir.



İnternete giden ve gelen bütün trafik güvenlik duvarı vekil sunucuları üzerinden geçmektedir. İnternete erişim için tek bir geçiş noktası vardır. Pratikte birkaç paralel güvenlik duvarı vekil sunucu olabilir. Bir vekil sunucu çok sayıdaki isteklere cevap vermekte yeterli olamayacaktır.

4.4.2. Vekil Zinciri

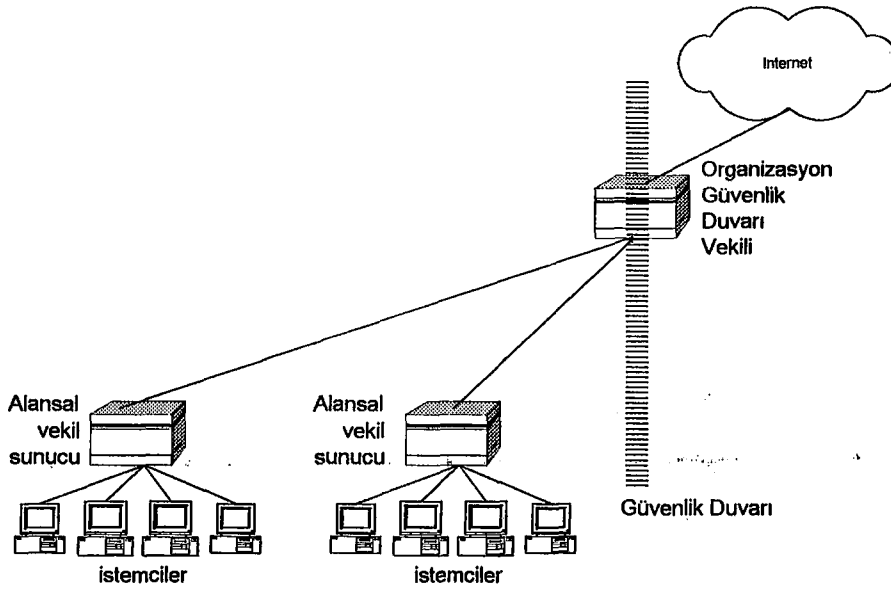
İstemciler, alansal vekil sunucuların oluşturduğu papatya zinciri güvenlik duvarı vekil sunucular üzerinden doküman isteğinde bulunabilirler. Papatya zinciri, alansal vekil sunucuların bir vekil sunucuya direk olarak bağlı olduğu ve isteklerini bu vekil sunucuya ilettiği yapıdır. Alansal vekiller kendi depolarını oluşturabilir ya da ana vekil sunucunun deposunu (cache) kopyalayabilir.

Vekil zinciri ana güvenlik duvarı vekil sunucuları kendi depolarından isteklere yanıt vererek üzerindeki yükü hafifletirler. Alansal vekil sunucudan karşılanamayan isteklerin bir kısmı ana güvenlik duvarı vekil sunuculara yönlendirilir. Ana güvenlik duvarı vekili alansal vekillerden bütün istekleri alır böylece depodan (cache) bulabilme şansı arttırılmış olur.

4.4.3. Alansal Vekil Sunucular

Alansal vekil sunucular genel vekil sunuculardır. Güvenlik duvarı vekil sunucularına benzerler ama kullanıcı tabanı daha dardır. Büyük bir organizasyonunda bir bölüm(alan) için kurulan vekil sunucu yazılımı, güvenlik duvarı üzerinde kullanılan yazılımla aynıdır. Sadece konfigürasyon ayarları farklı olabilir. Örneğin bazı bölümlere giriş diğerlerine göre daha zordur. Bu bölümün erişim kontrolü diğer alanlara göre daha detaylı belirlenmelidir. Alansal vekil sunucular, güvenlik duvarı vekil sunucularla papatya zinciri oluşturmakta olup iki katmandan oluşurlar.

Şekil 4.1 buna bir örnektir.



Şekil 4.1. Alansal Vekil Sunucu

4.4.4. Kişisel Vekil Sunucular

Sadece kişisel kullanıcılar için tasarlanmış vekil sunuculardır. Tipik özelliği aynı konak(host) üzerinde istemci yazılımı olarak çalışırlar. İstemci yazılım ile kişisel vekil özellikleri arasındaki fark belirsizdir. Aslında kişisel vekil sunucu yazılımı tamamen istemci yazılım ile tamamen birleştirilmelidir.

Kişisel vekil sunucu tarafından yerel depolama (caching), aktif depo güncelleme, değişiklikleri alma, yerel arama ve kişisel liste yönetimi (hot list) sağlanır.

4.4.5. Özelleştirilmiş Vekil Sunucular

Özelleştirilmiş vekil sunucular farklı bir gruptur. Bu tür vekil sunucular hedef ortamlar için özelleştirilmiş fonksiyonlar sağlarlar. En iyi örnek, el bilgisayarları (palmtop) üzerinde vekil sunucu istemci yazılımları verilebilir. Bu tip vekil sunucular, alınan resimler üzerinde değişiklik yapar, renk çözünürlüğünü düşürür ve onu el bilgisayarında gösterilebilir formata dönüştürür. Bu şekilde bant genişliği ihtiyacı azaltılmış olur.

Özelleştirilmiş vekil sunucular vekil sunucu ile istemci arasında bulunur.

Bu tür vekil sunuculara başka bir örnekte gerçek vekil sunucuların önüne konulan vekil sunuculardır. Bu tür vekil sunucular java applet filtreleme ve virüs izleme fonksiyonları için kullanılmaktadır.

Yine bu tür sunucular gerçek sunucular önüne konularak kayıtlama (caching) işlemi yaparlar. Bu şekilde ağ performansı arttırılmış olur.

4.5. Dinamik İçerik

Dinamik içerik, doküman ya da doküman da ki bileşenler (resim) ilgili bir terim olup dinamik olarak oluşturulmaları anlamındadır. Dinamik içerik oluşturmanın birkaç yolu vardır. Bunlar;

server-parsed HTML

CGI script

Özelleştirilmiş sunucular

Özel sunucu API uygulamaları

CGI scriptlerin vekil sunucular üzerinde çalıştırılabilir olmaları yaygın bir yanlış anlayıştır. CGI scriptler daima merkez sunucular (origin server) üzerinde çalışırlar ve sadece sonuçlar vekiller üzerinden istemciye iletilir. Vekil sunucular asla CGI script lerin kaynak koduna erişmezler. Bu zaten güvenlik problemlerini ortaya çıkartırdı. Eğer zararlı kullanıcılar CGI kaynağını alabilselerdi, onlar olabilecek güvenlik delikleri için bunları inceleyecek ve sunucu üzerinde çalışan bir CGI scriptin güvenlik deliklerini yayınlayacaklardır.

Hatta eğer vekil sunucuların CGI kaynağına ulaşmasına izin verilmiş olsaydı, pratikte hayatta kalması mümkün olmazdı. CGI scriptler, web sunucular üzerindeki uygulama, veritabanı, dosyalar gibi ihtiyaç duyduğu şeylere güvenirlir.

5. GYTE-ProS VEKİL SUNUCU

5.1. Giriş

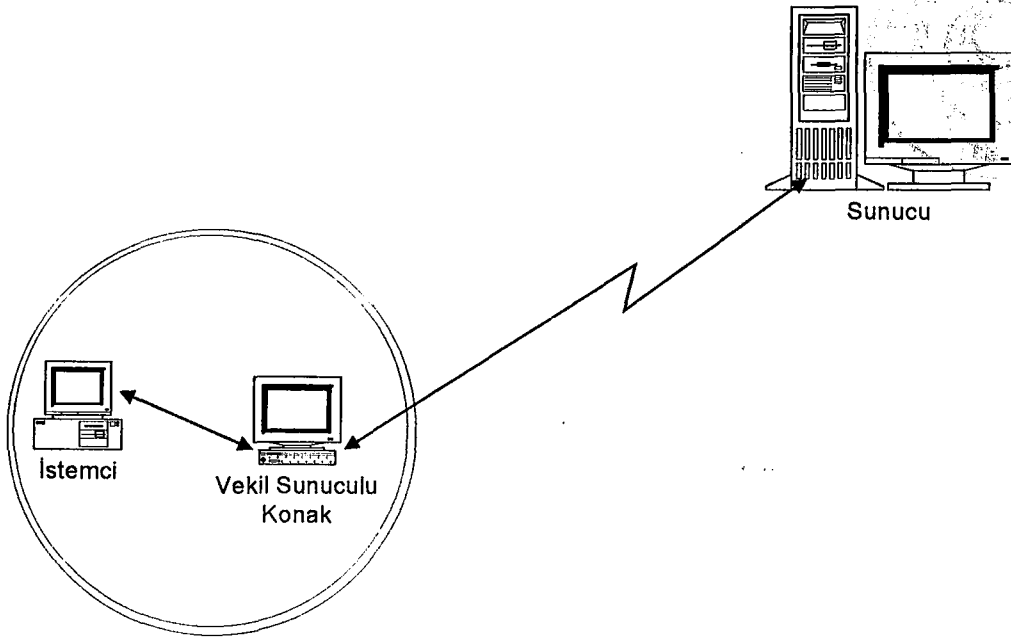
Vekil sunucular daha önceki bölümlerde anlatıldığı gibi sunucu ile istemci arasında bulunmalıdır. İstemciden gelecek isteklerini sunucuya ve yanıtlarını da istemciye geri iletecektir. Vekil sunucu, aktardığı istek ve mesaj içerikleri üzerinde kaydetme, izleme, filtreleme ve erişim denetimi işlemlerini yapacaktır. Bir vekil sunucunun ağ içindeki yeri aşağıdaki şekil 5.1’de gösterilmiştir.

