T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR AĞLARINDA TCP/IP PROTOKOL AİLESİNİN KULLANILMASI VE GÜVENLİK DENETİMLERİ

YÜKSEK LİSANS TEZİ

Bilgisayar Müh. Ümit ERSÖZ

Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ
Enstitü Bilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ

MAYIS 2003

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR AĞLARINDA TCP/IP PROTOKOL AİLESİNİN KULLANILMASI VE GÜVENLİK DENETİMLERİ

YÜKSEK LİSANS TEZİ

Bilgisayar Müh. Ümit ERSÖZ

Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ
Tez Danışmanı : Prof. Dr. Erol EMRE

Bu tez 7/5/2003 tarihinde aşağıdaki jüri tarafından Oybirliği/~~Oyçokluğu~~ ile kabul edilmiştir.

Prof. Dr. Erol Emre	Doç. Dr. Hüseyin Ertuğ	Yrd. Doç. Dr. Nejat Yumuşak
Jüri Başkanı	**Jüri Üyesi**	**Jüri Üyesi**

ÖNSÖZ

Globalleşme ve rekabet etme sürecinde kişi ve kuruluşların ayakta kalabilmesi, hızla gelişmekte olan ve her geçen gün yeni bir ürün, yeni bir hizmetle karşımıza çıkan bilişim sektörünün yakalanmasıyla mümkündür. Güncel bilgiye hızlı erişim, bilginin etkin ve verimli bir şekilde yönetimi, teknolojilere entegrasyonu zorunlu kılmaktadır. Bu teknolojik gelişmelerin sağladığı faydaların yanı sıra, kötü amaçlı kullanımlardan, kısıtlanmayan bilgi erşimlerinden zarar görüleceği de unutulmamalıdır. Bu nedenle bilgisayar sistemlerinin güvenliği sağlanmalı ve gelişen teknolojik imkanlara paralel olarak güvenlik sistemleri güncellenmelidir. Bende tezimde Internetin yapı taşı TCP/IP ve bilgi güvenliği çözümleri üzerinde durdum.

Tez çalışmamın içeriğini belirlemede ve geliştirilmesinde yardımlarını esirgemeyen danışman hocam sayın Prof. Dr. Erol Emre'ye ve bu konuda yaptığım uygulamalarda yardımcı olan çalışma arkadaşlarıma teşekkürlerimi sunarım.

İÇİNDEKİLER

SİMGELER LİSTESİ

ACK : Acknowledge
ACL : Access Control List
ARP : Adress Resulation Protokol
DF : Don't Fragment
DHCP : Dynamic Host Configuration Protocol
DMZ : Demilitirazed Zone
DNS : Domain Name Service
FTP : File Transfer Protocol
HTML : Hypertext Markup Language
HTTP : Hypertext Transfer Protocol
ICMP : Internet Control Message Protocol
IGMP : Internet Group Management Protocol
ISS : İnitial Send Sequence
LAN : Local Area Network
MAC : Media Access Control
MF : More Fragment
MTU : Maximum Transfer Unit
NFS : Network File System
NTFS : Network File System
OS : Operation System
OSI : Open Systems Interconnection
PDU : Protocol Data Unit
QoS : Quality of Service
RARP : Reverse Adress Resulation Protokol
ROS : Router Operating System
RPC : Remote Procedure Call

SEQ	: Sequence Number
SMTP	: Simple Mail Transfer Protocol
SNMP	: Simple Network Management Protocol
SSL	: Secure Socket Layer
SYN	: Synchronize
TCB	: Transfer Control Block
TFTP	: Trivial File Transfer Protocol
TOS	: Type of Service
TTL	: Time to Live
TCP/IP	: Transmission Control Protocol/Internet Protocol
UDP	: User Datagram Protocol
WAN	: Wide Area Network

ŞEKİLLER LİSTESİ

TABLOLAR LİSTESİ

ÖZET

Anahtar Kelimeler: Ateş duvarı, Saldırı tespit sistemleri, Zayıflık tarama sistemleri.

Bilişim teknolojileri alanında son yıllarda yaşanan hızlı gelişmeler, bilgi ve iletişim için temel araçların bilgisayarlar ve bilgisayar ağları olduğu bir dünyaya yönelişi sağlamıştır. Ben de bu çalışmamda bilişim teknolojilerindeki bu gelişim sürecinde bilgi güvenliğinin önemi ve yapılması gereken çalışmalar üzerinde durdum.

Bilişim teknolojilerinden çok farklı alanlarda yararlanılması kısa bir zaman zarfında gerçekleşmiştir. Bu kadar hızlı geçiş bazı konularda hazırlıksız yakalanılmasına, gerekli alt yapı çalışmalarının tamamlanmadan çözümlerin üretilmesine neden olmuştur. Bu konuların başında hizmetlerin güvenlik destekleri gelmektedir. Bilgi güvenliğinin sağlanmasına yönelik çalışmalar, başta yerel ağların güvenliğini sağlamaya yönelik ateş duvarı yazılım ve donanımları, saldırıları önceden sezmeye yönelik, saldırı tespit sistemleri ve güvenlik açıklarını tespit amaçlı zayıflık tarama sistemleri bilişim sektöründe güvenlik politikaları olarak kendisini göstermektedir.

Çalışmamda, saldırganların gözüyle ağımızı incelememizi sağlayan, ağımızda mevcut bulunan güvenlik açıklarını hızlı bir şekilde raporlayan Nessus zayıflık tarama yazılımı incelenmiş ve uygulama sonuçlarınada değinilmiştir.

USAGE OF TCP/IP PROTOCOL FAMILY IN COMPUTER NETWORKS AND SECURITY CONTROLS

SUMMARY

Keywords – Firewall, Intrusion detection systems, Vulnerability scanner systems.

The recent fast developments in the field of data processing have provided a passage to a world where the basic devices for information and communication are the computers and computer networks. In my study, I have pointed out the importance of information security in this development progress in the data processing technologies and the studies should be done.

Utilization of information processing technologies in various fields has been realized in a short period of time. Such fast transformation has caused to be caught unready in some subjects and to produce solutions without completing the infrastructure studies. The main subject is the security supports of the services. Studies for obtaining information security have showed itself as the security policies in the data processing sector, such as firewall software and hardware for obtaining the security of local networks, intrusion detection systems for sensing the intrusions before, and vulnerability scanner systems for detecting of security openings.

In our study, Nessus vulnerability scanner software, which enables us to examine our network from the view of the intruders and reports the security malfunctions in our system, has been examined and application results were also reported.

BÖLÜM 1. GİRİŞ

1980'li yılların başında sadece kurumlara hitap eden bilgisayar teknolojisi, kat ettiği yol ile şu anda günlük hayatımızın her alanında kendine yer edinmiştir. Eğitimden ticarete, devlet sektöründen özel sektöre, eğlenceden alış-verişe kadar bir çok alanda klasikleşmiş anlayışı değiştirmiş ve hayatımıza yeni bir yaşam tarzı getirmiştir.

Bilgisayarlar, ilk yıllarındaki kişisel kullanılırlığı geride bırakmış, bilginin paylaşıldığı ve ortak çalışmaların yürütüldüğü bilgisayar ağlarının parçası durumuna gelmiştir. Bilgisayar ağlarındaki hızlı gelişim ile birlikte, yerel ve geniş alan ağlarında da değişik iletişim altyapıları ile çok farklı hizmetler sunulmaya başlanmıştır.

Global bir bilgisayar ağı olan Internet, kullandığı alt yapı itibariyle sınır tanımadan tek bir bilgisayardan, kurumlara ait büyük bilgisayar ağlarına kadar pek çok bilgisayarı ortak bir iletişim ortamında buluşturmaktadır. Internet bugün 400 milyondan fazla kullanıcısı olan, 50.000'in üzerindeki fiziksel ağın bağlı olduğu dev bir iletişim ve bilgi paylaşım ortamı olarak kişilere ve şirketlere büyük bir bilgi hazinesi olmakta, azımsanamayacak faydalar sağlamaktadır.

Internet'in yaygınlaşması, onunla eşleşmiş olan TCP/IP protokolünün de dünya çapında kullanılan protokol ailesi halini almasını sağlamıştır. TCP/IP bir protokol olmayıp bir protokol ailesidir. Sunulan her hizmet için geliştirilmiş uygulama katman protokolünü bünyesinde bulundurur. TCP/IP açık kaynak kodlu dağıtılması nedeniyle üzerinde çalışmalar yapan kişi ve kuruluş sayısı da oldukça fazladır ve bu yüzden çok fazla hizmetin sunulabildiği bir protokol ailesi olmuştur.

Internet'in bu denli büyümüş olması ve her geçen günde büyümeye devam etmesi, bu ağa çok değişik amaçlar doğrultusunda bağlananlarıda beraberinde getirmiştir. Bu

yüzdendir ki; bilgi güvenliği ve güvenlik politikası kavramları ön plana çıkmıştır. Böyle bir ortamda bize ait hassas bilgiler, ağ ve diğer kaynaklarımız üzerindeki tehditleri bilmemiz gerekmektedir.

TCP/IP hizmetlerindeki açıklar, bilgisayar konfigürasyonun karmaşıklığı ile bir dizi diğer etken, iyi korunmayan merkezlerin güvenlik problemleri yaşamalarına neden olur. Bir başka gerçek de Internet'e bağlanan herhangi bir bilgisayarın ya da sunucunun fark edilerek, güvenlik açıkları üzerine ataklara geçilmesinin ortalama olarak 3 saat sürmesidir. Güvenlik uzmanları tarafından dile getirildiği gibi TCP/IP güvenlik açısından doğuşundan kaynaklanan zayıflıklar göstermektedir. Bunun sebebi, TCP/IP'nin bilgiyi paylaşmak amaçlı geliştirilmesinden kaynaklanmasıdır, saklamak amaçlı değil. Buna bağlı olarak Internet'in aslında doğuştan güvenlik aksaklıkları vardır.

Bu doğrultuda bilgi güvenliğinin sağlanması, hizmetlerin kullanılabilirliği, aksamadan devamlılığı için bir güvenlik politikası ortaya konmalıdır. Bir bilgi güvenliği politikasının başarısı teknik detaylarda saklı olduğu kadar aynı zamanda uygulanacak prosedürlerin ve süreçlerin başarısına da bağlıdır. Bu doğrultuda güvenlik politikası uygulanabilirlik arz etmeli ve uygulaması kontrol edilmelidir. Örneğin Internet bağlantılarının hangi kıstaslar dahilinde gerçekleşeceği, hangi hizmetlerin sunulacağı gibi kural tanımlamaları ortaya konulmalı ve belli bir denetim ve yaptırım altına alınmalıdır.

Bilgi güvenliğini tehdit eden saldırılar sadece dışarıdan gelmez. CERT'in (Computer Emergency Response Teams Coordination Center-CERT/CC) raporlarına göre saldırıların %80'i içeriden gelmektedir. Bu nedenle yerel ağlar üzerinde de önem arz eden sistem kaynakları için güvenlik tedbirlerine baş vurulmalıdır.

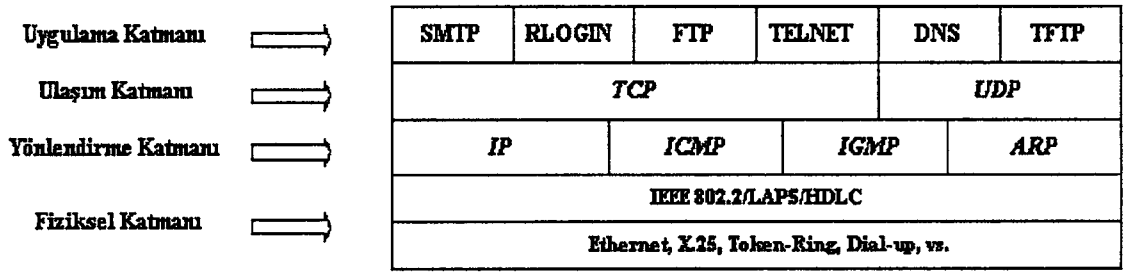
BÖLÜM 2. TCP/IP PROTOKOL AİLESİ

TCP/IP protokol ailesinin kökeni 1970'li yılların başında Amerikan Savunma Bakanlığına bağlı İleri Araştırma Projeleri Ajansı'nın (ARPA) yürüttüğü, askeri merkezlerin ve araştırma laboratuvarlarının iletişiminin sağlanması projesiyle ortaya çıkmıştır. Bu amaç doğrultusunda kurulan ve daha sonra ABD için ulusal ağ olmaktan çıkan Internet'in temelini teşkil eder. Internet üzerinde markadan bağımsız bilgisayar sistemlerinin iletişimi ve sunulan hizmetler TCP/IP protokol ailesi dahilindeki protokollerce sağlanmaktadır.

2.1. TCP/IP Mimarisi

TCP/IP protokol kümesi katmanlı yapıdadır; ancak OSI'de olduğu gibi yedi katman değilde yalnızca dört katman tanımlıdır. Bilgisayar ağlarındaki iletişim için gerekli bütün iş bu dört katmana yayılmıştır. Her katmanda yapılacak görevler bu katmanlar için tanımlanmış protokoller ile standarta sokulmuştur.

TCP/IP protokol kümesinin sahip olduğu dört katmanlı mimari Şekil 2.1'de gösterildiği gibi dizayn edilmiştir. En üstte uygulama katmanı ve altında sırasıyla ulaşım, yönlendirme ve fiziksel katmanlar yer alır.

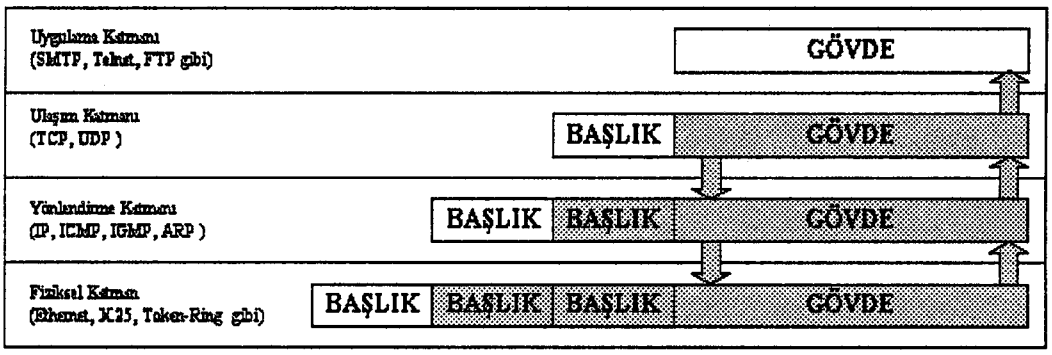


Şekil 2.1 TCP/IP protokol ailesinin katmanlı yapısı

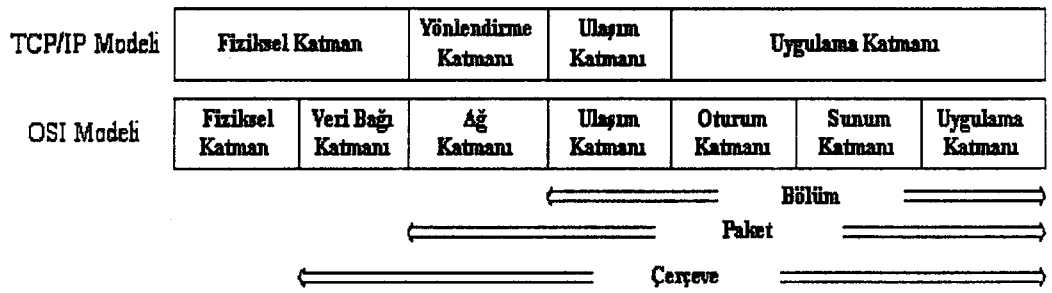
2.2. Veri Kapsülleme

Her katman açısından, ağa çıkarılmak üzere hazırlanan veri blok yapısı iki ayrı kısım da değerlendirilebilir; başlık ve gövde. Başlık kısmı, o anda bulunulan katmanda koşan protokole ilişkin bilgileri taşır. Gövde kısmı ise, yine o anda bulunulan katman için haberleşme bilgisi olarak kabul edilen kısmı içerir. Bu kısım esasında bir üstteki katman tarafından hazırlanmış ve/veya bir üst katmana iletilecek veri bloğu yapısıdır ve kendi içerisinde yine başlık ve gövde olarak ayrılacaktır. Olaya tersten baktığımızda ise, her katman, kendisinin üstünde yer alan katmandan alacağı ve/veya vereceği veri bloğu yapısını, kendi veri bloğu yapısnın gövdesi olarak kullanır, buna eklenecek bir başlık ile bu katmana özel yeni bir veri bloğu yapısını oluşturacaktır. Bu işlem veri kapsülleme olarak bilinir.

Ulaşım katmanında oluşturulan veri bloğu yapısına bölüm (segment) adı verilir. Ulaşım katmanından, yönlendirme katmanına aktarılan bölümler, bu katmanda oluşturulacak veri bloğu yapısı için gövde vazifesini görecek ve başlık kısmı olarak ağ adresleri eklenecektir. Yönlendirme katmanında oluşuturulan bu veri bloğu yapısına ise paket denir. Paketler, yönlendirme katmanından fiziksel katmana aktarıldığında bir başlık bilgisi eklenerek lojik işaret bloklarına dönüştürülür. Fiziksel katmanda oluşturulan lojik işaret bloklarına çerçeve (frame) denir. Şekil 2.2'de bölüm, paket ve çerçevenin katmanlar ile ilişkisi gösterilmiştir.



Şekil 2.2 Veri kapsülleme



Şekil 2.3 TCP/IP ve OSI referans modelinin kıyaslanması

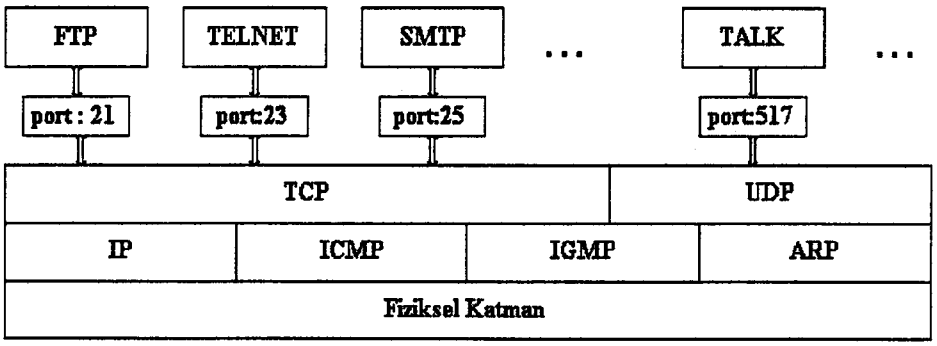
2.3. Port ve Soket Kavramları

Bilgisayar ve iletişim dünyası için "port" kavramının önemli bir yeri vardır. "Port" kavramı, hem fiziksel, hem mantıksal olarak iletişim için bağlantı noktası tanımıdır. TCP/IP ortamında, uygulama katmanı protokolleri ile ulaşım katmanı protokolleri arasında iletişim için bir geçit oluşturur. Uygulama katmanı protokollerini kullanan birden falza uygulamanın hizmet verdiği bir ortamda, gelen isteğin ulaşım katmanından doğru uygulamaya yönlendirilmesi, ancak mantıksal bir bağlantı noktası tanımı olan port numaraları ile mümkün olmaktadır.

Her port 16 bitlik bir numaradır. Dolayısıyla ulaşım katmanın TCP ve UDP protokollerinin her ikisi içinde 2^{16} adet port tanımlıdır. Hizmet sunma görevini yerine getiren uygulamalar için port numaraları daha önceden tanımlanmıştır. 0-255 arasındaki port numaraları çok kullanılan uygulamalar için ayrılmıştır. Örneğin, FTP için TCP port 21, TELNET için TCP port 23, TALK için UDP port 517 ayrılmıştır. Şekil 2.4'de soket yapısı izlenmektedir.

TCP/IP protokolü ile iletişimde uçlar arasında karşılıklı bağlantının kurulabilmesi için hem IP adresinin gerektiği, hemde hizmete ait port numarası gerekliliği açıktır. Buna göre hem IP adresi hem de port numarasının birlikte kullanılması ile ortaya çıkan adrese soket numarası denir. Örneğin, 144.122.156.104 IP adresine sahip makinada çalışan TELNET sunucu programına (23. "port" dan hizmet veren) bağlanmak için aşağıdaki satır yazılır.

telnet *144.122.156.104 23*



Şekil 2.4 Soket yapısı gösterimi

Daha önce de belirttiğimiz gibi bazı sunucu programların belirli portlardan hizmet verdiği bilindiği için, bu sunuculara bağlanmak istediğimizde, port numarasını vermeye gerek kalmaz. Bu durumda yukardaki satır

telnet 144.122.156.104 olarak kullanılabilir.

2.4. TCP/IP Protokol Ailesinin Katmanlı Yapısı ve Protokoller

TCP/IP protokol ailesi dahilinde bulunan, iletişimi düzenleyen standartlar katmanlı yapıda şekillenmiştir ve iletişim, her katmandaki gerekli protokolün uyarılması ile sağlanır. Uygulama katmanından, fiziksel katmana kadar olan her katmandan sadece bir protokol bu iletişim için gerekli bölüm, paket ve çerçevenin oluşturulmasında görev alır. Bundan sonraki kısımda bu protokoller, katmanlı yapıya uygun olarak sınıflandırılarak ele alınacaktır.

2.4.1. Uygulama katmanı

Kullanıcıların ortak süreçler yürütebilmesi için gerekli arayüzü sunan uygulamalar, ağ kaynaklarının paylaştırılması ve işletim sistemi servisleri tarafından koşturulan protokoller bu katman dahilinde yer almaktadır. FTP, SMTP, SNMP, TELNET vs birçok protokol tanımlıdır. Sayılarının oldukça fazla olması ve verilen servise özel olmaları, yaygın olanları dışınada tam bir standarda oturtulamamalarına neden olmuştur. Uygulama katmanı protokolleri için iki farklı çalışma ortamı söz

konusudur; kullanıcı ve sunucu. Sunucu ortamında, kullanıcılardan gelen istekler değerlendirilerek uygun cevaplar üretilir. Kullanıcı ortamında ise; gereken hizmeti almak için sunucuya bağlantı kurmaya çalışan uygulama programı tarafından koşturulur. Ancak bir sistem hem hizmet sunup, hem de hizmet alıyorsa bu sistem üzerinde ilgili uygulama protokolünün sunucu ve kullanıcı parçalarının ikiside çalıştırılır.

2.4.2. Ulaşım katmanı

TCP/IP'de ulaşım katmanında TCP ve UDP olmak üzere iki protokol tanımlıdır. Servis sağlayıcı olarak bilinen bu katman, uçtan uca haberleşmeden sorumludur. Eğer bağlantı yönlendirmeli ise, güvenilirlik ölçümleri ve bir ağ üzerinden akan tüm trafiği açıklayabilen mekanizmaları sunar.

Bu katmandaki segmentler ve datagramlar iki kısımdan oluşur; başlık ve gövde. IP, üzerinde koşan bu protokollere ait segmentler ve datagramlar bir paket yapısı haline getirerek aktarır. IP'nin de IP üzerinden koşması söz konusu olabilmektedir. Herkese yayın özelliğinin olmadığı IP ağlarında, IP paketlerinin başka IP paketleri içerisinde kapsüllenerek gönderilmesi böyle bir çalışma uygulamasıdır.

Bu katmanda yer alan protokollerden TCP bağlantı temelli, UDP bağlantısız basit bir protokoldür. Bağlantı temellide, gönderici ve alıcı sistem iletişim başlamadan önce birbirleriyle anlaşırlar (iki taraf iletişim yapma konusunda istek ve onaylarını birbirlerine gönderirler.) Özellikle ağ katmanı yeterli düzeyde güvenilir değilse (Intemet Ağ ortamı genelde güvenilir değildir) bu açık kapatılır. Bağlantısız protokolde ise iletişim başlamadan önce gönderici ve alıcı sistemin bir anlaşmaya varmalarına gerek yoktur. Kritik önemdeki uygulama katmanı protokolleri TCP'yi kullanırken; UDP daha çok sorgulama amaçlı kullanılır.

Bu katmanda UDP'nin oluşturduğu veri bloklarına "datagram", TCP'nin oluşturduğu veri bloklarına "segment" adı verilir. İkisi arasındaki temel fark, segmenti oluşturan veri grubunun başında sıra numarası bulunmasıdır.

2.4.2.1. Transmission control protocol (TCP)

TCP protokolü, uygulama katmanı protokollerinden aldığı verileri daha küçük parçalara (segment) bölerek ağ üzerinden iletilmesini sağlar. İki sistem arasında TCP iletişimi başlamadan önce bir oturumun kurulması gerekir. Yani TCP, bağlantı temelli (connection-oriented) bir protokoldür. Bunun yanında TCP full-duplex ve güvenilir bir protokoldür.

Bağlantı temelli protokol olması nedeniyle sistemler arasında iletişim başlamadan önce aralarında bir oturum (session) açarlar. Oturumun açılması sırasında sistemler kendi iletişim parametrelerini birbirlerine iletir ve aralarında senkranizasyon sağlar. Senkranizasyon, gidip gelen üç çerçeveyle gerçekleşir, karşılıklı tampon bellek miktarları ve TCP parametreleri aktarılır. Daha sonra iletişim bu parametreler ışığı altında güvenilir olarak kotarılır. Güvenilirlikten kasıt bilginin karşı tarafa gittiğinin onayının alınmasıdır (acknowledge, ACK). Kritik işler yapan, yani güvenilirlik gerektiren uygulama katmanı protokolleri verileri TCP ile iletirler. Örneğin, 21 nolu portu kullanan FTP, 23 nolu portu kullanan Telnet gibi...

TCP, üst katmanı olan uygulama katmanı, alt katmanı olan yönlendirme katmanı ile kendi arasında iki farklı arayüz oluşturur. Uygulama katmanı ile oluşturduğu arayüzde servis tanımlamalarını (primitive) kullanır. Bir uygulama protokolü tanımlanmış servis çağrıları (call) yoluyla veriyi TCP katmanındaki tampon (buffer) alana yerleştirir. TCP, gelen veriyi segmentler haline getirip herbir segmenti çağrı yoluyla IP'ye gönderip hedef sisteme iletilmesini ister.

TCP ile uygulama katmanı arasındaki arayüzde tanımlanmış çağrı fonksiyonları; uygulama protokolünün bir bağlantıyı açmasını (OPEN), kapamasını (CLOSE), veriyi gönderilmesini (SEND), almasını (RECEIVE), veri trafiği akışının kontrolünü (STATUS) gibi işlevleri yerine getirir.

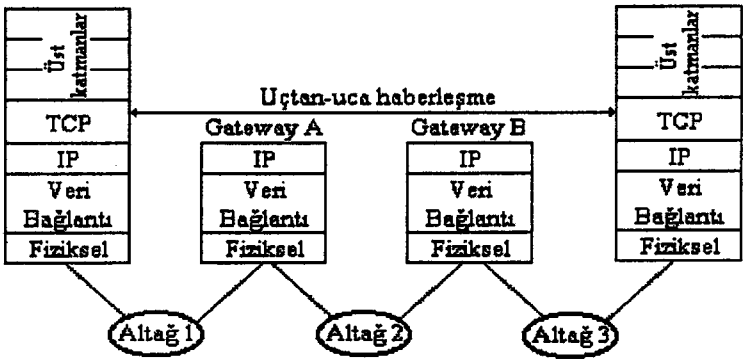
TCP ve alt katmanı olan yönlendirme katmanı (IP) arasındaki servis tanımlamaları TCP standartlarınca belirlenmemiştir. TCP işlemlerinde farz edilen; TCP ve alt katmanın birbirlerine bilgileri eşzamansız olarak iletebilmesidir. TCP, IP katmanının

bu arabirimi tanımlamasını bekler. TCP'nin IP ile olan arayüzü içinde çağrı fonksiyonları kullanılır.

Tablo 2.1 TCP kullanıcı arabirimi çağrı fonksiyonları

Komut	Parametreler
	Servis istek primitive'leri (Uygulama Katmanı Protokollerinden (UKP) TCP'ye)
UNSPECIFIED-PASSIVE	Yerel port, Uygulama katmanı protokolü (UKP) timeout (1)*, timeout aksiyonu (1), öncelik (1)
OPEN	Güvenlik (1), opsiyonlar (1) -> yerel bağlantı ismi
FULL-PASSIVE-OPEN	Yerel port, varış portu, UKP timeout(1), timeout aksiyonu (1), öncelik (1), güvenlik (1), opsiyonlar (1)
ACTIVE-OPEN	Yerel port, yabancı soket, UKP timeout (1) UKP timeout aksiyonu (1), öncelik (1), güvenlik (1), opsiyonlar (1)
ACTIVE-OPEN-WITH-DATA	Kaynak portları, varış adresi, UKP timeout (1), UKP timeout aksiyonu (1), öncelik (1), güvenlik (1), veri, veri uzunluğu, push bayrağı, acil bayrağı (1)
SEND	Yerel bağlantı ismi, tampon adresi, bayt sayısı, push bayrağı, acil bayrağı,UKP timeout (1), UKP timeout aksiyonu (1)
RECEIVE	Yerel bağlantı ismi, tampon adresi, bayt sayısı, push bayrağı, acil bayrağı
ALLOCATE	Yerel bağlantı ismi, veri uzunluğu
CLOSE	Yerel bağlantı ismi
ABORT	Yerel bağlantı ismi
STATUS	Yerel bağlantı ismi
	Servi cevap primitive'leri (TCP'den UKP'ye)
OPEN-ID	Yerel bağlantı ismi, yabancı soket, varış adresi
OPEN-FAILURE	Yerel bağlantı ismi
OPEN-SUCCESS	Yerel bağlantı ismi
DELIVER	Yerel bağlantı ismi, tampon adresi, bayt sayımı, acil bayrağı
CLOSING	Yerel bağlantı ismi
TERMINATE	Yerel bağlantı ismi, tanım
STATUS-RESPONSE	Yerel bağlantı ismi, kaynak portu ve adresi, yabancı port, bağlantı statüsü, alma ve gönderme penceresi, ACK ve fiş bekleme miktarı, acil modu, timeout, timeout aksiyonu
ERROR	Yerel bağlantı ismi, hata tanımı

*(1) notasyonu parametrelerin opsiyonel olduğunu gösterir.



Şekil 2.5 TCP protokolü uçtan-uca haberleşme şeması

TCP'nin sağladığı işlevler:

- Bağlantı-yönlendirmeli oturumun kurulması ve sonlandırılması
- Güvenilir veri transferi
- Akıcı-yönlendirmeli veri transferi
- Push fonksiyonları
- Yeniden-sıralama (resequencing)
- Bozulmuş ya da ikilenmiş verinin düzeltilmesi (error recovery)
- Akış kontrolü (kayan pencereler) ve veri taşması kontrolü
- Alıcı sistemde birçok uygulama arasında demultiplexing yapılması
- Full-duplex iletim
- Öncelik ve güvenlik

TCP bağlantı-yönlendirmeli bir protokoldür. Yani TCP modülüne giren veya çıkan her bir uygulama katmanı verisi akışıyla ilgili durum ve konum bilgilerini sağlar. Aynı zamanda bir ağ veya çoklu ağlar boyunca yerleşmiş sistemler arasındaki uçtan-uca veri transferinin kontrolünden de sorumludur. Şekil 2.5'de iki sistem arasında kurulan bir TCP oturumuyla verinin çoklu ağlar boyunca iletilmesi gösterilmiştir. TCP iletim yaparken sıra numaraları ve pozitif onaylar (acknowledgment-ACK) kullanır.

İletilen her bir bayt için bir sıra numarası atanır. Alıcı TCP modülü bir toplamsal-hata kontrolü kullanarak verinin iletim boyunca bir tahribata uğrayıp uğramadığını kontrol eder. Eğer veri kabul edilebilir ise, TCP gönderici-TCP modülüne bir pozitif

ACK gönderir. Eğer veri tahribata uğramış ise, alıcı-TCP veriyi yok eder ve bir sıra numarası kullanarak gönderici-TCP'ye sorun hakkında bilgi gönderir. TCP zamanlayıcıları tahribat onarım işlevleri yapmadan önce zaman gecikmesinin aşırı olmadığından emin olurlar. Onarım, alıcı sistemin ACK göndermesi veya gönderici sistemin doğrulama bilgisi zaman aşımına karar vermesi sonucunda gönderici sistemin veriyi yeniden göndermesiyle yapılır.

TCP, veriyi bir uygulama protokolünden akıcı-yönlendirmeli biçimde alır. Akıcı-yönlendirmeli protokoller ayrık karakterler (blok, çerçeve veya datagram değil) göndermek üzere tasarlanmamışlardır. Veri bir uygulama katmanı protokolünden akıcı temelli, yani bayt-bayt gönderilir. Baytlar TCP katmanına varınca, TCP segmentleri olarak gruplandırılır. Bu segmentler daha sonra diğer hedefe iletilmek üzere IP'ye (veya başka bir alt-katman-protokolüne) geçirilir. Segment uzunluğuna TCP karar verir, ancak bir sistem geliştiricisi TCP'nin bu kararı nasıl vereceğine karar verebilir.

TCP ayrıca ikilenmiş veri kontrolü yapar. Eğer gönderici, TCP veriyi tekrar yollarsa, alıcı TCP tüm ikilenmiş gelen veriyi yok eder. Örneğin, alıcı TCP ACK onayını belli bir zamanda gerçekleştirmezse, gönderici TCP veriyi yeniden gönderir ve veri ikilenmiş olur.

TCP, PUSH fonksiyonu kavramını destekler. Bir uygulama; alt katmandaki TCP'ye geçirdiği tüm verinin iletildiğinden emin olmak istediğinde PUSH çağrısını kullanır. Böylece, PUSH fonksiyonu TCP'nin tampon yönetimini ele geçirir. Uygulama katmanı protokolü PUSH'u kullanmak için, PUSH parametresi bayrağı 1'e ayarlanmış bir SEND çağrısını TCP'ye gönderir. Bu işlem TCP'nin, tüm tamponlanmış uygulama verisini bir daha fazla segment halinde hedefe gönderileck şekilde yönlendirme katmanına ilerletmesini sağlar. CLOSE bağlantı işlemi kullanarak da PUSH fonksiyonunu işlevleri sağlayabilir.

TCP, ACK'lar için sıra numaraları kullanır. Bu sıra numaralarını aynı zamanda, segmentlerin hedefe sırası ile varıp varmadıklarını kontrol etmek ve segmentleri yeniden-sıralamada da kullanır. TCP, bağlantısız bir alt katmanın üzerinde yer alır ki;

bu katmanda yapılacak dinamik çoklu yönlendirmelerin sonucunda hedefte ikilenmiş segmentlerin görülmeside muhtemeldir. TCP, bu ikilenmiş segmentleri teke indirgeyecek şekilde yok eder.

TCP her bir bayta sıra numarası verir. Daha sonra ilettiği bu baytlara karşılık ACK onayı bekler. Eğer belirli aralıklarla beklenen ACK'ları almazsa ACK almadığı kısımları yeniden hedef sisteme iletir. TCP olumsuz bir ACK onay mekanizması kullanmaz.

Alıcı TCP modülü, gönderici verisi üzerinde akış kontrolü yapabilir. Böylece tampon taşması ve alıcı cihazın doyması (saturation) gibi sorunlar engellenir. TCP'nin kullandığı kavramın, haberleşme protokollerinde kullanımı alışılmış değildir. Akış kontrolü göndericiye bir "pencere" değeri verilmesine dayanır. Gönderici bu pencere ile belirlenmiş sayıda bayt iletebilir, pencere kapanınca gönderici veri göndermeyi durduracaktır.

TCP sayesinde bir sistem üzerinde çoklu oturum gerçekleştirilmesi olanaklıdır. Çoğullama; TCP ve IP başlıklarından soket bilgilerinin elde edilmesi gerçekleştirilir.

TCP, iki TCP sistemi arasında full-duplex iletişim sağlar. Böylece bir dönüş işareti beklemeksizin (half-duplex'te gereklidir) eşzamanlı iki-yönlü iletim yapılır.

TCP kullanıcının bağlantı için güvenlik ve öncelik seviyeleri belirleyebilmesine olanak tanır. Bu iki özellik, tüm TCP ürünlerinde bulunmayabilir ancak TCP DOD (Departments of Defense) standardında tanımlanmışlardır. TCP iki kullanıcı arasında CLOSE özelliği sağlar. CLOSE, bağlantı koparılmadan önce tüm trafiğin ACK'larının oluşturulduğundan emin olunmasını sağlar.

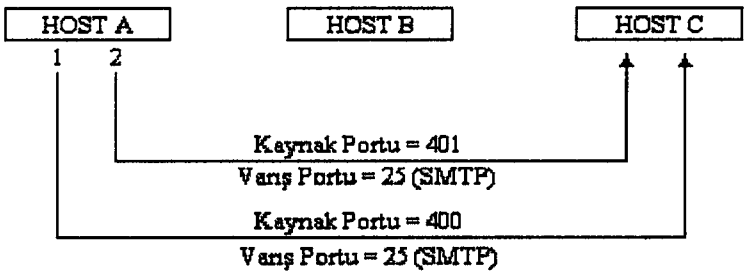
2.4.2.2. TCP protokolü ile uygulama protokolü arası bağlantı

TCP protokolü ile uygulama katmanı protokolleri arası bağlantı port numaraları üzerinden gerçekleşir. Hizmet sunan uygulama katmanı protokolleri ile eşleşmiş bilinen port numaraları mevcuttur ve hizmet isteyen istemci bu port numaralarını bilerek iletişim gerçekleştirir. Şekil 2.6'daki sistem üzerinde port numaralarının nasıl atandığı ve yönetildiği adım adım gösterilmiştir.

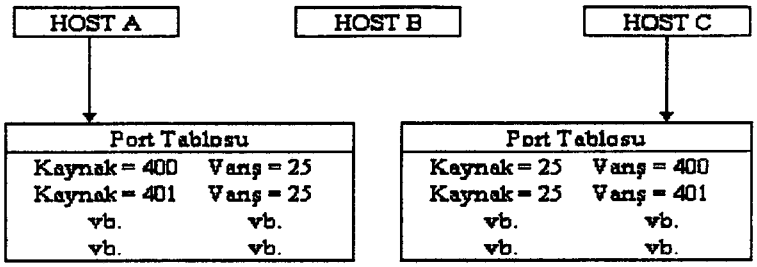
1. adım : A sistemi, C sistemine bir TCP bağlantı isteği göndermekte. Bağlantı isteği SMTP uygulama protokolüne yapıldığı için hedef port SMTP ile eşleşmiş olan 25. portadır. Kaynak port yerel sistem tarafından o an tahsis edilebilecek uygun bir port numarasıdır. Burada 400. port atanmıştır.

2. adım : A sisteminden C sistemine yine bir SMTP oturumu kurulması yönünde bir istek geliyor. Doğal olarak hedef port yine 25 olmakta iken kaynak port olarak 401 uygun port olarak atanmakta.

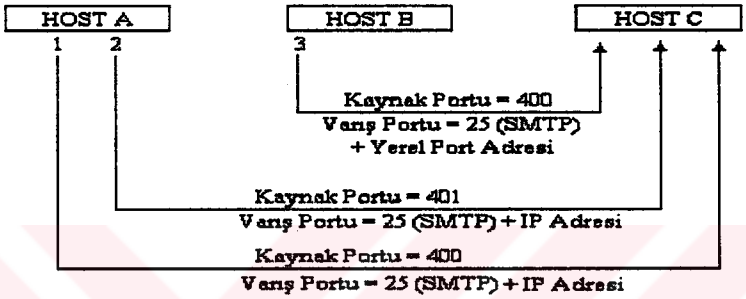
A ve C sistemleri TCP modülleri açısından oturumların durumları Şekil 2.7'de gösterilmiştir. A sisteminden, C sisteminde SMTP'yi hedef alacak şekilde açılan iki oturumun birbirine karışmadan devam ettirilebilmesi kaynak portlarının farklılığı ile mümkün olmaktadır. C sistemi TCP modülü açısından ise kaynak port adresi aynı fakat hedef port adresi farklı oturumlar gerçekleştirilmiştir.



Şekil 2.6 Uç sistemler arasında TCP bağlantısı



Şekil 2.7 Uç sistemlerin TCP bağlantı port tablosu



Şekil 2.8 Bir düğüme birden fazla TCP bağlantısı

3. adım : B sistemide C sistemine bir TCP bağlantı isteği yapmakta ve hedef port olarak yine SMTP protokolü yani port 25 tanımlanmakta. Hedef portun aynı olması olağandır, çünkü iyi bilinen uygulamalara erişim fazla olmaktadır. B sistemi bu bağlantı için kaynak portunuda 400 olarak belirlemiştir.

A-C sistemleri arasında kurulan bağlantı ile B-C sistemleri arasında kurulan bağlantılarda kaynak port ve hedef port bilgileri aynı olması nedeniyle çakışma gözükmektedir. Soket adresleme; yani port ve IP adreslerinin birlikte bağlantıyı belirlemede kullanılmasıyla ayırt edici adresleme gerçekleştirilmiş olur.

2.4.2.3. TCP protokolünde bağlantı mekanizmaları

TCP portları ile iki şekilde bağlantı kurulmasına izin verilir. Bunlar pasivve-open ve active-open'dır. Passive-open modunda; TCP ve işletim sistemi uzak sistemlerden uygulama katmanına gelecek bağlantıyı bekler. Yani uygulama katman protokolü

sunucu rolündedir. İşletim sistemi bağlantı isteği alınca, bu uca bir tanımlayıcı atayarak bir active-open gecikmesi ile karşılaşılmaksızın uzak kullanıcı ile bağlantının teşkil edilmesi sağlanır.

Passive-open isteyen bir uygulama prosesi, herhangi kullanıcıdan gelebilecek uygun bağlantı isteklerine cevap verir. Eğer hiç bir çağrı kabul edilebilir değilse, yabancı soket numarasının tümü 0'la doldurulur. Özelleşmemiş yabancı soketlere yalnızca passive-open'larda izin verilir.

Bağlantı kurulmasının ikinci şekli active-open'dır. Active-open, uygulama katmanı protokolü bir bağlantı kurulması için özel bir soketi görevlendirdiğinde kullanılır. Tipik olarak, active-open bir passive-open porta bir bağlantı kurulması için yayın yapar. İki aktif-open birbirleriyle aynı anda yayın yapsalar dahi, TCP bağlantıyı kurar. Bu özellik; uygulamaların, başka bir uygulamanın aynı zamanda bir open yayınlaması ile ilgilenmeksizin, herhangi bir zamanda open yayınlamalarına olanak sağlar.

TCP active ve passive-open'ların beraber kullanımına ilişkin anlaşmalar sağlar. Birincisi, bir aktif-open özel bir soket ve opsiyonel olarak, bu soketin öncelik ve güvenlik seviyelerini tanımlar. TCP'nin bir open'ı, eğer uzak soket eşleşen bir pasif-open'a sahipse veya eğer uzak soket eşleşen bir aktif-open yayınlamışsa, kabul eder.

TCP'nin her bağlantı için çeşitli parametreleri hatırlaması gerektiğinden, TCP bir iletişim kontrol bloğu (TCB) bilgileri saklar. Aşağıdaki bilgiler TCB'de saklanır:

- Yerel ve uzak soket numaraları
- Tamponları göndermek ve almak için işaretçiler
- Yeniden-iletişim sırası işaretçileri
- Bağlantının güvenlik ve öncelik değerleri
- Şimdiki segment
- TCB aynı zamanda gönderme ve alma sıra numaraları (sequence number) ile ilgili belirli değişkenler içerir. Bu değişkenler Tablo 2.2'de verilmiştir.

Tablo 2.2 TCB değişken tanımları

Değişken ismi	Amacı
	Gönderme sıra değişkenleri
SND.UNA	Gönderildi ama onaylanmadı
SND.NXT	Sıradakini gönder
SND.WND	Gönderme penceresi
SND.UP	Acil verinin son oktetinin sıra numarası
SND.WL1	Son pencere güncellenmesi için kullanılan sıra numarası
SND.WL2	Son pencere güncellenmesi için kullanılan ACK numarası
SND.PUSH	Push'lanmış verinin son oktetinin sıra numarası
ISS	İlk gönderilen sıra numarası
	Alma sıra değişkenleri
RCV.NXT	alınacak sıradaki oktetin sıra numarası
RCV.WND	Alınabilecek oktetlerin sayısı
RCV.UP	Alınan acil verinin son oktetinin sıra numarası
RCV.IRS	İlk alınan sıra numarası

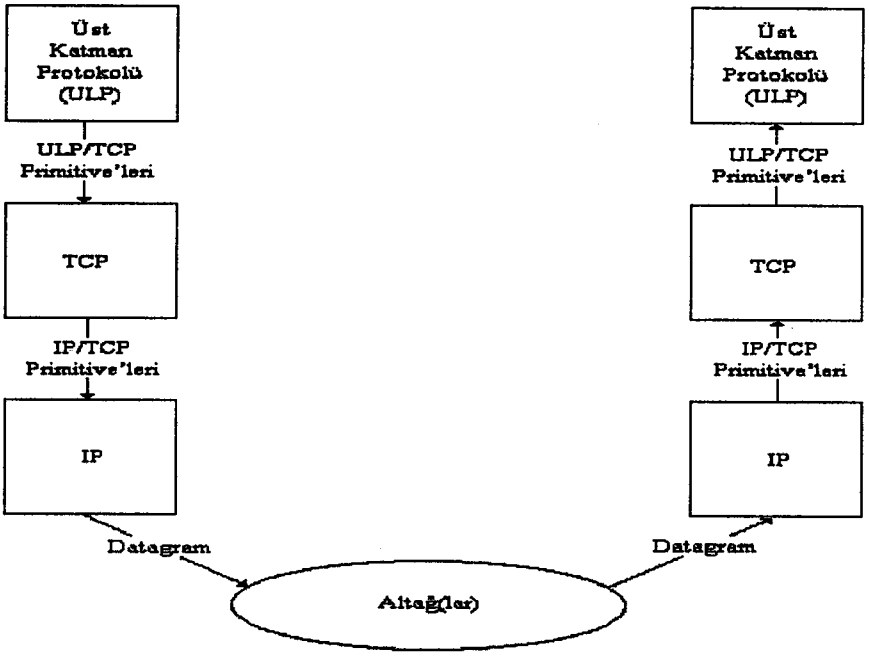
2.4.2.4. TCP segmentleri

Segment başlık ve veri olmak üzere iki parçaya ayrılır. Şekil 2.10'da gösterildiği gibi, uygulama katmanı verisi, başlığın ardından gelir. Başlık kısmı şu alanlardan meydana gelir :

Kaynak Port : Kaynak konak için TCP bağlantısı kurulacak uygulama katmanı protokolünü tanımlar.

Varış (Hedef) Port : Varış (Hedef) konak için TCP bağlantısının kurulacağı uygulama katmanı protokolünü tanımlar.

Sıra Numarası (sequence number (SEQ)) : TCP uygulama katmanından aldığı veriyi segmentlere böler. Bu segmentlerin her biri genellikle bir IP paketi içinde taşınır. TCP, her bir segmente bir numara vererek ağlar üzerinde dolaşan bu segmentlerin hedefe varış sıralarının karışması durumunda, hedef sistemde çalışan TCP protokolünün bunları tekrar uygun şekilde birleştirmesini sağlar



Şekil 2.9 Üst katman, TCP, ve IP'nin ilişkileri

Kaynak konaktaki TCP, hedef konaktaki TCP ile bağlantıyı ilk kurduğunda, ilk gönderdiği segmente bir numara verir. Bu numaraya başlangıç gönderi sırası (initial send sequence (ISS)) denir. Sıra numarası 0 ile 231 değeri arasında olabilmektedir. TCP, verideki baytları gruplayarak segmentleri oluşturur ve her bir segment ayrı bir numara ile numaralandırılır. Bir segmentin, aldığı numara içinde barındırdığı ilk bayta verilen numaradır. İçinde barındırdığı diğer baytlara ise bu numaraların artanları verilir. Bu segmentten sonra gelen segmentin alacağı numara, bir önceki segmentin içindeki en son baytın aldığı numaranın bir fazlası olacaktır. Bu sıra numaraları segment başlığı içinde taşınır.

Acknowledgment (ACK) : Önceden alınan verilerin onaylanmasını sağlamak için kullanılır. Bu alandaki değer, ileticiden gelmesi beklenen, bir sonraki baytın sıra numarası değerini belirtir. Bu numara beklenen bayt için ayarlanarak dahili bir onay kapasitesi sağlar. Yani bu değer bu numaraya kadar olan baytları onaylar (Onaylanan bayt sayısı ACK numarası-1 adettir).

Veri Offset Alanı : TCP başlığını oluşturan, 32-bit sıralı kelimelerin sayısını belirtir. Bu alan, veri alanının nerede başladığının tespitinde kullanılır.

Rezerve Alanı : Adından da anlaşıldığı üzere rezerve edilmiştir. 0'a set edilmesi gereken 6 bitten oluşur. Bu bitler gelecekte kullanılmak için saklanmaktadır.

Bayraklar : TCP'nin kontrol bitleri olarak kullanılırlar ve oturumlar sırasında kullanılan bazı servis ve işlemleri belirtirler. Bitlerin bazıları başlığın diğer alanlarının nasıl yorumlanacağını belirtir. Bu altı bit aşağıdaki bilgileri ifade eder:

URG: Bu bayrak, urgent işaretçisi (acil işaretçisi) alanının anlamlı olup olmadığını belirtir.
ACK: Bu bayrak, acknowledgment alanının anlamlı olup olmadığını belirtir.
PSH: Bu bayrak, modülün push fonksiyonunu işletip işletmeyeceğini belirtir.
RST: Bu bayrak, bağlantının resetlenmesi gerektiğini bildirir.
SYN: Bu bayrak, sıra numaralarının eşzamanlamasının oluşturulmaya çalışıldığını bildirir. SYN bayrağı, bağlantı-kurma segmentlerinde handshaking işlemlerinin oluştuğunu belirtmek için kullanılır.
FIN: Bu bayrak göndericinin gönderecek başka verisi kalmadığını belirtir.

Window (pencere) : Alıcı sistemin kaç bayt almayı beklediğini gösterir. Bu değer atanırken ACK alanındaki değere dayanılır. Window alanındaki değer, ACK alanındaki değere eklenir ve göndericinin iletmek istediği veri miktarı hesap edilir.

Checksum (hata kontrolü): Başlığın vericiden bozulmaksızın geldiğine karar vermektir. UDP'nin kullandığına benzer bir sözde-başlık kullanır.

<table>
<tr><td colspan="5">Kaynak Portu (16 bit)</td><td colspan="4">Varış Portu (16 bit)</td></tr>
<tr><td colspan="9">Sıra Numarası (32 bit)</td></tr>
<tr><td colspan="9">ACK Numarası (32 bit)</td></tr>
<tr><td>Veri Offset (4 bit)</td><td colspan="2">Rezerve (6 bit)</td><td>U R G</td><td>A C K</td><td>P S H</td><td>R S T</td><td>S Y N</td><td>F I N</td><td>Pencere (16 bit)</td></tr>
<tr><td colspan="5">Checksum (16 bit)</td><td colspan="4">Acil İşaretçisi</td></tr>
<tr><td colspan="5">Opsiyonlar (Değişken)</td><td colspan="4">Dolgu</td></tr>
<tr><td colspan="9">Veri (Değişken)
...</td></tr>
</table>

Şekil 2.10 TCP segment yapısı

Acil İşaretçisi (urgent pointer) : Bu alan yalnızca URG bayrağı set edildiğinde kullanılır. Acil işaretçisinin amacı acil verinin yerleştiği veri baytını belirtmektir. Acil veriye band-dışı veri de denir. TCP, acil verinin ne yapılacağına dikkat etmez; bu uygulamaya katmanına özeldir. TCP yalnızca acil verinin nereye yerleştirildiğini belirtir. İşaretçi aynı zamanda acil verinin nerede bittiğini de gösterir. Alıcı TCP, acil verinin geldiğini uygulama katmanına haber verir. Acil veri, interrupt'lar, checkpoint'ler, terminal kontrol karakterleri, vs. gibi kontrol işaretleri olabilirler.

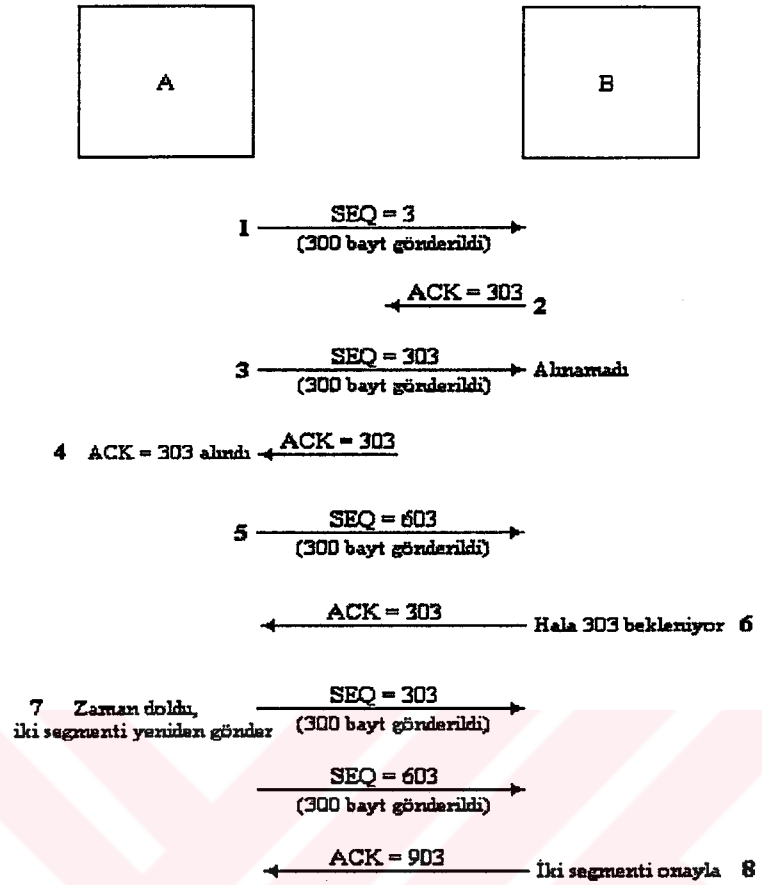
Opsiyon Alanı (değişken, seçenekler) : TCP'ye gelecekte yapılacak eklemeler düşünülerek tasarlanmıştır. IP datagramlarındaki opsiyon alanına benzer bir biçimde yapılandırılmıştır. Her bir opsiyon içeriği; tek bir bayttan oluşur ki, bu bayt bir opsiyon numarası, opsiyon uzunluğunu içeren bir alan, ve opsiyon değerinin kendisini içerir. Opsiyon alanının kullanımı oldukça sınırlıdır.

2.4.2.5. Yeniden iletim işlemleri

TCP protokolü; her bir bağlantının devamlılığını sağlamada ACK bilgisini kullanır fakat negatif ACK (NAK) gibi bir bilgi üretilmez. Ancak, ileten sistemin pozitif-ACK bilgisini almadığı veriler için bir zaman aşımı (timeout) ve retransmit (yeniden-iletim) verisi yayınlamasına dayanır. Bu kavram Şekil 2.11'de 8 adım halinde 900 byte pencere büyüklüğü ve 300 byte segment büyüklüğü için gösterilmiştir.

1. Adım : A sisteminin TCP katmanı, B sisteminin TCP katmanına 300 byte'lık birinci segmentini sıra (SEQ) numarası 3 değerini içerecek şekilde gönderir.

2. Adım : B sisteminin TCP katmanı, aldığı segmenti hatalar açısından kontrol eder ve 303 değerinde bir ACK'yı geri gönderir. Bu değer 0'dan 302. byte kadar (302 de dahil) tüm trafiği onaylayan bir ACK'dır. 2. adımda gösterilen oktan anlaşılacağı üzere, ACK bilgisi 3. adım olduğunda hala A sistemine varamamıştır.



Şekil 2.11 TCP yeniden-iletim şeması

3. Adım : A sisteminin TCP penceresi hala açık olduğundan, 303 numarası (303-603) ile başlayan başka bir veri segmenti gönderir. Çeşitli sebeplerden dolayı, bu segment B sistemine ulaşamamıştır.

4. Adım : 3. adımda iletilen ACK segmenti A sistemine ulaşarak B sisteminin 303 numarası ile başlayan bir segment beklediğini belirtir. Bu noktada, A sistemi, 3. adımda ilettiği segmentin alındığını veya bir Internet içerisindeki değişken gecikmelerle hala varmadığını veya bunların tersini bilemez.

5. Adım : A sistemi 603 numarası ile başlayan sıradaki segmenti yollar. Bu B sistemine hatasız varır.

6. Adım : B sistemi 5. adımda iletilen 603 numaralı segmenti başarı ile alır. B sistemi sonra ACK 303'lü bir segment geri yollar çünkü hala 303 numarasıyla başlayan segmenti beklemektedir.

7. Adım: Nihayet, A sisteminin TCP zamanı dolar ve ACK bilgisini alamadığı segmentleri yeniden gönderir. Bu örnekte, 303 ve 603'le başlayan segmentleri yeniden yollanmaktadır.

8. Adım: B sistemi 303 ve 603 ile başlayan segmentlerini alınca ve hata kontrollerini yapınca tüm segment trafiği açıklanmış (accounted) olur ve B sistemi, 903 değerinde bir ACK bilgisini geri gönderir.

7. adımda gerçekleştirilen uygulamanın avantajları ve dezavantajları vardır. Bu protokolü oldukça basit yapar, çünkü TCP son ACK bilgisi alamadığı segmente bakar ve tüm bunu izleyen segmentleri yeniden iletir. Diğer taraftan, hatasız giden segmentleride yeniden iletmesi söz konusudur, örneğimiz açısından 603 numarası ile başlayan segment gibi ki bu hatasız olarak B sistemine varmıştı. TCP basitlik uğruna bu şekilde çalışarak, azaltılmış akış riskini göze almıştır

2.4.2.6. TCP bağlantı yönetim işlemleri

TCP'nin bağlantı temelli yapısı ve iletişimde uymak zorunda olduğu kurallı yapı üç temel başlıkta incelenecektir.

TCP OPEN :
Şekil 2.12'da bağlantı kuran iki TCP varlığı arasındaki ana işlemler gösterilmiştir. TCP A'nın kullanıcısı TCP'ye bir active-open primitive'i göndermiştir. Uzak kullanıcı kendi TCP sağlayıcısına bir passive-open göndermiştir. Bu olaylar, sırası ile 1 ve 2 olayları olarak belirtilmiştir. Bu olaylardan her ikisi de diğerinden daha önce olmuş olabilir.

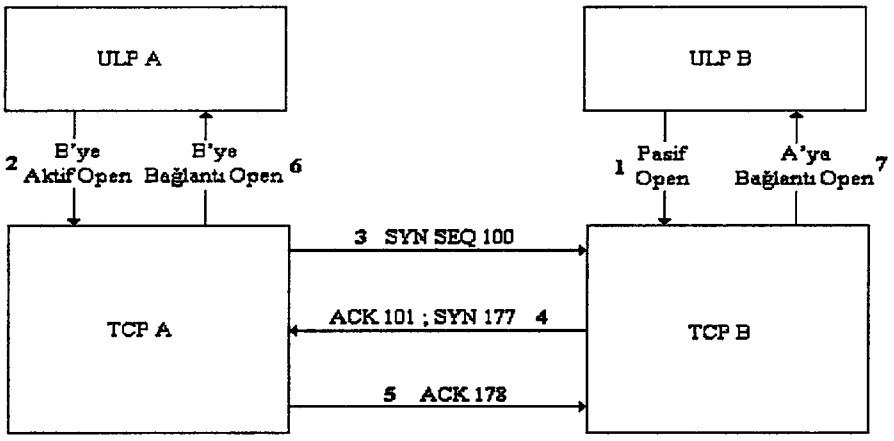
Bir active-open meydana getirmek için TCP A'nın; SYN bayrağı 1'e set edilmiş bir segment hazırlaması gerekir. 3. adımda SYN SEQ 100 olarak kodlanmış segment,

TCP B'ye gönderilir. ISS numarası olarak SEQ=100 kullanılmıştır. En yaygın yaklaşım ISS değerini 0 yapmaktır ancak ISS değeri 0-231 arasında herhangi bir sayı seçilebilir. SYN kodlaması basitçe SYN bayrağının 1'e set edildiğini gösterir.

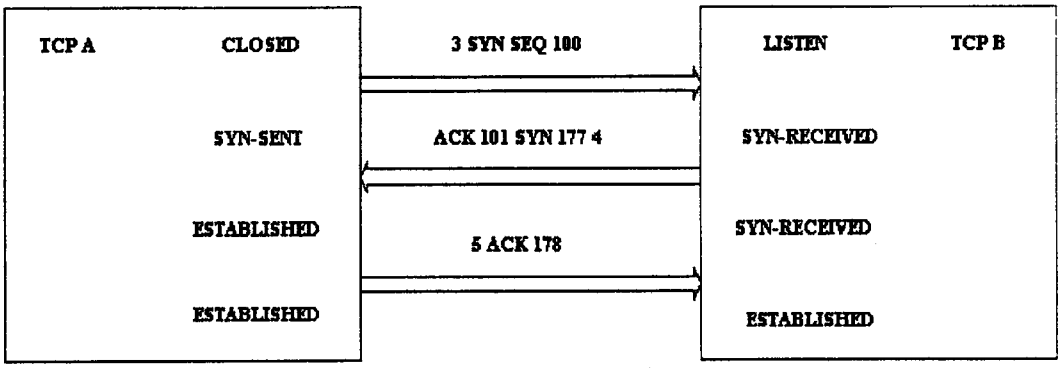
TCP B, SYN segmentini alınca 101 sıra numaralı bir ACK'yı geri gönderir. Aynı zamanda kendi ISS numarası olarak 177'yi belirlemştir. Bu olay, 4. adım olarak etiketlenmiştir. Bu segmentin alınması ile, TCP A ACK numarası 178'i içeren bir segmentle onay yollar (Şekil 2.12'daki 5 adım).

3, 4, ve 5 nolu adımlarla handshaking (el sıkışma) işlemleri oluşunca (ki buna üç-yollu handshake denir), iki TCP modülü, 6 ve 7 adımlarda olduğu gibi, kendi kullanıcılarına OPEN'lar gönderirler.

Şekil 2.12'deki işlemlerin konum diyagramı kuralları ile ilişkisi Şekil 2.13 ve Şekil 2.14'de gösterilmiştir. Bu şekiller yalnızca iki TCP varlığı arasındaki segment akışını içerir, her bir cihaz içindeki uygulama ve TCP katmanları arasındaki işlemleri içermez. Şekil 2.13'de 3 olarak etiketli işlemde TCP A'nın SYN SEQ 100 yayınladığı görülür. Bu segmentin iletimi öncesi, TCP A bu özel kullanıcı oturumu için CLOSED konumundadır. TCP A, segmenti TCP B'ye gönderdikten sonra konumunu SYN-SENT olarak değiştirir ve bağlantı için, konum diyagramında gösterildiği gibi, bir TCB girişi oluşturur (Şekil 2.14'de A-3 etiketli işlem).

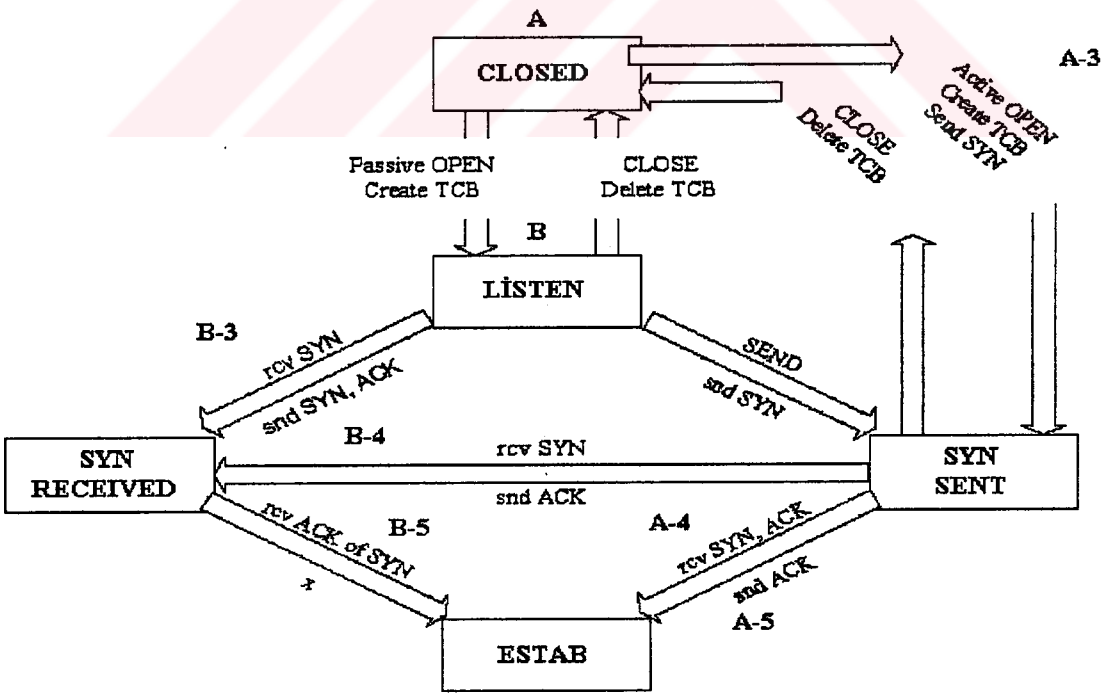


Şekil 2.12 TCP open işlemleri

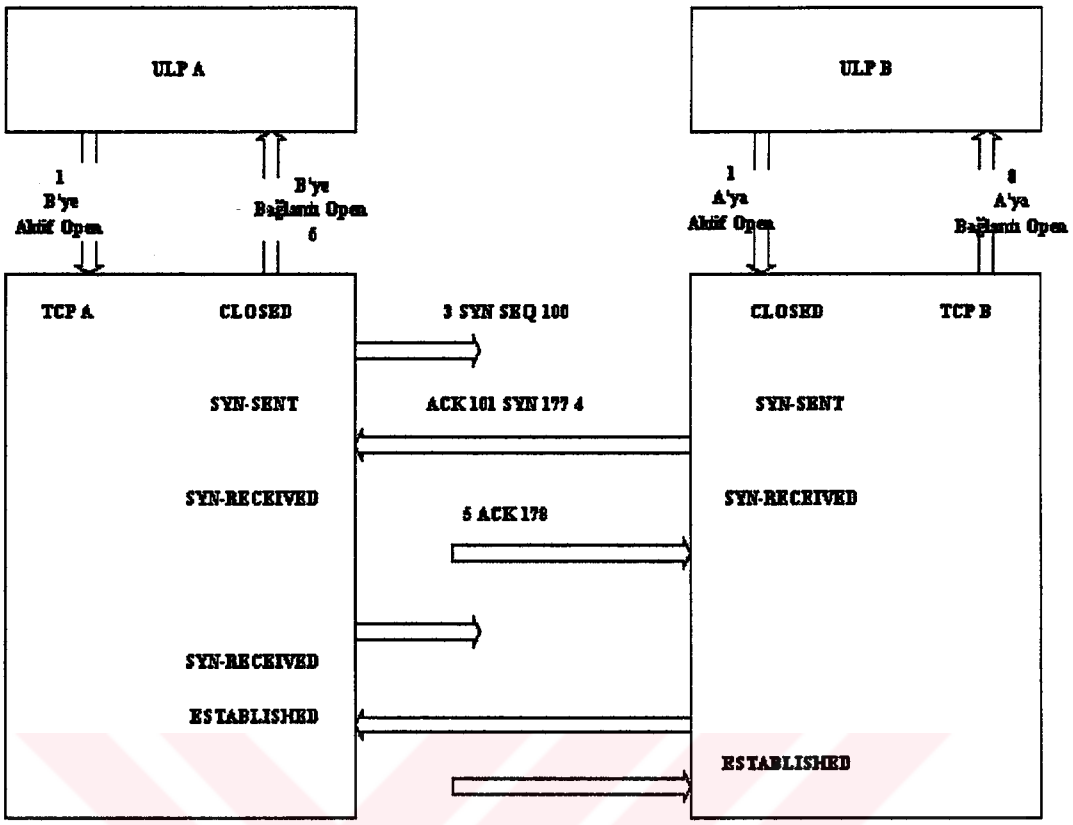


Şekil 2.13 İki uç sistem arasında TCP bağlantısının kurulması

Şekil 2.13'deki konum diyagramında görüleceği üzere TCP B LISTEN konumundadır ve SYN SEQ 100 segmentini alınca, TCP B SYN-RECEIVED konumuna geçer. Bu olaylar konum diyagramında B-3 ile etiketlenmiştir. Şekil 2.13'de 4. adım, Şekil 2.14 için B-4 etiketli adımda; TCP B'nin SYN ve ACK geri yolladığı görülüyor.



Şekil 2.14 TCP OPEN işlemleri, segment alışverişi, ve konum geçişlerinin ilişkisi



Şekil 2.15 Closed konumlara eşzamanlı open yayınlanması

TCP modülü kapalı bir TCP soketini başlatabilir mi? Yani, bir bağlantı oluşmadan önce active-open'a karşılık verecek bir passive-open olmalı mıdır? TCP kapalı soketlere OPEN yayınlanmasına müsaade eder. Kapalı bir TCP soketine bir OPEN yayınlandığında, ana gereksinimler şunlardır: OPEN çağrısı yerel ve yabancı soket tanımlayıcılarını içermelidir. OPEN çağrısı aynı zamanda öncelik, güvenlik ve kullanıcı zaman aşımı bilgilerinide içermelidir. Eğer bu bilgiler mevcut ise, TCP modülü SYN segmentini yayınlayacaktır. Bu durum ve eşzamanlı olarak iki TCP modülünden yayın yapıldığında TCP'nin OPEN'ları nasıl kabul ettiği Şekil 2.15'de gösterilmiştir. TCP A ve B'den yaklaşık olarak aynı anda OPEN yollanmıştır. Bu durumun gelişimi söyle olacaktır:

1. adım : TCP modülleri bu open'ları alınca, yeni iletim kontrol blokları oluştururlar.

2. adım : TCP A ve B SYN segmentlerini yaklaşık olarak aynı zamanda göndermişlerdir. Bu şekilde okların pozisyonu trafiğin göreceli zaman sırasını

göstermek için kullanılmıştır. Böylece TCP B'nin segmenti TCP A'ya ulaştığında daha TCP A'dan gönderilen SYN segmenti TCP B'ye varmamıştır.

3. Adım : Sonuçta TCP A'dan gönderilen SYN segmenti TCP B'ye varır. 2 adımdaki SYN segmentlerinin sonuçları iki TCP modülünün CLOSED'dan SYN-SENT'e ve SYN-RECEIVED'e geçmesidir.

4 & 5 Adım : İki TCP modülü de, SYN segmentlerini onaylamak üzere, birer ACK segmenti yayınlarlar. 5. adımdaki TCP B'nin segmenti 4. adımdaki TCP A'nın segmentinden önce varır. TCP'nin bu eşzamansız yönü bir Internet içerisinde değişken gecikmelere sebep olur. Gecikme her iki yönde de değişir.

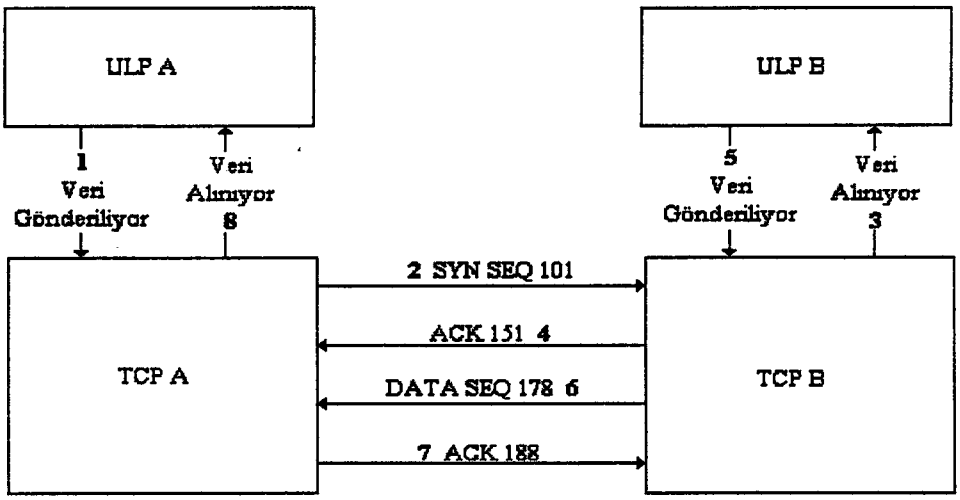
6. Adım : ACK'nın TCP A tarafından alınması ile (5. Adım), TCP A uygulama katman protokolüne bir bağlantı open işareti yollar.

7. Adım : TCP A'dan gönderilen ACK segmenti sonunda TCP B'ye ulaşır.

8. Adım : Bağlantıyı tamamlamak üzere, TCP B uygulama katman protokolüne bir bağlantı open gönderir.

TCP veri transfer işlemleri :
TCP oturumu başlatmış A ve B sistemleri Şekil 2.16'de gösterilmiştir. 1. adımda A sistemi uygulama katman protokolü, TCP A'ya iletim için bir SEND primitive'i ile veri göndermekte. Burada farz edilen segment büyüklüğü 50 bayt seçilmiştir. 2. adımda görüldüğü gibi, TCP A bir veriyi bir segment haline getirir ve segmenti TCP B'ye sıra numarası 101 ile gönderir. Sıra numarası kullanıcı veri akışının ilk baytını tanımlar.

3. adımda B sistemindeki TCP, kendisine ulaşan veriyi uygulama katman protokolüne teslim etmiştir. TCP B, 4. adımda gösterildiği gibi, veriyi 151 ACK numaralı bir segmentle onaylar. 151 ACK numarası, 2. adımda TCP B tarafından 50 baytlık segment alındığını beliritir.



Şekil 2.16 TCP veri transfer işlemleri

5. adımla TCP B'ye bağlı uygulama katman protokolünce veri gönderiliyor. Bu veri bir segment olarak paketlenir ve 6. adımda olduğu gibi iletilir. TCP B'den gelen başlangıç sıra numarası 177 idi; böylece, TCP sıralamasına 178 ile başlar. Gönderilen bu veri alanının 10 bayt olduğunu farz edilmiştir. TCP A, ACK numarası 188 olan bir segment geri döndürmekte ve 10 baytlık veriyi aldığını TCP B'ye bildiriyor. 8. adımla TCP A aldığı veriyi uygulama katman protokolüne teslim etmektedir.

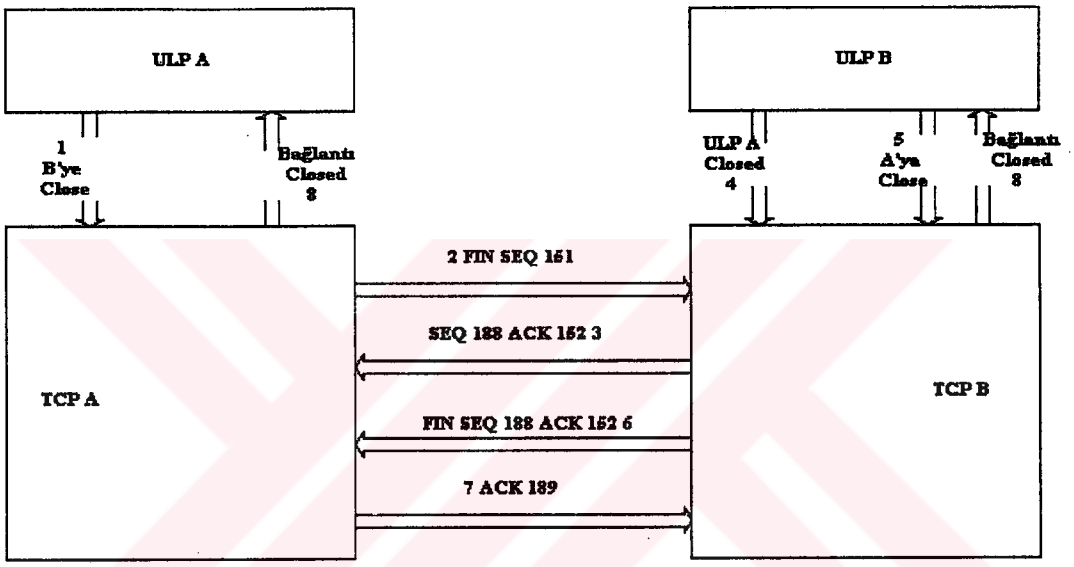
TCP CLOSE işlemleri :

Şekil 2.17'de bir CLOSE işlemi gösterilmiştir. 1. adımda; TCP A kullanıcısı, TCP B'deki eş uygulama katmanı protokolü ile işlemlerini bitirmek (CLOSE) istemektedir. 2. adımda TCP A, FIN biti 1'e set edilmiş bir segment yollar. Şekil 2.17'deki işlemlerin devamı olduğu düşünülerek 151 sıra numarası kullanılmıştır.

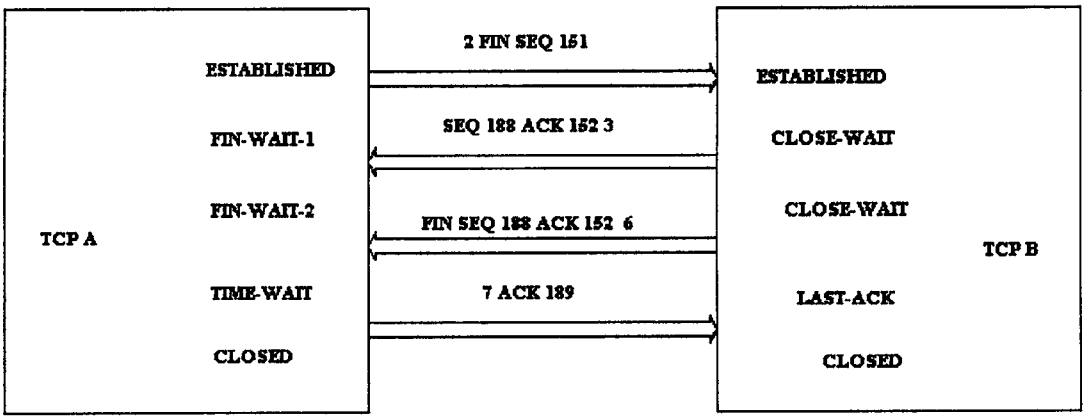
Bu segmentin TCP B'deki etkisi 3. adımda görülmektedir. TCP B, TCP A'nın FIN SEQ 151'ini onaylar. TCP B'nin segmentinde SEQ=188 ve ACK=152'dir. Bundan sonra, TCP B kendi kullanıcısına bir closing primitive'i yayınlar (4. adım).