Bir vekil sunucunun genel olarak sağladığı fonksiyonlar;

Erişim Kontrolü (Acces Control) : kullanıcı yada grup bazında onaylama yapılabilir. Böylece kullanıcıların erişimi kısıtlanmış oluyor. Ayrıca log dosyası tutularak, her kullanıcının ne kadar bant genişliği kullandığına dair istatistikler çıkartılabilir.

Kaydetme(Caching) : Vekil sunucudaki anlamı aynı dokümanı isteyen başka birine vekil sunucu tarafından yerel diske kopyalanmış dokümanın gönderilmesidir. Bu performansı artırır, gecikmeyi azaltır. Depolanan dokümanların dinamik olarak kontrol edilip yenilenmesi işlemleri yine vekil sunucuların fonksiyonlarındandır.

İzleme ve Filtreleme (Monitoring & filtering) : İstenmeyen isteklerin bloke edilmesi, zararlı (virus) içeriklerin izlenmesi ve isteniyorsa filtrelenebilir. Bu tez çalışmasında da vekil sunucunun bu fonksiyonları gerçekleştirilmiştir. En yaygın filtreleme türü URL filtrelemesidir.



Şekil 5.1.Vekil Sunuculu Ağ

Vekil sunucular OSI modelindeki uygulama katmanında çalışmaktadır. Her uygulama (http,ftp,...) için özel bir vekil aracı programı kullanılarak ağ haberleşmesi sağlanır. Bu vekil program belirlenen güvenlik politikasına göre kontrol yapar.

Vekil sunucular genelde çift arayüzlü konaklar üzerinde çalışırlar. Dolayısıyla her ağ arayüzü ayrı bir IP numarasına sahiptir. Vekil sunucu bu ağ adaptörlerinden biriyle iç ağa diğeriyle de internete bağlanır. Vekil sunucu, kurumsal politikaya uygun olarak bir ağdan gelen trafiği diğer ağa geçirir ya da iptal eder.

Bu tez çalışmasında amaçlanan, ilk planda izleme ve filtreleme (monitoring & filtering) fonksiyonlarını sağlayan ve http protokolü değerlendirmeleri yapan bir vekil sunucu gerçekleştirildi. Vekil sunucu nesne yönelimli olarak java dilinde yazıldı. Gebze Yüksek Teknoloji Enstitüsü Proxy Server (GYTE-ProS) adı verildi. Testleri 192.168.6.0 ağında 3 bilgisayar çoklu http isteğinde bulunarak yapıldı ve kullanıcıları rahatsız edecek bir performans kaybı görülmedi.

Çok sayıda bilgisayarın bağlı olduğu yerel ağlarda bir GYTE-ProS kullanılması performansı olumsuz olarak etkileyecektir. Böyle bir durumda, GYTE-ProS sayısının artırılması ve dağıtık vekil sunucu yapısına geçilmesi gerekecektir.



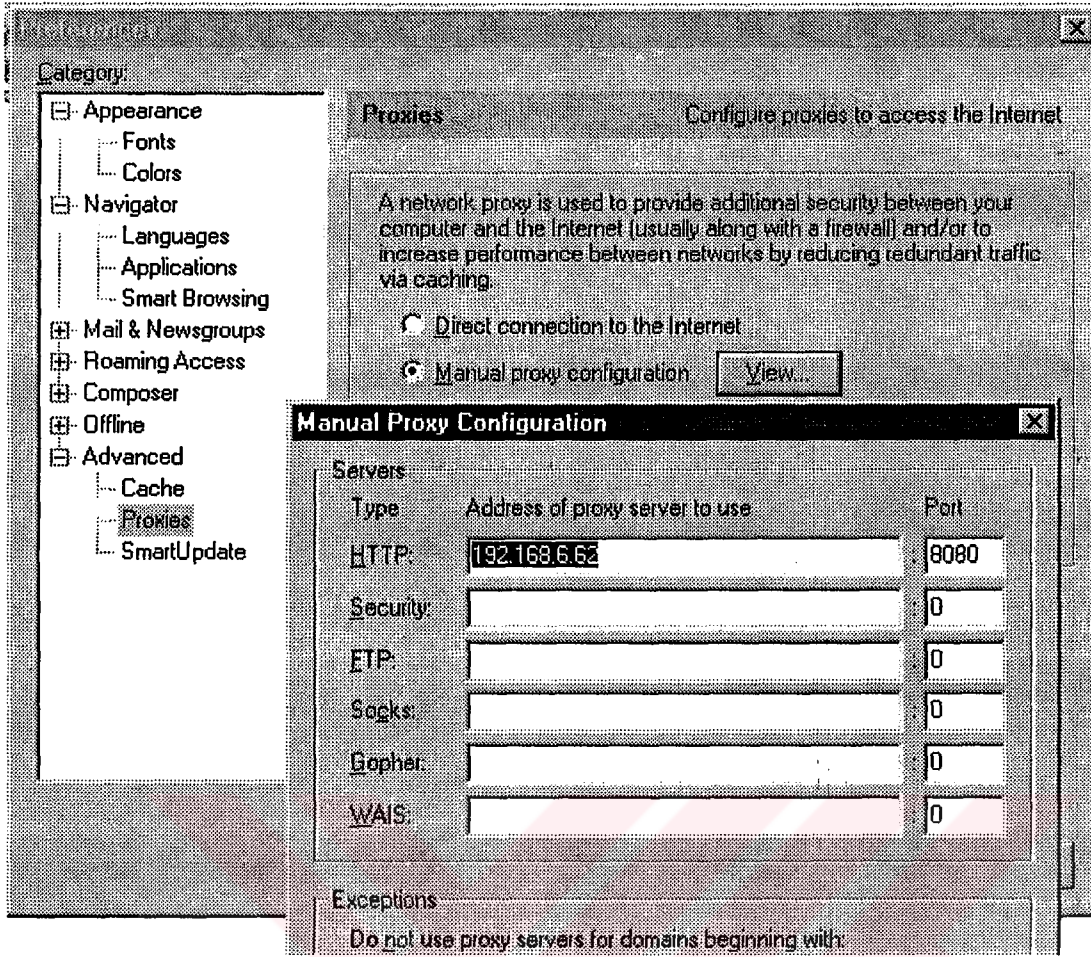
GYTE-ProS'a ileride eklenecek kaydetme mekanizmasıyla performansa olumlu gelişme sağlanacaktır.

5.1.1. Kullanıcı Ayarları

GYTE-ProS üzerinden internete girecek istemcinin, html izleyicisinde (browser) bazı ayarları yapması gerekir.

Netscape browser için;

Netscape menüsünden "Edit/Preferences..." seçeneğinden aşağıdaki pencere açılır. Açılan "Preferences" penceresinde "Category" listesinden "Advanced/proxy" seçeneği seçilir. Daha sonra pencerenin sağ kısmına gelen seçeneklerden "Manual proxy configuration" seçilir ve "view..." tuşuna basılarak vekil sunucu ayarlarının yapılacağı pencere açılır. Bu pencerede servisler, vekil sunucuları ve port numaraları görülmektedir. Bizim vekil sunucumuz http vekil sunucu olduğundan ayarları aşağıdaki gibi olacaktır.



Şekil 5.2. Kullanıcı İçin Vekil Sunucu Ayarları

5.1.2. Geliştirme Ortamı

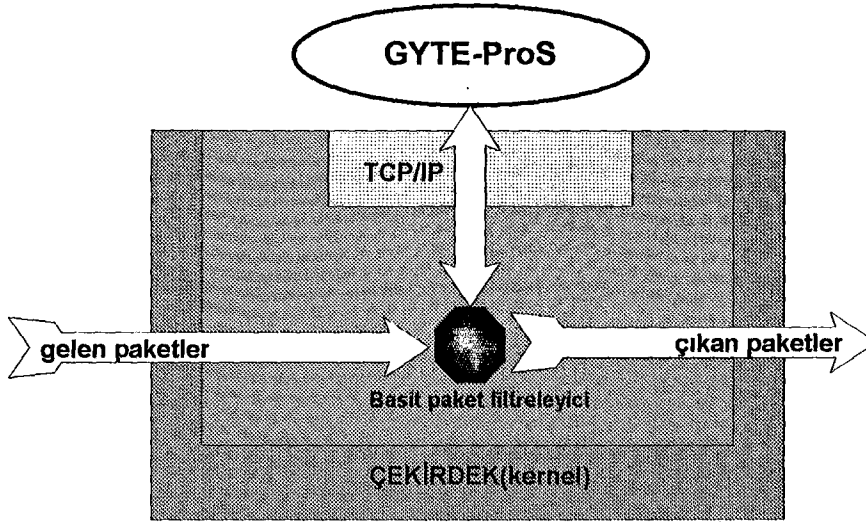
Proje tamamen java programlama dilinde yazılmıştır. Bu yüzden platform bağımsızdır. Borland Jbuilder 2.0 ortamında geliştirilmiştir. JDK 1.1.6 java geliştirme kiti kullanılmıştır.

İlerideki çalışmalar için esneklik sağlamaktadır. Java dilinin getirdiği teknolojilerden (agent, mobile code) faydalanılarak değişik çözümler üretilebilir.