Uygulama, işlemlerinin konumuna bağlı olarak, CLOSE'u kabul edebilir veya etmeyebilir. Bu örnekte, kullanıcı uygulaması, 5. adımda olduğu gibi, CLOSE'u

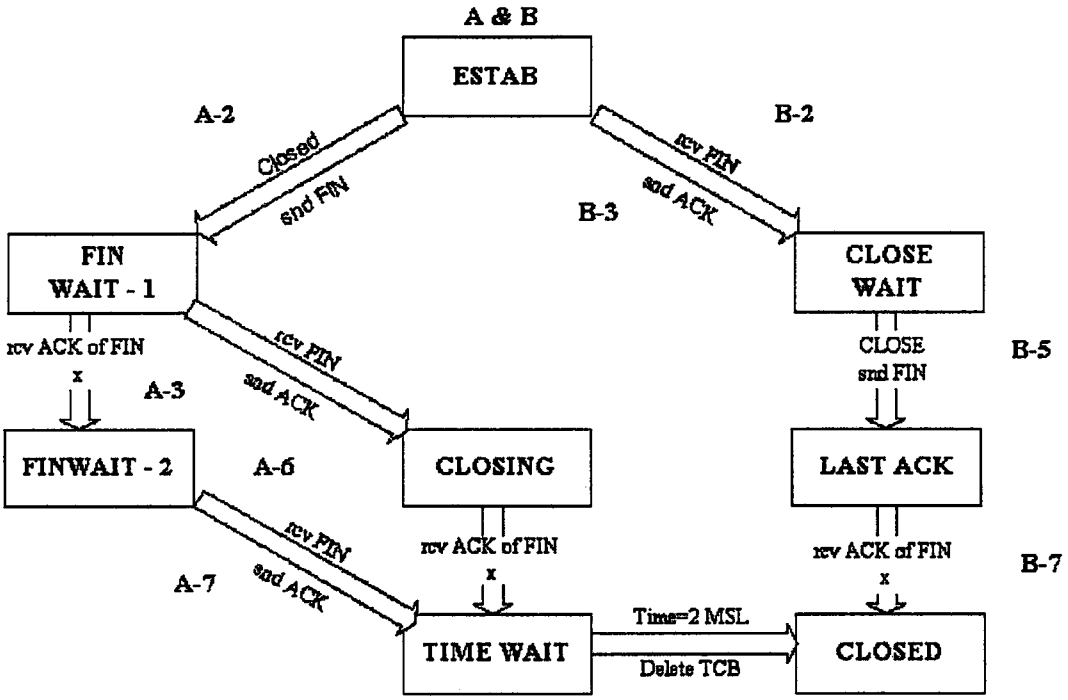
onaylar (acknowledges) ve kabul eder. Uygulama bunu takiben TCP B'ye bir "A'yı kapat" primitive'i yollar. Bu primitive TCP B tarafından FIN SEQ 188; ACK 152 olarak haritalanır, ki bu TCP B'nin yayınladığı son segmenttir(6. adım). FIN bayrağı 1'e set edilmiş, SEQ=188 ve ACK=152'dir. TCP A bu son segmenti ACK=189 ile onaylar (7. adım). Tüm bu işlemlerin sonucu 8. ve 9. adımlarda gösterilmiştir. Böylece bağlantı-closed işaretleri uygulamalara gönderilmiş olur. Bu şekilde bir CLOSE oluşturduğundan, TCP CLOSE sağlayan bir protokol olarak anılır.



Şekil 2.17 TCP CLOSE işlemleri



Şekil 2.18a TCP CLOSE işlemleri ve segment alışverişi



Şekil 2.18b TCP CLOSE konum geçişlerinin ilişkisi

2.4.2.7. TCP bağlantı tablosu

TCP bağlantı tablosu var olan her bir TCP bağlantısı ile ilgili bilgileri içerir. Şekil 2.19'da gösterildiği gibi, tablo beş sütun ve her bir bağlantı için bir satırdan oluşur. Bağlantı konumu sütunu her bir TCP bağlantısının konumunu tanımlar (closed, listen, fin wait 1, closing, vs.). Yerel adres sütunu her bir TCP bağlantısı için yerel IP adresini, yerel port sütunu her bir TCP bağlantısı için yerel port numarasını, uzak adres sütunu her bir TCP bağlantısı için uzak IP adresini ve uzak port sütunu her bir TCP bağlantısı için uzak port numarasını içerir.

	Bağlantı Konumu	Yerel Adres	Yerel Port	Uzak Adres	Uzak Port
1. Bağlantı					
2. Bağlantı					
3. Bağlantı					
...					
n. Bağlantı					

Şekil 2.19 TCP bağlantı tablosu

2.4.2.8. Güvenlik açısından TCP

Paket filtreleme açısından TCP Başlığı' nda yer alan önemli bilgiler şunlardır;

i) TCP kaynak portu.

ii) TCP hedef portu

iii) TCP bayrak alanları.

TCP bayrak alanlarından ACK biti, paket filtreleme açısından kullanılabilecek bir bilgi içerir.Bu bite bakmak suretiyle, aktarılmakta olan bir paketin, istemci tarafından sunucuya gönderilmekte olan ve bağlantıyı kurmaya yönelik ilk paket olup olmadığı anlaşılabilir. Eğer bu alan sıfır değerini taşıyorsa, inceleme konusu olan paket ilk pakettir. Bu alan, haberleşmenin bir tarafının diğer tarafa, daha önce gönderilen paketi alıp almadığını gösteren bir bilgidir. Dolayısıyla da, karşılıklı haberleşen iki sistem arasında gönderilen paketlerden sadece, istemcinin sunucuya gönderdiği ilk pakette bu alan sıfır olacak, diğer tüm paketlerde bir olacaktır.

2.4.2.9. User datagram protocol (UDP)

Diğer bir ulaşım katman protokolüdür; TCP'den farkı, sorgulama ve sınama amaçlı, küçük boyutlu verinin aktarılması için olmasıdır. Veri küçük boyutlu olduğu için parçalanmaya gerek duyulmaz, kaydı tutulmaz. UDP datagramların sıraya konulmasının gerekli olmadığı uygulama protokolleri için kullanılır. Avantajı bandgenişliğini fazlaca harcamamasıdır. Sonuçta TCP'nin sağladığı özellikleri vermez. Dolayısıyla UDP segmenti TCP segmentinden farklıdır; başlık bilgisi daha az alan içerir, daha az yük getirir.

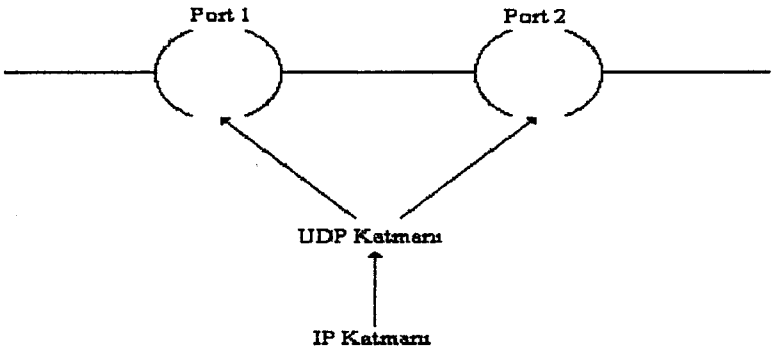
UDP, bağlantısız ulaştırma katmanı protokolüdür. Çünkü UDP normal olarak bağlantı temelli protokollerde geniş olarak kullanılan konum yönetim işlemlerini yürütmez.. UDP'de gönderilen verinin yerine ulaşıp ulaşmadığı kontrol edilmez. Sahip olduğu bu özellik gereği güvenirlilik ve akış-kontrol mekanizmaları sağlamaz. Aynı zamanda hiçbir hata bulma prosedürüde yoktur. Buna karşılık olarak UDP, TCP'den daha hızlı hizmet sunar.

UDP, IP'ye basit bir arabirim olarak hizmet verir. Güvenilirlik, akış-kontrol, veya hata-bulma ölçümleri olmadığından, prensip olarak, IP'nin alışverişi ve uygulamaların trafiği için bir port multiplexer/demultiplexer gibi hizmet eder. Yani tek bir haberleşme kanalı üzerinden birden fazla farklı ulaşım ve uygulama katmanı protokolü taşınabilmektedir. Bu olaya "multiplexing" denilmektedir. Bu tek kanaldan iletilen protokollerin hedef sistem üzerinde katmanları tırmanırken uygun şekilde ters işleme tabi tutulması gereklidir. Yani her protokol kendini ilgilendiren protokol kanalına sevk edilmelidir. Bu olaya da "demultiplexing" denir. Şekil 2.20'de UDP'nin IP'den gelen datagramları nasıl kabul ettiği gösterilmiştir.

UDP'de her datagram bağımsızdır, TCP'de olduğu gibi belli bir bağlantı üzerinden aktarılmazlar ve akış denetiminden yoksundurlar. Bunun sağladığı dezavantajlar ise; datagramlar, gönderildikleri sıradan farklı bir sırada hedefe ulaşabilirler. Ayrıca, aynı paketin birden fazla kopyası hedefe ulaşabilir. Buda değişik sorunlar doğurabilir.

2.4.2.10. UDP datagramının formatı

Bu protokolü açıklamak için en iyi yol belki de datagramını ve datagramdaki alanları incelemektir. Şekil 2.21'de gösterildiği gibi, format oldukça basittir ve aşağıdaki alanları içerir:



Şekil 2.20 UDP'nin çoğullanması

Kaynak Portu (16 bit)	Varış Portu (16 bit)
Uzunluk (16 bit)	Checksum (16 bit)
Veri (Değişken)	

Şekil 2.21 UDP datagramının formatı

Kaynak portu: Bu değer gönderici uygulama katmanı portunu tanıtır. Bu alan opsiyoneldir, ve eğer kullanılmazsa, buraya 0 değeri yerleştirilir.

Varış portu : Bu değer varış konak cihazındaki alıcı uygulama katmanı portunu tanıtır.

Uzunluk: Bu değer, başlık ve veri de içinde olmak üzere, kullanıcı datagramının uzunluğunu gösterir.

Hata kontrol toplamı (Checksum) : Bu opsiyonel değer; sözde-IP başlığı, UDP başlığı ve verinin 1'lerinin toplamının tümleyeninin, 16-bit 1'e tümlemesini içerir. UDP aynı zamanda herhangi bir doldurma için de bir hata kontrolü sağlar.

Sözde-başlık (aynı zamanda TCP'de de kullanılır) UDP veri biriminin doğru varış adresine varmasını sağlar. Sözde-başlık IP adresleri içerir ve checksum hesabına katılır. Son varış, sözde-başlığa (ve, tabii ki, UDP veri biriminin kalanına) tamamlayıcı bir checksum sağlayarak; trafiğin değişmediğini ve doğru varış adresine

vardığını kanıtlar. UDP birçok geçiş-tabanlı (transaction-based) uygulama sistemlerinde kullanılan servislerin minimal bir seviyesidir, şöyle ki eğer tüm TCP servislerine ihtiyaç yoksa UDP oldukça kullanışlı olur.

2.4.2.11. Güvenlik açısından UDP

Bazı paket filtreleme ürünleri, daha önceden gönderdikleri paketlerin üzerinde yer alan kaynak ve hedef port ve IP adres bilgilerini hatırlayarak, gelen paketleri buna göre filtrelerler. Böylece de, yapılan çalışmaya ait olmayan, beklenmeyen paketler filtrelenebilinir. Böyle filtrelemeler dinamik filtreleme olarak bilinir. Filtreleme kuralları dinamik olarak belirlenecek, nelerin filtrelenip nelerin filtrelenmeyeceğine dinamik olarak karar verilecektir. Bu da özellikle UDP protokolünü kullanan servisler için önemli bir güvenlik getirmiş olacaktır.

2.4.2.12. Remote procedure call (RPC)

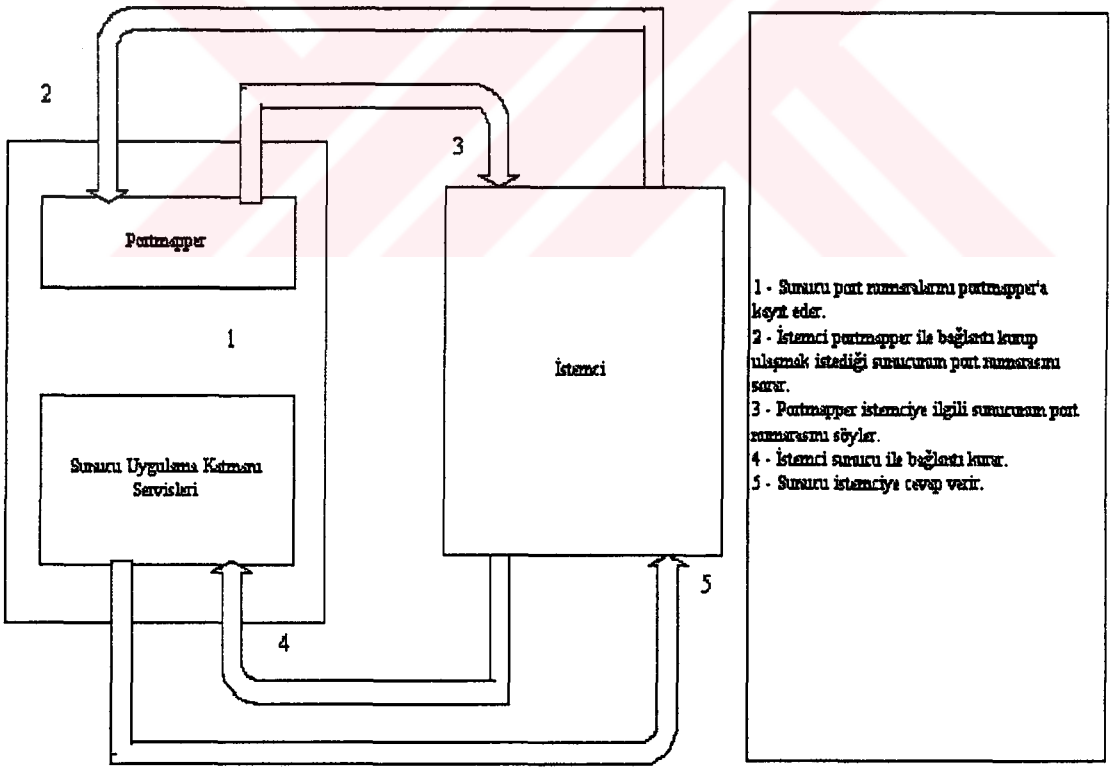
Ulaştırma katmanında IP üzerinde koşan, dolayısıyla bu protokolü kullanarak çalışan servis ve protokollerle ilgilenmemize karşın RPC daha çok TCP ve UDP protokollerinin sunduğu servisleri kullanarak işlemlerini gerçekleştirir. Yani katman olarak incelersek RPC, TCP veya UDP'nin bulunduğu ulaştırma katmanının üzerinde yer alır. NFS, NIS/YP gibi uygulamalar bu servisi kullanarak çalışmaktadır.

TCP ve UDP, her bir servise iki baytlık bir port numarası atar, işlemler bu port numarası üzerinden yapılır. Yalnız iki byte, toplam 65536 adet farklı port manasına gelmektedir ki bu da her servise bilinen, tekil bir port numarası atamanın mümkün olmayacağı manasına gelir, bunun için yeteri kadar port numarası yoktur. RPC'de bu ve benzeri sorunları aşmak üzere, dört bayttan oluşan port numaraları kullanılmıştır. Bu da 4,294,967,296 ayrı port numarası demektir ki, bununla her servise tekil bir port numarası vermek mümkündür.

Yukarıda da belirtildiği gibi RPC, TCP ve UDP protokollerinin sunduğu hizmetleri kullanarak çalışır.Bu da, RPC üzerinden çalışan uygulamalara atanmış port

numaraları ile, bunlara karşılık düşen TCP veya UDP port numaralarının belirlenmesi işleminin sürekli yapılmasını gerekli kılar. RPC port sayısı ile TCP ve UDP port sayısı eşit değildir, yani her seferinde RPC'ye ilişkin bir port, esasında başka bir TCP veya UDP portuyla eşleşmiş olabilecektir. RPC için port eşleştirme işlemini 'portmapper' denen bir sunucu gerçekleştirir.

Portmapper 111 numaralı port üzerinde koşan RPC'ye özel bir sunucudur ve RPC'ye ilişkin portlara, bir TCP veya UDP portu eşleştirmeyi garanti eder. RPC temelli bir uygulama çalışacağı zaman öncelikle kendisine bir TCP veya UDP portu alır (çalışma şekline göre hangisi gerekiyorsa, bazı durumlarda her ikisini de alır). Daha sonradan da aynı bilgisayar üzerinde çalışan portmapper sunucusuna başvurarak, kendisine tekil olarak tahsis edilmiş olan RPC numarası ile o anda kullanmakta olduğu TCP veya UDP port numaralarının eşleştirilerek kayıt edilmesini sağlar.



Şekil 2.22 RPC'nin çalışma yapısı

Daha sonradan RPC temelli bir istekçi, yine RPC temelli bir sunucuya ulaşmak istediğinde, hedef bilgisayar üzerinde, 111 numaralı portta koşan portmapper sunucusuna başvurarak, ulaşmak istediği ve tekil bir RPC port numarası olan sunucuya ilişkin kayıt bilgisini ister. Bu kayıt bilgisi, ilgili sunucunun o anda hangi TCP veya UDP portu üzerinden koştuğunu istemciye söyler. Çalışmanın bundan sonrasında istemci, elde ettiği bu port numarasını kullanarak doğrudan sunucu ile iletişimi gerçekleştirir.

2.4.2.13. Güvenlik açısından RPC

RPC temelli servisleri paket filtreleme yöntemini kullanarak kontrol etmek oldukça zordur çünkü bu servislere atanan TCP veya UDP port numaraları çalışmanın başlangıcında belli değildir ve bilgisayar kapatılıp açıldığında da büyük bir ihtimalle değişik bir port kullanacaktır. Filtreleme işleminin portmapper'a ulaşımlar için gerçeklemek yeterli olmayacaktır, çünkü korunmak istenen servisler diğer yanda TCP veya UDP portu üzerinden koşuyor olacaktır ve buna saldırmak isteyen bir kişi, portmapper'dan faydalanmadan da, kısa süreli bir deneme yanılma işlemi sonucunda aradığı servisin hangi port üzerinden koşuyor olduğunu belirleyebilecektir.

Bazı yeni paket filtreleme mekanizmaları portmapper ile irtibat haline geçerek, mevcut RPC servislerinin hangi TCP veya UDP portları üzerinden konuştuğunu belirleyerek buna göre paket filtreleme kuralları belirlemektedir. Ancak bu, özellikle UDP paketleri için aşırı yük getirecek, her UDP paketi için portmapper' dan port numarası sorgulaması yapılmasını gerektirecektir. TCP ise bağlantılı bir aktarım sunduğu için ilk paket için filtreleme çalışması yapmak yeterli olacaktır.

RPC temelli servislere filtreleme uygulama konusunda iki unsuru göz önünde bulundurmak, bir nebze de olsa çalışmaları rahatlatacaktır. Bunlardan ilki, güvenlik açısından zayıflıkları bulunan RPC temelli servislerin büyük bir kısmı UDP üzerinden çalışmaktadır. İkinci olarak, paket filtreleme uygulanarak kullanılacak diğer servislerin büyük bir kısmı da TCP temellidir. Buna gösterilebilecek en önemli istisnalar NTP, syslog ve Archie' dir. Bu iki unsur, RPC temelli servislerin de bulunduğu bir sistem için filtreleme uygulanacağında izlenmesi gereken çalışmayı

ortaya koyar: DNS, NTP, syslog ve Archie gibi, güvenlik açıkları artık bilinen ve önlemleri alınmış olan servislerin dışındaki UDP temelli servisler filtrelenmelidir.

2.4.3. Yönlendirme katmanı

Yönlendirme katmanı, ulaşım katmanından kendisine aktarılan veriyi kaynaktan varış noktasına taşımakla sorumludur. IP ve ICMP (Internet kontrol mesajı protokolü) protokolleri bu katmanda yer alır. Bu katman ağ cihazlarını ve ağları bir sistem içerisinde birbirlerine bağlamak için gerekli fonksiyonları sağlar. Yön bulma ve adres haritalama için kullanılan diğer destek protokolleri de bu katmanda IP ile birliktedirler.

2.4.3.1. Internet protocol (IP)

IP'nin açılımı Internet protokoldür ve Internet'te dahil bir çok TCP/IP ağda kullanılan yönlendirme protokolüdür. Amacı ise ulaşım katmanından gelen TCP segmentleri ya da UDP datagram yapılarını birbirine bağlı ağlar üzerinden iletmektir. Ulaşım katmanından gelen herbir segment ya da datagramın içeriği ile ilgilenilmeden IP tarafından bir başlık bilgisi eklenerek IP paketi haline getirilir ve herbir IP paketi birbirinden bağımsız varlıklar olarak varış (hedef) sisteme gönderilmek üzere yola (route) çıkarılır. Arada geçilecek sistemler ve geçiş yollarının bu paketi doğru yere göndermesi IP başlık bilgisi ile mümkün olmaktadır

Internet'teki tüm sistemler üzerinde IP protokol grubu çalışmak zorundadır. IP paketleri, ağlar arası dolaşımda pek çok sisteme uğrar ve her bir sistemde aynı kurallara göre çalışan IP katmanları ile karşılanır. IP'nin sağladığı fonksiyonlar şunlardır:

- Global adresleme yapısı
- Servis isteklerini tiplendirme
- Paketleri iletim için uygun parçalara ayırma
- Hedef alıcıda paketleri tekrar birleştirme

2.4.3.2. IP'nin ana özellikleri

IP bağlantısız servise bir örnektir. İki sistem arasında senkranizasyon kurulmaksızın veri alışverişi yapılmasını sağlar. IP bağlantısız olduğundan, paketler iki son kullanıcı istasyonu arasında kaybolabilir. Örneğin; IP yönlendirici bir maksimum kuyruk uzunluğunu zorlarsa, ve eğer kuyruk uzunluğu bozulursa, tamponlar taşar. Fazla paketler daha sonra ağda bertaraf edilir. Bu sorunların telafi edilmesi ulaşım katmanı protokollerinin (TCP gibi) oturumu düzenleme yetenekleri ile aşılır.

IP protokolü, alt katmanında bulunan fiziksel ağı üst katmanındaki protokollerden saklar. Bu açıdan IP üst katmanlar için hayali bir ağ meydana getirir. Bir IP geçitine (gateway) farklı tiplerde ağlar bağlanabilir. Bunun bir sonucu, IP'nin kurulması oldukça kolaydır ve bağlantısız yapısından dolayı oldukça sağlamdır. Ancak IP güvenilir olmayan, yüksek-zorluklu (best effort), datagram-tipi bir protokol olduğundan, hiç bir güvenilirlik mekanizması yoktur. Fiziksel katmanda yer alan ağ yapıları için hiçbir hata telafisi sağlamaz, akış-kontrol mekanizması bulunmaz. Ulaşım katmanından iletilen veriler (datagramlar veya segmentler) kaybolabilir, çiftlenebilir veya hedefe sırası bozuk olarak ulaşabilir. Bu problemlerin bir çoğu ile IP uğraşmaz. Bu sorunların çözümü ulaşım katman protokolünün asli görevidir.

IP fragmantasyon işlemini destekler. Fragmantasyon, bir üst katmandan gelen verinin daha küçük parçalara bölünmesi işlemine denir. Bu özellik oldukça kullanışlıdır çünkü ağların çoğu aynı büyüklükte veri parçaları (protocol data unit, PDU) kullanmazlar. Örneğin, X.25 tabanlı WAN'lar tipik olarak 128 bayt veri alanlı bir PDU kullanırken, Ethernet standardı, bir PDU büyüklüğü 1500 bayttır. Fragmantasyon kullanılmasaydı, ağlar arasındaki birbirine uymayan PDU büyüklükleri sorununu çözmek için bir yönlendirici tahsis edilmesi gerekecekti. IP yönlendiricilerde fragmantasyon yaparak ve alıcı sistemde yeniden birleştirmeyle bu problem çözülür.

2.4.3.3. IP ve altağlar

IP, fiziksel katman üstüne yerleşmiştir ve olabildiğince transparandır. Bu; IP, altındaki ağ veya ağların karakteristikleri ile çok az ilgilenir, altağları göreceli olarak kendisinden bağımsız tuttuğu anlamına gelir.

IP paketleri, bir IP yönlendiriciye geldiğinde, paketlerin IP adresleri bir yönlendirme tablosu ile eşleştirilir. Yönlendiricideki IP yönlendirme tablosu girişleri baz alınarak, paketler bir sonraki ağa veya doğrudan alıcı sisteme yönlendirlir. Transparanlık her katmanda yeniden paketleme yapılarak başarılır. Bir sistemin göndereceği veri IP paketine dönüştürülür ve IP başlığında alıcı sistemin IP adresi tanımlanır. IP paketi daha sonra transit geçeceği ağın yapısına göre fiziksel katman tarafından özel çerçeveler haline dönüştürülür. Örneğin transit ağ bir X.25 ağı veya Ethernet LAN'ı olabilir.

Transit ağ trafiği bir IP geçitine (gateway) vardığında, transit ağın kontrol bilgisi atılır. Geçit, daha sonra paket başlığındaki varış adresine bakarak paketi nereye yönlendireceğine karar verir. Paket yeni ağ ortamına göre fiziksel katman tarafından uygun çerçeve formatına sokulur. Bu işlem her bir yönlendiricide tekrarlanarak veri alıcısının bulunduğu ağa ve orada da alıcı düğüme teslim edilir.

Netice olarak IP transit geçeceği ağın çalışma yapısıyla ilgisiz tutulmaya çalışılmış mümkün olduğunca soyut tutumuştur.

2.4.3.4. IP paketi

Şekil 2.23'de bir IP paketi görülmektedir. IP paketinin sahip olduğu alanlar:

Versiyon Alanı : Versiyon alanı kullanımdaki IP sürümünü tanımlar. Çoğu protokolde bu alan vardır çünkü bazı ağ düğümlerinde protokolün son sürümleri bulunmayabilir.

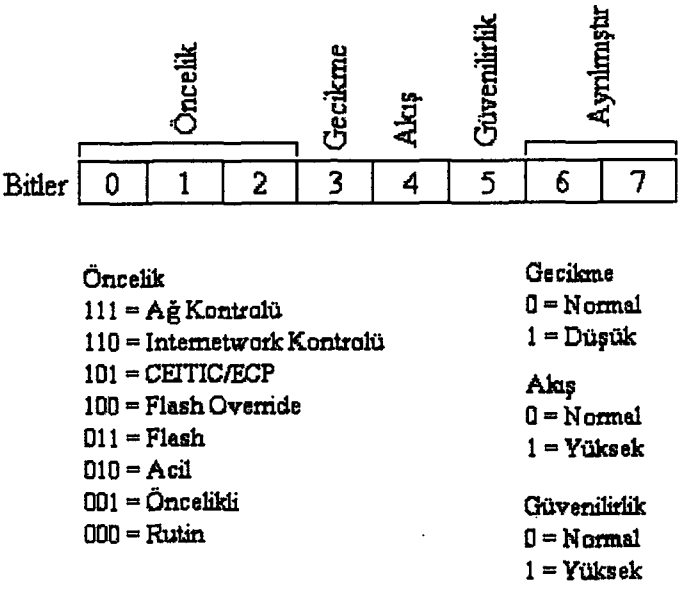
Versiyon (4)	Başlık Uzunluğu (4)
Servis-Tipi (TOS)	
Toplam Uzunluk (16)	
Tanımlayıcı (16)	
Bayraklar (3)	Fragman Offset (13)
Yaşama Zamanı (TTL)	
Protokol (8)	
Header Checksum (16)	
Kaynak Adresi (32)	
Varış Adresi (32)	
Opsiyonlar ve Dolgu (Değişken)	
Veri (Değişken)	

(n) = Alandaki bitlerin sayısı

Şekil 2.23 IP paketi

Başlık Uzunluğu (header lenght) : Başlığın 4 biti kullanılarak paketin başlığının uzunluğu gösterilir. Uzunluk 32-bit kelimelerle ölçülür. Tipik olarak, QOS (quality-of-service) opsiyonu olmayan bir başlık 20 bayt içerir. Böylece uzunluk alanındaki değer genelde 5 olur.

Servis Tipi (type-of-service (TOS)) : Bu alan bir ağın sağladığı belirli QOS fonksiyonlarını tanımlamak için kullanılabilir. Paketlere bu alan aracılığıyla önem düzeyi atanabilir, göndericinin ağdan beklediği güvenilirlik, hız ve gecikmenin düzeyini belirtilebilir. Ancak bu alanı mevcut yönlendiricilerin pek azı değerlendirmektedir. TOS alanı Şekil 2.24'de gösterilmiştir. Toplam 8 bit yer tutan beş girişe sahiptir. 0, 1 ve 2 bitleri bir öncelik bilgisi taşıyarak datagramın göreceli önemini gösterirler. Değerler 0'dan 7'ye değişir. 0'a ayarlanması bir rutin önceliği gösterir. Öncelik alanı tüm sistemlerde kullanılmaz. 7 değeri bazı uygulamalarda bir ağ kontrol paketini göstermek üzere kullanılır. Öncelik alanı aynı zamanda bir ağdaki akış kontrol ve tıkanıklık mekanizmalarının kurulmasında kullanılabilir. Bu geçit ve yönlendirici düğümlerin sıkışıklık durumlarında paketleri yok etme sırasına karar vermelerini sağlar.



Şekil 2.24 IP TOS alanı

Takip eden 3 bit diğer servisler için kullanılır. 3. biti gecikme bitidir. 1'e ayarlanması, ağ üzerinde kısa bir gecikme istediğini beliritr. Gecikme durumu standartlarda tanımlanmamıştır ve satıcı servisi geliştirmelidir. Sonraki bit akış bitidir (througput). Bunun 1'e ayarlanması ağ üzerinde hızlı bir akış isteği anlamı taşır. Bunun da standartlarla özel bir yolu belirlenmemiştir. 5. bit güvenilirlik bitidir ki kullanıcıya datagramı için yüksek bir güvenilirlik isteme olanağı tanır. 6 ve 7 bitleri şimdilik kullanılmamaktadır.

Toplam Uzunluk (total lenght) : IP paketinin toplam uzunluğunu gösterir. Bayt cinsinden ölçülür ve başlık ile veri uzunluğunu içerir. IP, toplam uzunluktan başlık uzunluğunu çıkararak veri alanı için ayrılmış alanın büyüklüğünü hesap eder. Bir paketin maksimum olası uzunluğu 65535 bayttır. IP paketlerine servis veren bir yönlendiricinin, kendine bağlı ağların paketlerine sağladığı maksimum PDU uzunluklarını desteklemesi gerekir.

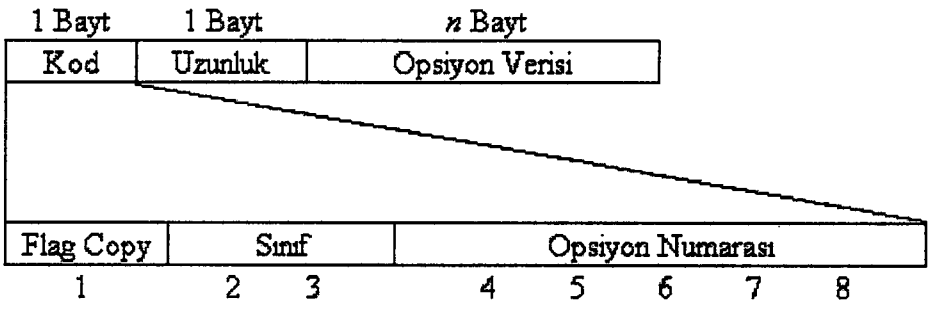
Bayrak Bitleri (Flags) : Üç tane olan bayrak bitlerinden ilki *D bit* (Don't Fragment, DF), içinde bulunduğu paketin kaç parçadan oluştuğunu belirtir, eğer 1 ise gönderilen verinin tek parçadan oluştuğu anlaşılır; alıcıya başkası yok bekleme anlamında mesaj

iletir, ikinci bayraksa MF (More Fragment, M Biti), parçalanıp birçok fragmanlar halinde gönderilen verinin en son olduğunu belirtir. Bir datagramın son fragment'i dışındaki tüm fragment'lerinde MF=l'dir. Üçüncüsü saklı tutulmuştur.

Yaşama Zamanı (time-to-live (TTL)) : Bu parametre bir paketin ağ üzerinde kalma süresinin ölçüsüdür. Her yönlendirici kendisine gelen paketin TTL alanına bakar ve eğer değer 0 ise paketi yok eder. Ayrıca bir yönlendirici üzerinden geçirdiği her paketin TTL alanını değerini 1 azaltır. Sonuçta TTL alanı paketin aşacağı düğüm sayısını belirtir. Böylece, bir IP paketi bir düğüm ilerleyince, TTL alanın değeri 1 azaltılır. IP uygulamaları bu alanda bir zaman sayıcısıda kullanabilirler ve azaltma saniyede bir olur. TTL alanı yalnızca sonsuz döngüleri engellemek için kullanılmaz. Aynı zamanda ağ yönetim protokollerince bilgi toplama amaçlıda kullanılır.

Protokol : Bu alan, alıcı sistemde IP üzerindeki katmanda bulunan protokollerden hangisinin paketin gövde kısmını alacağını tanımlar. Çokça kullanılan üst-katman protokolleriniyle eşleştirilmiş bir numaralandırma sistemi geliştirilmiştir. Örneğin TCP=6, ICMP=1, IGMP=2 vs.

Başlık Sınama Bitleri (Header Checksum) : Başlıkta olabilecek bozulmaları algılamak için kullanılır. Her yönlendiricide bu alandaki değer kullanılarak paket başlığının bozulup bozulmadığı araştırılır. Sonuç olumlu ise paket bir sonraki yönlendiriciye gönderilir. Bu arada başlıktaki bazı değerlerle birlikte (örneğin TTL) bu alandaki değer de gönderilen pakette yeniden hesaplanır.



Şekil 2.25 IP opsiyon alanı

Kaynak ve Varış Adresleri : IP paketinde, iki adres taşınır. Bunlar kaynak ve varış adresleri olarak etiketlenir ve paket yaşadıkça bu değerler aynı kalır. Internet adreslerini (IP numarası) içerirler.

Opsiyonlar : Bu alan belirli ek servisleri tanımlamada kullanılırlar. Opsiyonlar alanı tüm paketlerde kullanılmaz. Uygulamaların çoğu bu alanı ağ yönetimi ve teşhisler için kullanırlar. Şekil 2.25'de bir opsiyon alanının formatı gösterilmektedir. Tablo 2.3'da opsiyon alanı için tanımlanmış standart değerler gösterilmiştir. Opsiyon alanının uzunluğu değişkendir, çünkü bazı opsiyonların uzunluğu değişir. Her opsiyonun 3 alanı mevcuttur. İlk alan tek bir bayttan oluşan opsiyon kodunu içerir. Opsiyon kodunun kendisi de üç alandan oluşur. Bunların fonksiyonları aşağıdaki gibidir:

Flag Copy (1 bit):

0 = Fragmantasyonlu bir paket ve yalnızca ilk fragman içinde kopya opsiyonu vardır.

1 = Fragmantasyonlu bir paket tüm fragmanları içinde kopya opsiyonu vardır.

Sınıf (2 bit) : Opsiyon sınıfını belirler (Tablo 2.3) . Aldığı değerin anlamları ise 0= Kullanıcı veya ağ kontrol paketi, 1 = Ayrılmıştır (reserved), 2 = Hata giderme ve teşhis amaçlı, 3 = Ayrılmıştır.

Opsiyon numarası : Opsiyon numarasını tanıtır (Tablo 2.3).

Diğer bayt opsiyonun uzunluğunu içerir. Üçüncü alan opsiyonun veri değerlerini içerir.

Tablo 2.3 Opsiyon Kodları

Sınıf	Numara	Uzunluk	Tanım
0	0	0	Opsiyon listesinin sonu
0	1	0	İşlem yok
0	2	11	Güvenlik
0	3	Değişken	Esnek kaynak yönlendirme
0	7	Değişken	Yönü kaydet
0	8	4	Akış ID'si (eski bir alan)
0	9	Değişken	Sert kaynak yönlendirme
2	4	değişken	Internet zaman damgası

Padding (dolgu) : Paket başlığını 32-bit'e tamamlamak için kullanılır.

Veri Alanı : Ulaşım katmanından gelen veriyi içerir. IP, veri alanı ve başlığının toplamının 65535 baytı geçmesine izin vermez.

2.4.3.5. IP servisleri

IP başlık kontrolü ve yönlendirme : Bir yönlendirici IP paketi alınca, IP başlık kontrol modülüne iletir. Bu modül tarafından IP paket başlığı üzerinde bir takım düzenlemeler ve doğruluk testleri gerçekleştirilir. Başlık üzerinde aşağıdaki testler uygulanır:

- Geçerli IP başlık uzunluğu,
- Doğru IP sürüm numarası,
- Geçerli IP mesaj uzunluğu,
- Geçerli IP başlık sınaması,
- Sıfırlanmamış TTL geçerliliği.

Eğer IP paketi testleri geçemezse yok edilir. Eğer testler yapılır ve IP paketi testleri geçerse, paketin varış adresi incelenerek, bu yönlendiriciye mi? başka bir yönlendiriciye mi? ait olduğuna karar verilir. Eğer paket bu yönlendirici için değilse, IP yönlendirme modülüne atılarak yönlendirilir.

Yönlendirme : IP yönlendiriciler, yönlendirme tablolarına dayanarak yönlendirme kararları alır. Eğer hedef sistem başka bir ağda ise IP yönlendirici, paketi diğer ağa nasıl yönlendireceğine karar vermelidir.

Her bir yönlendirici, varış ağına gitmek için geçilmesi gereken diğer yönlendiricilerin bilgilerini bir yönlendirme tablosunda tutar. Tablo her bir erişilebilir ağ için bir IP adresi ve kendine komşu bir yönlendiricinin adresini saklar. Yönlendirme tabloları oluşturulmalarına göre statik veya dinamik olabilir. Dinamik tablolar daha yaygındır.