5.1.3. Mimarisi

GYTE-ProS OSI modeline göre TCP katmanı üzerinde çalışmaktadır. Şekil 5.3'den de anlaşılacağı gibi işletim sisteminin kendisine teslim ettiği paketler

üzerinde işlemler yapacaktır. Bu paketlerin ulaştırılabilmesi için gelen paketlerin hedef port ve IP numaraların doğru verilmiş olması gerekir.

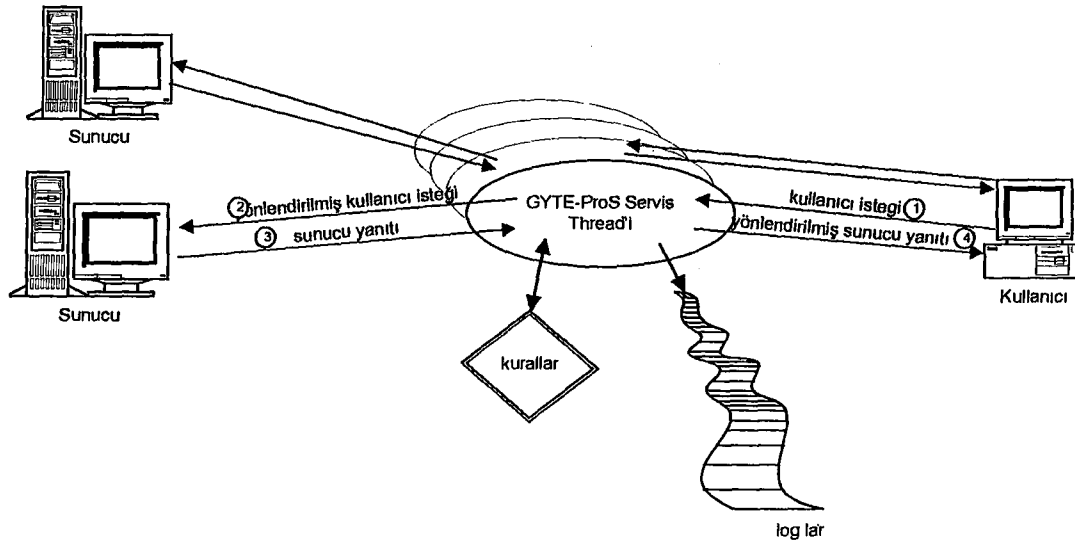


Şekil 5.3. İstemci, GYTE-ProS, Sunucu Mesaj Aktarımı

Çift arayüzlü korumalı konak makinada çalışan vekil sunucuda gelen tüm TCP paketleri alınmaktadır. İçerikleri incelendikten sonra filtrelenmesi gerekmiyorsa paketin geçmesine izin verilir. Aksi takdirde paketin gitmesi engellenir ve log dosyasına kaydedilir. Bunun için html izleyicide (browser) bazı parametre (port) ayarlarının yapılması gerekmektedir. Genel vekil sunucu ayarı, vekil sunucunun verdiği hizmetlerin (HTTP, FTP,...) IP adresi ve port numaraları verilerek yapılır. Kullanıcı ayarları bölümünde anlatılmıştır.

GYTE-ProS şu an için sadece HTTP vekil sunuculuğu yapmaktadır.

GYTE-ProS vekil sunucusunun dinleyici portu 8080 olarak belirlenmiştir. Ana pencere üzerinden GYTE-ProS başlatılmadan bu port numarası değiştirilebilir. GYTE-ProS bu porttan gelen bağlantı isteklerini kabul edecektir. Şekil 5.4'ten de görüleceği gibi bağlantıyı sağlaması için bir nesne atanır. Bağlantı yapıldıktan sonra erişim kontrol, izleme ve filtreleme işlemlerini bu nesne gerçekleştirir. İstemci bağlantıyı kestikten sonra nesne yok edilir.



Şekil 5.4. GYTE-ProS Servis Thread Bağlantı Yapısı

GYTE-ProS portuna, bir kullanıcıdan birden fazla ya da bir anda birden çok kullanıcıdan çok sayıda bağlantı isteği gelebilir. Bu durumda servis thread'i bu istekleri sırayla karşılayacaktır. Bu da cevapların yavaşlaması anlamına gelir.

5.1.4. Kaynak Kodlar

GYTE-ProS vekil sunucusunun çekirdek kaynak kodları www.reithshamer.org adresinden alınmıştır. Alınan kodlar vekil sunucu fonksiyonlarını gerçekleştirmektedir. Sadece bağlantı isteklerini alıp yönlendirme işlemlerini yapmaktadır. Bu kodlara yukarıda anlatılan vekil sunucu fonksiyonlarıyla birlikte kullanıcı arayüzü de ilave edilmiştir.

Yazılım nesne tabanlı olarak geliştirilmiştir. Nesneler arasındaki ilişki ve önemli sınıflar hakkında özet bilgiler aşağıda verilmiştir.

COM.reithshamer.proxy paketindeki sınıflar;

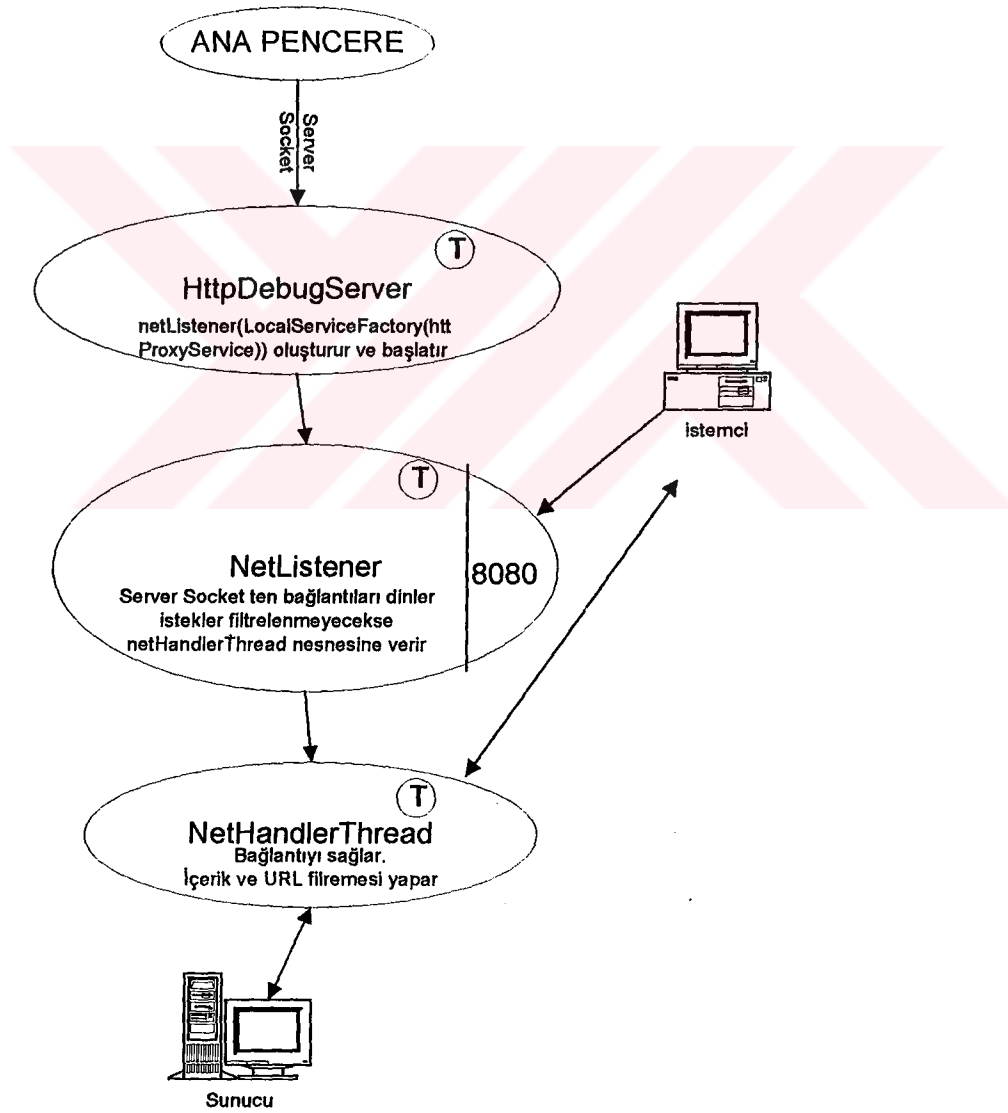
DataTransferThread.java : Bu sınıfta giriş akımından (input stream) alınan veriler çıkış akımına (output stream)' e aktarılır. Bir hata oluştuğunda SockBalanceService durdurulur ve kendini sonlandırır. Veri aktarımları bu sınıf üzerinden yapıldığı için kontroller ve filtrelemeler burada yapılmaktadır.

HttpProxyService.java : Bu sınıf **IdcService** sınıfından türemiştir. **SocketRequest** istekleri olan “default” servislere cevap verir. İstenen hedef URL isteğinin ilk satırını okur, hedef HTTP sunucu için bir socket açar ve düzenlenmiş ilk satırı ona gönderir. Basitçe istemci ile sunucu arasındaki veri transferi işini yapar.

HttpProxySvr.java: Main fonksiyonunu içerir. **NetListener** ve **HttpProxyService** ilklendirmesi için konfigürasyon dosyası ister.

COM.reithshamer.net paketindeki sınıflar;

NetListener.java : Bir sunucu socket açar ve bir porttan bağlantıları dinlemeye başlar. Ana program tarafından çağrılır. Thread ten türetilmiştir.



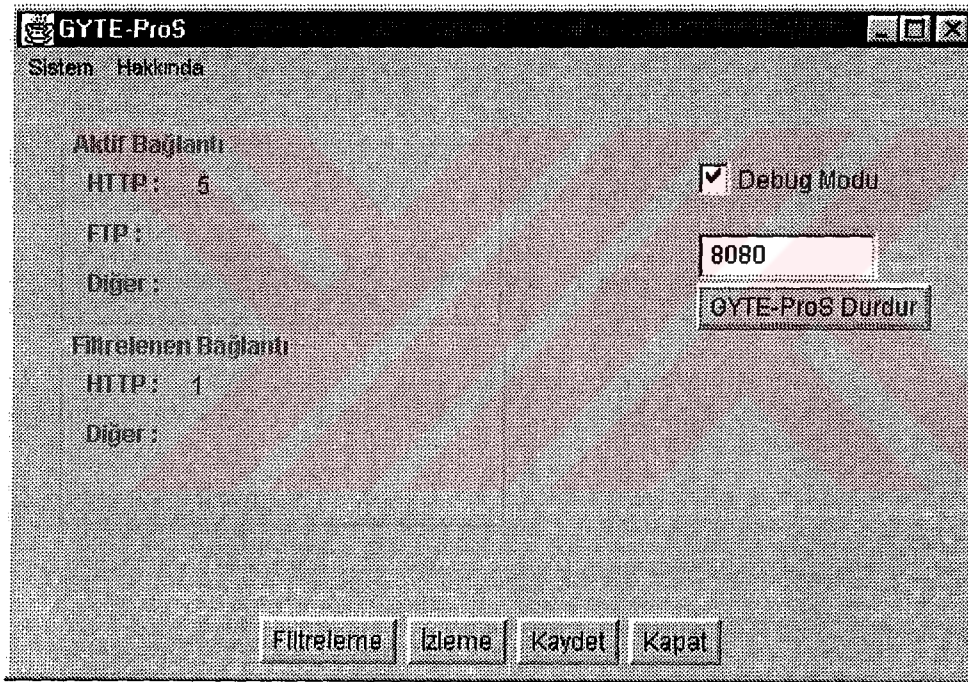
Şekil 5.5. GYTE-ProS Nesne İlişkileri

HHTPDebugServer ana fonksiyonu içeren vekil sunucuyu başlattığımız nesnedir. Bu nesneye ana pencereden port numarası parametre olarak geçilir. Ayrıca ana pencereden bir tuş yardımıyla başlatılıp durdurulabilmektedir.

5.1.5. Pencereleler

5.1.5.1. Ana Pencere

GYTE-ProS bu pencereden başlatılıp sonlandırılır. Aktif ve fitrelenen bağlantı sayıları pencere üzerinde gösterilmektedir. Filtreleme, izleme penceresi bu pencere üzerinden başlatılabilir. GYTE-ProS dinleme portu ve çalışma modu değiştirilebilir. Kapat tuşuyla program sonlandırılır.



Şekil 5.6. Ana Pencere

5.1.5.2. Filtreleme Penceresi

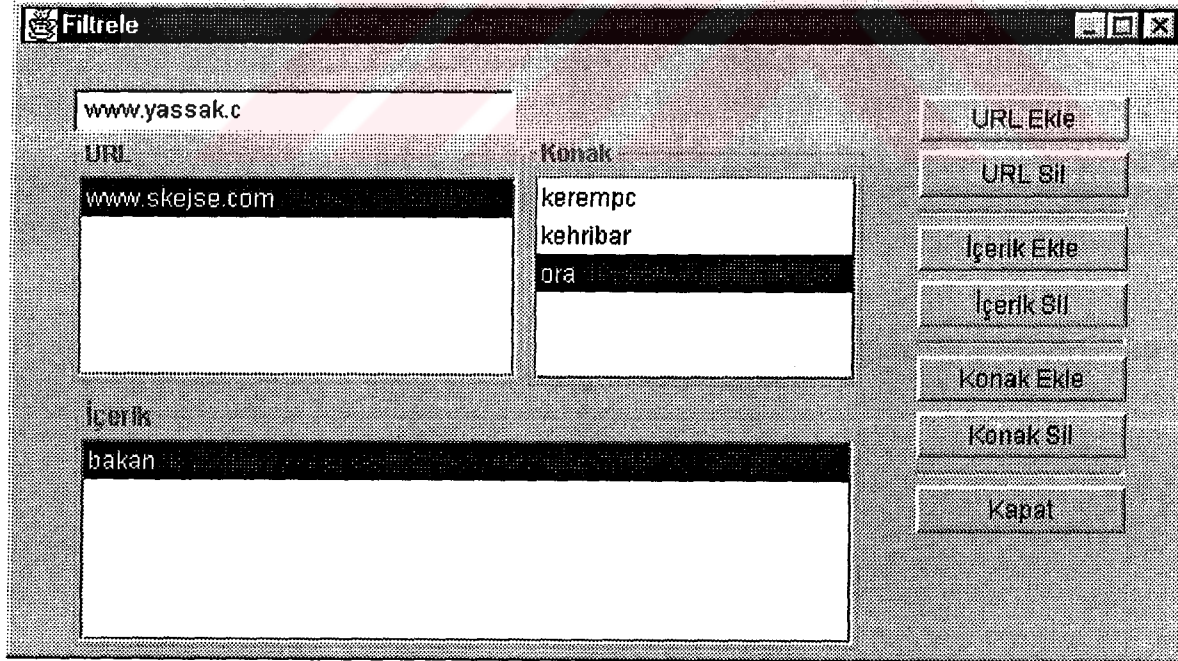
Vekil sunucu üzerinden yapılan bağlantılar buradaki kurallara göre filtrelenecektir. Bu pencereden, filtrelenmesi istenilen URL adresi, konak ismi ya da aktarılan verinin içeriği girilir.

Burada girilen kurallar GYTE-ProS için kuralları içeren bir el kitabı gibidir. Gelen bağlantı istekleri ve geçecek mesajların içerikleri burada girilmiş olan anahtar kelimeler arandıktan sonra geçmesine izin verilir.

Burada girilen kuralların az olması vekil sunucun performansını direk olarak etkileyecektir. Bundan dolayı anahtar kelimelerin az girilmesi gerekir. GYTE-ProS bu sayıyı 20 ile sınırlamıştır.

Konak adı kontrolü bilgisayardan alınarak yapılmaz. DNS'de kayıtlı olan isme göre yapılmaktadır. Bu yüzden bir yerel ağda vekil sunucuyu kullanarak dışarı açılan konakların DNS kayıtlarının mutlaka yapılması gerekir. Konak adı DNS'e kayıt edilmemiş konakların IP numaraları verilerek de kontrolleri yapılabilir. Konak adı yada IP vererek bir kullanıcının dışarı açılmasını engellemek oldukça zordur. Çünkü kullanıcı IP numarası ve konak adını rahatlıkla değiştirebilir. Hatta MAC adresini de ethernet kartını değiştirerek yapabilir.

Pencerenin sağ tarafında bulunan tuşlarla seçili olan satırlar silinebilir ve ekleme yapılabilir.



Şekil 5.7. Filtreleme Penceresi



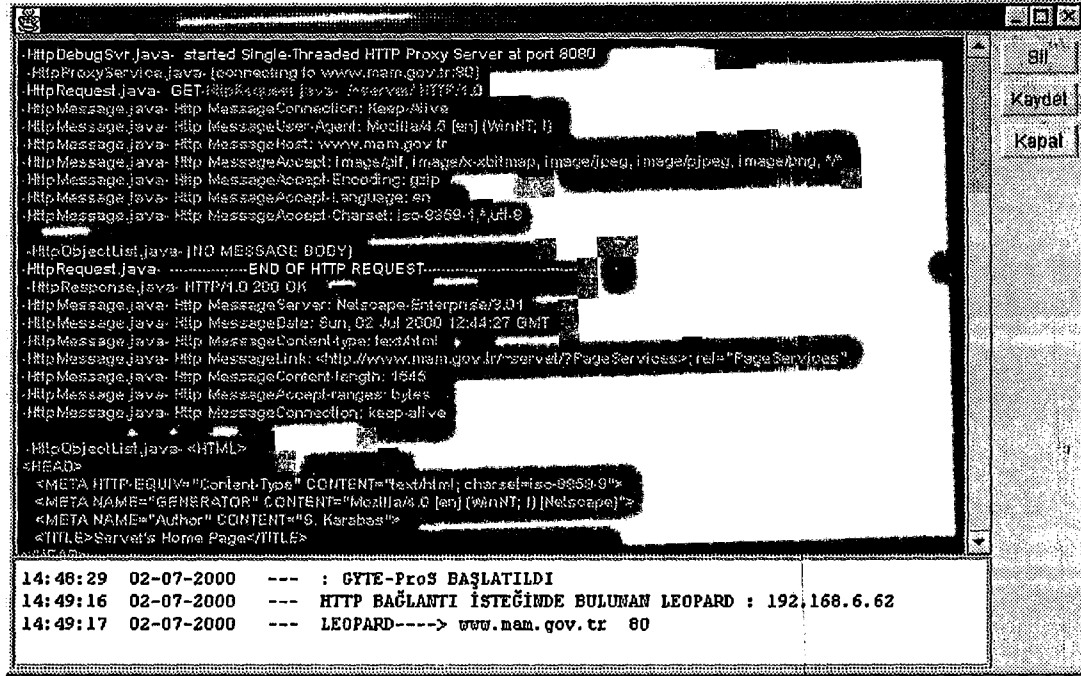
5.1.5.3. İzleme Penceresi

İzleme penceresinde iki temel ekran vardır. Üstteki ekranda istemci ve sunucular arasında akan veriler gösterilmektedir. Alt ekranda ise filtrelenen ya da bağlantı yapan konaklara ait bilgiler verilmektedir.