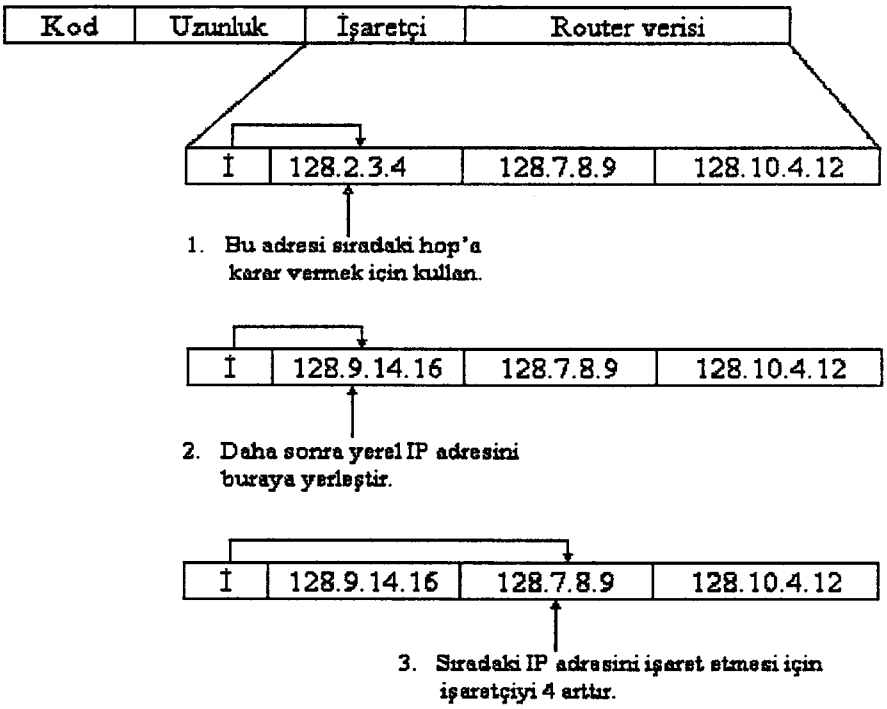
IP modülü, aldığı tüm paketler için bir yönlendirme kararı verirken yönlendirme için seçilen komşu yönlendirici varış ağı için en kısa yönlendirmedir. Eğer bir komşu yönlendirici için hiç bir adres yoksa, IP yönlendirme mantığı gereği yönlendiricinin doğrudan bu ağa bağlı olduğu çıkartılır.

Yönlendirici, yönlendirme tablosuna başvurur ve IP başlığında bulduğu varış ağ adresi ile yönlendirme tablosunda bulunan bir ağ girişini eşleştirmeye çalışır. Eğer eşleşme yoksa, paket yok edilir ve IP kaynağına bir ICMP mesajı iletilir. Bu mesaj "varış ulaşılamazdır" kodu içerecektir. Eğer yönlendirme tablosunda bir eşleşme mevcut ise yönlendirici bunu kullanarak çıkışın yapılacağı porta karar verir.

IP kaynak yönlendirme : Kaynak yönlendirme mekanizması, yönlendirme algoritmasının parçası olarak kullanabilir. Kaynak yönlendirme IP başlığının, IP yönlendiriciye nasıl yönlendirme yapacağını bildirmesiyle sağlanır. IP paketlerine verilen hedef adresler listesi, paketlerin hedefe giderken geçecekleri ara IP düğümlerini içerir. Listedeki son adres varılacak son ara düğümün adresidir.

IP yönlendirici modulü, bir paketi alınca, bir sonraki ara düğümü belirlemek için kaynak yönlendirme alanındaki adresleri kullanır. Şekil 2.26'da gösterildiği gibi bir IP, kendinden sonraki diğer IP adresini öğrenmek için bir işaretçi alanı kullanır. İşaretçi ve uzunluk alanları listenin tamamlandığını gösterene kadar yönlendirme için varış IP adresi bu listeden elde edilir. Eğer liste bitmemişse IP modülü işaretçinin gösterdiği adresi kullanır.

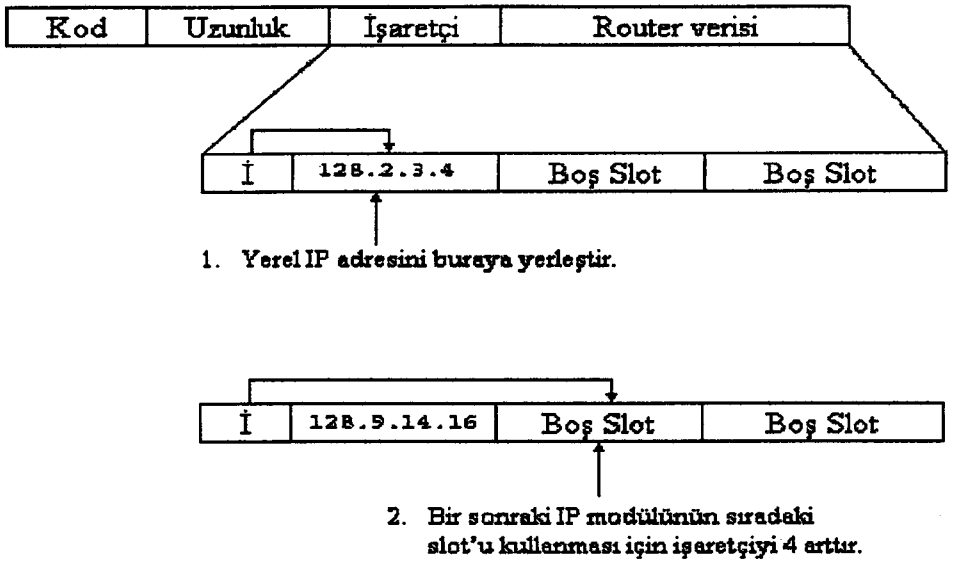
IP modülü sonra kaynak yönlendirme listesindeki değerin yerine kendi adresini yerleştirir. Daha sonra ise işaretçiyi, bir adres (4 bayt) arttırarak diğer düğümün sıradaki adresi kullanabilmesini sağlar. Bu yaklaşım ile paket dikte edilen yönlendirmeyi izler ve yönlendirilen yol boyunca geçtiği IP'ler kaydedilir.



Şekil 2.26 Kaynak yönlendirme

Yönlendirme kaydı : Kaynak yönlendirmeye benzer biçimde çalışır, ancak aynı zamanda kayıt özelliğini kullanır. Böylece her bir IP modülü bir paket alınca, pakete adresini, yönlendirme kayıt listesi olarak ekler. Yönlendirme kayıt işleminin yapılması için alıcı IP modülü, işaretçi (pointer) ve uzunluk alanlarını inceleyerek yönlendirmeyi kayıt etmek için boş yerinin olup olmadığına karar verir. Eğer yönlendirme kayıt listesi dolu ise IP modülü pakete adresini kayıt edemeden onu ilerletir. Eğer dolu değilse işaretçi kullanılarak ilk boş tam-oktet slotuna adres yerleştirilir ve sonra IP modülü işaretçiyi sıradaki IP slotunu göstermek üzere arttırır.

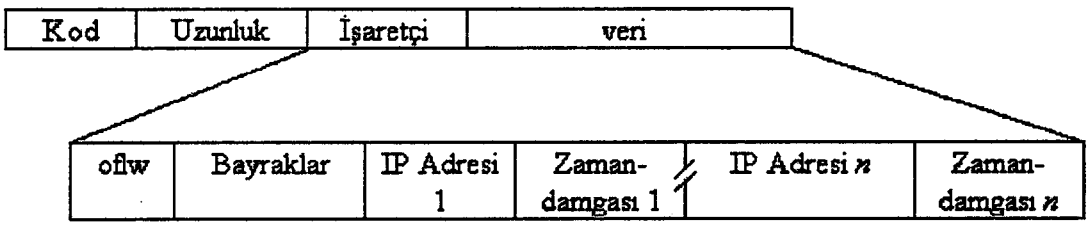
Şekil 2.27'de yönlendirme kaydı ile ilgili bir örnek gösterilmiştir. İlk adımda IP işaretçi kullanılarak yönlendirme verisi alanındaki sıradaki adrese yani 128.2.3.4'e kilitlenilir. Bu adrese dayanarak bir yönlendirme kararı verilir. İkinci adımda, şimdiki varış adresini belirtmek üzere yönlendirme verisi alanındaki 128.2.3.4 adresi yerine yönlendirici kendi adresini koyar. Üçüncü adımda, işaretçi değeri arttırılır.



Şekil 2.27 Yönlendirme kaydı

Zaman-damgası opsiyonu (time-stamp option) : IP'deki kullanışlı seçeneklerden biri; paketin ağ içerisinde her bir IP modülünü geçerken, paketin zaman-damgasının görülebilmesidir. Bu fikir bir ağ yöneticisinin paketin ağdaki yönlendirmesine karar vermesini sağladığı gibi aynı zamanda her bir IP modülünün paketi işleme zamanını bilmesini de sağlar. Bu özellik ile yönlendiricilerin, ağların ve yönlendirme algoritmalarının etkinlikleri karşılaştırılabilir.

Şekil 2.28'de zaman-damgası işlemlerinin opsiyon alanlarının formatı gösterilmektedir. Diğer opsiyonlarda olduğu gibi bir IP adresini ve onun ilgili zaman damgasını doğru slota yerleştirmek için uzunluk ve işaretçi alanları kullanılır. Oflw alanı yalnızca bir IP modülü kaynak yetersizliği veya çok-küçük opsiyon alanı gibi nedenlerle bir zaman-damgası oluşturamamışsa kullanılır. Böyle bir sorunu olan her bir modül için bu değer arttırılır.



Şekil 2.28 Zaman-damgası opsiyonu

4-bitlik bayrak alanı her bir IP modülüne zaman-damgası işlemleri için rehberlik sağlamak amacı ile kullanılır. Bu alandaki değerleri şöyle açıklarız:

0 = ardışık 32-bitlik kelimelere yalnızca zaman-damgası kaydı ve saklanması yapılacaktır.

1 = her bir zaman damgası, ilgili modülün IP adresinin ardında yer alacaktır.

3 = IP adresleri hali hazırda kaynak modülünce tanımlanmıştır ve yönlendirici zaman-damgasını kendi ilgili IP adres bölgesi içine kaydetmekle görevlidir.

Zaman-damgasında kullanılan zaman milisaniyeler mertebesindedir ve evrensel zamana (Greenwich Mean Time) göredir. Açıkçası, evrensel zamanın kullanımı cihazlar arasında tam doğru zaman damgalarını garanti etmez, çünkü cihazların saatleri biraz değişebilir. Bununla beraber çoğu ağda milisaniyeler mertebesindeki evrensel zaman fark edilir bir doğruluk derecesi sağlar. Ağ zaman protokolü (NTP) bu opsiyon için kullanışlı bir araçtır.

Fragmantasyon ve yeniden birleştirme : Bir IP paketi farklı PDU (protocol data unit) büyüklükleri kullanan çeşitli ağlardan geçebilir. Her ağın bir maksimum PDU büyüklüğü vardır ve buna maksimum iletim birimi (MTU) denir. Bu yüzden IP, bir büyük IP gövdesini, daha küçük parçalara bölecek yöntemler içerir. Üst katman protokolü, IP'yi fragmantasyon yapabilir veya yapamaz diye şartlayabilir.

Bir IP yönlendirici modülü, transit altağın iletebileceğinden daha büyük bir paket alınca fragmantasyon işlemlerini kullanır. Paketi; 8-oktetlik sınırlarla dizecek biçimde iki veya daha çok parçaya böler. Her bir fragmana tanımlayıcı (identifier), adres ve orijinal pakate ait tüm opsiyon bilgilerini içeren bir başlık eklenir.

Şekil 2.23'u referans alırsak, bayraklar (3-bit) şöyle kullanılır.:

Bit 0 : rezervedir

Bit 1 : 0 = fragmantasyon

1 = fragmantasyon yapma

Bit 2 : 0 = son fragman

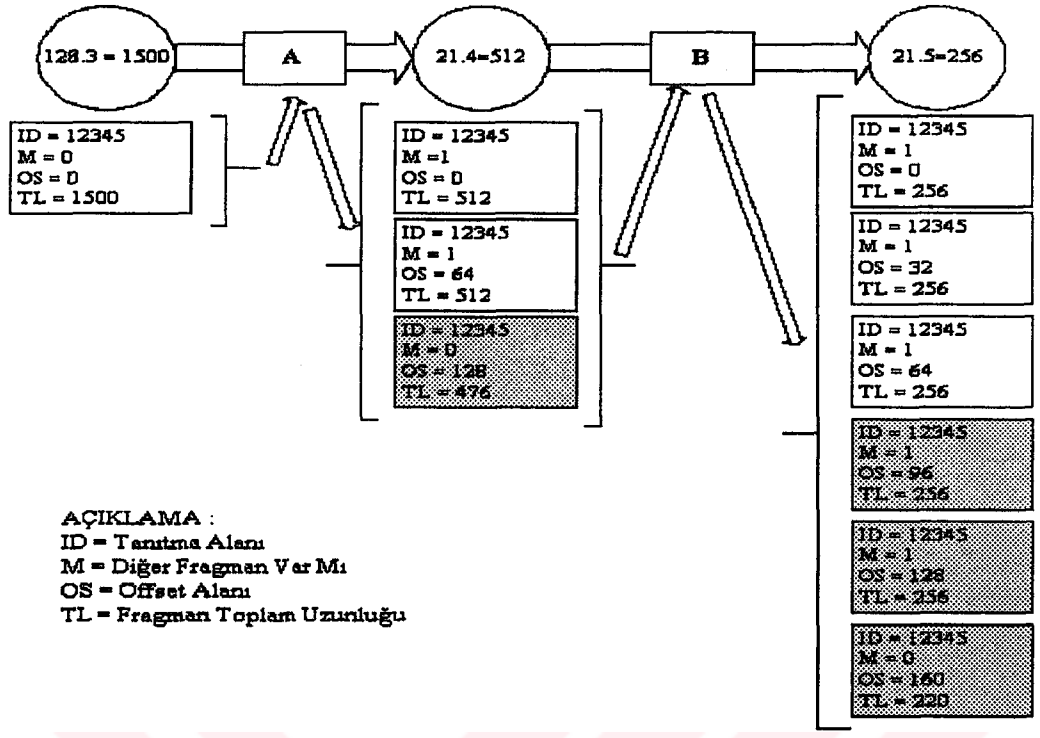
1 = daha fragman var

Bayrak alanları kullanılarak, paketin fragmantasyona müsait olup olmadığına karar verilir. Eğer müsaitse, bitlerden biri set edilerek bu fragmanın paketin son fragmanı olup olmadığına karar verilebilir. Fragmantasyon offset alanı, fragmanın orijinal pakete olan göreceli pozisyonunu belirten değeri içerir. İlk değer 0'dır, sonra yönlendirici fragmantasyon yaparsa sıradaki değerler set edilir. Değer sekiz oktet birimi ile ölçülür. İlginçtir ki IP, her bir fragmanı ayrık olarak ele alır. Şöyle ki, fragmanlar son varışa gitmek için farklı yönlendiricilerden geçebilirler ve eğer geçtikleri ağ daha küçük veri birimleri kullanıyorsa tekrar fragmantasyona uğrayabilirler. Fragmanlarda offset değeri bunu belirtmek ve fragman sıralarının karışmasını önlemek üzere ayarlanır. Şekil 2.29'da iki yönlendirici boyunca yapılan çoklu fragmantasyon gösterilmiştir.

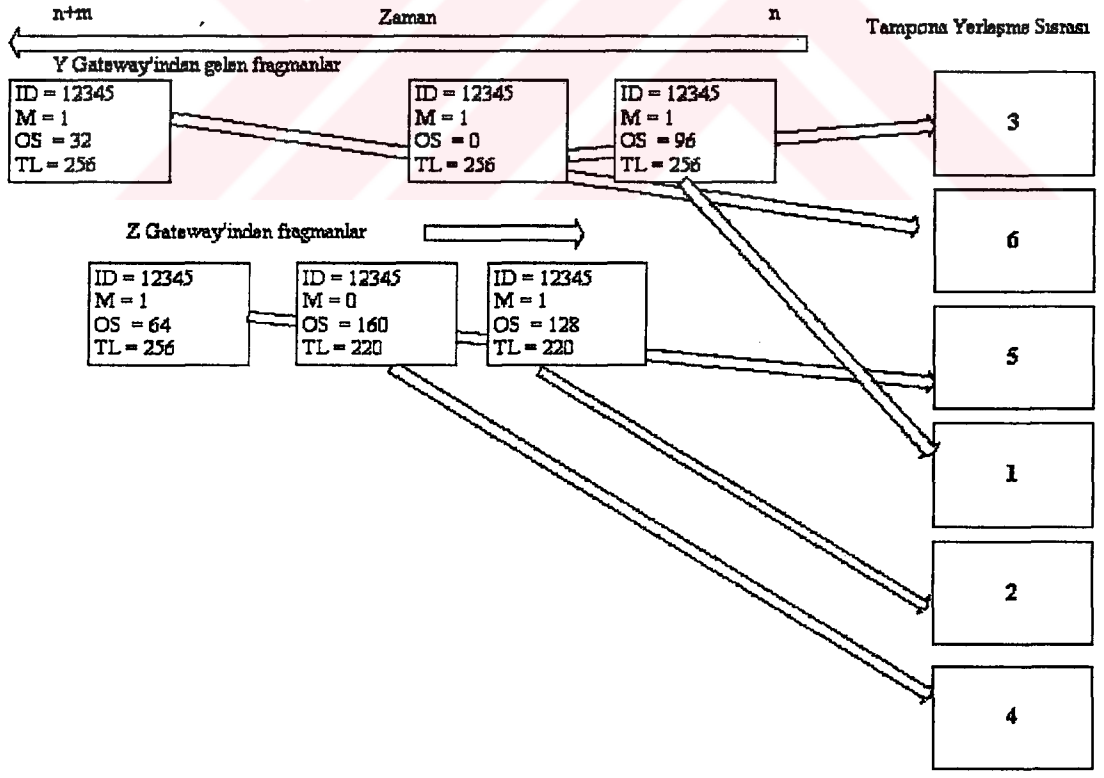
Altağ 128.3 1500-bayt PDU büyüklüğü kullanır. 128.3 altağı bu veri birimini A yönlendiricisine geçirir. A yönlendiricisi PDU'yu 21.4 altağına yönlendirmeye karar verir. 21.4 altağı 512-bayt PDU büyüklüğünü desteklemektedir. Yönlendiriciler 1500'lük veri birimini 512, 512 ve 476 baytlık 3 küçük parçaya ayırır (1500 = 512 + 512 + 476). 476 baytlık son parçaya, 8'in tam katına ulaşmak için 0'lar eklenir. Böylece bu veri alanı 480 bayt (476 + 4 = 480, ki 8'in tam katıdır) olur.

A yönlendiricisi veriyi 21.4 altağına iletir. Veri buradan da B yönlendiricisine iletilir. Bu yönlendirci, fragmanları 21.5 altağına göndermeye karar verir. B yönlendiricisi 21.5'in 256 bayt büyüklüğünde PDU'lar kullandığını bildiğinden, yeni bir fragmantasyon gerçekleştirir. 512 bayt büyüklüğündeki fragmanları daha küçük veri birimlerine böler ve gelen üç fragmanın offset değerlerini kullanarak gereğince yeni veri birimlerinin offset değerlerini ayarlar. Offset değerleri B yönlendiricisinde değiştirilmiştir ve yeni değerler önceki fragmanların offset değerleri kullanılarak türetilmiştir.

Şekil 2.30'da, alıcı sistemde paketlerin yeniden-birleştirilmesi gösterilmektedir. IP modülü ilk fragmanı alınca tampon boşluğunu hazırlar. Her bir fragman için bir tampon ayrılır ve fragman göreceli olarak orijinal paketteki yerine göre tampondaki yerine yerleştirilir. Tüm fragmanlar gelince IP modülü veriyi, gönderici üst katmanın yollandığı orijinal düzeni ile üst katmanına geçirir.



Şekil 2.29 Gateway'lerdeki fragmantasyon işlemleri



Şekil 2.30 Fragmanların yeniden birleştirilmesi

Konuyu tartışabilmek için Şekil 2.30'da görülen paket fragmanlarının başka yönlendiricilere (Y ve Z'ye diyelim) yönlendirildiğini varsayalım. Analize devam etmek için Şekil 2.30'de "zaman oku"nun gösterdiği sırayla (en önce n zamanı, en son n + m zamanı) Y ve Z yönlendiricilerinden fragmanların birleştirilecekleri sisteme vardıklarını söyleyelim. Böylece fragmanlar aşağıdaki sırayla gelirler:

1. : offset değeri 96 olan fragman
2. : offset değeri 128 olan fragman
3. : offset değeri 0 olan fragman
4. : offset değeri 160 olan fragman
5. : offset değeri 64 olan fragman
6. : offset değeri 32 olan fragman

Alıcı sistem için fragmanları yerleştirmek oldukça kolay bir iştir. IP modülü, fragmanı tamponun hangi slotuna yerleştireceğine karar vermek için offset değerini 8 ile çarpar. Örneğin ilk gelen fragmanın tampondaki göreceli pozisyonu hafiza adresi 768 olarak hesap edilir (96×8=768).

Şekil 2.30'daki birleştiren konak, dördüncü fragman gelene kadar IP paketinin tam uzunluğunu bilmez. Dördüncü fragman M=0 biti (son fragman), offset değeri ve fragman uzunluğu bilgilerini içerir. Offset değeri 160 ve uzunluk 220 bayt olduğundan konak, artık orijinal paketin 1500 bayt [(160 offset değeri) × (8 bayt her değerde) + (220 bayt son fragman)] olduğunu bilir.

Fragmandaki uzunluk alanı orijinal paketin büyüklüğü yerine fragmanın büyüklüğünü gösterdiğinden, orijinal uzunluğu (ve son fragmanı) kararlaştırmanın tek yöntemi M=0 (M biti) işaretçisidir.

Eğer bazı fragmanlar varışa gelmezlerse veya TTL parametresini aşarlarsa, IP varışa gelen fragmanları yok eder. Ayrıca ilk fragmanın gelmesi ile alıcı bilgisayar bir birleştirme zamanlayıcısı çalıştırır. Ağ yönetici tarafından kurulan bu zamanlayıcı, tüm fragmanların belirli bir sürede gelmesini zorunlu kılar. Eğer zamanlayıcının verdiği süre dolana kadar tüm fragmanlar gelmezse, gelen fragmanlar yok edilir.

Eğer kullanıcı, fragmantasyon oluşmasını istemezse, fragman bayrağı 1 yapılır. Bu, datagramlar üzerinde fragmantasyon yapılamayacağını belirtir. Bu alan böyle set edilirse ve MTU geçeceği bir altağ kapasitesini aşarsa paketler yönlendirici tarafından yok edilir.

2.4.3.6. Internet control message protocol (ICMP)

TCP/IP protokol ortamında iki yada daha fazla sistem arasındaki veri transferi sırasında meydana gelebilecek hataları ve kontrol mesajlarını idare eder. TCP/IP ağ problemlerinin çözümünü tespit etmekte önemli bir protokoldür.

ICMP mesajları şu amaçlarla kullanılır:

- Bir yönlendirici, IP paketinin TTL süresi dolduğu zaman yok eder. Paketin yok edildiği, paketi gönderene bir ICMP paketiyle bildirilir.
- Yönlendirici kendisine gönderilen paket için yeterli tampon alana sahip değilse bu paket yönlendirici tarafından yok edilir. ICMP paketiyle gönderen sistem bu durumdan haberdar edelir.
- Yönlendirici DF bayrak biti "1" olan bir paketi parçaladığında ICMP paketi gönderen sistemi bilgilendirir.
- Yönlendirici veya bir sistem paketin IP başlığında bir dizilim hatası bulduğunda, hatayı bulan sistem tarafından paketi gönderen sisteme ICMP paketi sayesinde bilgi verilerek paket yok edilir.
- Yönlendirici üzerinde geçerli varsayılan yönlendirici tanımı yoksa ve yönlendirici kendisine gelen paketi göndereceği yol bilgisini yönlendirme tablosunda bulamıyorsa, bu yönlendirici tarafından ICMP paketi aracılığıyla paketi gönderen sistem bilgilendirilir.
- Yönlendirici kaynak konağa daha kısa yol olan başka bir yönlendiricinin kullanılmasını önereceğinde bunu ICMP paketleri aracılığıyla yapar.

ICMP paketleri, TCP/IP ortamda bir geri besleme (feedback) sağlar. Bu sayede yönlendirme katmanı daha güvenilir kılınır. Bu yolla ciddi problemler, iletişim birimlerine bildirilerek bir hata tespit mekanizması oluşturulmuştur. Ancak buradan

ICMP'nin IP'yi güvenilir bir protokol haline dönüştürme amacıyla geliştirildiği yargısı çıkarılmamalıdır.

ICMP mesajı, IP paketlerinin veri bölümünde taşınır. Bu yüzden ICMP paketlerinin dağıtım güvenilirliği, IP paketlerinin dağıtım güvenliliği ile sınırlı kalmaktadır. Buradan ICMP paketlerinin güvenilir iletilemeyeceği ve hedefe varmasının garanti edilemeyeceği sonucu çıkarılabilir.

2.4.3.7. Internet group management protocol (IGMP)

Multicast gruplarını belirlemek için kullanılır.Bir ağda mesajlar üç şekilde gönderilir. Mesaj ya bütün makinalara (broadcast mesaj), ya bir gruba (multicast), yada doğrudan bir makinaya (directed) gönderilebilir.

TCP/IP kullanan bilgisayarlar kendi IP multicast grup üyelik bilgilerini IGMP destekli yönlendiricilere bildirmek için IGMP'yi kullanıyorlar. IGMP yönlendirici bileşeni LAN'de bulunan istemcilerin multicast grup üyeliklerini takip eder. IGMP vekil bileşeni ise IGMP grup üyelik paketlerini dış bağlantısından (ya da bağlantılarından) göndermeye yarıyor. IGMP'nin ana amacı LAN istemcilerinin IP multicast uygulamalarını çalıştırabilmeleridir. Multicast teknolojisi giderek yayılıyor çünkü bu teknoloji yüksek bantgenişliği gerektiren uygulamaların çok verimli bir şekilde çok sayıdaki istemciye dağıtılabilmesine olanak sağlıyor.

2.4.3.8. Adress resulation protokol (ARP)

Yerel ağda iletişim kotarılabilmesi için üst katman protokol adreslemelerinden fiziksel katman (MAC) adrese geçilmesi gerekir. TCP/IP protokol kümesinin kullanıldığı ağlarda, 32 bitlik bir sistem IP adresine karşı düşen fiziksel adres ARP protokolüyle elde edilir.

ARP ile adres çözümlemek istendiği zaman tüm ağa bir ARP istek mesajı yayınlanır ve bu IP adresini gören ya da bu IP adresine giden yol üzerinde bulunan yönlendirmeyi gerçekleştirecek sistem bu isteğe cevap verir ve kendi fiziksel adresini

gönderir. ARP isteğinde bulunan makine, bu adresi alarak fiziksel katmandan iletişim yoluna çıkaracağı çerçevelerin başlıklarına ekler.

Sonuçta bilgisayarlar, yönlendirme katmanında (OSI için ağ katmanı) IP adresleriyle haberleşiyorlar gibi görünseler de gerçekte bu adresler sadece rehberdir ve son hedef noktayı simgeler. Hedef uçlar arasındaki düğümler (köprüler, yönlendiriciler, switchler ..) aralarındaki haberleşmede IP adresi kılavuz olarak kullanılmakta ve gerçek haberleşme için MAC adreslerinden yararlanmaktadırlar. Gerçek haberleşme fiziksel adresler kullanılarak fiziksel katmanda (OSI için veri-bağı) gerçekleştirilir.

Bazı düğümler fiziksel adres öğrenme süreçlerini azaltmak için, diğer sistemlerin ARP sorgulamalarını sürekli dinleyerek kendi ARP tablolarını güncel tutabilirler. Böylece kendisi daha önce herhangi bir aktarım yapmasa bile, diğer sistemlerin IP fiziksel adres dönüşüm bilgisine sahip olurlar.

Eğer ARP isteğine cevap gelmez ise hedef sistem cevap vermiyor mesajı oluşturulur. Bu mesaj ARP'ce değil uygulama katmanı tarafından oluşturur.

ARP, IP'nin hizmetlerini kullanmaz o nedenle IP başlığı içermez. ARP paketi sadece yerel ağ üzerinde hazırlanıp gönderilir. Uzak ağlardaki (yönlendiricilere bağlı ağlar) konakların fiziksel adresi konak için bir anlam ifade etmez. Çünkü yönlendiriciler fiziksel adrese göre değil ağ katmanı mantıksal adresine göre (IP Adresi) yönlendirme yaparlar.

2.4.3.9. Reverse adress resulation protokol (RARP)

ARP prokolünün yaptığı işin tersini yapar. Elde bulunan fiziksel adresin IP karşılığını bulmaya çalışır. Özellikle disksiz konaklar tarafından kullanılır. Paket formatı ARP ile aynıdır. Bu protokolün kullanılabilmesi için ağda en az bir RARP server olmalıdır.

BÖLÜM 3. BİLGİSAYAR AĞLARINDA GÜVENLİK

Günümüz, bilginin çok önem kazandığı, çok hızlı üretildiği ve kısa zamanda güncelliğini yitirdiği bir çağ olmuştur. Bilginin bu kadar önem kazanması, bilgi teknolojilerininde çok hızlı gelişmeler göstermesine sebep olmuştur. Bilgisayarlar, bu gelişmelerin ışığında tek başına kullanılır olmaktan çıkmış, Internet ve kuruluşların Intranet'i gibi büyük ağ yapılarının birer parçası durumunu almıştır. Bilgi teknolojilerinin bu gelişimi yaşamımızın her alanında kendini göstermesine yol açmıştır; banka hesapları, sağlık kayıtları, alış-veriş vb. Hedeflenilen nokta ağ üzerindeki hizmetlerin her hangi bir zamanda herhangi bir yerdeki kullanıcılara güvenli ulaştırılabilmesidir.

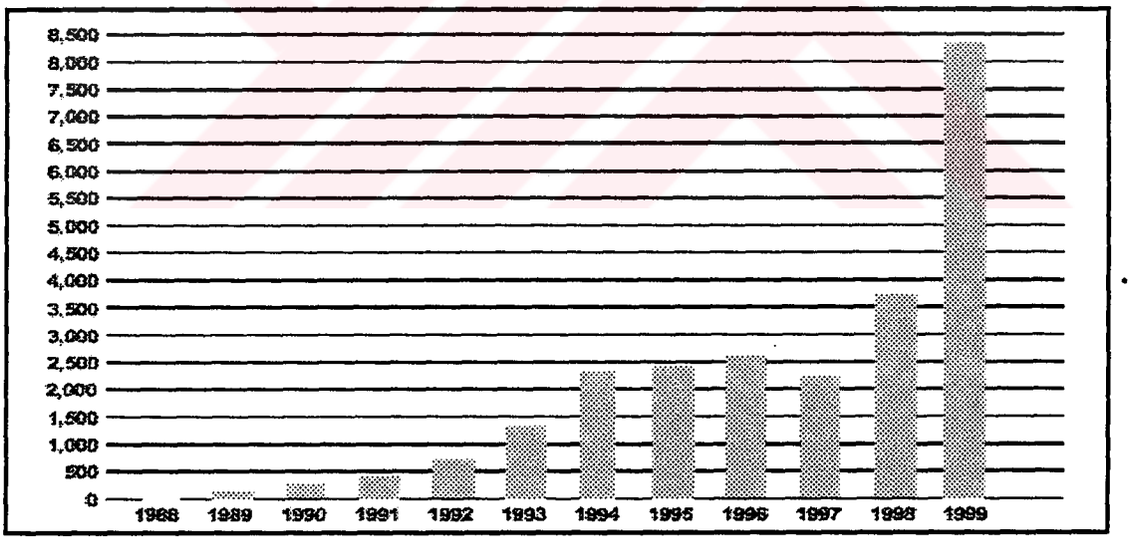
Bilgi teknolojilerinin, Internet gibi global ağlar üzerinde bu denli farklı alanlarda hizmet vermesi çok kısa bir zaman zarfında gerçekleşmiştir. Internet'in temeli, hatta Internet ile özdeşleşmiş TCP/IP protokol aileside, bu hizmetleri karşılayacak birçok değişik amaçlı protokole sahip olmuştur. Bu kadar hızlı geçiş bazı konularda hazırlıksız yakalanılmasına, gerekli alt yapı çalışmalarının tamamlanmadan çözümlerin üretilmesine neden olmuştur. Bu konuların başında hizmetlerin güvenlik destekleri gelmektedir. Internet'in ilk yıllarında güvenlik konusu önemsiz olarak görülmüş ve gereken ciddi çalışmalar yapılmamıştır. TCP/IP protokol ailesinin çoğu protokolündeki güvenlik açıkları bu yaklaşımı doğrulamaktadır. Bu ağa bağlı kurum sayısının her geçen gün daha da artması güvenliğin önemli bir problem olarak görülmesini sağlamıştır.

1988 yılında Morris Worm'unun Internet üzerinde çok sayıda bilgisayara bulaşması ve bu sistemleri çalışmaz duruma getirmesi, güvenlik konusuna dikkatleri çekmiş ve teknik önlemlerin alınması yolunda çalışmaların başlamasını sağlamıştır.[2]

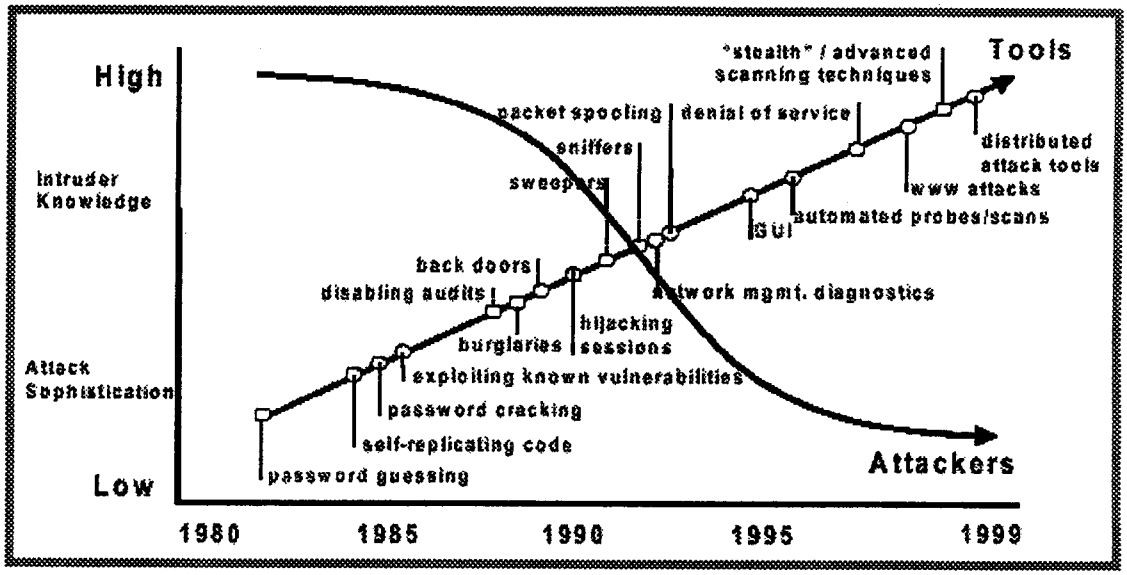
Ağ yapıları büyüdükçe, dağıtıklığı ve karmaşıklığı arttıkça bu sistemlerin doğruluğunu, eksiksizliğini denetlemek ve hizmetlerin sürekliliğini sağlamak zorlaşmaktadır. Sistemlerin işlerliğini koruyabilmek ve işlenen bilgilerin güvenliğini sağlamak artık çok önem kazanmış ve güvenlik politikaları oluşturulmasını zorunlu kılar duruma gelmiştir.[2]

Uluslar Arası Bilgisayar Acil Durum Müdehale Ekipleri Koordinasyon Merkezi (Computer Emergency Response Teams Coordination Center-CERT/CC) yaptığı istatistiklere göre 1997 ve 2000 yılları arasında rapor edilen bilişim suçlarının yıllara göre sayısı geometrik olarak artmaktadır.[3]

Güvenlik problemleri sadece iletişim hizmet protokollerinden kaynaklanmaz. Fiziksel olarak kaynakların korunması, kullanıcı cahilliklerinin giderilmesi, kullanıcı isim ve şifre bilgilerinin saklanmasıda güvenliğin sağlanması için çok önemlidir.



Şekil 3.1 CERT/CC'ye rapor edilen bilişim suçlarının yıllara göre dağılımı



Şekil 3.2 CERT/CC'nin raporuna göre saldırı ve saldırgan kalitesi değişimi

3.1. Güvenlik Tedbirleri ile Neler Korunmaktadır

Bir kuruluş Internet'e bağlanmakla güvenliğini riske atmış olmaktadır. Kurumun sahip olduğu veriler, kaynaklar ve saygınlığı saldırılara karşı tehdit altında bulunmaktadır.

Veriler : Veriler ile ilgili gizlilik, bütünlük, kullanıma hazırlık güvenlikle ilgili üç temel esastır.

Gizliliğe, kuruma ait ve üçüncü şahıslar tarafından erişilmesi istenmeyen; kurumun finansal bilgileri, yeni ürün tasarımları, organizasyon bilgileri ve faliyet gösterdiği alana ilişkin raporlar v.b. veriler için gereksinim duyulur. Bu bilgisayarların Internet ortamından ayrı olması düşünülebilir. Böylece gizli bilgilerin ayrılması ve Internet ortamından sadece gizli olmayan verilerin erişime açık olması sağlanalabilir.

Veriler gizli olmasa dahi yetkisiz kişilerce değiştirilmesi veya yok edilmesi istenmez. Bu tip saldırılara maruz kalınması maddi kayıpların doğmasına sebep olacaktır.

Ayrıca verilere ihtiyaç halinde ulaşılabilir olması istenir. Yetki sınırlandırılması ileriye yönelik iyi bir planlama ile yapılmalıdır.

Kaynaklar : Internet'e bağlanılmakla sistem kaynaklarıda riske atılmış olacaktadır. Gerekli güvenlik sağlanılamazsa veri depolama alanları, işlemci gücü, bellek kullanımı yetkisiz kişilerin müdehalesine maruz kalacaktır.

Saygınlık : Kurumun gizli verilerinin ortaya çıkması veya veri kayıplarının olduğunun anlaşılması maddi zararlara neden olduğu gibi kurumsal kimliğine olan güveni zayıflatacaktır.

Kurumun sistemine olan bir sızma sistem kaynaklarının kötü amaçlar doğrultusunda kullanılmasına neden olabilir. Sızılan sistemlerin farklı kuruluşlara saldırı amaçlı kullanımı saldırıların sızılan sistem üzerinden geldiği görüntüsünü vereceğinden kuruluşun imajı üzerinde düzeltilmesi zor yalnış anlaşılmaları doğuracaktır.

Kurumun veya kişisel isim hakkı kullanılarak yanıltıcı bilgilerin yayılması düzeltilemeyecek kötü sonuçları ortaya çıkarabilir. Kuruma ait mail sisteminin bu amaçla kullanılması karşılaşılabilecek bir saldırı tipidir.

3.2. Saldırıların Meydana Geliş Şekilleri

Saldırganların sistemler üzerine gerçekleştireceği saldırılar sistemin güvenliği ve saldırganın amaçı doğrultusunda farklı gruplara ayrılabilir.

Davetsiz misafir türündeki saldırılarda istenmeyen kişilerin sisteme girmesi şeklinde gerçekleşir. Bu saldırılar genellikle sistem üzerinde yetkili bir kullanıcının, kullanıcı adı ve şifresi denenerek elde edilmesi sonucunda meydana gelmektedir. Böylece sistemde yetkili bir kullanıcı davranışını sergileyerek amaçlarına ulaşacaklardır.