Pencerenin üstünde bulunan ekranda, istemci sunucu arasındaki mesajlar istemci cevabı ve sunucu yanıtı olarak verilmektedir. Http isteğinde bulunan kullanıcının istek mesajı http başlık yapısına göre parçalanıp gösterilmiştir. Ekranda http mesaj akışını rahatlıkla izleyebilirsiniz. Bu http mesaj yapısının ve çalışmasının anlaşılması için yeterlidir. Ayrıca bu ekranda mesajların hangi nesnelerden verildiği de belirtilmiştir. Kaynak kod üzerinde değişiklik yapacak kısma ait sınıf bu ekrandan kolaylıkla tahmin edilebilir. Aktarılan ikilik koddaki dosyalar anlaşılamayacağı ve üzerinde inceleme yapılmayacağı için pencerede gösterilmemiştir.

Pencerenin altında bulunan alt ekranda daha çok özet bilgiler sunulmuştur. Vekil sunucu ne zaman açıldığı, konakların hangi url adreslerine bağlandıkları burada gösterilmektedir. Filtrelenen bağlantı varsa hangi içerikten kimlerin bağlantılarının engellendiğini yine bu pencereden izlenebilir.

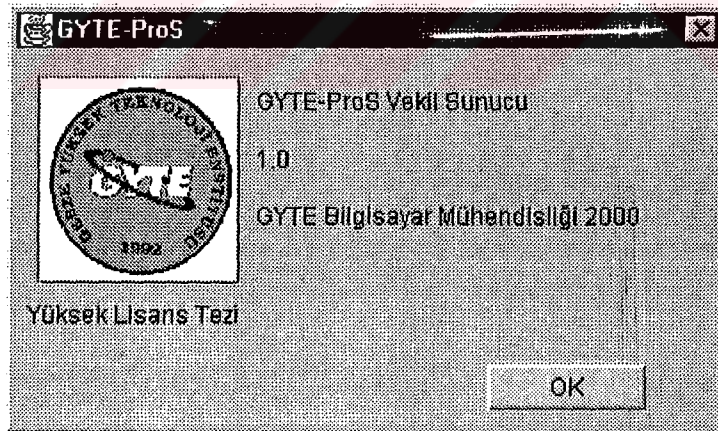
Sil tuşuyla ekran temizlenir. Kaydet tuşuyla ekranlar dosyaya kaydedilir. Kapat tuşuyla da ekran kapatılır.



Şekil 5.8. İzleme Penceresi

5.1.5.4. Hakkında Penceresi

Program Borland Jbuilder 2.0 da tamamen nesne tabanlı olarak java diliyle gerçekleştirilmiştir. Çekirdek kod www.reithshamer.org adresinden alınmıştır.



Şekil 5.9. Hakkında Penceresi

6. SONUÇ

Bu çalışmada internet üzerinden verilen hizmetler anlatılmış ve bu hizmetlerin güvenlik problemleri üzerinde durulmuştur. Oluşan güvenlik problemlerine karşı üretilen güvenlik duvarı mimari çözümleri de anlatılmıştır. Örnek olarak bir uygulama seviyesi güvenlik duvarı gerçekleştirilmiştir.

Bir çok ticari yazılım ve donanım aracından oluşan bir ağı korumak oldukça zordur. Zamanla bu yazılım ve donanımlarla ilgili güvenlik problemleri çıkmaktadır ve çıkacaktır. Bu nedenle her kurumun mutlaka bir güvenlik politikasının olması ve ciddiyetinin anlaşılması gerekmektedir. Bu konuda çok ciddi çalışmalar vardır ve devam etmektedir. Tez çalışmasında güvenlik politikası üzerinde de durulmuştur.

Tez çalışmasında gerçekleştirilen uygulama seviyesi güvenlik duvarı (GYTE-ProS vekil sunucusu) tasarımı java dilinde yazılmış olup basit paket filtreleme ve izleme fonksiyonlarını sağlamaktadır. Vekil sunucu üzerinden geçen paketler ekrandan izlenebilmektedir. Böylece istemci ile sunucu arasındaki paket haberleşmesi gözlenebilmekte ve verilerin aktarım mantığı anlaşılmaktadır. Ayrıca belirlenen politikaya paralel olarak verilen kurallara göre paketler vekil sunucu tarafından filtrelenirler.

Bu çalışma ile vekil sunucuların performansı çokta düşürmeden güvenli bir erişim sağladığı ayrıca yaptığı kayıtların internet bağlantısı üzerinde çalışan kişilere yararlı olduğu görülmüştür. Bu vekil sunucu ile kullanıcıların, kurumsal politikalara uygun olarak internet ortamına rahatça girmeleri sağlanmıştır. Kurumsal politikaya uymayan kişilerin bağlantıları kesilmekte ve yetkililer tarafından incelenebilecek şekilde kaydedilmektedir.

GYTE-ProS Vekil sunucusunun ilerideki çalışmalar için esnek bir yapısı vardır. Vekil sunucuya kaydetme (caching) özelliği eklenebilir. Vekil sunucu sayısı artırılıp, javanın getirdiği yeni teknolojilerden de faydalanarak güvenli ve verimli bir mimari oluşturulabilir. GYTE-ProS sayısı bir ağ içinde artınca, bunların izlenmesi ve

yönetimi gerekecektir. www.besiex.org adresinde çoklu vekil sunucular için çeşitli dağıtık sistemler önerilmektedir. GYTE-ProS'ta geliştirilen dağıtık yapılara açıktır.



KAYNAKLAR

- 1- Ari Luotonen, "Web Proxy Servers", Prentice Hall, 1998
- 2- AXENT Technologies, "Everything you need to know about network security", <http://www.axent.com>, 1998
- 3- AXENT Technologies, "Information Security Begins with Sound Security Policies", <http://www.axent.com>, 1999
- 4- C. P. Pfleeger, *Security in Computing*, Prentice Hall, 1997. Sayfa: 426-437.
- 5- CERT Advisory CA-94.15, NFS Vulnerability, <http://www.cert.org>
- 6- CERT Advisory CA-96.21, TCP SYN Flooding and IP Spoofing Attacks, <http://www.cert.org>
- 7- CERT Advisory CA-98.01, "smurf" IP Denial of Service Attacks, <http://www.cert.org>
- 8- D. Brent Chapman, Elizabeth D. Zwicky, D. Russell, *Building Internet Firewalls*, O'Reilly & Associates, 1995
- 9- ElGamal, T., A public key cryptosystem and a signature scheme based on discrete logarithms, *Advances in Cryptology CRYPTO '84*, LNCS 196, Springer Verlag, 1985
- 10- Marcus Gonzalves, "Firewalls Complete", McGraw-Hill 1998
- 11- NIST Information Technology Lab., Internet Security Policy, NIST special publ. 800-XX, 1997
- 12- S. W. Lodin, C. L. Schuba, "Firewalls Fend off Invasions from the Net", *IEEE Spectrum*, February 1998, Vol. 35 Num. 2.

- 13- Simson Garfinkel, Gene Spafford, "*Practical Unix & Internet Security*", O'Reilly & Associates", 1996
- 14- Stallings W., *Network and Internetwork Security*, Prentice Hall, 1995
- 15- T. Köprülü, el. al., "Interceptor: Bir Güvenlik Duvarı Tasarımı", *Bilgisayar Ağları Sempozyumu 98 (BAS'98)*, 1998
- 16- www.besiex.org
- 17- www.microsoft.com
- 18- www.reithshamer.org
- 19- www.tis.com



ÖZGEÇMİŞ

Servet Haydar KARABAŞ, 1976 Osmaniye/Bahçe doğumludur. Adana Erkek Lisesi'nden sonra 1996 yılında Uludağ Üniversitesi, Müh-Mim Fakültesi, *Elektronik müh. bölümü*'nü bitirmiştir. Halen TÜBİTAK-UEKAE' de (Ulusal Elektronik ve Kriptoloji Araştırma Enstitüsü) araştırmacı olarak çalışmaktadır. Çalışma konuları cihaz sürücüler, ağ yönetimi ve güvenliği üzerinedir.





EK 1. GYTE-ProS KAYNAK KODLARI

AnaPencere.java

```
//Title: GYTE-ProS Proxy Server Projesi
//Version:
//Copyright: Copyright (c) 1999
//Author: Servet Haydar Karabaş
//Company: GYTE
//Description: Proxy Server yuksek lisans tezi olup
//aşağıdaki fonksiyonları gerçeklemektedir
//1. Monitoring
//2. filtering
//3. logging
package gui;
import COM.reitshamer.proxy.*;

import java.awt.*;
import borland.jbcl.layout.*;
import com.sun.java.swing.border.*;
import borland.jbcl.control.*;
import com.sun.java.swing.*;
import java.awt.event.*;

public class AnaPencere extends JFrame {

    AnaPencereMenu anaPencereMenu = new AnaPencereMenu(this);

    Border emptyBorder = BorderFactory.createEmptyBorder(20,20,20,20);
    TitledBorder titledBorderAktif = BorderFactory.createTitledBorder("Aktif Bağlantı");
    TitledBorder titledBorderFiltre = BorderFactory.createTitledBorder("Filtrelenen Bağlantı");

    JPanel jPanel1 = new JPanel();
    JPanel jPanel2 = new JPanel();
    ButtonControl buttonControlFiltre = new ButtonControl();
    ButtonControl buttonControlIzleme = new ButtonControl();
    ButtonControl buttonControlKaydet = new ButtonControl();
    ButtonControl buttonControlKapat = new ButtonControl();
    BorderLayout borderLayout1 = new BorderLayout();
    XYLayout xYLayout1 = new XYLayout();
    JPanel jPanel3 = new JPanel();
    JPanel jPanel4 = new JPanel();
    JLabel textControl1 = new JLabel();
    JLabel textControl2 = new JLabel();
    JLabel textControl3 = new JLabel();
    XYLayout xYLayout2 = new XYLayout();
    XYLayout xYLayout3 = new XYLayout();
    JLabel jLabelDigerAktif = new JLabel();
    JLabel jLabelHttpAktif = new JLabel();
    JLabel jLabelFtpAktif = new JLabel();
    JLabel jLabelDigerFiltre = new JLabel();
    JLabel textControl4 = new JLabel();
    JLabel textControl6 = new JLabel();
    JLabel jLabelHttpFiltre = new JLabel();

    FiltreFrame filtreFrame = new FiltreFrame();
    IzlemePenceresi izlemePenceresi = IzlemePenceresi.ornek;

    HttpDebugSvr GYTE_ProS = new HttpDebugSvr();
```

```

ButtonControl buttonControlProS = new ButtonControl();
TextFieldControl textFieldControlProxyPort = new TextFieldControl();
CheckboxControl checkboxControlDebugMod = new CheckboxControl();
TextControl textControlHTTPSayisi = new TextControl();
TextControl textControlFiltrelenenHTTPSayisi = new TextControl();

public AnaPencere() {
    try {
        jbInit();
    }
    catch (Exception e) {
        e.printStackTrace();
    }
}

public static void main(String[] args) {
    try {
        //UIManager.setLookAndFeel(new com.sun.java.swing.plaf.windows.WindowsLookAndFeel());
        //UIManager.setLookAndFeel(new com.sun.java.swing.plaf.motif.MotifLookAndFeel());
        UIManager.setLookAndFeel(new com.sun.java.swing.plaf.metal.MetalLookAndFeel());
    }
    catch (Exception e) {
    }
    AnaPencere anaPencere1 = new AnaPencere();
    anaPencere1.pack();
    anaPencere1.setVisible(true);
}

private void jbInit() throws Exception {
    this.getContentPane().setLayout(borderLayout1);
    jPanel1.setLayout(xYLayout1);
    this.setTitle("GYTE-ProS");
    this.setMenuBar(anaPencereMenu);
    jPanel1.setBorder(emptyBorder);
    jPanel2.setBackground(new Color(192, 192, 255));
    buttonControlFiltre.setLabel("Filtreleme");
    buttonControlFiltre.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(ActionEvent e) {
            buttonControlFiltre_actionPerformed(e);
        }
    });
    buttonControlIzleme.setLabel("İzleme");
    buttonControlIzleme.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(ActionEvent e) {
            buttonControlIzleme_actionPerformed(e);
        }
    });
    buttonControlKaydet.setLabel("Kaydet");
    buttonControlKaydet.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(ActionEvent e) {
            buttonControlKaydet_actionPerformed(e);
        }
    });
    buttonControlKapat.setLabel("Kapat");
    buttonControlKapat.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(ActionEvent e) {
            buttonControlKapat_actionPerformed(e);
        }
    });
}

```

```

jPanel3.setLayout(xYLayout2);
jPanel3.setBorder(titledBorderAktif);
jPanel4.setBorder(titledBorderFiltre);
jPanel4.setLayout(xYLayout3);
textControl1.setText("HTTP : ");
textControl2.setText("FTP : ");
textControl3.setText("Diğer : ");
jLabelDigerAktif.setText(" ");
jLabelDigerFiltre.setText(" ");
textControl4.setText("HTTP : ");
textControl6.setText("Diğer : ");
jLabelHttpFiltre.setText(" ");
buttonControlProS.setLabel("GYTE-ProS Başlat");
buttonControlProS.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        buttonControlProS_actionPerformed(e);
    }
});
textFieldControlProxyPort.setText("8080");
checkboxControlDebugMod.setChecked(true);
checkboxControlDebugMod.setLabel("Debug Modu");
textControlHTTPSayisi.setText("5");
textControlFiltrelenenHTTPSayisi.setText("1");
jLabelHttpAktif.setText(" ");
jLabelFtpAktif.setText(" ");
this.getContentPane().add(jPanel1, BorderLayout.EAST);
jPanel1.add(jPanel3, new XYConstraints(0, 0, 224, 100));
jPanel3.add(textControl1, new XYConstraints(10, 0, -1, -1));
jPanel3.add(textControl2, new XYConstraints(10, 25, 40, 15));
jPanel3.add(textControl3, new XYConstraints(10, 50, 40, 15));
jPanel3.add(jLabelDigerAktif, new XYConstraints(90, 50, 40, -1));
jPanel3.add(jLabelHttpAktif, new XYConstraints(90, 0, 40, -1));
jPanel3.add(jLabelFtpAktif, new XYConstraints(90, 25, 40, -1));
jPanel3.add(textControlHTTPSayisi, new XYConstraints(64, 1, -1, -1));
jPanel1.add(jPanel4, new XYConstraints(0, 100, 226, 100));
jPanel4.add(jLabelDigerFiltre, new XYConstraints(90, 25, 40, -1));
jPanel4.add(textControl4, new XYConstraints(10, 0, -1, -1));
jPanel4.add(textControl6, new XYConstraints(10, 25, 40, -1));
jPanel4.add(jLabelHttpFiltre, new XYConstraints(90, 0, 40, -1));
jPanel4.add(textControlFiltrelenenHTTPSayisi, new XYConstraints(62, 1, -1, -1));
jPanel1.add(buttonControlProS, new XYConstraints(319, 77, 117, -1));
jPanel1.add(textFieldControlProxyPort, new XYConstraints(320, 52, 89, -1));
jPanel1.add(checkboxControlDebugMod, new XYConstraints(320, 14, 102, -1));
this.getContentPane().add(jPanel2, BorderLayout.SOUTH);
jPanel2.add(buttonControlFiltre, null);
jPanel2.add(buttonControlIzleme, null);
jPanel2.add(buttonControlKaydet, null);
jPanel2.add(buttonControlKapat, null);
}

void buttonControlKapat_actionPerformed(ActionEvent e) {
    System.exit(0);
}

void buttonControlKaydet_actionPerformed(ActionEvent e) {
}

void buttonControlIzleme_actionPerformed(ActionEvent e) {
    izlemePenceresi.openForm(); // formu uygun boya getiriyor (içindeki componentlere uygun hale getiriyor).
}

```

```

    }

    void buttonControlFiltre_actionPerformed(ActionEvent e) {
        filtreFrame.openForm(); // formu uygun boya getiriyor (içindeki componentlere uygun hale
        getiriyor).
    }

    public void hakkındaCalistir(){
        HakkindaPenceresi dlg = new HakkindaPenceresi(this, "GYTE-ProS", true);
        dlg.setLocation(100,100);
        dlg.show();
        dlg.dispose();
    }

    void buttonControlProS_actionPerformed(ActionEvent e) {
        if( buttonControlProS.getLabel() == "GYTE-ProS Başlat" ){
            buttonControlProS.setLabel("GYTE-ProS Durdur");
            GYTE_ProS.baslat( new String[] {textFieldControlProxyPort.getText()});
            // filtrelenecek url mesaj yada hostlar String dizilerine alınır
            // Mesaj Filtrelemesi htmlMesaj.javada
            // url filtrelemesi HttpProxyService.java
            // konak filtrelemesi ... da yapılmaktadır
            filtreFrame.filtreStringOku();
            filtreFrame.filtreStringKaydet();
        }
        else{
            buttonControlProS.setLabel("GYTE-ProS Başlat");
            GYTE_ProS.durdur();
        }
    }
}

```

FiltreFrame.Java

```

package gui;

import java.awt.*;
import java.io.*;
import com.sun.java.swing.border.*;
import com.sun.java.swing.*;
import borland.jbcl.layout.*;
import borland.jbcl.control.*;
import java.awt.event.*;
import jclass.bwt.*;

public class FiltreFrame extends JFrame {

    Border emptyBorder = BorderFactory.createEmptyBorder(20,20,20,20);
    TitledBorder titledBorderUrl = BorderFactory.createTitledBorder("URL");
    TitledBorder titledBorderIcerik = BorderFactory.createTitledBorder("İçerik");
    TitledBorder titledBorderKonak = BorderFactory.createTitledBorder("Konak");

    JPanel jPanel1 = new JPanel();
    JPanel jPanel2 = new JPanel();
    VerticalFlowLayout verticalFlowLayout1 = new VerticalFlowLayout();
    ButtonControl buttonControlUrlEkle = new ButtonControl();
    ButtonControl buttonControlIcerikEkle = new ButtonControl();
    BorderLayout borderLayout1 = new BorderLayout();
}

```



```

XYLayout xYLayout1 = new XYLayout();
JPanel jPanel3 = new JPanel();
TextFieldControl textFieldControl = new TextFieldControl();
ListControl listControlUrl = new ListControl();
ListControl listControlIcerik = new ListControl();
ButtonControl buttonControlKapat = new ButtonControl();
JPanel jPanel4 = new JPanel();
ButtonControl buttonControlUrlSil = new ButtonControl();
ButtonControl buttonControlIcerikSil = new ButtonControl();
jclass.bwt.JCSeparator jCSeparator1 = new jclass.bwt.JCSeparator();
jclass.bwt.JCSeparator jCSeparator2 = new jclass.bwt.JCSeparator();
GridLayout gridLayout1 = new GridLayout();
GridLayout gridLayout2 = new GridLayout();
JPanel jPanel5 = new JPanel();
GridLayout gridLayout3 = new GridLayout();
ListControl listControlKonak = new ListControl();
ButtonControl buttonControlKonakSil = new ButtonControl();
ButtonControl buttonControlKonakEkle = new ButtonControl();
jclass.bwt.JCSeparator jCSeparator3 = new jclass.bwt.JCSeparator();