Bir başka saldırı şekli olarak, servis kilitleme saldırıları izlenmektedir. Bu saldırı yöntemiyle sistemin yetkili kullanıcılar tarafından kullanılamaması sağlanmaktadır. Sistem üzerinde sürekli mesajlar ve istekler oluşturulması ile sistem kaynaklarının

zamanları boşa harcanır böylece sistem servis veremez duruma getirilir. Saldırıları engellemek adına alınmış bazı tedbirlerde saldırganlar tarafından kullanılmaktadır. Örneğin geçerli kullanıcı ve şifrenin, erişim hakkını engellemek için yapılan başarısız teşebbüsler kullanıcının girişinin kilitlenmesine neden olacaktır.

Bazı saldırı şekillerinde saldırganlar kuruluşun sistemine doğrudan girmeden istedikleri bilgileri elde edebilirler. Bu saldırılar genellikle bilgi vermeye yönelik hazırlanmış hizmetlerin kullanılması ile olur. Bazı hizmetler ise yerel alan ağlarında kullanılmaya yönelik tasarlanmıştır ve Internet üzerinden kullanılması güvenli olmamaktadır. [5]

3.3. Saldırgan Tipleri

Saldırganların amaçları farklı olsada sergiledikleri bazı genel tutumlar mevcuttur. Bir saldırgan asla yakalanmak istemeyecektir. Saldırganlar girmeyi başardığı bir sisteme sürekli erişebilmek için farklı ulaşım yolları oluşturmaya çalışırlar. Yakalansalar bile erişim yollarını gizli tutmaya çalışırlar. Bu genel özellikler dışında saldırganlar gruplanabilir:

Eğlence İçin Saldıranlar : Belirgin bir hedefi olmayan, girdikleri sistemlerin önemli bilgiler içerebileceğini düşünerek bunlara ulaşmaktan keyif alan, ama zarar vermeyen saldırgan tipleridir. Çok bilinen Internet sitelerine girmek onlar için başarıdır.

Zarar Vermek İçin Saldıranlar : Saldırdıkları sistemlere zarar veren saldırgan tipleridir. Girdikleri sistemleri tahrip edmeye ve yıkıcı zararlar vermeye çalışırlar.

Skor Tutucular : Ulaşabildikleri sistem sayısı ve çeşitliliğine göre puan topladıklarına inanan saldırgan tipleridir. Sistemlere zarar verme amacı içinde değildirler ama daha sonra erişebilmek için kendilerine yeni ulaşım yolları hazırlarlar. Çok bilinen veya iyi korunan sistemlere ulaşmak onlar için daha önemlidir.

Bilgi Hırsızlığı İçin Saldıranlar : Bu tip saldırganlar ulaşabildikleri sistemlerden paraya dönüştürülebilcek verileri alırlar. Bir çeşit casusluk olarakta nitelendirilebilecek saldırgan tipleridir. Ulaştıkları sistemlere zarar vermeden sadece istedikleri bilginin bir kopyasını alırlar. [4]

3.4. Güvenliğin Sağlanması İçin Yapılabilecekler

Güvenliğin sağlanmasında yapılabilcek en basit yöntem kullanılan ürünlerin sağladığı güvenlik tedbirlerine güvenmektir. Bu yaklaşımın tehlikeler düşünüldüğünde ve ürünlerin açıklarını kapatmak için çıkarılan yamalar görüldükçe pek uygun olmadığı anlaşılacaktır. Açıkların kapatılmasından önce uğranılan saldırlar sistem üzerinde çok ciddi zararların oluşmasına sebep olacaktır.

Diğer bir basit güvenlik tedbiri ise sistemin hiç kimse tarafından bilinmemesinden istifade etmektir. Ama bu yaklaşım nadiren uzun süre çalışır. Internet'te dahil olmak üzere bir ağda üzerinde hizmet sunabilmek ve alabilmek için sistem bilgilerinin bir merkezde kaydının bulunması gerekir. Saldırganlar bir hizmet sunulduğu veya alındığı taktirde yeni sistemlerin farkına varacaklar ve henüz güvenlik tedbirlerinin yetersiz olduğunu düşünerek bu sistemlere erişmeyi deneyeceklerdir. Dolayısıyla bu yaklaşım güvenli bir çalışma ortamı sağlamayacaktır.

Çok kullanılan bir güvenlik yaklaşımı ise konak bazında alınan güvenlik yaklaşımıdır. Bu yaklaşımda her bir konak makinanın güvenliği ayrı ayrı ele alınır. Konağın sunduğu hizmet veya alacağı hizmet doğrultusunda bilinen güvenlik problemlerini bertaraf etmek için gerekli tedbirler alınır. Konak güvenliği her makina ve her işletim sistemi için farklı tedbilerin alınmasını gerektireceği için çok sayıda makina bulunan ağlar için uygulanabilirliği düşüktür. Ama aynı ağ içerisinde daha yüksek güvenlik seviyesi gerektiren konaklar olduğunda uygulanması faydalı olacaktır.

Sistemlerin büyümesi ve sunulan hizmetlerin çoğalması durumunda, konak bazında güvenlik yaklaşımından ağ güvenliği yaklaşımına geçmek daha akılcı olacaktır. Ağ güvenliği yaklaşımı ile ağdaki değişik konaklar ve onların sunduğu hizmetler

üzerinde yoğunlaşılır. Bu yaklaşımda ağları korumak için ateş duvarları (firewall), özel kullanıcı belirleme mekanizmaları, şifreleme teknikleri sıralanabilir. Ayrıca ağ güvenliğinden yeterince emin olabilmek ve gelebilecek saldırıları tespit etmek için saldırı tespit sistemleride güvenlik konusunda yardımcı olacaktır. Bunun yanında zayıflık tarama sistemleri ile sistemimizde almış olduğumuz önlemlerin bilinen saldırı teknikleri karşısındaki tutumu incelenerek, açıklar konusunda daha doğru çözümler üretilmesi sağlanalabilir.

3.5. Ateş Duvarı (Firewall)

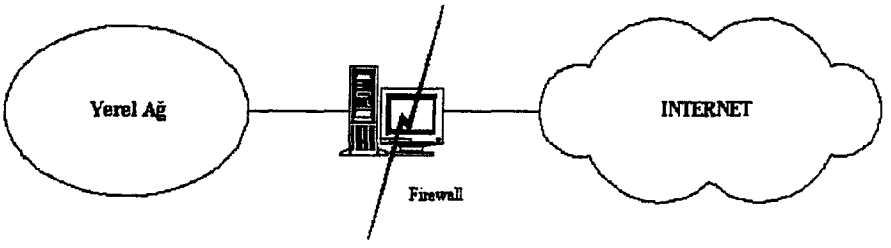
Firewall, bir sistemin özel bölümlerini halka açık bölümlerinden ayıran, hizmetlerden faydalananların kendilerine tanınan haklardan daha fazlasını almalarını engelleyen uygulamalardır. Harici ağlardan erişilebilir yerel ağ "risk bölgesi" olarakta düşünülebilir. Bir firewall olmadan yerel ağımızın tümü risk bölgesi olacaktır. Bir firewall ile harici ağdan erişilebilen daha küçük bir alan tanımlayarak risk bölgemizi küçültürüz. Bu küçük risk bölgesi için saldırı ve sızma kontrolleri gerçekleştiririz.

Firewall'ların çeşitli bileşenler ve ayarlamalar kullanan birçok çeşiti vardır. Sadece eleyen bir yönlendirici kullanarak, güvenlik amaçlı olarak endişelenmemiz gereken alanı küçülten basit bir koruma duvarı kurmuş oluruz. Güvenliğin gerekliliği ön plana çıktıkça sadece firewall vazifesinde donanımlar üretilmeye başlanmıştır. Bu donanım üzerinde koşacak yazılımlar da firewall'ların birer parçasıdır.

Firewall'lar, harici ağdan yerel ağa ve yerel ağdan harici ağa giden tüm trafiği üzerinden geçirmek zorundadır. Ancak böylece tüm trafik kontrol altında tutulabilir. Bu trafikten ağ için oluşturulmuş güvenlik politikasının izin verdiği ölçüler içinde olanların geçişine izin verirler. Sonuçta firewall'lar harici ağ ile yerel ağ arasında bölümleme yapar, trafik kısıtlaması getirir ve trafik üzerinde analizler yapma imkanı sağlar.

Firewall'lar ile ağ trafiğinin kontrolünde iki temel yaklaşım vardır;

- Harici ağ bağlantısında tam olarak izin verilmişlerin dışındaki hizmetleri engelleyen bir güvenlik duvarı planlamak.



Şekil 3.3 Firewall ile yerel ağ için güvenlik oluşturulması

- İkinci yaklaşım bunun tam tersidir. Tam olarak kısıtlanmış olmayanlara izin veren bir güvenlik duvarı planlamak.

Buradaki fark, ilk durumda firewall herşeyi engellemek üzere tasarlanmıştır ve hizmetlere dikkatli bir risk değerlendirmesinden sonra izin verilir. İkinci durumda, sistem yöneticisi güvenlikte zayıf noktaları belirlemeli ve açık bırakılmaları çok riskli olacak olan hizmetleri kapatmalıdır. Kullanıcılar genellikle ilk yaklaşımı daraltıcı görüyorlar ve güvenlik duvarına üretkenliği engelleyici gözüyle bakıyorlar. İkinci yaklaşım, kullanıcılara harici ağ kaynaklarını kullanmaları için daha çok serbestlik sağlarken, güvenlik delikleri oluşturulması riskinide arttırmaktadır.

Firewall'ların yapabilecekleri :

- Yerel ağ ile harici ağ bağlantısı tek bir noktadan geçmek zorunda bırakılacak ve bu noktada güvenlik politikası uygulanması mümkün olacaktır.
- Güvenlik açıklarına sahip servisler, firewall üzerinde tanımlanacak güvenlik politikası ile ya güvenilir kılınacak ya da kullanılamaz duruma getirilecektir.
- Firewall'lar sayesinde yerel ağdan harici ağa ve harici ağdan yerel ağa kurulabilecek servis ve kullanıcı bağlantıları kısıtlanabilecek, kontrol mekanizması getirilecektir. İstenildiği taktirde bağlantı kayıtları da tutulabilecektir.
- Ayrıca firewall'lar yerel ağdaki bölümler arasında da kullanılarak güvenliği daha yüksek bölümler oluşturulabilecektir.

Firewall'ların yapamayacakları:

- Konak güvenliği ve kullanıcı bilgisizliğinden kaynaklanan güvenlik sorunlarına karşı çare olamazlar.

- Yerel ağ içerisindeki kötü niyetli kullanıcıların verecekleri zararlara karşı hiçbir şey yapamazlar. Yalnız bu kişilerin harici ağ üzerinden gerçekleştirebilecekleri saldırılara karşı güvenlik teşkil ederler.
- Yerel ağın tek çıkış noktaları olması durumunda etkilidirler. Farklı harici ağ bağlantıları mevcutsa ve bunların herbiri için firewall kullanılmamışsa güvenlik yine yetersiz kalacaktır.
- Mevcut güvenlik açıkları ve saldırı imzalarına göre geliştirilirler. Yeni saldırı teknikleri karşısında yetersiz kalabilirler.
- Firewall'lar, veri paketlerinin sahip olduğu başlık bilgilerine dayalı trafik denetimi gerçekleştirir. Bu nedenle paketin veri alanındaki, uygulama katmanı düzeyindeki saldırılara (virus gibi) karşı genelde yetersiz kalırlar.

3.5.1. Firewall mimarileri

Temel olarak 3 çeşit firewall mimarisinden söz edilebilir. Bunlar :

- Çift Ağ Arayüzlü Konak Mimarisi
- Denetlenen Konak Mimarisi
- Denetlenen Alt Ağ Mimarisi

Bu mimariler kendi aralarında değişik varyasyonlar ile birleştirilerek daha farklı mimarilerde elde edildiği görülmektedir. Bu mimarilere geçmeden birkaç kavrama açıklık getireceğiz:

Konak (Host) : Ağ üzerinde mevcut olan bir bilgisayar.

Korumalı Konak (Bastion Host) : Güvenlik politikalarının belirlediği sınırlar ölçüsünde saldırılara karşı güvenliği arttırılmış bilgisayardır. Yerel ağın harici ağlar tarafından bilinen, gelen isteklerin karşılandığı ve bu nedenle saldırıya en çok maruz kalan bilgisayarlarıdır. Bu nedenle bu bilgisayarlar üzerinde güvenlik tedbirleri dikkatlice alınmalıdır. Korumalı konak tasarım ve kurulmasında iki temel ilke vardır; basitlik ve hazırlık.

Korumalı konak ne kadar basit tutulursa güvenliğini sağlama o kadar kolay olacaktır. Konağın vereceği hizmetlere ait yazılımlar ya da konfigürasyon hatalı olabilir. Bu nedenle konağın vereceği hizmetler ne kadar az olursa güvensiz olma ihtimalide o kadar düşük olacaktır.Korumalı konak üzerindeki güvenlik önlemleri ne kadar iyi olursa olsun aşanlar olabilecektir. Bu tip kötü durumlara karşı yerel ağ için sağladığı hizmetler gözden geçirilmeli ve neler olabileceği belirlenmelidir.

Korumalı konaklar, güvenli olmayan hizmetlerin kullanılmasını sağlar. Çift arayüzlü bir korumalı konak ile bağlantıları arasında trafik akışı geçirilmeden vekil sunucu davranışıyla bir firewall vazifesi üstlenirler.

Birçok firewall mimarisinde ana korumalı konak yanında dahili korumalı konaklarda mevcuttur. Ana korumalı konaklar dahili korumalı konaklar ile etkileşim içindedir. Örneğin elektronik postalar dahili sunucuya aktarılabilir, dahili bir DNS ile koordineli çalışabilirler. Bu makinalar etkin olarak ikincil korumalı konaktır ve diğer makinalara göre daha iyi korunmalıdır.

Korumalı konakların özel ağ trafiğini taşıyan ağ kesimlerine konması sakıncalıdır. TELNET, FTP ya da rlogin oturumlarında ağ üzerinden akan paket trafikleri dinlenerek kullanıcı ismi, şifre gibi bilgiler elde edilebilir. Bu nedenle korumalı konakların yerel ağdan soyutlanması uygundur.

Korumalı konaklar harici ağlara erişim için gereken hizmetleri sağlar. Bu hizmetler dört sınıfta toplanabilir:

- Güvenli hizmetler : Bu kategoriye giren hizmetler paket filtreleme ile sunulabilir.
- Normalde güvensiz fakat güvenilebilir yapılabilen hizmetler : Bu tip hizmetler korumalı konak üzerinden verilebilir.
- Normalde güvensiz ve güvenilir yapılamıyan hizmetler : Bu hizmetler ya kullanılmaz ya da ihtiyaç doğrultusunda vekil sunucu üzerinden sağlanır.
- Hiç kullanılmayan ya da bağlanılan harici ağda kullanılmayan hizmetler deve dışı bırakılır.

Çift Ağ Ara Yüzlü Konak : Üzerinde iki tane ağ bağlantı arayüzüne sahip konaktır.

Vekil Sunucu (Proxy Server) : Ağdaki diğer konaklar üzerindeki istemci yazılımlara ait istekleri gerçekleştirir. İstemciler, isteklerini vekil sunucuya bildirir, o da istekleri gerçek kaynağına istemciler adına iletir ve cevabı istemcilere aktarır.

Paket Filtreleme : Paket trafiğinin belirtilen kıstaslara uygun olmayanlarının geçişini engellenmesidir.

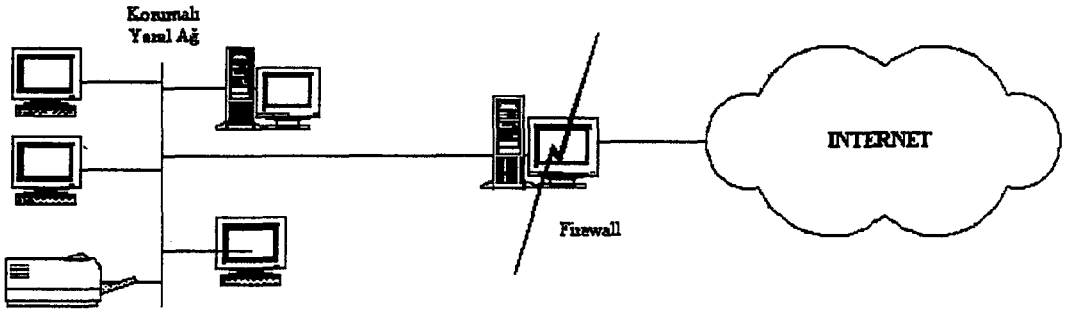
3.5.1.1. Çift ağ arayüzlü konak mimarisi

İki ağ arayüzüne sahip bir konak vardır ve bu konak etrafında inşa edilir. Bu konak iç ağ ile dış ağ arasında yönlendirici gibi işlev görür ama yönlendirme özelliği kullanılmaz, vekil sunucu görevini yerine getirir (Şekil 3.4). Yani iç ağ ile dış ağ doğrudan iletişim içinde olmaz, bu iletişim çift ara yüzlü konak tarafından kotarılır. Ağlar arasındaki IP trafiği tamamen bloke edilmiş durumdadır ve yüksek seviyede denetim uygulanmaktadır.

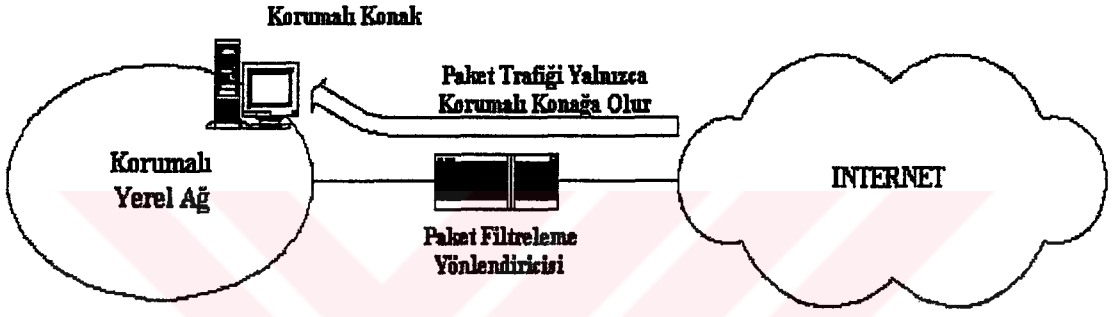
3.5.1.2. Denetlenen konak mimarisi

Bu mimaride harici ağ hizmetleri yerel ağ dahilinde bulunan bir konak üzerinden sağlanır (Şekil 3.5). Bu mimaride güvenlik kontrolü asıl olarak paket filtreleme ile gerçekleştirilmektedir. Korumalı konak, yerel ağda dahilinde bulunur. Paket filtreleme, dış yönlendirici üzerinde ve güvenlik politikasına uyan harici ağ hizmetlerinin sadece korumalı konağa erişebileceği şekilde kotarılır. Dışarıdan gelen servis istekleri sadece korumalı konağa yönlendirileceği için bu konağın üst düzeyde güvenliği sağlanmalıdır.

Paket filtreleme yönlendiricisi üzerindeki kısıtlama iki şekilde olabilir; yerel ağdaki korumalı konak harici konakların belirli servisler için dış ağa bağlanabilmelerine izin verir, ya da yerel ağdaki konakların dış ağa yapacakları tüm bağlantılar yasaklanır ve tüm bağlantıların korumalı konağın sunacağı vekil sunucu hizmetiyle yapılması sağlanır.



Şekil 3.4 Çift ağ arayüzlü konak mimari yapısı



Şekil 3.5. Denetelenen konak mimari yapısı

Bazı durumlarda iki kısıtlama yönteminde aynı anda kullanılması gerekebilir. Bazı servisler için vekil sunucu hizmeti verilemeyebilirki; bu duruma neden olan servisler için paket filtreleme yönlendiricisi harici ağa ulaşıma izin verecek, diğer servisler korumalı konağın sağladığı vekil sunucu hizmetiyle alınacaktır.

Çift ağ arayüzlü konak mimarisi ile kıyaslanacak olursa; paket trafiğinin yerel ağa geçmesi nedeniyle daha riskli yorumu yapılabilir. Fakat paket filtreleme yönlendiricisi bu amaçla düzenlenmiştir ve korumalı konağa göre güvenliğinin sağlanması daha kolay olacaktır.

Bu mimaride bazı dezavantajlar söz konusudur. Saldırgan korumalı konağa ulaştığı durumda, ağın geri kalan konaklarına ulaşması için engel kalmayacak. Paket filtreleme yönlendiricisi her hangi bir sebeple aşılırsa yine ağın diğer konaklarına doğrudan erişim imkanı doğacaktır.

3.5.1.3. Denetlenen alt ağ mimarisi

Bu mimari, denetlenen konak mimarisine, yerel ağı harici ağdan ayıran bir çevre ağ eklenerek elde edilir (Şekil 3.6). Eklenen çevre ağ güvenlik kademesi oluşturacaktır.

Bu yapılanma ile hedef durumdaki korumalı konak, yerel ağdan ayrıldığı için her hangi bir güvenlik problemi karşısında yerel ağ saldırıyla yüz yüze bırakılmaz. Yerel ağ ile çevre ağ arasında paket filtreleme yeteneği bulunan yönlendirici yer almaktadır. Bu yönlendirici üzerinde yerel ağ kullanıcılarının hem korumalı konağa hem harici sunuculara erişim yetkileri, güvenlik politikaları göz önüne alınarak verilir. Bu yönlendirici, korumalı konağın aşılması durumunda yerel ağ için ikinci bir güvenlik mekanizması oluşturur. Yerel ağın dışında kurulan çevre ağın, dış dünya ile bağlantısı yine paket filtreleme yeteneğine sahip bir yönlendirici üzerinden kotarılır. Bu yönlendirici üzerinde hem yerel ağı hemde korumalı konağı korumaya yönelik bir güvenlik politikası izlenir. Korumalı konak aynı zamanda yerel ağın vekil sunuculuğunuda üstlenecek şekilde düzenlenebilir. Böylece yerel ağın yapacağı istekler korumalı konağa olmakta ve korumalı konakta onlar adına harici sunuculardan hizmet talep etmektedir.

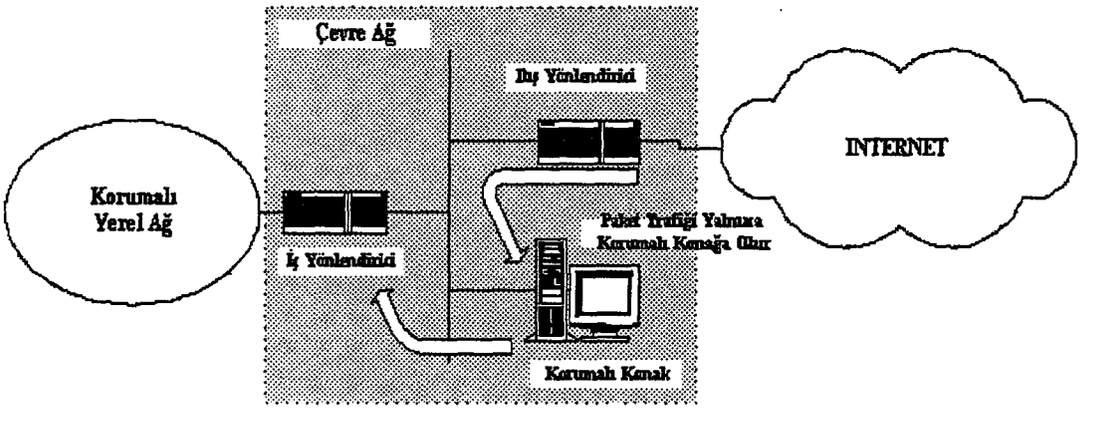
Çevre Ağ (Demilitirazed Zone, DMZ) : Bu mimaride, harici bir ağdan gelecek saldırı için yerel ağ ile harici ağ arasındaki çevre ağ güvenlik kademesi vazifesi görür. Saldırının hem dış paket filtreleme yönlendiricisini hemde iç paket filtreleme yönlendiricisini aşması gerekir.

Bazı bilgisayar ağlarında, bir bilgisayar üzerinden tüm ağ trafiğini dinlemek mümkündür (Ethernet, Token Ring, FDDI). Böyle bir ağ üzerindeki bir konağa erişen saldırgan ağdaki trafiği dinleyerek kullanıcı isimleri, parolaları ve gizliliğe sahip bilgilere ulaşması mümkün olacaktır. Saldırganın açık hedef durumundaki korumalı konağa ulaşması durumunda, saldırgan çevre ağ üzerindeki trafiği görebilecek yerel ağdaki konaklara erişemeyecektir. Çevre ağda korumalı konağın dış ağ ve yerel ağ ile olan trafiğine ilişkin paketler dinlenebilir.

Vekil sunucu kullanılmayacaksa, yerel ağ konakları için, iç ve dış yönlendirici paket filtreleme kuralları uygun hale getirilerek doğrudan harici ağ sunucularına erişimi sağlanmalıdır.

İç Yönlendirici : Yerel ağı hem çerve ağ, hem harici ağdan gelecek saldırılardan korur ve paket filtreleme mekanizmasının büyük bölümü bu yönlendirici üzerinde gerçeklenir. Yerel ağdaki konakların, istek yaptığı servislerden hangilerinin korumalı konak üzerindeki vekil sunucudan, hangilerinin doğrudan harici ağ sunucularına erişilerek alınacağı bu yönlendirici üzerindeki güvenlik politikası doğrultusunda, yapılacak paket filtreleme düzenlemeleriyle sağlanır. Korumalı konakların yerel ağa verecekleri servisler dışında, yerel ağa geçişide kısıtlanmalıdır. Çünkü, korumalı konağa yapılan saldırının başarılı olması durumunda saldırganın, yerel ağa girebilmesi için açık kapılar oluşur.

Dış Yönlendirici : Yerel ağı ve çevre ağı koruyacak paket filtreleme kurallarını işletir. Asıl amaç çevre ağı ve iç yönlendriciyi korumaktır. Paket fitreleme kuralları, çevre ağdaki korumalı konakların, harici bağlantılara izin verecek esnekliktedir. İç yönlendiricideki yerel ağı korumaya yönelik paket filtreleme kuralları benzerlik gösterir. Genelde harici ağdan gelen, fakat kendini yerel ağ adresiyle gösteren paketlerin yakalanmasını sağlayacak filtreleme kuralları içermelidir.



Şekil 3.6 Denetlenen alt ağ mimari yapısı

3.5.2. Firewall mimarilerinin birlikte kullanılması

Firewall mimarilerine temel teşkil edecek üç durum incelendi fakat günlük hayatta bu üç mimarinin özelliklerinin güvenlik politikası, ağ mimarisi, bütçe gibi faktörlerle bileştirilmesi gerekmektedir. Bu durumda yeni firewall mimarileri ortaya çıkacaktır. Oluşturulabilecek diğer firewall mimarileri:

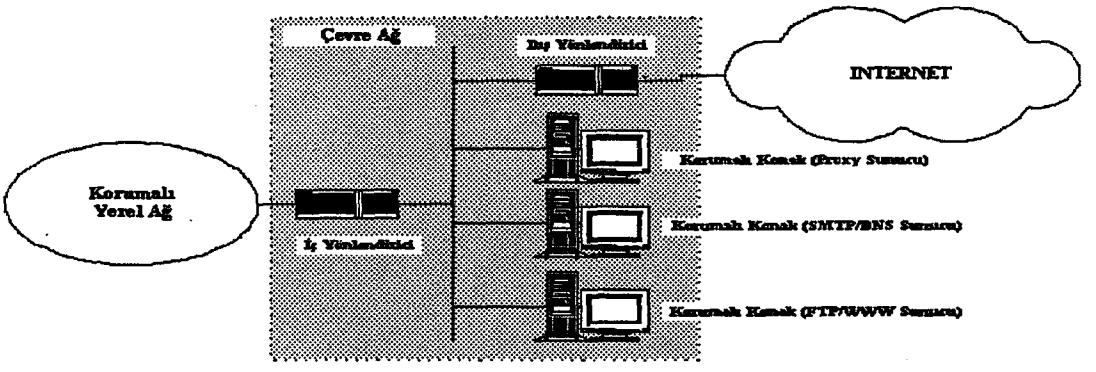
3.5.2.1 Birden fazla korumalı konak kullanımı

Denetlenen alt ağ mimarisinde, birden fazla korumalı konağın çevre ağda kullanılması ile gerçekleştirilir. Birden fazla korumalı konak kullanılması ile hizmetler sunucular üzerinde dağıtılır. Bu da performans artışı ve yedekleme imkanları sağlar (Şekil 3.7).

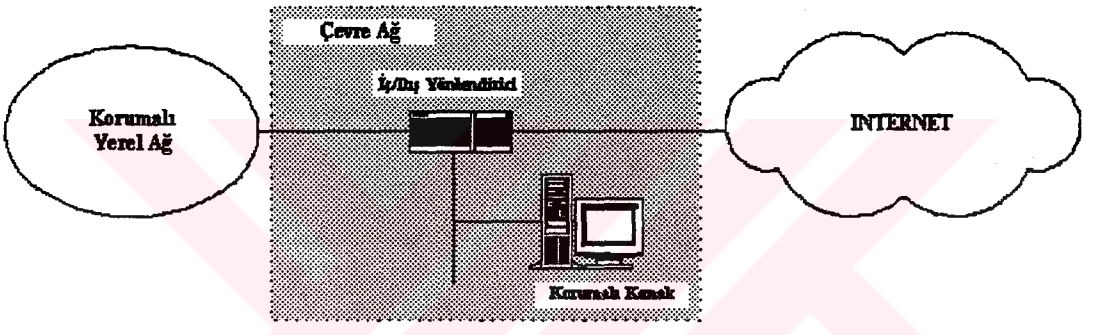
Korumalı konaklardan bir tanesi yerel ağ konakları için vekil sunucu, mail sunucu gibi hizmetleri gerçekleştirirken diğeri Internet üzerinden gelen isteklere (www, ftp v.s.) cevap verebilir. Ayrıca konaklar birbirlerinin yedeği olarak kullanılabilir. Bu durumda bir konak devre dışı kalsa bile hizmetin devamlılığı yedek konak ile gerçekleştirilebilecektir.

3.5.2.2. İç ve dış yönlendiricilerin birleştirilmesi

Yeterli özelliklere sahip bir yönlendiriciyle iç ve dış yönlendiriciler tek bir yönlendiricide toplanabilir. Yönlendiricinin harici ağa, yerel ağa ve çevre ağa bağlantısı bulunacaktır (Şekil 3.8). Bu yönlendiricininde güvenlik politikası dahilinde yerel ağ ile harici ağ (doğrudan harici sunuculardan alıncak hizmetler), yerel ağ ile çevre ağ (genelde vekil sunucu istekleri), çevre ağ ile harici ağ (harici ağdan gelecek bağlantı istekleri) arasında gerçekleşecek bağlantı istekleri üzerinde paket filtreleme gerçekleştirilebilmelidir.



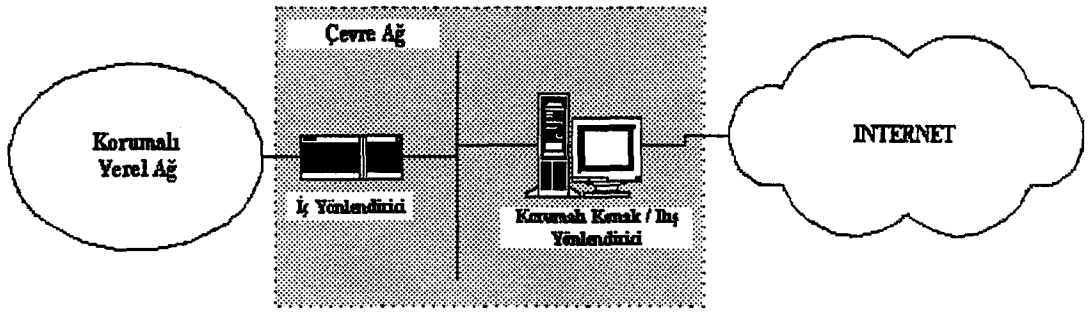
Şekil 3.7 Birden fazla korumalı konağın kullanıldığı mimari yapı



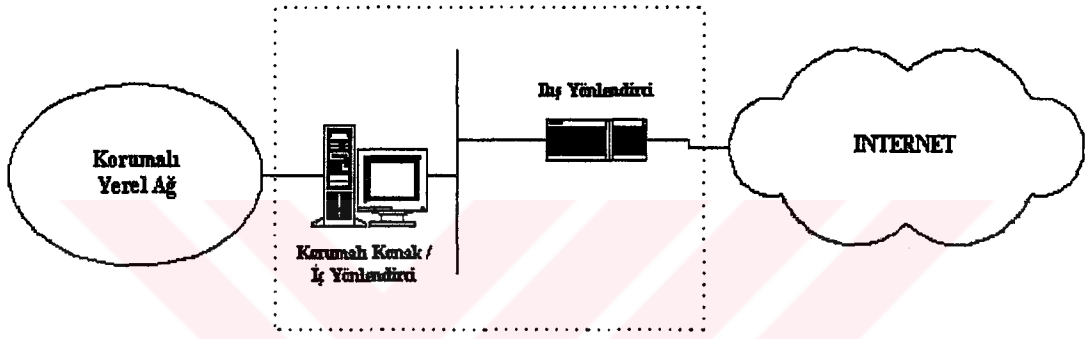
Şekil 3.8 İç ve dış yönlendiricilerin birleştirildiği mimari yapı

3.5.2.3. Korumalı konak ile dış yönlendiricinin birleştirilmesi

Çift ağ arayüzlü bir konağın hem korumalı konak hem dış yönlendirici olarak kullanıldığı mimaridir (Şekil 3.9). Bir konağın yönlendirici vazifesi görmesi bu iş için gerçeklenmiş bir donanıma göre performans ve esneklik kaybı doğuracaktır. Fakat harici ağ bağlantı hızının düşük olduğu durumlarda problem teşkil etmeyecektir. Korumalı konak üzerinde koşan işletim sistemi ve yazılım yetenekleri doğrultusunda filtreleme yapılabilecektir. Korumalı konağın doğrudan harici ağ bağlantısı olacağı için güvenliğine yeterince dikkat edilmesi gerekir.



Şekil 3.9 Korumalı konak ile dış yönlendiricinin birleştirildiği mimari yapı



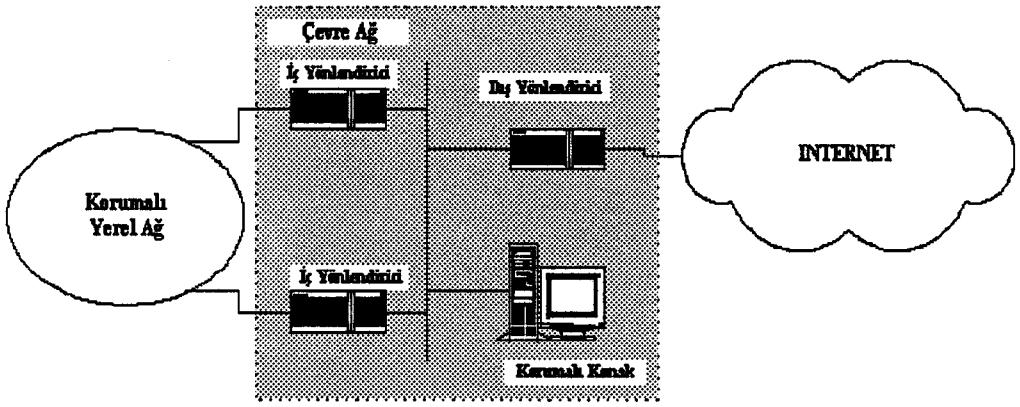
Şekil 3.10 Korumalı konak ile iç yönlendiricinin birleştirildiği mimari yapı

3.5.2.4. Korumalı konak ile iç yönlendiricinin birleştirilmesi

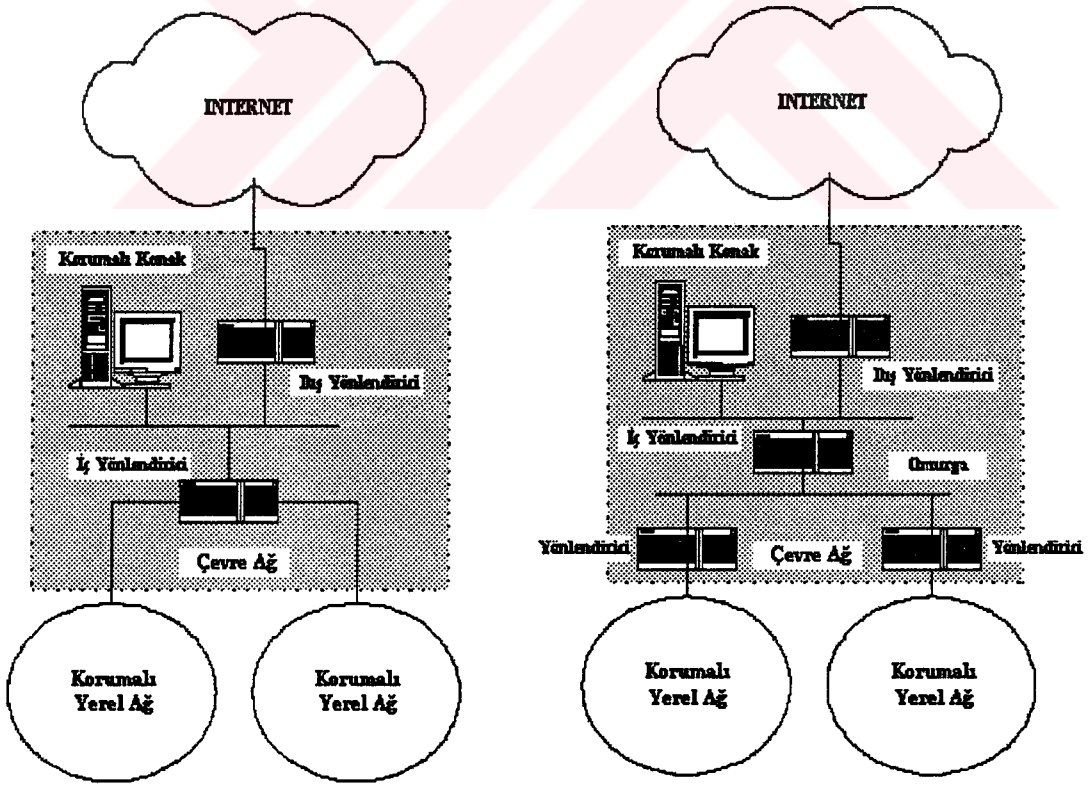
Bu mimari çevre ağ kullanımının kaldırılması anlamına gelirki, ortaya çıkan mimari denetlenen konak mimarisinin özelliklerini yansıtacaktır (Şekil 3.10). Çevre ağ kullanımı ile güvenlik kademesi oluşturulmakta ve saldırının korumalı konağa ulaşması durumunda dahi yerel ağ trafiğinin dinlenmesini engellenecektir. Çünkü çevre ağ ile yerel ağ arasında paket filtreleme yönlendiricisi mevcut olacaktır. Oysa bu mimari ile korumalı konağa erişildiği taktirde yerel ağ trafiğide dinlenebilecek bu da bir güvenlik zaafi oluşturacaktır.