```

```

public String fitrelekonakAdi[] = new String[20];
public String fitrelemesajIcerigi[] = new String[20];
public String fitrelewebadres[] = new String[20];

```

```

public FiltreFrame() {
    try {
        jbInit();
    }
    catch (Exception e) {
        e.printStackTrace();
    }
}

```

```

public static void main(String[] args) {
    FiltreFrame filtreFrame1 = new FiltreFrame();
    filtreFrame1.pack();
    filtreFrame1.setVisible(true);
}

```

```

private void jbInit() throws Exception {
    this.getContentPane().setLayout(borderLayout1);
    jPanel2.setLayout(verticalFlowLayout1);
    jPanel1.setLayout(xYLayout1);
    this.setTitle("Filtrele");
    jPanel2.setBorder(emptyBorder);
    buttonControlUrlEkle.setLabel("URL Ekle");
    buttonControlUrlEkle.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(ActionEvent e) {
            buttonControlUrlEkle_actionPerformed(e);
        }
    });
    buttonControlIcerikEkle.setLabel("İçerik Ekle");
    buttonControlIcerikEkle.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(ActionEvent e) {
            buttonControlIcerikEkle_actionPerformed(e);
        }
    });
    buttonControlUrlSil.setLabel("buttonControl3");
    jPanel3.setLayout(gridLayout2);
}

```

```

jPanel3.setBorder(titledBorderUrl);
listControlUrl.setAutoInsert(false);
listControlUrl.setAutoEdit(false);
listControlUrl.setEditInPlace(false);
listControlUrl.setGrowEditor(false);
listControlUrl.setShowFocus(false);

```

```

buttonControlUrlSil.setLabel("URL Sil");
buttonControlIcerikSil.setLabel("İçerik Sil");
jPanel5.setBorder(titledBorderKonak);
listControlKonak.setEditInPlace(false);
listControlKonak.setGrowEditor(false);
listControlKonak.setShowFocus(false);
buttonControlKonakSil.setLabel("Konak Sil");
buttonControlKonakEkle.setLabel("Konak Ekle");

```

```

buttonControlKonakEkle.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        buttonControlKonakEkle_actionPerformed(e);
    }
});
buttonControlKonakSil.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        buttonControlKonakSil_actionPerformed(e);
    }
});
listControlKonak.setAutoInsert(false);
listControlKonak.setColumnName("");
listControlKonak.setAutoEdit(false);
jPanel5.setLayout(gridLayout3);
buttonControlIcerikSil.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        buttonControlIcerikSil_actionPerformed(e);
    }
});
buttonControlUrlSil.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        buttonControlUrlSil_actionPerformed(e);
    }
});
listControlIcerik.setEditInPlace(false);
listControlIcerik.setGrowEditor(false);
listControlIcerik.setShowFocus(false);
jPanel4.setBorder(titledBorderIcerik);
listControlIcerik.setAutoInsert(false);
listControlIcerik.setColumnName("");
listControlIcerik.setAutoEdit(false);
this.getContentPane().add(jPanel1, BorderLayout.CENTER);
jPanel1.add(jPanel3, new XYConstraints(25, 47, 228, 125));
jPanel3.add(listControlUrl, null);
jPanel1.add(textFieldControl, new XYConstraints(26, 22, 221, -1));
jPanel1.add(jPanel4, new XYConstraints(27, 178, 395, 125));
jPanel4.add(listControlIcerik, null);
jPanel1.add(jPanel5, new XYConstraints(252, 46, 171, 127));
jPanel5.add(listControlKonak, null);
this.getContentPane().add(jPanel2, BorderLayout.EAST);
jPanel2.add(buttonControlUrlEkle, null);
jPanel2.add(buttonControlUrlSil, null);
jPanel2.add(jCSeparator1, null);

```

```

jPanel2.add(buttonControlIcerikEkle, null);
jPanel2.add(buttonControlIcerikSil, null);
jPanel2.add(jCSeparator2, null);
jPanel2.add(buttonControlKonakEkle, null);
jPanel2.add(buttonControlKonakSil, null);
jPanel2.add(jCSeparator3, null);
jPanel2.add(buttonControlKapat, null);
buttonControlKapat.setLabel("Kapat");
buttonControlKapat.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        buttonControlKapat_actionPerformed(e);
    }
});
jPanel4.setLayout(gridLayout1);
}

void buttonControlUrlEkle_actionPerformed(ActionEvent e) {
    //listControlUrl.sh();
    listControlUrl.addItem(textFieldControl.getText());
}

void buttonControlIcerikEkle_actionPerformed(ActionEvent e) {
    listControlIcerik.transferFocus();
    listControlIcerik.addItem(textFieldControl.getText());
}

void buttonControlKonakEkle_actionPerformed(ActionEvent e) {
    listControlKonak.transferFocus();
    listControlKonak.addItem(textFieldControl.getText());
}

void buttonControlUrlSil_actionPerformed(ActionEvent e) {
    listControlUrl.removeItem(listControlUrl.getSubfocus());
}

void buttonControlKonakSil_actionPerformed(ActionEvent e) {
    listControlKonak.removeItem(listControlKonak.getSubfocus());
}

public void filtreStringOku(){
    readFromFile("URLFiltre.txt", listControlUrl);
    readFromFile("IcerikFiltre.txt", listControlIcerik);
    readFromFile("KonakFiltre.txt", listControlKonak);
}

public void filtreStringKaydet(){
    int tooo= listControlUrl.getCount();
    for(int i=0;i<tooo; i++)
        fitrelewebadres[i]= listControlUrl.get(i).toString();
    for(int i=tooo; i<20 ; i++)
        fitrelewebadres[i]= "";

    tooo= listControlIcerik.getCount();
    for(int i=0;i<tooo; i++)
        fitrelemesajIcerigi[i]= listControlIcerik.get(i).toString();
    for(int i=tooo; i<20 ; i++)
        fitrelemesajIcerigi[i]= "";

    tooo= listControlKonak.getCount();
    for(int i=0;i<tooo; i++)
        fitrelekonakAdi[i]= listControlKonak.get(i).toString();
    for(int i=tooo; i<20 ; i++)

```

```

        fitrelekonakAdi[i] = "";
    }

```

```

void saveToFile(String fileName, ListControl listControl){
    try {
        File file = new File (fileName);
        FileWriter out = new FileWriter(file);
        for(int i=0;i<listControl.getCount(); i++)
            out.write(listControl.get(i).toString()+"\n");
        out.close();
    }
    catch (IOException e) {
    }
}

```

```

void readFromFile(String fileName, ListControl listControl){
    try {
        listControl.removeAllItems();
        File file = new File (fileName);
        FileReader in = new FileReader(file);

        LineNumberReader lnr = new LineNumberReader(in);
        int mln = 10;
        //int mln = lnr.getLineNumber();
        while(lnr.ready())
            listControl.addItem(lnr.readLine());
        lnr.close();
    }
    catch (Exception e) {
        e.printStackTrace();
    }
}

```

```

public void openForm(){
    textFieldControl.setText("");
    readFromFile("URLFiltre.txt", listControlUrl);
    readFromFile("IcerikFiltre.txt", listControlIcerik);
    readFromFile("KonakFiltre.txt", listControlKonak);
    pack();
    show();
}

```

```

void buttonControlKapat_actionPerformed(ActionEvent e) {
    saveToFile("URLFiltre.txt",listControlUrl);
    saveToFile("IcerikFiltre.txt",listControlIcerik);
    saveToFile("KonakFiltre.txt",listControlKonak);
    setVisible(false);

}

```

```

void buttonControlIcerikSil_actionPerformed(ActionEvent e) {
    listControlIcerik.removeItem(listControlIcerik.getSubfocus());
}

}

```

IzlemePenceresi.Java

```

package gui;

import java.awt.*;
import com.sun.java.swing.*;
import borland.jbcl.control.*;
import borland.jbcl.layout.*;
import java.awt.event.*;
import java.lang.Object.*;
import java.util.Hashtable;
import com.sun.java.swing.text.*; // DefaultStyledDocument
import java.text.*;
import java.util.*;

public class IzlemePenceresi extends JFrame {
    public static IzlemePenceresi ornek=new IzlemePenceresi();
    SimpleDateFormat dt = new SimpleDateFormat("HH:mm:ss dd-MM-yyyy ' --- '");

    JPanel jPanel1 = new JPanel();
    JPanel jPanel2 = new JPanel();
    VerticalFlowLayout verticalFlowLayout1 = new VerticalFlowLayout();

    JTextPane jTextAreaHtmlMesaj;
    //JTextPane jTextAreaHtmlMesaj = new JTextPane();
    JScrollPane scp=new JScrollPane(); //
    DefaultStyledDocument doc;
    StyleContext styles=new StyleContext();
    Hashtable runAttr = new Hashtable();

    ButtonControl buttonControlSil = new ButtonControl();
    ButtonControl buttonControl2 = new ButtonControl();
    ButtonControl buttonControl3 = new ButtonControl();
    BorderLayout borderLayout1 = new BorderLayout();
    XYLayout xYLayout1 = new XYLayout();
    JTextPane jTextPaneHtmlMesaj = new JTextPane();
    JScrollPane jScrollPane1 = new JScrollPane();
    JTextPane jTextPaneGenel = new JTextPane();
    BorderLayout borderLayout2 = new BorderLayout();

    public IzlemePenceresi() {
        try {
            //jPanel1.add(scp);
            //jTextPaneHtmlMesaj.setEditable(false);
            jblInit();
        }
        catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