Dış yönlendirici ile korumalı konak birleştirilmesi bu denli bir zaafa yol açmaz. Çünkü çevre ağ hala mevcut durumdadır ve yerel ağa erişim için paket filtreleme yeteneği olan iç yönlendiricininde aşılması gerekecektir. Yani korumalı konak dış

yönlendirici birleşiminin sigortası mevcut olacaktır. Oysa korumalı konak iç yönlendirici birleşiminin böyle bir sigortası mevcut olmayacaktır.



Şekil 3.11 Birden fazla iç yönlendiricinin birleştirildiği mimari yapı



Şekil 3.12 Tek iç yönlendirici tercih edilerek gerçekleştirilen yerel ağlar çevre ağ bağlantıları

3.5.2.5. Birden fazla iç yönlendirici kullanılması

Yerel ağ ile çevre ağ arasına yerleştirilecek birden fazla yönlendirici birçok probleme neden teşkil edecek ve güvenlik açısından iyi bir mimari olmayacaktır (Şekil 3.11).

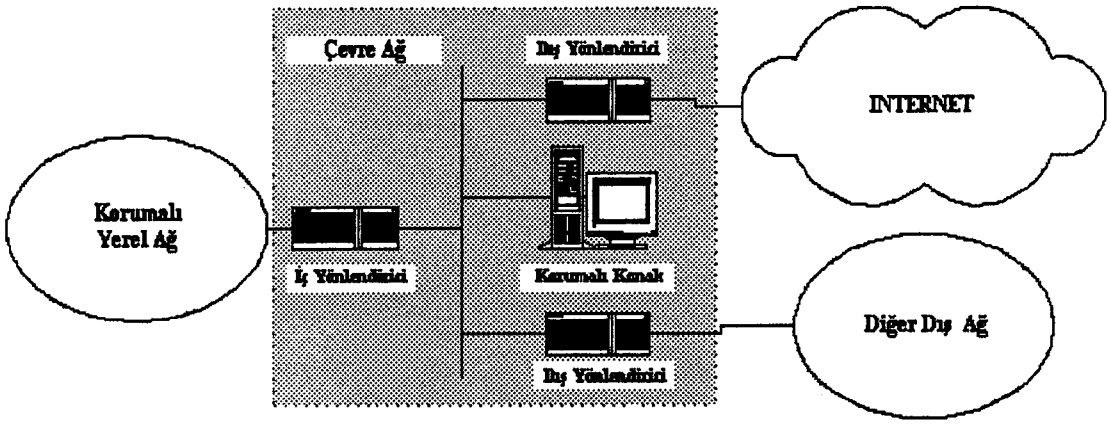
Yerel ağ içinde gerçekleşmesi gereken trafik akışı için, en iyi yolun çevre ağ üzerinden geçmesi kararını veren bir iç yönlendirci yerel ağ trafiğinin çevre ağa çıkmasına neden olabilir. Bunu engellemek, iç yönlendiriciler üzerinde yerel ağa yönelik paket filtreleme mekanizması ile mümkün olur. Ancak bu filtrelemede yaşanacak eksiklik yerel ağ için istenmeyen güvenlik açığı oluşturacaktır. Çevre ağı dinleyebilen bir kişinin aynı yolla yerel ağa geçişi mümkün olacaktır.

İç yönlendiricilerdeki paket filtreleme önemlidir ve karmaşıklık içerir. Birden fazla iç yönlendirici kullanılması daha karmaşık durumlara neden olabileceği için çok dikkat edilmelidir.

İç yönlendiriciler genelde performans problemleri oluşturmaz. Ama yerel ağ ile çevre ağ arası trafik çok yoğun ve bunların hepsi harici ağa yönelik değil ya da dış yönlendirci iç yönlendiriciye göre daha hızlıysa iç yönlendirci performans problemine yol açabilir. Bu durumda birden fazla iç yönlendirici kullanmak yerine daha güçlü bir iç yönlendirici şeçilmelidir.

Teknik, organizasyonel ve politik sebeplerle yerel ağ birden fazla parçadan oluşabilir. Bu durumda ise yine tek bir iç yönlendirici kullanılması için çözümler aranmalıdır. Bunlardan biri, yerel ağlara birer portun tahsil edilebileceği çok portlu bir yönlendirici şeçilmesidir. Bir diğer çözüm yöntemide bu yerel ağları bir omurga üzerinde toplamaktır. Bu yapıda omurganın iç yönlendirici üzerinden çevre ağa bağlanması gerçekleştirilir.

Değişik yerel ağlar üzerinde uygulanacak güvenlik politikaları farklılık arz ediyorsa bu ağların çevre ağa bağlantıları farklı iç yönlendiriciler üzerinden gerçekleştirilebilir. Bu durumda yerel ağlar arası tek bağlantı çevre ağ olmalı ve yerel ağlar arası trafik için güvensiz dış ağ trafiği davranışı uygulanmalıdır.



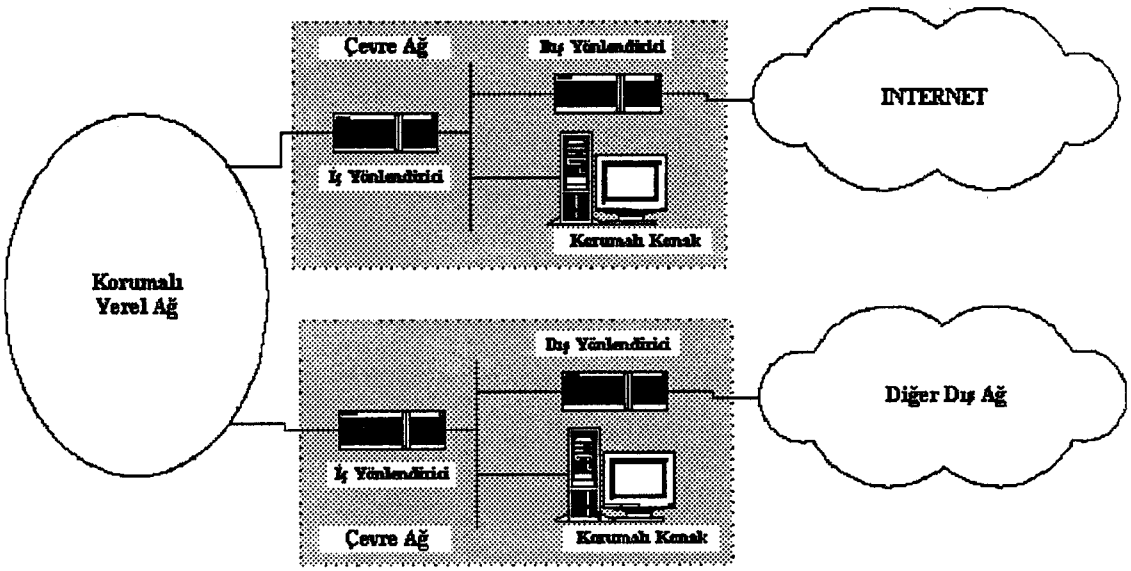
Şekil 3.13 Birden fazla dış yönlendiricinin kullanıldığı mimari yapı

3.5.2.6. Birden fazla dış yönlendirici kullanılması

Birden fazla dış yönlendricinin çevre ağa bağlandığı mimaridir (Şekil 3.13). Bu mimari birden fazla harici ağa bağlantı olması durumunda kullanılır.

Bağlantılar aynı harici ağa ise büyük güvenlik sorunları yaşanmaz. Bazı küçük paket filtreleme farklılıkları dikkatle düzenlenerek aşılabilir. Dış yönlendirici paket filtrelemesi çok önemli güvenlik sorununa neden olmadığı için bunlarda çok önemli olmayacaktır. Ayrıca aynı harici ağa bağlantılar ayrı dış yönlendiriciler yerine birden fazla harici ağ bağlanmasına sahip tek bir dış yönlendirici üzerinden yapılabilir.

Ancak dış yönlendiriciler farklı ağlara açılıyorsa işlemler daha karışık olacaktır. Bu bağlantılarda herhangi bir dış ağdan korumalı konağa bağlanan kişi diğer dış ağlardan gelen trafiği dinleyebilirki bu istenmeyen bir durumdur. Bunu önlemek için dış ağları önem derecesine göre farklı çevre ağlar kullanarak bu ağlara bağlamak daha güvenli bir çözüm olacaktır.

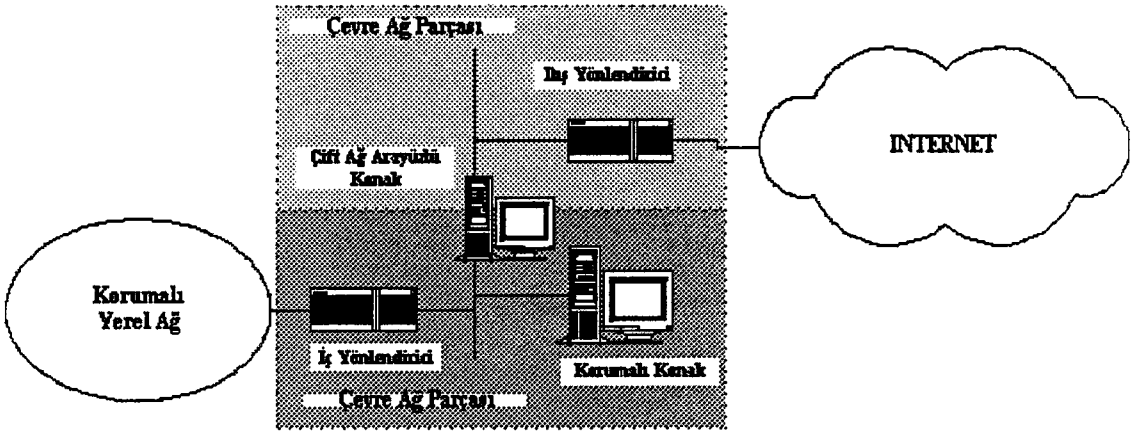


Şekil 3.14 Birden fazla çevre ağın kullanıldığı mimari

3.5.2.7. Birden fazla çevre ağ kullanılması

Birden fazla çevre ağ kullanımı bağlantıların bir birinden bağımsızlığını sağlama, yedekleme ve farklı güvenlik seviyeleri oluşturması açısından faydalı olacaktır (Şekil 3.14). Herbir çevre ağ, bir harici ağ bağlantısı içermesi durumunda herhangi bir çevre ağda yaşanacak problem diğer çevre ağın ve buna bağlı harici ağların bağlantısını etkilemeyecek, işlerlik devam edecektir.

Aynı harici ağ için, farklı çevre ağlar üzerinden bağlantılar oluşturularak yedeklemede gerçekleştirilebilir. Bazı durumlarda da farklı güvenlik seviyeleri oluşturulması açısından çevre ağ kullanılması gerekebilir. Böylece gizliliği yüksek seviyede olan bilgilerin erişiminde daha güvenli ortamlar oluşturacaktır. Örneğin, Internet erişimi bir çevre ağ üzerinden ve diğer çevre ağ üzerinden başka bir harici ağ bağlantısı gerçekleşsin. Bu durumda geniş erişime açık Internet çevre ağı için daha sıkı bir güvenlik politikası izlenirken, diğer çevre ağ için güvenlik politikası, daha çok hizmetin kullanılması yönünde olabilecektir.



Şekil 3.15 Çift ağ arayüzlü konak ile denetlenen alt ağ mimarilerinin birlikte kullanılması

Bu tip bir mimari oluşturulması ve bakımı daha zorlu olacaktır. Çünkü daha fazla sayıda yönlendirici kullanılacak ve herbiri üzerinde güvenlik politikasının gerçekleştirilmesi gerekecektir.

3.5.2.8. Çift ağ arayüzlü konak ile denetlenen alt ağ kullanılması

Bu iki mimarinin birlikte kullanılması güvenliği önemli ölçüde arttıracaktır. Çevre ağ bölünerek bu bölünen noktaya çift ağ arayüzlü konak yerleştirilir. Çift ağ arayüzlü konak vekil sunucu işlevi gördüğü için paket filtrelemeye nazaran daha iyi bir güvenlik kademesi oluşturur (Şekil 3.15). Paket filtrelemeyi aşabilecek yanıltmalar çift ağ arayüzlü konak üzerinde tespit edilip bertaraf edilebilecektir. Sonuçta iyi konfigüre edilmiş bir çift ağ arayüzlü konak daha sağlam, kademeli bir güvenlik sağlayacaktır.

3.5.3. Dahili firewall'lar

Genellikle firewall'lar, yerel ağ ile harici ağlar (Internet ve diğer kuruluş ağları) arasında güvenlik sorunları meydana getirmemek için güvenlik politikası sınırlarını çimek için kullanılır. Fakat bazı durumlarda yerel ağın bölümleri arasın da güvenlik sorunları meydana gelebilir. Bu sorunların aşılması içinde yine firewall'lar kullanılabilir. Firewall ile ayrılabilecek ağ parçaları :

- Test ve laboratuar çalışmalarının yapıldığı ağ parçalarını, geri kalan ağ parçaları için güvenlik sorunlarına neden olmayacak şekilde ayırmak.
- Diğer ağ parçalarından daha az güvenli ağ parçalarını ayırmak. Daha az güvenli olmasına neden ise eğitim çalışmalarının yürütüldüğü, sunumların yapıldığı, misafirlere açık ağ parçaları olmasıdır.
- Diğer ağ parçalarından daha fazla güvenliğe sahip ağ parçaları oluştrumak gerekebilir. Bu ağ parçalarında gizli projeler yürütülüyor, parasal bilgiler tutuluyor olabilir. Bu nedenlerle bu ağ parçalarının daha az güvenli ağ parçalarından ayrılması firewall'lar ile yapılabilir.

3.5.4. Firewall uygulama çeşitleri

Firewall'lar üç çeşittir.

- Paket filtreleme yapan firewall'lar
- Vekil (Proxy) firewall'lar
- Uygulama düzeyli firewall'lar

Paket filtreleyen firewall'lar bilgisayara girecek olan paketleri kontrol ederek belirlenen kurallar çerçevesinde filtrelerinden geçirerek paketlerin akıbetine karar verir. Bu akıbet üç şekilde olur. Paket ya kabul edilir, ya kabul edilmediğine dair paketi gönderen sisteme bir cevap gider ya da hiçbir cevap verilmeden paket bloke edilir.

Vekil firewall'ların çalışma prensibi vekaleten iş yapmaktır. Yerel ağ dahilinde yer alan bir konak adına harici ağdaki herhangi bir sisteme hizmet isteği yapar ve ondan gelen cevapları da bizim makinamıza taşır.

Uygulama düzeyli firewall'lar; OSI başvuru modeline göre uygulama katmanı düzeyindeki, uygulama protokolleri üzerinde güvenlik denetimi sağlar.

3.5.4.1 Paket filtrelemeye yönelik firewall

Bilgisayar ağlarında veriler, küçük parçalara ayrılarak paketler şeklinde iletilir. Verilerin paketler şeklinde iletimi ağın birçok sistem tarafından paylaşılmasına müsade eder. Paketlerin iletimi esnasında hedeflerine ulaşabilmesi, paketlerde yer alan IP başlık bilgilerinin yönlendirici cihazlar tarafından okunup, doğru yönde aktarılması ile mümkün olur. Paket filtreleme ise bu yönlendiriciler üzerinde koşturulan filtereleme amaçlı yazılımlar ile gerçeklenir.

Paket filtrelemeye yönelik firewall'lar ağdaki trafik akışını paket bazında çok sıkı bir denetimde tutan mekanizmalardır. Bu firewall'lar kendi üzerinden geçen trafiği kontrol altında tutarak, bunlardan sadece kabul edilebilir olanların geçişine izin verirler. Kabul edilebilirlik sınırı ağdaki servisler ve güvenlik politikası doğrultusunda belirlenir. Mesela bir ağ üzerinden e-posta hizmeti sunuluyor ise SMTP paketlerinin geçişi firewall politikasında izin verilecek şekilde belirlenmelidir.

Paket filterelemeye dayalı firewall kullanımında dikkat edilmesi gereken önemli nokta, tüm ağ trafiğinin sadece bu mekanizma üzerinden geçtiğinden emin olmaktır. Başka çıkış noktaları olan bir ağda, bu tip bir mekanizmayı kurmak, güvenlik konusunda problemler doğuracaktır. Çünkü saldırılar diğer bağlantı noktaları üzerinden geçebileceği için paket filtreleme mekanizmaları işe yaramayacaktır. Paket filtelemeye dayalı firewall ile;

- Yerel ağdan dış ağlara giden paket trafiği sınırlandırılabilir.
- Dış ağlardan yerel ağa gelen paket trafiği sınırlandırılabilir.
- Ağ trafiği hakkında bilgi edimek mümkün olur.

Bir paket filtereleme yazılımı IP katmanı seviyesinde kontrol gerçekleştirir ve IP başlığında yer alan aşağıdaki bilgiler doğrultusunda davranış sergiler.

- Kaynak IP Adresi
- Hedef IP Adresi
- Protokol Tipi

- Kaynak Port
- Hedef Port
- Eğer ICMP mesajı ise mesaj tipi
- Paketin giriş yaptığı ağ arayüzü
- Paketin hangi ağ arayüzüne iletileceği

Son iki bilgi paket yönlendirme mekanizması tarafından sağlanırken diğer bilgiler IP paketi içinde saklıdır.

Paket filtreleme yazılımları paketin taşıdığı veri ile ilgilenilmez ve dolayısıyla kullanıcı bazında denetim gerçekleştirmezler.

Bir firewall mimarisinde paket filtreleme tüm ağ trafiğinin yöneldiği her aşamada yapılabilir. Sadece tek yönlendirici içeren mimarilerde paket filtrelemenin yapılacağı yer bu yönlendiricilerdir. Bunun yanısıra konak bazında da paket filtreleme yapılabilir.

Paket filtreleme mimarisinde statik ve dinamik olmak üzere iki temel teknik kullanılır. Statik paket filtreleme; gelen ve giden paketleri sadece başlık bilgisi doğrultusunda inceler ve bu değerler doğrultusunda erişim iznini sorgular. Örneğin bir http isteği eline geldiğinde erişmek isteği portun 80, protokolün TCP ve geldiği yerin 1.2.3.4 IP adresinin olduğunu görür ve çevre ağda mevcut korumalı konak sunucuya ulaşmasına izin verilmişse, bu paketin sunucuya gitmesine izin verir.

Bu tekniğin en büyük zayıflığı paketleri ilk gönderen sistemi yani oturumu ilk başlatan sistemi saptayamıyor olmasıdır. Kaynak portu taramaları ve bağlantıları bu teknik için risk teşkil edecektir.

Örneğin yerel ağımızdaki bir çalışanın FTP portundan iletişim kurabilmesi için izin verilmiş olsun. FTP oturumun işleyişi gereği önce çalışanın TCP 21 portunu hedef port olarak belirleyerek bir sunucuya dosya isteği göndermesi ile başlar, hedef sunucu, kaynak portu TCP 20 olan paketler ile çalışana dosya transferi yapar. Bu durumu değerlendiren bir saldırgan ağa kaynak portu TCP 20 olan bir paket

gönderdiğinde firewall sistemi bu paketi görecek ve yerel ağdan istek gelmeseydi bu paket gönderilmezdi mantığıyla paketin geçişine izin verecektir. Firewall'un, paketin hedef portuna bakmaması sebebiyle saldırgan kaynak portu TCP 20 olan paketlerle yerel ağdaki herhangi bir sistemin örneğin TCP 139 portuna ulaşabilecektir. Böylece firewall etkisiz kılınacaktır.

Dinamik paket filtreleme (stateful inspection) teknolojisi statik paket filtreleme teknolojisinin zaaflarını girecek şekilde geliştirilmiştir. Bu teknoloji, oturumun baştan sona takip edilmesini, kimin ne istediğini ve kimin ne gönderdiğini, bir tabloda tutacak ve karşılaştıracaktır. Sonuçta klasik paket filtrelemenin yanısıra oturumu takip etme özelliği de karar verme mekanizmasında kullanılacaktır.

Firewall'lar oturumu nasıl izleyecekler; temel olarak TCP oturumları bir başı, ortası ve sonu olan oturumlardır. Hiçbir oturum başından veya ortasından kurulamaz. Bu durumda firewall kuralları sadece SYN bayrağı set edilerek gönderilen paketlere (nereden gönderildiği önemli değil) uygular ve geriye kalan paketler oturumun tutulduğu tabloya bakılarak takip edilir. Böylece FIN veya SYN/ACK bayraklı paketlerin bir oturumun devamı olmadığında geçişi engellenebilir. Oturumun SYN bayraklı paketler ile başlayacağını düşünerek tasarlanan bu sistemin, kuralları bu paketlere uygulaması oldukça mantıklı ve güvenilirdir.

TCP için olan bu oturum izleme işlemi UDP ve ICMP paketlerinede uygulanabilmektedir.

3.5.4.2. Vekil (Proxy) firewall'lar

Vekil, yerel ağ ile dış dünya arasında yer alan ve başta bilginin güvenli bir şekilde temini ve paketlerin depolanmasını sağlayan uygulamadır.

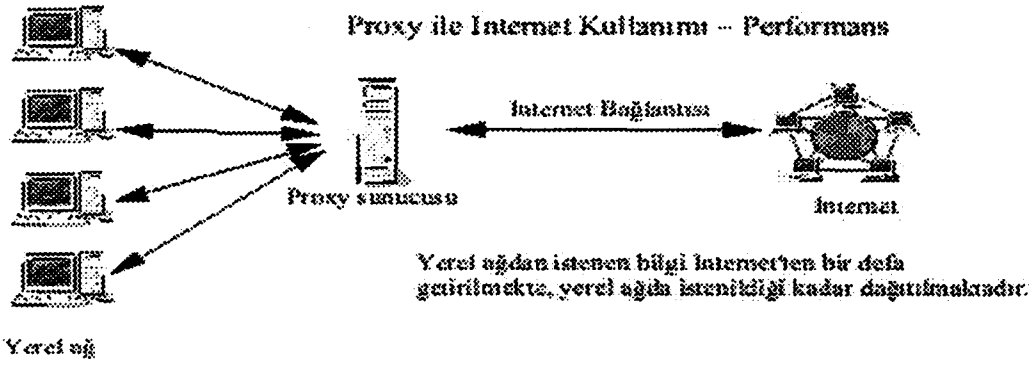
Vekil kavramı tüm konaklara erişim varmış gibi görünüyorken bir ya da birkaç tane konağa erişim sağlar. Özel bir protokol ya da protokol listesi için bir vekil sunucu çift arayüzlü konak veya korumalı konak üzerinde çalışır. İstemci programı Internet üzerindeki gerçek sunucu yerine vekil sunucu ile bağlantı kurar. Vekil sunucu isteğin

güvenlik politikası dahilinde izinli olup olmadığına baktıktan sonra eğer izin veriliyor ise gerçek sunucu ile bağlantı kurar ve haberleşme boyunca istemci ile sunucu arasında yer alır. Böylece iki sistem arası tamamen yalıtılır ve firewall paketlerin gerek içeriklerine, gerek hedef ve kaynak portlarına gerekse de gönderenin IP adresi doğrultusunda iletişime müdahale edebilir. Vekil, gerçeklemede özel bir donanım gerektirmez ancak birçok hizmet için özel yazılım gerektirir.

Vekil ön belleği yardımıyla yerel ağdaki istemci, dışarıdaki bir sunucuya doğrudan bağlanmadan yine yerel ağdaki sunucu üzerinden servis alır. Vekil sunucu bir istek aldığı anda eğer bu isteği kendi disk alanından karşılayabiliyorsa istemciye doğrudan gönderir aksi taktirde asıl sunucuya bağlanır ve bu sunucudan aldığı bilgileri istemciye yollar. Bu sırada yolladığı paketleri, ileride gelebilecek istekleri karşılayabilmek amacıyla depolar.

Böylelikle birkaç bilgisayarın aynı anda aynı bilgileri almak için mevcut band genişliğini kullanmaları önlenmiş olur. Ayrıca yerel ağdan daha hızlı bilgi transferi gerçekleştirilebilir.

Yeni geliştirilmiş ya da yaygın kullanılmayan hizmetler için vekil sunucu yazılımı bulmak zordur. Diğer bir dezavantajı; istemci ve sunucu arasında vekilin saydam olması ve isteklerin filtrelenmesi için protokol bazında farklı vekil sunucular gerektirmesidir.



Şekil 3.16 Proxy sunucu üzerinden Internet erişimi

Vekilin çalışma şekli TCP/IP hizmetleri arasında farklı şekilde gerçeklenir . Her hizmet vekil desteği ile dizayn edilmediği için sunucu ve istemci bazında farklı işlemler gerektirir. İstemciler vekil sunucuyla temasa geçtiğini ve gerçek sunucuyu bildirmek zorunda olduğunu bilmelidir. Bu desteği veremeyen istemciler için yama yazılımlar üretilmiştir. İstemcinin vekil sunucuya bağlantı desteği yoksa kullanıcı başka bir yöntemi takip ederek vekil sunucuya, gerçek sunucuyu bildirmek durumundadır.

3.5.4.3. Uygulama düzeyli firewall'lar

Uygulama düzeyli firewall'lar en sıkı koruma yapan mimaridir. OSI başvuru modeline göre uygulama katmanı düzeyinde çalışır yani paketlerin başlık bilgisi yanında veri alanı bilgiside kontrolden geçirilir; dolayısıyla tam denetim yapma imkanı kazanılır. Genel olarak güçlü bir iş istasyonu üzerine yüklenen yazılımla gerçekleştirilir. Bu tür firewall'lar, vekil firewall'lara benzer ama oturum kurulduktan sonra bile paketlerin sınanması yapılır. Bu da beklenmedik saldırılara karşı korumayı kuvvetlendirir. Genellikle uygulamaları vekil firewall'lar ile gerçekleştirilir.

Paketlerin içeriğini kontrol edebilme bu firewall'ların en büyük artılarındandır, böylece istenmeyen komutlar (HTTP paketlerinde POST komutunun kullanılmaması gibi) veya içerik (Java, ActiveX gibi) filtrelenebilir. Özellikle FTP protokolü kesinlikle bu firewall'lar üzerinden hizmet vermelidir, aksi taktirde FTP protokolünün pasif FTP seçeneği ile saldırgan FTP sunucusundan içerideki sistemlere ulaşabilir. FTP oturumu uygulama düzeyli firewall veya vekil firewall üzerinden geçirilirse bu tür isteklerin firewall tarafından filtrelenmesi sağlanılabilir.

Bu yöntem, ağ yöneticisine, paket filtrelemeli ve vekil firewall'a göre daha güvenli, daha sıkı bir koruma imkanı sağlar. İstenen programların çalışmasına izin verilirken, yasak olanlar engellenir. Ancak bu tür koruma çok çeşitli uygulamaların bulunmasından dolayı her uygulama için hayata geçirilemez, geçirilebildiği durumlarda ise sisteme çok yük getirir. Sadece belirli uygulamalar için böyle bir

koruma sunulur. Bu tür güvenlik duvarı kullanılması durumunda ağ yöneticisine büyük bir sorumluluk düşer; gerekli olan konfigürasyonu kendisi yapmalıdır.

3.5.5. Firewall yazılımlarının çalışma mantığı

Firewall yazılımlarını çalıştıracağımız işletim sistemlerinin, bu desteği sunabilecek çekirdek yeteneklerine sahip olması gerekir. Bu yetenekler ile güvenlik konusunda değişik çalışmalar yapılabilir. Bu çalışmalar; IP paketleri ile ilgili sayma, kaydetme işlemleri, paket filtreleme desteği verilmesi, vekil firewall desteği verilmesi, iletilen paketlerin maskelenmesi gibi işlemeler sayılabilir. Çekirdeğin sunduğu bu destekler, yine çekirdekte yer alan dört farklı liste ile sağlanmaktadır. Bu listelerde çekirdeğin yeteneklerine ait değişik kural tanımlamaları yer alır. Her bir kural, paket başlıklarındaki bilgiler üzerinde değerlendirmeler yapar. Kural tanımı ile paket değerleri uyuştuğu taktirde kurala ilişkin belirlenmiş işlemler yerine getirilir. Kural listeleri:

Kayıt Tutma Listesi : Ağ arayüzleri üzerinden aktarılan paketlerin sayılması ve bayt değerlerinin tutulmasıyla ilişkilidir. Bir arayüze gelen ya da giden paket, kayıt tutma listesinde yer alan tüm kurallar ile karşılaştırılacak, uyuşan bir kural bulunduğunda bu kurala ilişkin paket ve bayt sayıcı değerleri uygun şekilde arttırılacaktır. Bu sayede ağ trafiği hakkında daha bilgi edinilmesi mümkün olacaktır.

Giriş Kural Listesi : Ağ arayüzleri üzerinden kabul edilebilecek ya da edilemeyecek paketlerin tanımlanması sağlanır. Ağ arayüzünden alınan bir paket bu listedeki kurallar ile karşılaştırılacak ilk uyuşma sağlandığında bu kurala ilişkin işlem yerine getirilecektir. Eğer listedeki kurallar ile uyuşma sağlanamaz ise kural listesi için belirtilen geçerli politika uygulanır.

Çıkış Kural Listesi : Ağ arayüzünden çıkan paketler üzerinde uygulanacak kural tanımlarına sahiptir. Ağ arayüzünden, ağa çıkmaya hazır hale gelen paket, listede yer alan kurallarla karşılaştırılarak uyuşan bir kural bulunmaya çalışılır. Uyuşan bir kural bulunursa gerektirdiği işlemler yerine getirilir. Eğer uyuşan bir kural bulunamazsa kural listesi için belirtilen geçerli politika uygulanır.

İletim Kural Listesi : Bu liste bir ağ arayüzünden bilgisayarımıza gelen ve başka bir hedef bilgisayara iletilecek olan paketler üzerinde uygulanan kural tanımlarını içerir. İletim için alınan paket, iletim kural listesiyle karşılaştırılarak uyuşan bir kural bulunmaya çalışılacaktır. Bulunacak ilk uyuşan kural için geçerli işlem yürütülecektir. Eğer uyuşan kural tanımı yoksa liste için geçerli politika uygulanır.

Kural listelerinde paketin akıbetiyle ilgili karar verilirken paket sayıcı ve bayt sayıcı değerleride uygun değerler ile değiştirilir.

Kural listeleriyle, paket başlık bilgileri kıyaslanırken uyuşan kural tanımı bulunması sonucunda uygulanan işleme kurala ait politika denir. Kural listesindeki hiçbir kural ile uyuşma sağlanamaz ise paketin akibeti için yürütülecek işleme ise kural listesi için geçerli politika denilir. Paket üzerinde uygulanabilecek üç farklı politika söz konusudur; paketin kabul edilip geçişine izin verilmesi (accept), paketin kabul edilmeyip reddedilmesi (deny) ve paketin kabul edilmeyip reddedildiği durumlarda pakete kaynak teşkil eden bilgisayara durumun ICMP paketi ile bildirilmesi (reject).

3.5.5.1. Paket üzerinde kural listelerinin uygulanması

Firewall'umuzun bir ağ arayüzüne ulaşan paketin kural listelerini takibi şu şekilde gerçekleşecektir:

- Paket ilk önce kayıt tutma listesinden geçirilerek paket sayıcı ve bayt sayıcı değerleri değiştirilir.
- Paket daha sonra giriş kural listesindeki kurallar ile karşılaştırılarak kabul edilip edilmeyeceğine karar verilecektir. Kabul edilen paketlerin bu aşamadan sonra çalışma şeklimiz doğrultusunda yerel bir sokete yönlendirilmesi (redirect) (vekil veya uygulama firewall özelliği kullanılıyorsa) veya hedef adres doğrultusunda iletilmesi gerçekleşecektir. İletilen paketler için bir sonraki adım iletim kural listesi olur.
- İletimi yapılacak paketler iletim kural listesindeki kurallar ile karşılaştırılacak ve uyuşan kural için gereken işlem gerçeklenecektir.

- İletim kural listesinden geçirilen ya da yerel bilgisayarca üretilen paketler çıkış kural listesindeki kurallar ile karşılaştırılır. Uyuşan kural doğrultusunda gereken işlem yerine getirilir.
- Çıkış kural listesinden geçerek aktarılmasına karar verilen paketler, kayıt tutma listesinden geçirilecek, paket sayıcı ve bayt sayıcı değerler uygun olarak değiştirilecektir.

Paketler üzerinde gerçekleştirilecek işlemler bu sıra ile vuku bulmaktadır. Firewall uygulama yazılımları kural listelerinin değiştirilmesini ve yeni kural tanımlarının eklenmesini sağlar.

3.6. Saldırı Tespit Sistemleri (Intrusion Detection Systems)

Bilgi güvenliği için güvenlik politikasının yaptırımının sağlanması kadar, ihlallerin tespitide önem arz etmektedir. Saldırı tespit sistemleri, yerel ağdan veya bağlı bulunulan harici ağlardan gelebilecek ve ağımızdaki sistemlere zarar verebilecek, çeşitli paket ve verilerden oluşabilen saldırıları tespit ederek kayıt tutmak ve uyarı mesajları üretmek görevini üstlenmişlerdir.

Saldırganların bir kısmı bu iş için otomatize edilmiş program veya programcıklarla saldırır, uzman seviyesindeki saldırganlar ise saldırdıkları hedefe göre değişebilen çeşitli yöntemler kullanır. Yerel ağı koruma amaçlı olan firewall ve anti-virüs gibi sistemler sadece ilk tip saldırganları engelleme imkanı sunmaktadır. Korunmak için kurulan bu sistemler aslında saldırganları sadece yavaşlatır. Yavaşlatma aşaması, saldırıları tespit edip yakalayabilmek ve zararı mümkün olduğunca aza indirmek için güvenlik mekanizmasının sadece ilk katmanını oluşturur. Saldırının başarısı ile pasif duruma düşürülen firewall ve anti-virüs sistemlerine ek olarak güvenlik mekanizmasının ikinci katmanını saldırı tespit sistemleri ile teşkil etmek geekir. Böylece gerektiğinde aktif olabilecek bir savunma aracı kazanılmış olur.

Saldırı tespit sistemleri ile ağa yapılabilecek saldırıları belirleme ve düzenli olarak kayıtlar tutma yeteneği elde edilir. Ürettikleri sonuçlar ile potansiyel olarak tehlike arzeden güvenlik zaafları saldıranların tepkilerinden belirlenebilir. Gelen saldırıların

karakteristiğini saptama ve ağırlık verilmesi gereken noktaların daha iyi analiz edilmesi imkanı kazanılır. Gerekiyorsa bu sonuçlar değerlendirilerek engelleme imkanı kazanılır.

Saldırı tespit sistemlerini çalışma mantığı ve çalışma mimarisi açısından bölümleyerek inceleyebiliriz. Çalışma mantığı itibariyle saldırı tespit sistemleri aktif ya da pasif olabilir. Çalışma mimarisi olarakta sunucu tabanlı ya da ağ tabanlı olarak ayırabiliriz. Bu ikisini birleştirdiğimizde bir saldırı tespit sistemi aktif/suncu tabanlı, pasif/sunucu tabanlı, aktif/ağ tabanlı ya da pasif/ağ tabanlıdır.

Saldırganları sahte ortamlara yönlendiren honeypot ve honeynet adı verilen sistemlerde saldırı tespit sistemleri dahilinde gösterilmektedirler. Honeypot'lar, üzerlerine saldırmaları için tasarlanmış, saldırganları aldatmaya yönelik sistemlerdir. Atak girişimlerini tespit edip gerekli yerlere uyarı mesajları yollamak için kullanılırlar. Bir honeynet ise güvenlik duvarının arkasında kalan ve muhtemel bir saldırı hakkında yararlı bilgiler edinilmesine yönelik dizayn edilen bir ağdır. Bu ağ sayesinde, potansiyel saldırganların güdüleri, kullandıkları araç ve taktikler ortaya çıkar.

3.6.1. Saldırı tespit sistemlerinin çalışma mantıkları

Pasif çalışma mantığını kullanan sistemler, anti-virüs sistemlerinde olduğu gibi oluşturulmuş çeşitli imzaları paketleri incelemek ve saldırıları saptamak için kullanır. Aktif çalışma mantığında ise sistemlerin ve ağın işleyişi belirli bir düzenle özdeşleştirilmiştir, bu düzende olabilecek herhangi bir anormallik saldırının tanımlanmasını sağlamaktadır.

Pasif saldırı tespit sistemleri belirlenen çeşitli kurallar çerçevesinde, ağ üzerinde yakalanan paketleri incelemeye aldıkları için her saldırının izlerinin tanımlanmış olması gereklidir. Genelde bu tür sistemlerde saldırıların çokluğu, her saldırı varyasyonu için ayrı kurallar koyma, işi bir miktar zorlaştırmaktadır. Ancak ticari yazılımlarda otomatik olarak Internet'ten hergün yeni saldırı imzaları indirilebilmektedir. Ticari olmayan yazılımlarda da benzeri bir durum geçerlidir.

Aktif saldırı tespit sistmlerinde durum bir miktar daha akla yatkındır. Ağda ya da çeşitli sunucularda düzenli olarak yapılmakta olan işlemleri takip ederler ve farklı ya da olağandışı hareketler gördüklerinde ise rapor ederler. Bu tür sistemlerin normal olarak nitelendirilebilecek hareketleri öğrenmeleri oldukça fazla zaman almaktadır. Ayrıca bu hareketlerin zaman içerisinde değişebilirliği, kurulduğu sistemlerin yeniden yapılandırılması veya ağa yeni sistemler eklemek işi daha da zorlaştırmaktadır.

3.6.2. Saldırı tespit sistemlerinin çalışma mimarileri

Ağ tabanlı saldırı tespit sistemleri ağa yapılabilecek olası saldırıları raporlamak üzere tasarlanmıştır. Ağ tabanlı sistemler olarak tanımlanan grup, yerel ağdaki tüm bilgileri yakalayabilen bir sisteme kurulur ve ethernet kartı promiscuous moda geçirilerek ağdaki tüm trafik dinlenir Yakalayabildikleri tüm paketleri incelerler ve paketlerin ağa giriş izni olup olmadığına, saldırı imzası içerip içermediğine, bir anormallik teşkil edip etmediğine karar vererek rapor eder, uyarı yayınlar ya da gerektiğinde bir sunucuya açılmakta olan oturumu engeller. Kuruldukları sistemde dinleyecekleri ağ sayısı kadar kaliteli ethernet kartı bulunmalıdır. Performans kaybına yol açmamak için genel olarak posix tabanlı sistemlerde ya da kendi özel işletim sistemlerinde çalışmaktadırlar. Ağ parçalarını dinlerken bazı saldırı tespit sistemleri tek bir sistem ile bu işlemi tamamlarlar, bazı sistemler ise her ağ parçasını kendisinden farklı her biri yardımcı (agent) olarak adlandırılan sistemler ile dinlemektedir.

Sunucu tabanlı saldırı tespit sistemleri tüm ağı değilde sadece üzerine kurulduğu sunucuya yönelik saldırıları tespit etmek veya önlemek şeklinde çalışırlar. Bulundukları sistemlerin konfigürasyon dosyalarını incelemek, sistem ile ilgili kayıtların tutulduğu dosyaları izlemeye almak, sistemin bütünlüğünde meydana gelebilecek değişiklikleri incelemek ve sisteme yönelik kötü niyetli kullanımları engellemek görevlerinden başlıcalarıdır. Kuruldukları sistemlere tam olarak uyum sağlayabilmeleri konusunda zorlukları vardır. İşletim sistemlerinin doğası gereği birbirleriyle uyumluluk göstermeleri nadirdir ve bu durum saldırı tespit sistemlerinin o işletim sistemine özel yazılmış olması, o sistemin zayıflıklarına uygun

yapılandırılmış olması gibi zorunlulukları ortaya çıkarmaktadır. Özel bir sunucu yazılımı için üretilmiş olanlarıda vardır.

3.6.3. Saldırı tespit sistemlerinin zayıflıkları

Her saldırı tespit sisteminin yerleşim yönteminden ya da çalışma yönteminden kaynaklanan çeşitli zayıflıkları ve engelleri vardır. Bunlara ek olarak tüm saldırı tespit sistemlerinin ortak problemleride vardır.

Ortak problemlerine bakılırsa, çok fazla hatalı kayıt ürettikleri görülmektedir. Henüz tam anlamıyla gerçek ve güvenilir kayıtlar üretememektedirler, üretilen kayıtlardan büyük bölümü bir saldırıya ait değildir. Alınan yalnış bir alarm sonucu verilecek tepkide gerekli gereksiz bağlantılar kesilebilir, çeşitli adreslerden gelen istekler reddedilebilir. Bu durum saldırganlarca kötüye kullanılabilir. Bir saldırgan gönderdiği sahte adresli paketler ile hizmet alınan bir ağın bağlantısını koparabilir.

Ayrıca kurulan saldırı tespit sistemlerinin birbirleriyle haberleşememesi ya da sadece aynı firmaya ait sistemlerin haberleşebilmesi de bir diğer ciddi zaafıdır. Bu konuda ortak bir dil oluşturma çalışmaları devam etmektedir.

Bir diğer zayıf yönleri ise şifrelenmiş veri trafiği konusunda çözümsüz olmaları, örneğin SSL ile kurulmuş bir oturum içerisindeki paketleri farkedememesi ciddi engellerindendir. Genel olarak saldırganlar ele geçirdikleri sistemlere basitçe bir şifreleme (mesela blowfish) yöntemiyle verileri şifreleyen trojanlar yerleştirerek saldırı tespit sistemlerini etkisiz hale getirebilmektedirler.

Aktif çalışma mantığını kullanan sistemlerde öğrenme aşaması uzun sürmektedir. Bu durum öğrenme sürecinde saldırı tespit sisteminin işe yaramayacağı anlamına gelmektedir. Ağa eklenecek yeni bir sunucu, sunulacak ya da alınacak yeni bir hizmet, sistemi tekrar öğrenme sürecine sokacaktır. Bu öğrenme süreci içerisinde sürekli olarak yalnış kayıtlar üretilecektir.

Yine bir başka zayıflık ise saldırganların bu öğrenme sürecinde sistemi düzenli olarak incelemesi ve olağan sayılabilecek hareketler ile hareket edebilmesi veya baştan olağan olan ancak daha sonra yavaş yavaş arttırılan haklarla sisteme müdahale edebilmesidir.

Kural tabanlı yöntemle çalışan sistemlerde ise anti-virüs sistemlerine benzer zaaflar mevcuttur. Tanımlanamayan bir saldırı başarılı olabilmekte ya da kayıt edilememektedir. Tüm kuralların sürekli olarak güncellenmesi gerekmektedir, haliyle bu işlem ya otomatize olarak yapılmalı ya da düzenli olarak bir yönetici tarafından yapılandırılmalıdır. Anti-virüs sistemleri kadar gelişmiş olmayan imza veritabanları ancak çeşitli kurallar tanımlanmasıyla anlam kazanabilmektedir. Ayrıca belirlenen kurallar içerisinde dikkat edilmesi gereken birkaç ayrıntı da bulunmaktadır. Bir kural yazılırken saldırı olarak tespit edilecek paketin tüm özellikleri, paket boyu, hedefi, içeriği, kaynağı, protokolü, hedef ve kaynak portu tam olarak tanımlanmalıdır. Böylece aynı imzayı taşıyan fakat bir saldırı olmayan paketlerin yanlış alarm olarak gelmemesi sağlanmış olur.

Saldırı tespit sistemleri için kural yazılımında küçük varyasyonlarda dikkate alınmalıdır. Mesela "http://www.deneme.com/dene.idq" diye bir bilgi saldırı olarak tanımlanıyorsa ve gelen giden paketlerin veri kısmında bu bilgi aranıyorsa "http://www.deneme.com/./dene.idq" şeklindeki bir bilgi aynı sonucu vermesine karşın saldırı tespit sisteminden kaçmış olacaktır. İmzalarda çeşitli oynamalar yaparak amacına ulaşmaya çalışan saldırganlar bulunmaktadır.

Ağ tabanlı sistemlerde genel sorunlar ağın yapısı ve donanımlarla ilgilidir. Ağdaki tüm trafiği izlemesi istenilen saldırı tespit sistemi üzerinde 100Mbit bir ethernet kartı bulunuyorsa ve izlenilecek veri trafiğinin aktığı switch üzerinde ise 12 port bulunuyorsa, 11 portun 1 porta kopyalanması şeklindeki ayar değişikliği switch'in 11x100Mbit trafiği saldırı tespit sisteminin 100Mbit ethernet kartına yönlendircek olması ciddi oranda kaçan veri trafiğine neden olacaktır. Böyle bir durumda yapılabilecek çok fazla önlemde bulunmamaktadır. Ancak son zamanlarda geliştirilen projelerde, bu sorunu aşmak için saldırı tespit sistemlerinin switch

içerisine gömülmesi düşünülmektedir, böylece trafiğin büyük bölümü yakalanabilecektir.

Yüksek miktarda veri trafiğini incelemede ikinci bir sorunda saldırı tespit sistemlerinden en iyisinin bile 60Mbit üzerindeki trafiği kaçırıyor olmasıdır. Bu da incelenecek veri miktarını iyice düşürüyor. Ayrıca tek engel veri miktarı da değil, açılan oturumların sayısıda saldırı tespit sistemini zorlamaktadır. Örneğin bolca ICQ mesajının aktığı ağda topu topu 30Mbit veri akmakta iken saldırı tespit sistemi ciddi sorunlar yaşayacaktır.

Bu sorunları iyi kullanan saldırganların geliştirdikleri yöntemler içerisinde parçalanmış paketler ile saldırı gerçekleştirmek en yöntemlerindendir. Ağ yoğun çalışan bir ağ ise saldırganın parçalayarak gönderdiği paketlerden bir kısmı saldırı tespit sistemi tarafından yakalanabilmekte iken bir kısmıda hiç yakalanamayacaktır. Birleştirilemeyen paket içindeki imzalar doğrulanamayacağından saldırı tanımlanmayacaktır; ancak hedefin paketleri birleştirip içerisine bakması durumunda saldırı gerçekleşecek ve kayıtlarda da gözükmeyecektir.

Sunucu tabanlı sistemlerin ise daha çok çalıştığı platform ve taşınabilirlik problemleri vardır. Genel olarak girdileri işletim sisteminin oluşturduğu kayıtlar olmasına rağmen, bazı sunucu tabanlı sistemler ethernet kartını promiscuous moda geçirerek, paketlerden sisteme veya özel bir servise ulaşmak isteyenleri engellemeye çalışırlar.

Girdilerin sunucunun kendi kayıtlarından alınması her işletim sisteminin kendine ait bir kayıt tutma özelliği ve bazı işletim sistemlerinde önemli olan dosyaların diğer işletim sistemlerinde aynı önemi arzetmemesi ya da bulunmaması taşınabilirliklerini ve kurulduğu işletim sistemi platformlarında güvenlik oranın sürekli aynı olmasını engeller.

3.6.4. Saldırı tespit sistemlerinin yerleşimleri

Saldırı tespit sistemleri tek bir şekilde yerleştirilmezler, ağ tabanlı ise ağ trafiğinin dinlenilmesi istenen konumuna ya da sunucu tabanlı sistemler için saldırıya karşı korunmak istenen sunucular üzerine konuşlandırılırlar. Saldırı tespit sistemlerinin kullanılmasında yazılım ya da donanım üreticisinin talimatları da dikkate alınmalıdır. Şekil 3.17 ve 3.18'de harici ağ bağlantılı ağlar için saldırı tespit sistemlerinin nasıl konuşlandırılabileceği örneklenmiştir.

Şekil 3.17'de Internet bağlantısı bir yönlendirici üzerinden gerçekleştirilmiş ve çevre ağ (DMZ) üzerinde sunucuları konuşlandırılmış bir ağ mimari gözlenmekte. DMZ bölgesinde web ve mail sunucusu olmak üzere 2 adet sunucu mevcut.

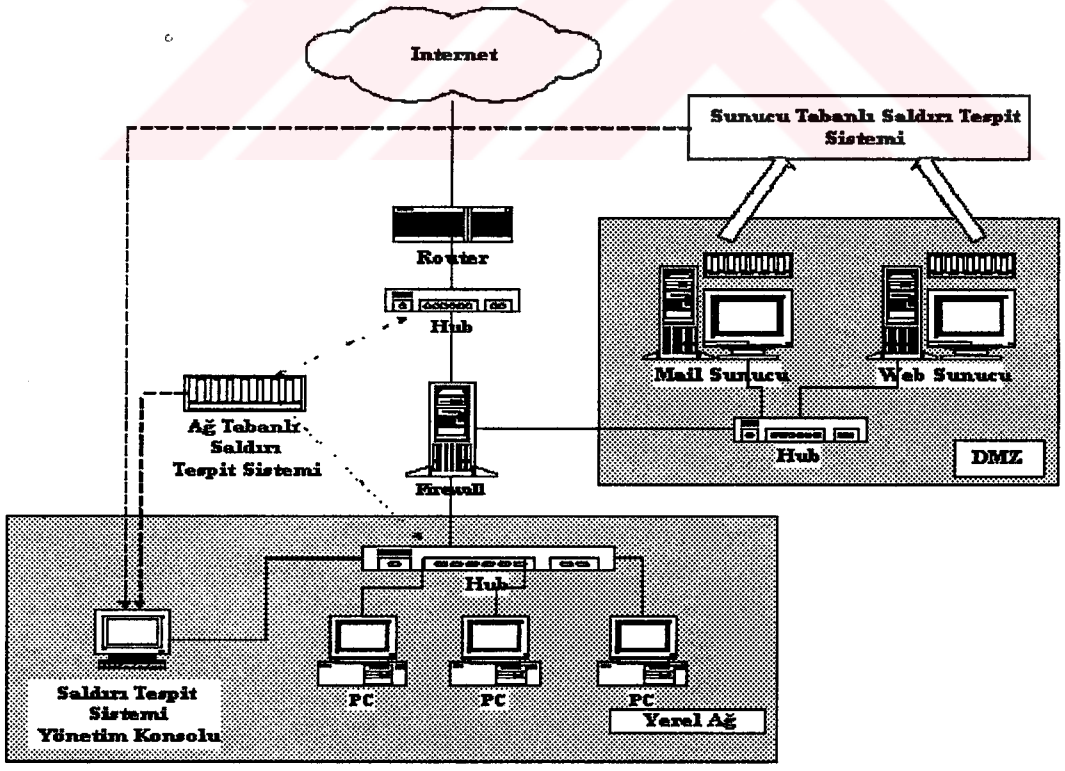
Ağ tabanlı saldırı tespit sistemiyle bu ağ için trafiğin dinlenebileceği iki uygun nokta belirlenmiştir. Ağ tabanlı saldırı tespit sisteminin birinci bağlantısı firewall ile yönlendirici arasındaki trafiği dinlemek üzere firewall ve yönlendirici arasındaki hub'a yapılmıştır. İkinci bağlantı ise yerel ağda yer alan hub'a yapılmıştır. Bu bağlantılar sniffer modunda çalıştırılarak birincisiyle yerel ağ ve DMZ bölgesinin Internet ile olan trafiği, ikincisiyle firewall'un arkasındaki yerel ağ trafiği dinlenebilecek ve saldırı imzalı paketler yaklanabilecektir. Hub yerine switch kullanılıyorsa (muhtemelen kullanılıyor) portlar ağ tabanlı saldırı tespit sisteminin portuna aynalanması gerekmektedir.

Bu yapıda ağ tabanlı saldırı tespit sistemi raporları tutmak ve yönetimi greçekleştirmek amacıyla yönetim konsoluna çeşitli bilgiler göndermekte ve almaktadır. Firewall ile yönlendirici arasını dinleyen ilk bağlantı genelde bir IP adresine sahip olmaz (çünkü saldırıya açık hale gelmesi böylece mümkün olabilir), diğer bağlantıda ise yerel bir IP adresi bulunur. Bu IP üzerinden yönetim gerçekleştirilir. Gerektiği durumda iki bağlantıdan da sunuculara ya da diğer kullanıcılara bir TCP bağlantısı sonlandırma isteği gönderebilir.

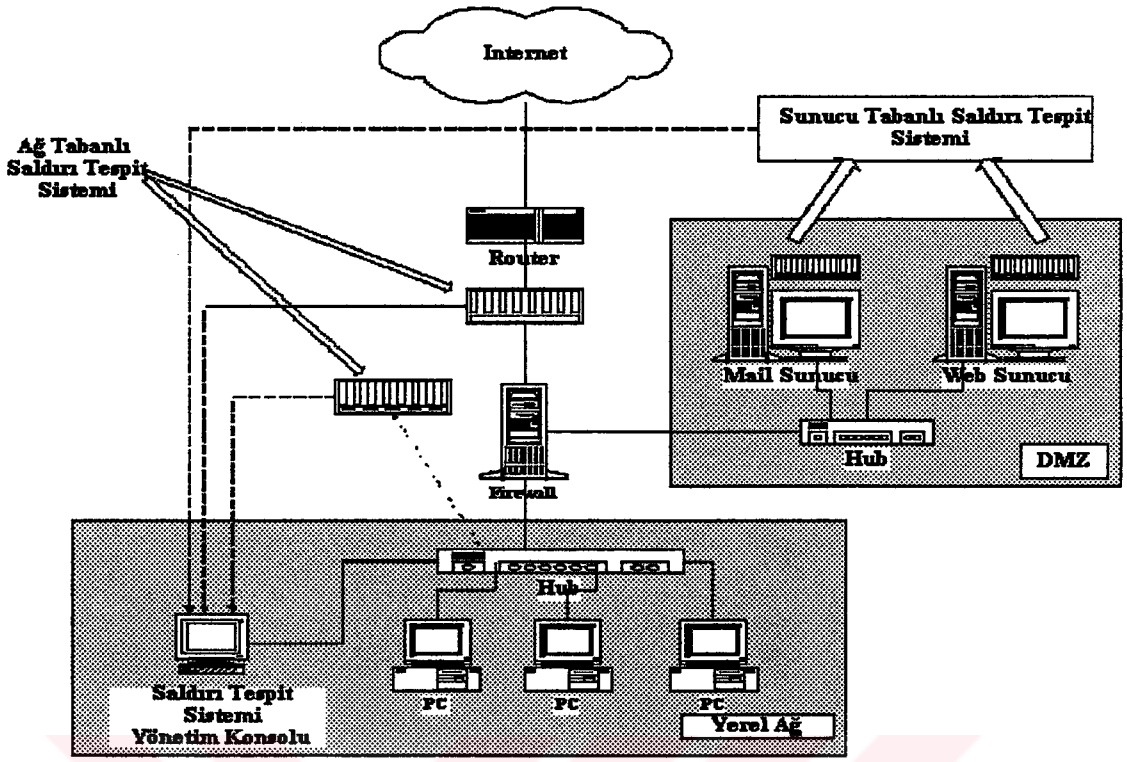
Sunucu tabanlı saldırı tespit sistemi ise yüklü bulunduğu mail ve web sunucularından ürettiği raporları yine yönetim konsoluna göndermektedir. Yönetim konsolu

aracılığıyla yönetilmesi için sunucuların IP'leri kullanılır. Bu yönetim işleminin gerçekleşeceği portlara firewall üzerinde yerel ağ için izinler tanımlanmış olmalıdır. Internet'ten gelen istekler için ise bu portlar firewall üzerinden kapalı olarak düzenlenmelidir.

Şekil 3.18'deki ağ mimarisi aynı olup farkı eklenen yeni bir ağ tabanlı saldırı tespit sistemi yardımcısıdır. İki adet yardımcı aracılığıyla Şekil 3.17'de anlatılan işlemler yine tam olarak yapılmaktadır. Aradaki tek fark ilk yardımcının firewall ve yönlendirici arasına bir köprü gibi ilave edilmesidir. Böylece traceroute istekleriyle saptanması mümkün olmayacaktır. Bir adet yönetim IP'si dışında üzerinde IP tanımlı değildir. Genellikle tespiti güç olsun düşüncesiyle tasarlanan bir sistemdir. Ancak bu durumda bazı zararlarıda vardır, örneğin dışarıdan gelen tüm trafik saldırı tespit sistemi üzerinden geçmektedir, böyle bir durumda saldırı tespit sistemi limitlerinin aşılması (60Mbit gibi yada oturum sayısının çok fazla olması gibi) geçmekte olan trafiğin sorunlu olmasını sağlar. Doğal olarak Internet bağlantısında bazı problemler oluşabilir, sunucular Internet'te hizmet veriyorsa hizmetlerinde aksamalar olabilir.



Şekil 3.17 Saldırı tespit sistemlerinin yerleşim uygulaması



Şekil 3.18 Saldırı tespit sistemlerinin yerleşim uygulaması

Bu durumun bir diğer sakıncası da saldırı tespit sisteminin ele geçirilebilirlik ihtimalidir. Çünkü üzerinden bir trafik geçmektedir ve bu trafik içerisinde saldırı tespit sisteminin bazı zaaflarını kullanan paketler mevcut ise saldırı tespit sistemi kilitlenebilir, böylece Internet bağlantısı kesilmiş olur ya da bu sistemin kontrolü ele geçirilerek direk olarak bu sistemin haklarıyla ağa ulaşma imkanı doğabilir. Böyle bir durumda firewall üzerinden trafiğin akmaması ihtimali dahi vardır, muhtemelen yönetim konsolu için ayrı bir bağlantı sistem üzerinde bulunur ve saldırgan bu kartı kullanabilir.

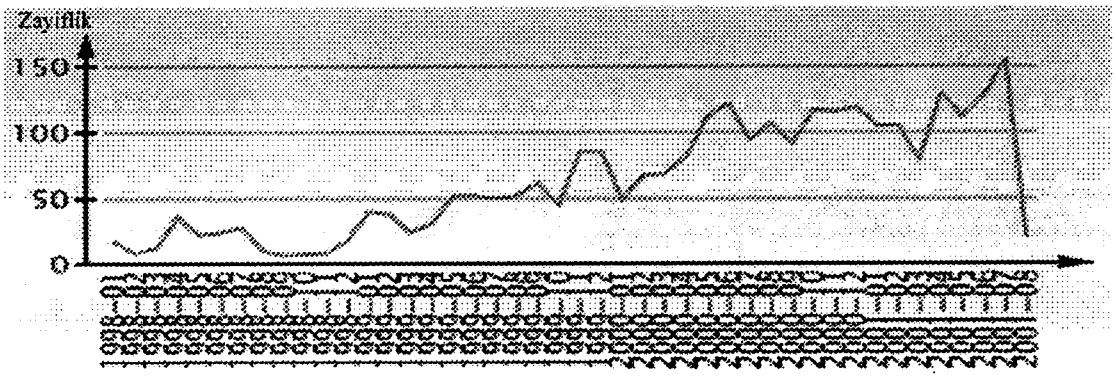
3.7. Zayıflık Tarama Sistemleri

Bilgi güvenliğini bir zincir gibi düşünürsek bu zincirin sağlamlığı, zinciri oluşturan halkalardan en zayıfıyla eş değerdir. Bilgi güvenliği zincirini oluşturan halkalar; sistemin yapısı (yazılım ve donanım), savunma uygulamaları, kullanıcı eğitimi ve yedekleme uygulamalarıdır. Bu öngörü çerçevesinde zayıf halkanın tespit edilmesi

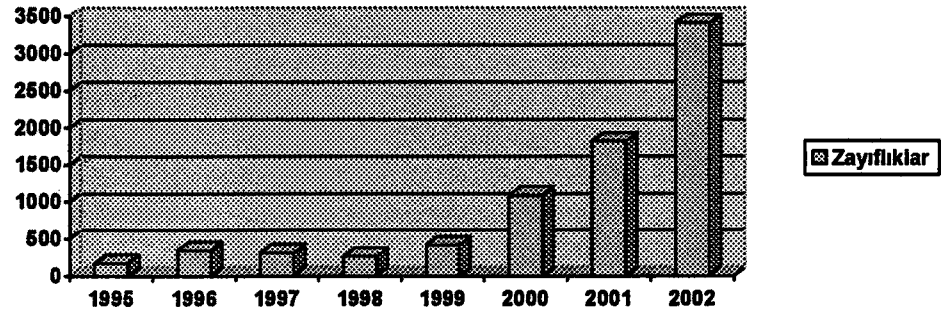
ve güçlendirilmesi çok önemlidir. Bütünsel güvenlik için zayıf noktaların belirlenmesi birkaç yöntem ile sağlanabilir. Ağa sızma testleri, yerel ağ analizi ve zayıflık tarama sistemleri bu yöntemlerdendir.

Security Focus ve Cert/CC'den alınan istatistikler son yıllarda bulunan güvenlik zayıflıklarının takip edilmesi güç rakamlara ulaştığını göstermiştir. Özellikle Internet kullanımındaki artış, programların Internet ortamına taşınması, Internet servislerinin artışı sistemler için potansiyel bir risk teşkil eder duruma getirmiştir. Yazılım üreticileri kendi aralarında ve açık kaynak kodlu serbest yazılımlar ile eskiye oranla daha fazla rekabete girmişlerdir. Bunun sonucunda ürünlerinin güvenlik testlerinden tam olarak geçirilmemesi, varolan ürünlerin Internet ortamına güvenli aktarımı yerine hızlı aktarımının tercih edilmeside güvenlik açısından büyük riskler doğurmuştur. Saldırganlar, her yazılımda bulunabilecek güvenlik zayıflığının ağın ele geçirilmesine yardımcı olabileceğini fark etmişler ve Internet ortamı ile kullanılabilecek tüm yazılımlarda güvenlik zayıflığı aramaya başlamışlardır.[3]

Ağa dahil sistemlerden herhangi birinde bulunan güvenlik zayıflığı tüm ağ için tehlike oluşturur. Ağ tasarımının güvenlik politikası çerçevesinde güvenlik zayıflığına olanak vermeyecek şekilde yapılandırılması gerekir. Ama zaman içinde ağ içinde olabilecek değişiklikler, eklentiler güvenlik risklerine sebep olabilir.



Şekil 3.19 Güvenlik zayıflıklarının yıllara göre artışı (Security Focus araştırma sonucu)



Şekil 3.20 Yayınlanan zayıflıkların yıllara göre değişimi

Bahsedilen bu güvenlik zayıflıklarının düzenli takibi ve raporlanması otomatik zayıflık tarama sistemleri ile düzenli olarak gerçekleştirilebilir. Zayıflık tarama sistemleri, bilgisayar sistemlerini ve ağ aktif cihazlarını dışarıdan ya da sistemlerin kendi üzerinden inceleyerek doğrudan ya da dolaylı olarak güvenlik problemlerine neden olabilecek noktaları işaret etmeye çalışan, bulunan zayıflıkların kapatılma yöntemleriyle ilgili referans ve açıklamaların listelenmesini sağlayan yazılımlardır. Genel olarak bu yazılımlara inceleme öncesinde ağ üzerindeki kaynaklarla ya da ağ topolojisi ile ilgili herhangi bir bilgi verilmez. Sıfır önbilgi olarak nitelendirilen bu durumda, yazılımın ortalama ya da ortalamanın üstünde beceriye sahip bir saldırganın tespit edebileceği kadar zayıflığı tespit etmesi beklenir.

Üç ayrı sistem ile çalışırlar.

- Yerel sistem zayıflıklarının taranması,
- Uzak sistem zayıflıklarının taranması,
- Uygulamaya özel zayıflıkların taranması.

Bu yazılımlar bir zayıflık veritabanı yardımıyla çalışmaktadırlar. Zayıflık veritabanlarında bulunan zayıflıkları, hedef gösterilen sistemlerde deneyerek rapor ederler. Zayıflık veritabanlarını Internet üzerindeki sitelerinden sürekli olarak güncellemek gerekmektedir. Bu zayıflıklara ek olarak yeni zayıflıkların tanımlanması, yazılımlara yardımcı programcıklar veya konfigürasyon dosyaları

aracılığıyla tanımlanabilir. Bu yazılımlarda zayıflık tanımlamaları için bu amaçla üretilmiş özel bir programlama dilide bulunabilir.

3.7.1. Zayıflık tarama sistemlerinin çalışma mantıkları

Zayıflık tarama sistemlerinin büyük bir kısmı, incelemenin ilk adımı olarak hedef sistemin tüm açık bağlantı noktalarının bir listesini oluşturur ve bu listeyi elindeki bilindik hizmetler (well-known services) listesi ile karşılaştırır. Böylece karşı sistem üzerinde çalışan hizmetleri kesin olmasada tespit etmeye çalışır. Bunu daha sonra yapacağı zayıflık testlerinde baz alacaktır.

Hedef sistem üzerinde yapılan hizmet testinin yanında tespit edilen hizmetlerin hangi uygulamalarca ve hangi sürümlerince verildiği, hedef sistemin kullandığı işletim sistemide anlaşılmaya çalışılır. Böylece zayıflık sorgulamasında ilgili işletim sistemi ve hizmet uygulmasına karşı doğrudan kullanılabilcek zayıflık tanımlamaları aktif hale getirilecetir. Bazı zayıflık testleri ise belirli bir hizmetin verildiği zannından yola çıkarak çok kullanılan ve çok bilinen bazı zafiyetlerin denenmesi şekilnde gerçekleşir. Örneğin Web sunuculardа kullanılan bazı popüler CGI scriptleri buna örnek olarak verebiliriz.

Zayıflık tarama sistemleri yaptıkları bu testlerin sonuçlarını rapor olarak sunabilecek yeteneklerde üretilmekte ve böylece istenilen metin tabanlı ya da grafik tabanlı formatta zayıflık sorgulama sonuçlarını elde etmek mümkün olmaktadır.

3.7.2. Zayıflık tarama sistemlerinin özellikleri

Kullanımlarının getireceği olumlu özellikler ve yararları:

- Düzenli olarak ağlar üzerinde güvenlik zayıflıklarının ortaya çıkarılmasını sağlar.
- Ağlar çok geniş bir alana dahi yayılsa zayıflık tespiti yapılabilir ve gözden kaçırılabilecek zayıflıklar kolayca tespit edilirler.

- Ortalama bir saldırgan seviyesinde sistemler üzerinde olabilecek güvenlik zayıflıklarını test eder ve raporlar. Raporlama sistemleri PDF, XML, HTML, Metin formatlarında olabilir.
- Buldukları zayıflıkları raporlamaları, risk seviyelerini göstermeleri, bu zayıflıkların nasıl kapatılabileceğini önermeleri diğer artılarındandır.
- Yardımcı programlama dilleri sayesinde özel saldırılar oluşturma imkanıda sunulmaktadırlar.
- Çok sayıda işletim sistemi üzerine kurulabilmektedirler (Unix, Linux, Windows NT/2000). Zayıflık veritabanları kolaylıkla Internet üzerinden güncellenebilmekte ve son zayıflıklar içinde teste imkan sağlamaktadır.

Kullanımlarından kaynaklanan olumsuzluklar ve eksiklikler:

- Henüz yayınlanmamış güvenlik zayıflıklarını bulamazlar.
- Gerçek bir saldırgan gibi saldıramazlar. Ancak veritabanlarında tanımlandığı çerçevede saldırılar gerçekleştirirler.
- Paket kayıplarının yoğun olduğu ağlar üzerinde doğru sonuçlar üretemez ve yanıltıcı olabilirler.
- Uygulamaların açılış mesajları veya verilebilecek sahte yanıtlar zayıflık tarama sistemlerini yanıltabilirler.
- Bir bağlantı noktasını dinmekle hizmetin verildiği sonucuna ulaşıltıkları için hatalı hizmet tespiti yapabilirler.
- Belirli bir hizmet uygulamasına ya da işletim istemine göre uygulanacak zayıflık testlerinde karşılama mesajlarından (service banner) faydalanmaları yalnış tarama sonuçları üretmelerine sebep olabilir.

3.7.3. Zayıflık tarama sistemlerinin seçim kriterleri

Zayıflık tarama sistemleri, kendi aralarında da çeşitli avantaj ve dezavantajlara sahiptirler.

- Seçim kiterlerinin başında maliyet ve lisanslama gelmektedir. Bazı zayıflık tarama sistemleri bir ağ aralığı için lisanslanmakta iken bazıları sunucu başına, bazıları ise sınırsız olarak lisanslanmaktadır. Nessus, Sara ve Saint gibi açık kodlu tarama sistemleri ise serbestçe kullanılabilirler.

- Her zayıflık tarama sistemi eşit kapasitede zayıflık veritabanına sahip değildir, bazı işletim sistemleri için zayıflıkları kapsamayabilirler, zayıflık veritabanları geç güncellenmektedir veya yerel bir sistem için alınacaksa her sistemde eşdeğer güvenlik seviyesi sunmazlar.
- Bazı tarama sistemleri çeşitli kuruluşların yayınlanan zayıflıkları onaylamasını beklerken, bazı sistemler gönüllü kişiler ile veritabanını sürekli güncellemektedir. Örneğin daha sonra uygulamasını gerçekleştireceğimiz Nessus zayıflık tarama sisteminin zayıflık veritabanı gönüllü kişilerce, düzenli olarak, NASL dili ile yazılmış script'ler aracılığıyla güncellenmektedir.
- Artımlı tarama yapabilmek, yani daha önce yapılan testleri uygulamayıp, sadece yeni testleri uygulamak, daha önce yapılan tüm testleri bir veritabanında saklayarak karşılaştırmak ve geriye dönük işlemler yapabilmek yine fark oluşturan etkenlerdendir.
- Bazı tarama sistemleri çok kullanıcılı ortamları ve sunucu/istemci mimarisi ile yönetim sağlanmasını desteklemektedir.

Konumuz gereği ağlardaki zayıflık taraması üzerinde durulacak ve uzak sistem zayıflık tarama yazılımı Nessus'un kurulum ve kullanımı açıklanacaktır. Nessus uygulamasının çalışma ortamı Linux Redhat seçilmiş ve bu doğrultuda uygulama ve açıklamalar yapılmıştır.

3.8. Saldırı Örnekleri ve Yapılabilecekler

Harici ağlar ya da Internet üzerinden ağımıza gelebilecek çok kullanılan bazı saldırı teknikleri üzerinde durulacak. Ayrıca bu saldırılar karşısında yapılabileceklerede değinilecektir.

3.8.1. IP spoofing

Internet üzerinde çok görülen ve diğer saldırılarında destekleyicisi olarak kullanılan bir saldırı şeklidir. IP adresi bilgisi doğrultusunda kullanıcılarını tanıyan ve yetkilendiren hizmetler üzerinde oldukça etkilidir.

TCP/IP protokolü üzerinden iletişim yapan sistemler karşılıklı birbirlerinin IP adreslerini bilmek zorundadır. İletişimde bulunulan sistemin IP adresini bilmek güvenlik açısından da önemlidir. Bazı Internet siteleri sadece belirli IP'lerden gelen isteklere cevap verirler. Bazı programlar IP adresi bilgisi doğrultusunda hizmet ve yetkilendirme sunar. Mesela Unix r komutları olarak bilinen rlogin, rsh gibi programlar bu tiptedir. Sistemlerin birbirine güvenmesine dayandığı için ayrıca bir kullanıcı tanımlaması gerekmez. Bu şekilde gerçekleştirilen bir çalışma bazı rahatlamalar, kolaylıklar sağlasada güvenlik sorunlarını birlikte getirir.

Bir kullanıcı bilgisayarının IP adresini değişik bir adres olarak göstermesi, iletişimde bulunduğu bilgisayarı aldatması mümkündür. Sistemlerin veri iletişiminde kullandıkları IP paketlerinin başlık kısımlarında bulunan ve paketin kimden geldiğini gösteren kaynak adres bölümünü değiştirmek suretiyle bu işlem yapılabilecektir. Bunun doğuracağı sonuçlar değindiğimimiz gibi, IP adresi bilgisinin ne oranda ve hangi işlemlerde kullanıldığına bağlı olarak değişecektir.

Böyle bir saldırıyı tam anlamıyla önleyebilecek bir yöntem yoktur. Gelen bir IP paketinin kaynak adres kısmında yazan bilginin, gerçekten paketi gönderen bilgisayara ait olup olmadığını belirleme şansı her zaman için mevcut değildir. Yinede bu tür paketlerin bir kısmını belirleme şansımız mevcuttur. Bunlardan en etkili olanı harici ağdan ya da Internet ortamından yerel ağda bulunan bir bilgisayara, yine kendisini yerel ağdaki bir bilgisayar gibi gösterdiği durumlara ilişkin olan paketlerin belirlendiği yöntemdir. Yerel ağ üzerinde bulunan bilgisayarların büyük kısmı birbirlerine güvendikleri için IP adresine göre yetkilendirmeyi aralarında kullanacaklardır. Böyle bir saldırıyı belirlemenin yolu, yerel ağın harici ağa ya da Internet'e açılan noktasına ulaşan paketlerin kaynak ve hedef adreslerine bakmaktır. Eğer paketlerin hem kaynak adresleri hem hedef adresleri yerel ağa ilişkin IP adresler içeriyorsa bu bir saldırı işaretidir ve engellenmelidir.

Böyle bir durumun belirlenemeyeceği durumda oluşabilir. Yerel ağımızdaki bilgisayarların harici ağ ya da Internet üzerindeki bazı IP'lere güvenmesi gerekiyorsa bu durumda firewall'umuz üzerinde güvenilecek IP'lerin geçişi kısıtlanamayacak. Saldırgan küçük bir çalışma yaparak bu IP'leri tespit ederse ve gönderdiği paketlerin

kaynak adres kısmına bu IP'lerden birini koyarsa firewall'umuzu geçerek yerel ağda yer alan bilgisayara erişebilecek ve bu IP'ye tanınan hizmet ve yetkileri kullanabilecektir. Maalesef böyle bir saldırı anlaşılamaz. Bu tip bir saldırıyı önlemenin temel yolu; yerel ağda yer alan bilgisayarların harici ağ ya da Internet üzerindeki diğer bilgisayarlara güvenmemesidir. Bu durumda harici ağ ile yerel ağ arasına konulacak paket filtreleme yaklaşımı ile güvenlik iyi bir şekilde sağlanmış olacaktır.

Bu tip saldırıları engelleyecek diğer bir yol ise kaynak yönlendirmeli paketlerin geçişine izin vermemektir. Normalde IP paketlerinin hedefe giden yol üzerinde nasıl bir yol izleyecekleri ağdaki düğümlerce karar verilir. Kaynak yönlendirmeli pakeler ile çalışma çok kısıtlıdır. Dolayısıyla böyle paketlerin firewall üzerinden engellenmesi yapılarak gelebilecek saldırılar için önlem alınmış olacaktır.

IP adresi yoluyla yanıltma yapılarak gerçekleştirilen saldırıların çoğunda saldırgan, sunucu tarafından gönderilen cevap paketlerini çoğu zaman alamaycaktır. Çünkü cevaplar gerçek IP sahibine gidecektir. Ama çoğu saldırı için cevap paketlerine ihtiyaç yoktur, saldırgan cevap paketlerini elde etmeden yapmak istediklerini gerçekleştirebilir. Sunucu tarafında oturumun açık tutulabilmesi için TCP sıra numarası tahminide gerekecektir.

Bu saldırı tip saldırıları engellemek için bir firewall üzerinde nasıl tanımlamalar yapmak gerekir bunu örnekleyeceğim. Bunun için Ipchains yazılımını baz alacağım. Ipchains -i [interface] bu parametre ile paketin makinaya hangi arayüz aracılığıyla geldiğini kontrol edebiliriz; lo, ethx, pppx bu arayüzlerden biri olabilir. Bu şekilde beklemediğimiz yerden gelen paketleri de kontrol edebiliriz. Şekil 3.21'deki gibi ağ için IP spoofing engellenme için aşağıda belirtilen Ipchains kuralları yazılmalıdır. (eth1 harici ağa bakan ağ arayüzümüzün ismidir)

```
#ipchains –A input –i eth1 –s 213.238.128.0/24 –d 0/0 –j DENY
#ipchains –A input –i eth1 –s 127.0.0.0/8 –d 0/0 –j DENY
#ipchains –A input –i eth1 –s 10.0.0.0 /8– 0/0 –j DENY
#ipchains –A input –i eth1 –s 172.16.0.0/16 – 0/0 –j DENY
```

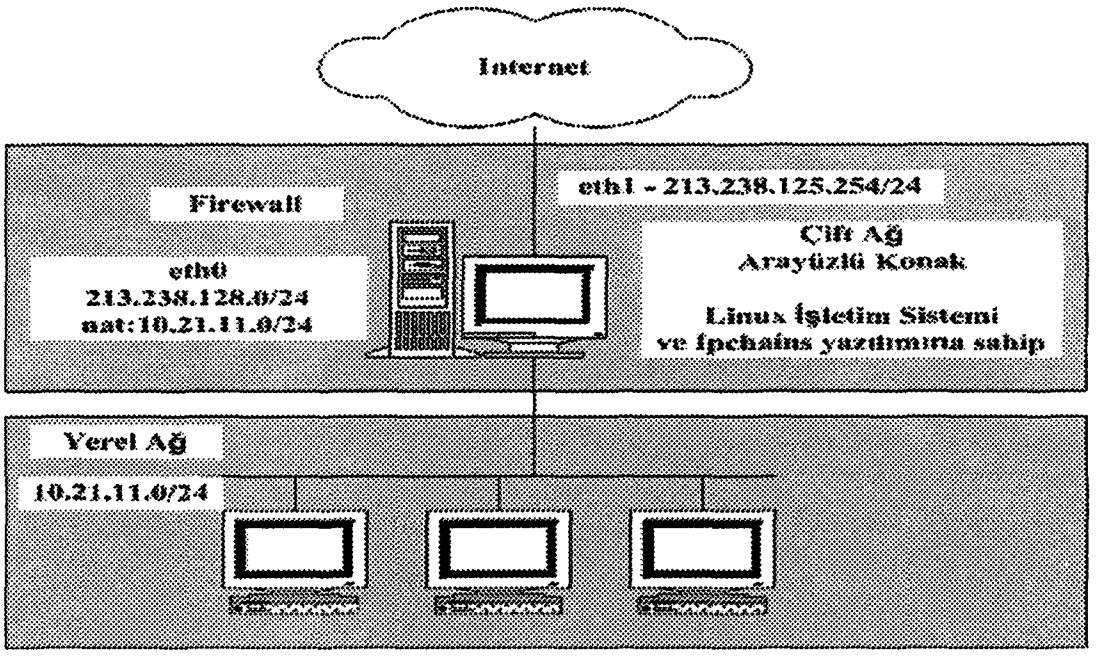
#ipchains –A input –i eth1 –s 192.168.0.0/16 – 0/0 –j DENY

Eklenen son dört kural özel anlam içeren ağ adresleri (loopback, reserved address gibi) için değerlendirilir.