```

public static void main(String[] args) {
    IzlemePenceresi izlemePenceresi1 = new IzlemePenceresi();
}

private void jblnit() throws Exception {
    jTextPaneHtmlMesaj = createEditor();

// jTextPaneGenel = createEditor();

    jTextPaneHtmlMesaj.setVisible( true );
    jTextPaneGenel.setVisible( true );

    //scp.setPreferredSize(new Dimension(40, 100));
    jPanel2.setLayout(verticalFlowLayout1);
    jPanel1.setLayout(borderLayout2);
    //jPanel1.setPreferredSize(new Dimension(800, 400));
    //jPanel2.setPreferredSize(new Dimension(100, 100));
    //jTextPaneHtmlMesaj.setPreferredSize(new Dimension(30, 30));
    jTextPaneHtmlMesaj.setBackground(Color.black);
    jScrollPane1.setPreferredSize(new Dimension(10, 75));
    jTextPaneGenel.setText("");
    buttonControlSil.setLabel("Sil");
    buttonControlSil.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(ActionEvent e) {
            buttonControlSil_actionPerformed(e);
        }
    });
    buttonControl2.setLabel("Kaydet");
    buttonControl3.setLabel("Kapat");
    buttonControl3.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(ActionEvent e) {
            buttonControl3_actionPerformed(e);
        }
    });
    jTextPaneHtmlMesaj.setText("");
    //jPanel1.add(jTextPaneHtmlMesaj, BorderLayout.CENTER);

// scp.add(jTextPaneHtmlMesaj);
//jPanel1.add(jTextPaneHtmlMesaj, BorderLayout.CENTER);
jPanel2.add(buttonControlSil, null);
jPanel2.add(buttonControl2, null);
jPanel2.add(buttonControl3, null);
scp.setPreferredSize(new Dimension(200, 200));
jTextPaneHtmlMesaj.setEditable(false);
this.getContentPane().add(jPanel1, BorderLayout.CENTER);
jPanel1.add(scp, BorderLayout.CENTER);
scp.getViewPort().add(jTextPaneHtmlMesaj, null);
jPanel1.add(jScrollPane1, BorderLayout.SOUTH);
jScrollPane1.getViewPort().add(jTextPaneGenel, null);
this.getContentPane().add(jPanel2, BorderLayout.EAST);
}

void buttonControlSil_actionPerformed(ActionEvent e) {
    jTextPaneHtmlMesaj.setText("");
}

}

public void openForm(){
    pack();
}

```

```

    show();
}

void buttonControl5_actionPerformed(ActionEvent e) {

}

// JTextPane fonksiyonları

public JTextPane createEditor() {
    createStyles();
    doc = new DefaultStyledDocument(styles);
    return new JTextPane(doc);
}

void createStyles() {
    Style s = styles.addStyle(null, null);
    StyleConstants.setForeground(s, Color.gray);
    runAttr.put("normal", s);

    Style def1 = styles.addStyle(null, null);
    //styles.getStyle(StyleContext.DEFAULT_STYLE);
    StyleConstants.setBold(def1, true);
    StyleConstants.setFontSize(def1, 10);
    runAttr.put("bold", def1);

    s = styles.addStyle(null, null);
    StyleConstants.setBold(s, true);
    StyleConstants.setFontFamily(s, "System");
    StyleConstants.setFontSize(s, 10);
    StyleConstants.setForeground(s, Color.red); //new Color(51,102,153));
    runAttr.put("red", s);

    s = styles.addStyle(null, null);
    StyleConstants.setBold(s, true);
    StyleConstants.setFontFamily(s, "System");
    StyleConstants.setFontSize(s, 10);
    StyleConstants.setForeground(s, Color.green); //new Color(51,102,153));
    runAttr.put("green", s);

    Style def = styles.getStyle(StyleContext.DEFAULT_STYLE);
    StyleConstants.setBold(def, true);
    StyleConstants.setFontFamily(def, "System");
    StyleConstants.setFontSize(def, 10);
    StyleConstants.setForeground(def, Color.yellow); //new Color(51,102,153));
    //StyleConstants.setAlignment(def, StyleConstants.ALIGN_CENTER);
    runAttr.put("blue", def);
} // create styles

public synchronized void appendText(String inStr){
    int last=0;
    int prev=0;
    String sty,mesaj;
    Style s = null;
    int a;

    while (last != -1) {
        prev=last+3;
        last=inStr.indexOf("<s>",prev);
        sty=inStr.substring(prev,last);
        prev=last+3;
    }
}

```

```

last=inStr.indexOf("<s>",prev);

if (last!=-1)
    mesaj=inStr.substring(prev,last);
else
    mesaj=inStr.substring(prev,inStr.length());

s = (Style) runAttr.get(sty);
try {
    a=getBoy();
    if (a>50000) {doc.remove(1,500);}

    doc.insertString(doc.getLength(), mesaj, s);
    jTextPaneHtmlMesaj.setCaretPosition(doc.getLength());
} catch (BadLocationException e) {System.err.println("Internal error: " + e);}

} //while

} // append text

public synchronized void appendInformation(String inStr){
    String zaman = dt.format(new Date(System.currentTimeMillis()));
    jTextPaneGenel.setText( jTextPaneGenel.getText()+ zaman + inStr);
} // appendInformation

```

```

public int getBoy(){
    int boy= jTextPaneHtmlMesaj.getDocument().getLength();
    return boy;
} // get boy

void buttonControl4_actionPerformed(ActionEvent e) {
}

void buttonControl3_actionPerformed(ActionEvent e) {
    setVisible(false);
}
}

```

HttpDebugSvr.java

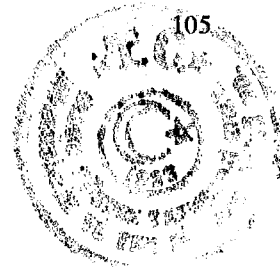
```

package COM.reitshamer.proxy;

import java.util.*;
import COM.reitshamer.util.*;
import COM.reitshamer.net.*;
import COM.reitshamer.error.*;
import COM.reitshamer.http.*;

import gui.*;
import java.text.*;

```



```
public class HttpDebugSvr
{
    /**
     * Creates an HttpDebugSvr and starts it.
     */
    public static HttpDebugSvr mySvr = new HttpDebugSvr();
    IzlemePenceresi izlemePenceresi = IzlemePenceresi.ornek;
    SimpleDateFormat dt = new SimpleDateFormat("HH:mm:ss dd-MM-yyyy");

    public static void main(String[] args)
    {
        if (args.length < 1)
        {
            System.err.println("usage: java COM.reitshamer.proxy.HttpDebugSvr <port>");
            System.exit(1);
        }
        HttpDebugSvr mySvr = new HttpDebugSvr();
        try
        {
            mySvr.init(args);
        }
        catch(Exception e)
        {
            Debug.print("exception: " + e);
        }
    }

    // proxy server ana pencereden başlat
    public void baslat(String[] args)
    {
        if (args.length < 1)
        {
            System.err.println("usage: java COM.reitshamer.proxy.HttpDebugSvr <port>");
            System.exit(1);
        }
    }

    // HttpDebugSvr mySvr = new HttpDebugSvr();
    try
    {
        String zaman = dt.format(new Date(System.currentTimeMillis()));
        izlemePenceresi.appendInformation(" GYTE-ProS BAŞLATILDI");

        mySvr.init(args);
    }
    catch(Exception e)
    {
        Debug.print("exception: " + e);
    }
}

    public void durdur(){
        String zaman = dt.format(new Date(System.currentTimeMillis()));
        izlemePenceresi.appendInformation( zaman+ " GYTE-ProS DURDURULDU ..");
    }

    public void init(String[] args) throws Exception
    {

```



```
int iListenPort = 8080;
String sPort = args[0];
// sPort = "8080";

if ( !sPort.equals( "" ) )
{
    try
    {
        iListenPort = Integer.parseInt(sPort);
        if ( iListenPort == 0 )
        {
            throw new Exception("");
        }
    }
    catch(Exception e)
    {
        throw new HmrException("Invalid listen port");
    }
}

//
// Create local service factory.
//

LocalServiceFactory localFactory = new LocalServiceFactory();

//
// Add an HttpProxyService to the local factory.
//
HttpProxyService httpSvc = new HttpProxyService();
localFactory.addService(httpSvc);

// Create a LogFile.
LogFile lf = new LogFile("out.txt");
lf.setCopyToStdout(true);
LogFile.s_lf = lf;

//
// Start a NetListener.
//
NetListener svr = new NetListener(iListenPort, localFactory,
    new HttpTrans(), true);
svr.start();
lf.writeln("started Single-Threaded HTTP Proxy Server at port "
    + iListenPort);

izlemePenceresi.appendText("<s>blue<s>"+"-HttpDebugSvr.java- started Single-Threaded
HTTP Proxy Server at port "+ iListenPort+"\r\n");

lf.useTimestamp(false);
if ( (args.length > 1) && (args[1].equals("0")) )
{
    System.out.println("not copying log to stdout");
    lf.setCopyToStdout(false);
}
}
```