Harici ağ bağlantısı IP adres maskelemesi (masquerading) ile gerçekleşiyor. Makelemede IP kontrlü yanı sıra arayüz kontrolüde eklenirse IP spoofing saldırılarının önüne bir ölçüde geçilebilir. Şekil 3.21'deki 10.2l.11.0 ağına eth1 aracılığıyla maskeleme yaptığımızı düşünürsek, maskeleme tanımında arayüz belirtmek; kendini bu adrestenmiş gibi gösteren birinin; bu ağa tanıdığımız haklardan yararlanmasını engelleyecektir.

ipchains -A forward -s 10.21.11.0/24 -d 0/0 -i ethi -j MASQ
ipchains -A forward -s 10.21.11.0/24 -d 0/0 -i eth+ -j MASQ
eth+ == eth1 eth2 .....
ppp+ = pppi ppp2 ..... seklinde de olabilir.

IP spoofing engellenmesi için daha başka yollar da vardır. Firewall genelde bu işi tam karşılamaz. Mesala çekirdek 2.1 den sonrası IP'si 127.x.x.x gibi davranan paketleri kabul etmez. Benzer şekilde çekirdekte kaynak adres kontrolü (Source Adres Verification) özelliği kullanılabilir. /proc/sys/net/ipv4/conf/all/rp-filter 'a değer yüklenebilir. Böylece kaynak yönlendirmeli paket geçişi filtrelenmiş olur.
echo "l" > /proc/sys/net/ipv4/conf/all/rp-filter



Şeki 3.21. Maskelenerek Internet'e açılan ağ üzerinde IP spoofing saldırılarını engelleme

3.8.2. TCP bağlantı isteklerinin engellenmesi

Bilindiği üzere TCP protokol olarak çok kullanılmakadır. TCP bağlantı temelli bir protokol olduğuna ve iletişimin karşılıklı onaya dayalı ilişki üzerine kurulduğuna önceki bölümde değinilmişti. FTP, Telnet vs., TCP paketleri ile yapacakları veri aktarımlarında TCP'nin üç-kurallı-onaylama (three way hanshaking) olarak adlandırılan karşılıklı anlaşma metodunu kullanır. Eğer bir makinaya gerçekleştirilecek TCP bağlantı istekleri kontrol edilmek isteniyorsa sekranizasyon ve onay bitini kullanarak bu gerçekleştrilebilir. Firewall ile bunu yapabilmek için ipchains de "–y" parametresiden faydalanılır.

Bu parametre ile içinde SYN biti 1 olan ve aynı zamanda ACK ve FIN bitleri 0'e eşit olan (SYN=1 && ACK=0 && FIN=0 durumu) paketler yakalanabilir. Parametrenin ! (tersi) opsiyonu da vardır. (bit durumlarının tersi değil kural sağlanmasının tersi). Ve de doğal olarak sadece TCP protokolü için kullanılır.

```
# ipchains -A input -p tcp -s 0/0 -d 213.238.128.202/32 -y -j DENY
```

Komutu ile bilgisayara (213.238.128.202 IP sunucuya) gelen tüm SYN_REQ'ler red edilmiş olur. Böylece dışardan hiçbir şekilde bilgisayara TCP bağlantısı yapılamaz. Ve bu bilgisayarın harici ağa TCP bağlantı istekleri bundan etkilenmez.

3.8.3 TCP SYN paketi ile gerçekleştirilen saldırılar

IP adresi yoluyla yanıltmaya dayalı gerçekleştirilen bir saldırı tipidir. Saldırılan sistem üzerindeki TCP servisini kullanılamaz hale getirmeyi hedef edinen bir saldırı şeklidir. Kısaca hizmet dışı bırakma saldırısıdır.

TCP hizmeti isteyecek sistem sunucuya SYN bayraklı (SYN=1) bir IP paketi gönderir. Bu paketi alan sunucu SYN ve ACK bitleri kurulu bir IP paketi ile karşılık verecektir. Sunucu, istemciden ACK bayraklı bir cevabı oturumun açılması için beklerken, tamamlanmamış istekleri yani yarı açık bağlantıları bir kuyruğa alır. İşte bu saldırı, yarı açık bağlantılar oluşturarak kuyruğu doldurmaya yani yeni gelen bağlantı isteklerinin reddedilmesine ve böylece sunucuyu hizmet veremez duruma getirmeyi amaç edinmiştir. Sunucu yarı açık bağlantıları belli bir süre sonunda kuyruktan çıkaracaktır fakat bu süre saldırganın yeni yarı açık bağlantılar oluşturma süresinden çok daha uzundur.

Ayrıca saldırganlar bu saldırılarda IP adresi yoluyla yanıltma yönteminide eklerler. Böylece saldırganlar kaynak adreslerini gizler ve sunucu cevabının o anda kapalı ya da hiç kullanılmayan bir IP adresine gitmesine neden olurlar ve cevabın sunucuya dönmemesini garanti ederek kuyruğu doldururlar.

Bu tip bir saldırıya karşı alınabilecek tedbir olarak; TCP protokolün geliştirilmesi ve bu açıklarının giderilmesi (Kuyruğun uzatılması, yeni SYN paketi alındığında kuyruktaki en eski yarı açık oturumun çıkarılması). Paket filtreleme düzeyinde ise bir SYN paketinin servis isteyen gerçek kullanıcıya mı yoksa saldırda bulunan bir bilgisayara mı ait olduğunu belirlemek her zaman mümkün değildir. Dolayısıyla firewall düzeyinde yapılacak çalışma, bu saldırıda kullanılan ve paket filtreleme ile önlenebilecek noktalara karşı önlemler almaktır. Bu doğrultuda IP adresi yoluyla yanıltma saldırılarına karşı alınabilecek olan önlemleri almaktır. Ayrıca ağ üzerinde

yoğunlaşan SYN bayraklı istemci isteklerinin saldırı tespit türü sistemlerce yakalanarak gerekli urarıların yayınlanması sağlanabilecek diğer bir tedbirdir.

3.8.4 TCP sıra numarası tahmini yoluyla gerekleşen saldırılar

IP adresi yoluyla yanıltma yönteminin birlikte kullanıldığı bir saldırı tipidir. TCP/IP protokolünün gerçeklenmesinde zayıf bırakılan bazı unsurları kullanır ve IP adresine göre yetkilendirme yapan programları kendisine hedef seçer.

TCP protokolünün bağlantı temelli olmasına dayalı bir saldırı tekniğidir. Bir örnek üzerinden ilerlersek; A, B ve C bilgisayarlarımız olsun. A bilgisayarının IP adresi yanıltma yöntemiyle kendisini C bilgisayarı gibi göstererek B sunucu bilgisayarına bağlanmaya çalıştığını düşünelim. TCP'nin üç-kurallı-onaylama mekanizması gereği A bilgisayarı B sunucusuna bir SYN bayraklı paket gönderecek. Gönderdiği bu paketi sanki C bilgisayarından geliyor gibi göstermek üzere kaynak adres alanını değiştirecek ve pakete kendi sıra numarasını (SN_A) ekleyecektir. Bu paketi alan B sunucu bilgisayarı SYN(SN_B), ACK(SN_A+1) bilgilerini taşıyan paketi kendisiyle haberleşmek istediğini düşündüğü C bilgisayarına gönderecektir. Üç-kurallı-onay metodu gereği ACK(SN_B+1) paketinin B sunucu bilgisayarına gönderilmesi gerekmektedir. Oysa ki SN_B bilgisi kendisini C gibi gösteren A bilgisayarına değil C bilgisayarına gönderilmiştir. Dolayısıyla A bilgisayarının kendisini C bilgisayarı gibi göstererek B sunucu bilgisayarı üzerinde işlem yapabilmesine imkan tanıyacak bağlantıyı tamamlayabilmesi için SN_B değerini belirlemesi ve ACK(SN_B+1) paketini, yine C bilgisayarından geliyormuş gibi göstererek B sunucu bilgisayarına göndermesi gerekir. Bunu başara bildiği taktirde üç-kurallı-onay gerçekleşecek ve B sunucusu C ile haberleştiğini düşünecektir. Eğer bu saldırı, IP adresiyle yetkilendirme yapan bir servise gerçekleşiyorsa, B'nin C'ye güvendiği oranda A bilgisayarı B sunucusu üzerinde işlem yapabilecektir.

Saldırının gerçekleşmesindeki en kritik nokta, A'nın SN_B'yi belirlemesidir. Eğer A bu değeri doğru belirleyebilirse saldırı büyük oranda gerçekleşebilecektir.

Bu noktada sıra numarası bilgisinin nasıl belirlendiği önemlidir. Bir bağlantıda kullanılacak sıra numarası bilgisi için bir başlangıç değeri belirlenir, daha sonra da her veri aktarımında bu değer, aktarılan veri miktarı kadar arttırılır. Başlangıç değeri belirlenirkende, bu işlem için kullanılan, 32 bitlik sayıcıdaki değer kullanılır. Bu sayıcıdaki değer, belirli aralıklarla ve belli durumlarda sayıcı değeri artış algoritmasınca belirlenecektir. Bu değerin tahmin edilmesi mümkündür. Örneğimizde A bilgisayarı B sunucusunun verdiği cevaptaki bağlantı sıra numarası değerini tahmin edebilir. Daha önce A bilgisayarı B sunucusuna bir bağlantı gerçekleştirerek verdiği sıra numarası hakkında bilgi sahibi olur ve daha sonra yapmaya çalıştığı illegal bağlantıda bu sayıyı tahmin edebilir. Kullanılacak sıra numarası tahmin edildiği anda bağlantı onaylanacak ve saldırı için gerekli zemin teşkil edilmiş olur.

Bir başka problemde B sunucusunun C bilgisayarına SYN(SN_B) ACK(SN_A+1) paketi göndermesi durumunda oluşur. C sunucusu böyle bir bağlantı talep etmediğini bildiren RESET paketini B sunucusuna göndererek bağlantıyı sonlandırmasını sağlayabilir. Bu durumda ya C'nin kapalı olduğu durumlar gözlenecek ya da C'nin gönderdiği paketin ilişkili olduğu servise ait kuyruk doldurularak bu paketin işlem görmesi engellenecektir.

Bu tip saldırılar karşısında yapılabilecek en iyi çözüm sıra numarası tahminini zorlaştırmaktır. Bu değerin belirlenmesinde kullanılan algoritmanın, sayıcının sıklığını saniyede 250000 defa değiştirmesi TCP tanımlamalarında belirtilen değerdir. Diğer bir yöntemde algoritmanın karmaşıklığının arttırılmasıda rasgeleliği arttırıcak ve tahmin edilmesini zorlaştıracaktır.

Firewall açısından yapılabilecek önlemler ise, IP adresi yanıltma yoluyla gelen saldırıların engellenmesidir. Çünkü bu saldırıya temel IP adresi yanıltmasıdır. Saldırıdaki diğer unsur sıra numarasının tahmin edilmesidir fakat buna firewall düzeyinde bir çözüm getirilemez.

3.8.5. Fragmentation saldırıları

IP paketlerinin büyüklüğü konusunda bir üst sınır mevcuttur. Bu da IP paket başlığında yer alan, 16 bitlik toplam uzunluk alanı ile belirlenir. 16 bitlik alanın sahip olabileceği en büyük değer 65536'dır. Buna IP paketi başlık ve veri alanları dahildir. Paketler taşınırken kullanılan ortamın bir defada taşıyabileceği paket boyutuda önemlidir. MTU ile tabir edilen (Maximim Transmission Unit) belli boyutlara uyulmak zorundadır. MTU değeri FDDI ortamları için 4000 bayt iken Ethernet ortamı için 1518 bayttır.

Mesala FDDI bir ağ 4000 baytlık paketler gönderebilirken buraya bağlı olan bir Ethernet ağ ise ancak 1518 bayt olarak bu paketleri alabilir. Bu durumda büyük gelen paket bölünür (fragmentation). Ayrıca ağ üzerinde her iletilmek istenen veride ortam boyutlarında olmaya bilir, bu takdirde bilgisayarca ağa çıkarılacak TCP segmentleride, parçalara ayrılarak IP paketlerine yerleştirilecektir. Bölünen paketler Fragment Header denilen bir yapı ile tanınırlar. Bu şekilde parçalarına ayrılan IP paketleri hedefe kadar bir araya getirilmez. Parçaların her birinde IP başlığı yer alırken, TCP başlığı sadece ilk pakette yer alacaktır.

Bu nedenle TCP başlığında yer alan bilgileri kullanarak filtreleme yapan bir firewall, ilk parça dışındakiler üzerinde değerlendirme yapamayacaktır. Bu duruma ilişkin genel yaklaşım olarak, ilk parça için karar verme mekanizması yürütülürken diğer parçaların geçişine müsade edilir. Bu durum güvenlik açısında problem teşkil etmeyecektir. Çünkü parçalardan birinin eksik olması durumunda TCP segmenti oluşturulamayacaktır. Parçalar hedefe ulaştıkça bir süre bekletilecek ve zaman aşım süresi içinde tüm parçalar ulaşmazsa diğerleride yok edilecektir.

Yalnız saldırı amacı taşıyan paketler oluşturulamasa da sistem kaynakarının doldurulması ile hizmet aksatmaları meydana getirilebilir. Yoğun olarak hedefe gönderilen büyük paketler olduğunu düşünelim. İlk parça (TCP başlığı içeren) firewall ile engellensede diğer parçalar hedefe ulaşacak ve parçaların ulaşımının tamamlanması için belirlenen zaman aşım süresi sonuna kadar sistem kaynakarını

işgal edecektir. Bu şekilde sıklıkla gönderilen, parçalara ayrılmış TCP paketleri hizmet sunumuna engel oluşturacaktır.

Bu problem karşısında ürerilebilecek çözüm; firewall'lar üzerinde parçalar halinde iletilen TCP segmentleri için durum tabloları oluşturmaktır. TCP paketinin ilk parçasını taşıyan IP paketi üzerinde yapılan karşılaştırma gereği paket geçişinin mümkün olup olmadığı kontrol edilecek ve eğer geçiş izni varsa firewall üzerinde bir durum tablosu oluşturulacaktır. Bu doğrultuda TCP segment parçalarını içeren diğer IP paketlerininde geçişine bu durum tablolarının kontrolü ile karar verilir. Durum tablosunda yer almayan parçalara ait IP paketleri geçirilmez.

Durum tabloları içinde belirli kıstaslar belirlenerek firewall için sorun oluşturması engellenmelidir. Çünkü durum tablolarının fazlaca büyümesi kaynaklarının işgal edilmesine ve firewall'lar için istenmeyen aksaklıklara sebep verecektir. Bu nedenle durum tabloları oluşturulurken ya tablodaki eleman sayısı için bir sınır getirilir ya da tablo elemanları için bir zaman aşım süresi tespit edilir. Durum tablosu için eleman sınırı konulması durumunda, tablo dolunca ilk eklenen, parçalı geçiş yapan IP paketi bilgileri çıkarışır. Zaman aşımıda aynı şekilde belirli süre bitiminde hala geçişini tamamlayamamış parçalı geçiş yapan IP paket bilgilerinin çıkartılamasını sağlar. Bu şekilde tablonun illagal büyümesi ve sistem kaynaklarını işgal etmesi önlenmiş olacaktır.

Durum tablosu ile karar verilmesinin bir dezavantajıda vardır ki; ilk parçası kabul edilmemiş parçalar kabul edilmez. İlk parçadan önce firewall'a ulaşan diğer parçalara ait IP paketleri geçirilmediği için tekrar bu paketlerin istenmesi gerekecektir. Ama güvenlik önceliği düşünüldüğünde çok nadirde olsa gerçekleşen bu durumlar önemsenmez.

Bu durumu Ipchains açısından ele alalım. Sadece ilk parça Ipchains tarafından kontrole tabi tutulabilir. Diğer paketler kontrole tabi tutulamazlar. Bunu sağlamak için Ipchains için -f parametresi kullanılmalıdır. Yalnız bu durumda bölünmüş olan paketin ikinci veya daha sonraki parçaları anlaşılabilir. Port adresi ise okunamaz. Bu sebeple de bu parametre ile port numarası kullanılmaz.

Özellikle ICMP paketleri kullanan exploitler için iyi bir savunma mekanizmasi oluşturulabilir. ICMP paketleri bölünmeyecek (fragment) kadar küçük paketlerdir, (ayrıca bölünmeyi üzerlerindeki "don't fragment" biti de engeller). Yani bölünmüş bu tür paketleri reddetmek yararlıdır.

ipchains -A -p icmp input -s x.x.x.x/x -d y.y.y.y/y -f -j DENY

3.8.6. "smurf" saldırıları

Yalnış kaynak adres bildirimi içeren ICMP paketleriyle gerçekleştirilir. ICMP mesaj tiplerinden "echo request" kullanılarak hedef bilgisayarın kilitlenmesine, saldırıda ara hedef pozisyonundaki ağın trafiğinde oluşturulan aşırı yüklenmeyle performansının düşürülmesi sağlayan bir saldırı çeşitidir. Adını bu tip saldırıları gerçekleştirmek için kullanılan bir programdan almıştır.

ICMP mesaj tipi olarak seçilen "echo request", hedef sistemin hizmet verip vermediğini anlamakta kullanılan bir mesaj tipidir. Hedef sistem eğer faal ise "echo reply" mesaj tipi ile hizmet vermekte olduğunu kendisini sorgulayan, kaynak adresin sahibi sisteme iletir.

Saldırıyı yapacak sistem, kendisine asıl hedef olarak şeçtiği sisteme ait IP adresini kaynak adres olarak kullanarak, ara hedef pozisyonundaki ağa ICMP mesajlarını yayın adresi kullanarak gönderir. Yayın adresi kullanıldığı için ara hedef ağdaki tüm sistemler bu paketleri alacak ve "echo request" sorgusuna cevap olarak "echo reply" paketlerini kaynak adreste yazılı sisteme yani saldırıya hedef olan sisteme göndereceklerdir. Bu durumda neler olacaktır;

Ara hedef ağ dahilinde bulunana her sistem alıdğı ICMP sorgusuna cevap oluşturacağı için ara hedef ağda istenmeyen yüksek bir ağ trafiği gözlenecektir. Üstelik ICMP paketlerinin boylarıda büyütülürse ağ performansı çok daha kötü etkilenecektir.

Hedef sistemki, saldırgan tarafından asıl zarar görmesi istenen pozisyonundadır. Ara hedef üzerinden kendisine gelen yüksek paket trafiği karşısında sistem kaynaklarının aşırı yüklenmesi ve bu nedenle hizmet veremez duruma gelmesi söz konusudur. Hatta bazı durumlarda kilitlenmelerde yaşanabilir. Ayrıca hedefin dahil olduğu ağda, yüksek trafikten olumsuz etkilenecektir. Eğer, ara hedef olarak kullanılacak ağ sayısı, birden fazla seçilirse ortaya çıkacak performan kaybı daha da yüksek olacaktır.

Saldırının oluşturduğu yoğun trafik, ara hedef ağ, hedef sistem ve hedefin dahil olduğu ağ üzerinde yoğun trafik oluştruduğu gibi ağları birbirine bağlayan ISP'ler veya diğer kuruluşlar içinde yoğun band genişliğini kullanımı gerçekleşmiş olacak ve performans düşecektir.

Bu saldırı karşısında ne gibi önlemler alınabileceğini; ara hedef ağ, hedef ağ ve saldırı için kaynak teşkil eden ağ açısından ele alalım.

Ara hedef ağ için bu tip bir saldırıdan zarar görülmemesi Internet gibi harici ağlar üzerinden gelen yayın adresli IP paketleri filtreleyerek yapılabilir. Böyle bir kısıtlama çoğu durumda bir soruna yol açmaz. Saldırganlar ara hedef ağ durumundaki ağa sızarak buradaki bir bilgisayar üzerindende yayın yoluyla ICMP paketleri yayabilir. Bu durum itibariyle paket filtreleme aşıldığı için bir çözüm olmayacaktır. Bu nedenle ağda yer alan bilgisayarların yayın ICMP isteklerini cevaplamaması sağlanmalıdır.

Ara hedef ağa gelen yayın ICMP paketlerinin Ipchains kullanılarak kısıtlanması:
#ipchains –A input -i ethX –p icmp –s 0/0 –d 213.238.128.255/24 –j DENY

Ara hedef ağımız 213.238.128.0 ve bu ağ için yayın adresimiz 213.238.128.255'tir. Bu ağa, firewall'umuzun ethX ağ arayüzü üzerinden gelecek, harici ağ ICMP isteklerini engelliyoruz.

Hedef ağa yoğun olarak gelen ICMP "echo reply" cevapları paket filtrelemeyle aynı ara hedef ağ üzerinde olduğu gibi engellenebilir. Fakat hedef ağa gelen paketler, hedef ağın harici ağ bağlantı noktasında kadar geleceği için, hedef ağın harici ağ bağlantılarında performas düşüklüğüne sebebiyet verecektir. Buna karşı yapılabilecek

herhangi bir engelleme maalesef yoktur. Ancak ara hedef ağ yetkililerine durum bildirilebilir.

Bu tür saldırıların kaynaklanmasının önüne geçebilmek için, saldırıya kaynaklık eden ağ üzerinden IP adresi yanıltması yoluyla gönderilecek paketler engellenmelidir. Bu da paket filtreleme ile engellenebilir.

3.8.7 UDP portlarını kullanan saldırılar

Smurf saldırısındaki ICMP paketleri yerine, UDP paketleri kullanılarak gerçekleştirilen saldırılardır. Bir sistem üzerinde ya da birkaç sistem arasında, UDP portlarına yöneltilecek yoğun paket akışıyla gerçekleştirilir.

Bir birleriyle iletişimde olan iki UDP servisinden birisi veya her ikiside üreteceği paketler ile karşısındaki servisin aşırı yüklenmesine ve servisin hizmet veremez duruma gelmesine neden olabilir. Buna neden ise UDP protokolünün yapısıdır. UDP bağlantısız bir iletişim protokolüdür ve karşılıklı kontol bilgilerinin değiştirilip, el şıkışılmasını gerektirmez.

UDP "echo" servisi (port 7), UDP "chargen" (port 19) servisleri bu saldırılarda en çok kullanılan servislerdir. "echo" servisini çalıştıran bir sistem, iletişim kurduğu bilgisayardan aldığı verileri aynen geriye gönderir. "chargen" servisinde ise iletişim içinde olduğu bilgisayardan aldığı her paket için, rasgele sayıdaki karakterlerden oluşan paketi geri gönderir.

Bu iki servise ilişkin UDP portlarının aynı bilgisayar ya da farklı iki bilgisayar arasında bir birine bağlanması sonu gelmeyecek trafik akışı oluşturur. Bu servisi veren bilgisayar/bilgisayarlar ve trafiğin aktığı ağ üzerinde performans düşecektir.

Bu saldırı karşısında ilk yapılması gereken kullanılmayan ya da gereksiz UDP servislerinin sistemler üzerinde kapatılması olmalıdır. Böylece bu tür saldırılara meydan verilmemiş olur. Ama bazı gerekli servisler (DNS servisi) olabilir ki, bu durumda paket filtreleme kuralları ile bu servislere erişim kısıtlanmalıdır. Ayrıca ağın,

bu durumlara karşı saldırı tespit sistemleri ile dinlenmesi ve anormalliklerinin tespit edilmeside faydalı bir çözüm olur. Bu saldırılarda, IP adresi yanıltması yöntemide kullanılacağından bu tip paketlerin geçişi firewall'lar üzerinde filtrelenmelidir.

UDP "echo" ve "chargen" servislerinin Ipchains kullanılarak kısıtlanması:

```
#ipchains –A input  -i ethX –p udp –s 0/0 –d 213.238.128.0/24  7  –j DENY
#ipchains –A input  -i ethX –p udp –s 0/0 –d 213.238.128.0/24 19 –j DENY
#ipchains –A input  -i ethY –p udp –s 213.238.128.0/24 7    -d 0/0 –j DENY
#ipchains –A input  -i ethY –p udp –s 213.238.128.0/24 19   -d 0/0 –j DENY
```

Ağımıza ait IP adresi 213.238.128.0 ve bu ağ için kullandığımız firewall'umuzun ethX ağ arayüzü harici ağa , ethY ağ arayüzü yerel ağımıza bağlıdır. İlk iki satırımızda harici ağ üzerinden gelen UDP "echo" ve "chargen" hedefli paketler, daha sonraki iki satırımızda ise yerel ağdan harici ağa giden UDP "echo" ve "chargen" kaynaklı paketler engellenmektedir.

Paket filtrelemenin, bu saldırıların aynı ağ üzerinde yer alan iki bilgisayarın servislerini birbirine bağlayarak ya da aynı bilgisayar üzerinde bu servislerin bir birine bağlanmasıyla yapılmasına karşı bir çözüm olmayacağı unutulmamalıdır.

3.8.8. NFS saldırıları

NFS, ağ üzerindeki bilgisayarların dosya sistemlerini paylaşmasını, sanki yerel bir sürücüsü gibi kullanmasına imkan tanıyan bir servistir. Ancak bu protokol üzerindeki bazı açıklar nedeniyle saldırılara hedef olmakta. Hatta uzak bilgisayardan erişen kullanıcının sunucu üzerinde süper kullanıcı yetkisiyle işlem yapması dahi mümkün olmaktadır.

Firewall çözümleriyle bu servise harici ağdan ya da Internet üzerinden erişim kısıtlanabilir. Fakat yerel ağ üzerinden gelebilecek saldırılara karşı bu bir çözüm oluşturmaz. Firewall çözümlerinin yanı sıra, NFS servisinin son güncellemeleri takip edilmeli, son güvenli uygulamaları kullanılmalıdır.

NFS, RPC servisi üzerinden hizmet vermektedir. Bu nedenle firewall ile portmapper (111 ve 2049 nolu portlar) erişimleri üzerinde getirilecek kısıtlama ile NFS servisine harici ağ üzerinden erişimler engellenir.

```
#ipchains –A input  -i ethX –p tcp –s 0/0 –d 213.238.128.0/24 111 –j DENY
#ipchains –A input  -i ethX –p udp –s 0/0 –d 213.238.128.0/24 2049 –j DENY
#ipchains –A input  -i ethX –p tcp –s 0/0 –d 213.238.128.0/24 111 –j DENY
#ipchains –A input  -i ethX –p udp –s 0/0 -d 213.238.128.0/24 2049 –j DENY
```

Ağımıza ait IP adresi 213.238.128.0 ve bu ağ için kullandığımız firewall'umuzun ethX ağ arayüzü harici ağa bağlıdır.

3.8.9. Diğer saldırılar

Bunlar dışında da saldırı çeşitleri mevcuttur. Fakat bu saldırılara karşı firewall teknelojileri ile güvenlik tesis edilmesi çok zor ya da olanaksızdır. Çünkü diğer güvelik açıkları sunucu servislerinden, işletim sistemlerinden, uygulama eksikliklerinden kaynaklanmaktadır.

Mesela bir servisin üzerinde bellek taşması oluşturacak bir veri iletimi yapılması, farklı komutların yürütülmesine ve servisin sahip olduğu kullanıcı yetkileri doğrultusunda sunucu üzerinde işlemler gerçekleştirilebilmesine olanak tanıyacaktır (Talkd). Linux çekirdeğinin 2.0.36 sürümünden öncesinde, TCP/IP yığınının gerçekleme sorunu nedeniyle, TCP sıra numarası tahmin etmeden bile saldırların gerçekleştirilmesi mümkün olmaktaydı. Microsoft ISS'in script açıkları nedeniyle, Internet browser aracılığıyla sunucu üzerinde sanki konsol üzerinden komut çalıştırır gibi işlemler yapılabiliyordu. İşte bu tür güvenlik açıklarına karşı uygulama ve işletim sistemlerinin son sürümleri ve açıklarını gideren yamaları takip edilerek güncellenmelidir. Nispeten bu sayede daha güvenli bir sistem elde etmek mümkün olacaktır.

BÖLÜM 4. NESSUS ZAYIFLIK TARAMA SİSTEMİ

Nessus, Nisan 1998'de uzak sistem zayıflık tarama sınıfı bir yazılım olarak geliştirilmiştir. Halen gelişimini sürdüren yazılım için 2.0.5 sürümü mevcuttur ve uygulamada bu sürüm üzerinde gerçekleştirilmiştir. Nessus, açık kaynak kodlu, serbestçe dağıtılabilir ve kolayca kullanılabilir bir yazılımdır.

4.1. Nessus Çalışma Mimarisi ve Ağ Yerleşimi

Nessus istemci sunucu mimarisini kullanır. Bu sayede aynı anda birden çok oturumun yürütülmesine izin verir. İstemci ve sunucu parçalarının farklı platformlarda bulunmasını destekler. İstemci bölümü Windows, Java, Unix tabanlı platformlarda çalışabilirken, sunucu bölümü ise Unix tabanlı platformlarda çalışabilmektedir.

Nessus sunucusu, bulunduğu ağ dahilinde ya da bulunuduğu ağ üzerinden erişebileceği sistemler üzerinde tanımlanmış erişim yetkileri kıstasında zayıflık sorgulamaları gerçekleştirir. Zayıflık sorgulamalarının gerçekleştirilme özellikleri, kulanılacak eklentiler, eklenti bilgileri istemci oturumları ile belirlenir.

Nessus sunucusu, kullanıcı tanımlama, kullanıcılara farklı yetkiler belirleme, kullanıcıların sunucuya bağlanabileceği ağ parçasını sınırlama gibi özelliklere de sahiptir. Bu tanımlamalar her kullanıcı için oluşturulan, kullanıcı kural tablolarında belirtilir. Örneğin;

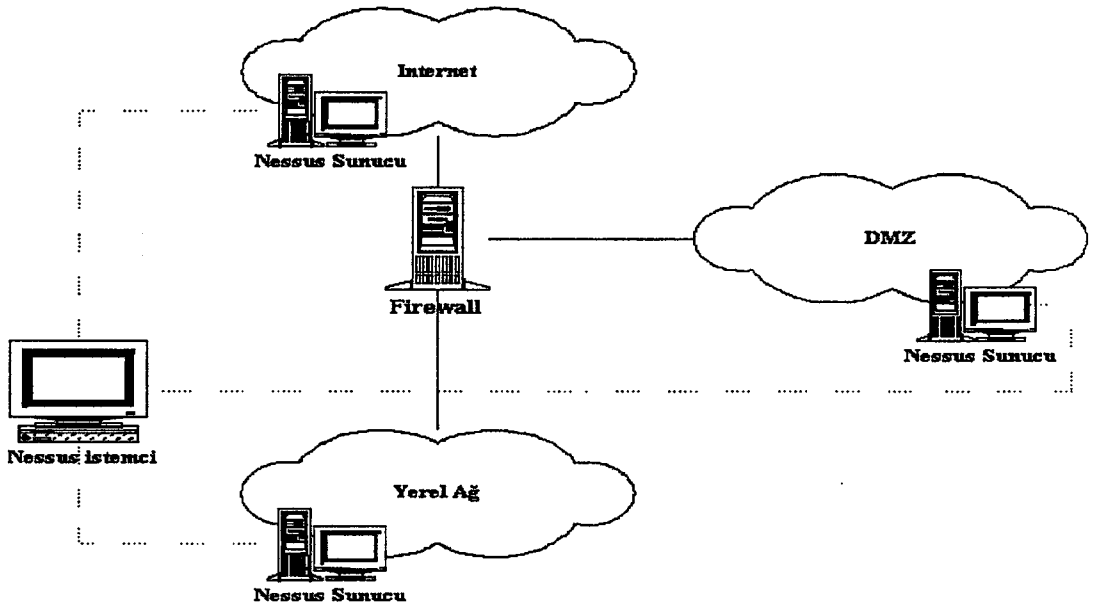
Accept 10.21.11.0/24
Acccet 192.168..5.0/24
Default deny

Bu kullanıcı kural tablosuna bakılarak 10.21.11.0/24 ve 192.168.5.0/24 altağları için kullanıcının zayıflık tarama yetkisi olduğu diğer ağlar içinse yetkiye sahip olmadığı görülür.

Tek bir istemci ile farklı noktalardaki Nessus sunucularına bağlanılarak daha güvenli ve gerçekçi taramalar yapılabilir. Örneğin, yerel ağın DMZ'ye erişimi kısıtlanmıştır. Yerel ağ dahilindeki bir Nessus sunucu ile tarama yapılırken bu kısıtlara bağlı kalınır. Bu nedenle taramanın eksik sonuçlar üreteceği aşikardır. Oysa DMZ bölümündeki bir Nessus sunucusu taramaları hiçbir kısıta bağlı kalmaksızın, gerçekçi ve hızlı bir şekilde tamamlar. Bunun için gereken istemcinin DMZ bölgesindeki sunucuya erişimi için gerekli portuna (1241) izin verilmesidir. Taramalarda güvenilirlik ve gerçekçilik esas olduğuna göre, sistemin değişik erişim kısıtlamalarına sahip bölümlerine, sadece bu bölümü tarayacak sunucular yerleştirmek gerekir. Şekil 4.1'de bir istemci üzerinden değişik konumlardaki sunuculara yapılan bağlantılar gösterilmiştir.

Nessus aynı anda birçok sistemin güvenlik denetimini paralel olarak yürütebilir. Denetimi gerçekleştirilen her sistem için sunucu sistem belleğinde 2 MB'lık bir alan kullanılmaktadır. Ayrıca paralel tarama sayısı sunucu üzerinde kısıtlanabilir.

Daha önceden yapılmış tarama sonuçlarını sunucu üzerinde kayıt edebilme, tarama esnasında durdurarak kayıt edebilme, taramalar arasındaki farkları analiz edebilme ve artımlı taramalar yapabilme imkanları mevcuttur. Artımlı tarama özelliğiyle, daha önceden yapılmış zayıflık sorgulamalarının uygulanmaması, sadece yapılmamış sorgulamaların uygulanması sağlanır.



Şekil 4.1 Ağın farklı bölümlerinde Nessus sunucularının kullanımın şeması

4.2. Nessus Zayıflık Veritabanı

Nessus'da diğer zayıflık tarama sistemleri gibi güvenlik denetimlerini bir zayıflık veritabanı sayesinde gerçekleştirir. Zayıflık veritabanı Nessus'a ait Internet sitesi "http://www.nessus.org/scripts.html" adresinden indirilerek veya "nessus-update-plugins" yardımcı programı ile Nessus sunucusu üzerinden otomatik olarak güncellenebilir.

Zayıflık veritabanı NASL (Nessus Attack Scripting Language) ve C dili kullanılarak oluşturulmuştur. NASL Nessus zayıflık tanımlamaları için geliştirilmiş özel bir script dilidir. C dili ile hazılanan zayıflık tanımlamaları içinse C-NASL dönüştürücüsü mevcuttur. Yazılmış tanımlamaların %99 NASL ile gerçekleştirilmiştir. Nessus kullanıcıları ya da katkı vermek isteyen herkes, tespit ettiği ya da çeşitli kaynaklarca belirlenen güvenlik açıklıklarını NASL ile sorgulanabilir birer eklenti haline getirebilir ve zayıflık veritabanına ekleyebilir.

Nessus zayıflık veritabanındaki zayıflık eklentileri 24 başlık altında toplanmıştır. Yazılmış tüm ekler bu başlıklardan birine dahil edilmiştir. Yazılacak yeni eklenti

içeriklerinin bu başlıklarda sınıflandırılmış eklentiler haricinde, farklılıklar içermesi durumunda yeni başlıkların oluşturulması mümkündür. Nessus 2.0.5 sürümüne kadar oluşturulmuş başlıklar:

Arka kapılar (Backdoors) : VNC, Radmin gibi uzaktan yönetim hizmeti sunan programlar ve çeşitli trojan yazılımlarının varlığını tespit etmek amacıyla kullanılan ekler bu başlığa dahil edilmştir.

CGI kötüye kullanımları (CGI abuses) : CGI scriptlerini çalıştıran sunucularda, script dilinin veya script yorumlayıcısının işletim hatalarını tespit eden eklentiler grubudur.

CISCO : CISCO ağ cihazlarının kullandığı IOS yazılımlarının değişik paketler karşısındaki zafiyetlerini test eden eklentileri içermektedir.

Geçerli Unix hesapları (Default Unix Accounts) : Unix tabanlı sistemler üzerinde varsayılan olarak gelen, belirli imtiyazlarla donatılmış hesapların varlığını ve erişim durumlarını test eden eklentileri sunmaktadır.

Hizmet engelleme (Denial of Service) : Sunucular üzerinde, hizmet veren uygulamalara erişimleri engelleyebilecek açıklıkları test eden eklentileri içerir.

Finger kötüye kullanımları (Finger abuses) : Uzak sistemde yer alan eski finger servislerinin sebep olduğu kötüye kulanım açıklıklarının sorgulanmasını gerçekleştiren eklentiler.

Firewalls : Ateş duvarı yazılımlarının eksiklikleri ya da yalnış konfigürasyonlarının doğuracağı zayıflıkları değerlendiren eklentiler gurubudur.

FTP (Dosya Transfer Protokolü) : FTP sunucu hizmeti zaaflarının oluşturduğu zayıflıklara karşı geliştirilmiş eklentiler.

Uzaktan bir kabuğa ulaşmak (Gain a shell remotely) : Kabuk düzeyinde çalışan uygulamaları test etmek ve zaaflarının saldırganlar açısından değerlendirilebilir olup olmadığını sorgulayan eklentiler.

Kaynakları uzaktan elde etmek (Gain root remotely) : Hizmet sunan uygulamalara uzun parametreler veya özel biçimlere sahip dosyalar gönderilmesi ile sunucunun üzerinde sahip olduğu kaynak ve hakların kötüye kullanılabilirliğini tespit etmek için geliştirilen testler bulunur.

Genel (General) : Sistemler üzerindeki mevcut istemci yazılımlar, uzak sunucu yönetim araçları ve yerel ağ yönetim araçlarının oluşturduğu zayıflıklar hakkında bilgi toplayan eklentiler.

Misc (Yalnışlar) : Yalnış ya da eksik bırakılmış konfigürasyonlar, verilmemiş şifreler, herkesin kulanımına açık kalmış kaynakların sistemde oluşturacağı zatıflıkları sorgulayan eklentiler ailesidir.

Netware (Netware İşletim sistemi) : Netware tabanlı sunucular üzerinde test edilebilecek zayıflıklar bu eklenti ailesine dahil edilmektedir. Netware sunucu üzerinde çalışabilen uygulamalar ve OS'un yapısından kaynaklanan zayıflık taramaları.

NIS (Ağ Bilgi Sistemi) : NIS RPC servisinin yapısından kaynaklanan açıklıkların sistem üzerinde erişilebilirliğinin testi için geliştirilmiş eklenti grubudur.

Port tarayıcıları (Port scanners) : Uzak sistem üzerindeki portların taranması işlemiyle ilgilenen eklentiler mevcuttur.

Uzak dosya erişimi (Remote file access) : Dosya sistemine erişimi sağlayan servislerin yetkilendirme problemleri, dosya sistemi üzerinde çalışan sunucu yazılımlarının sahip olduğu zayıflıkların testini gerçekleştiren eklentiler yer alır.

RPC (Uzak Yöntem Çağrıları) : RPC üzerinden hizmet vermekte olan güvenlik zayıflığı içeren servislerin varlığını tespit etmek üzere geliştirilen eklenti ailesidir.

Ayarlar (Settings) : SMTP, HTTP, SMB gibi protokolleri kullanan uygulama ayarlarında yapılan yapılandırma eksikliklerinin uzak sistem üzerinde neden olabileceği zayıflık düzeylerini sorgulayan eklentiler gurubudur.

SMTP problemleri (SMTP problems) : SMTP servisinin kullanan uygulamaların yani e-mail sunucu yazılımlarının SMTP protokol sürecini yürütürken düşebilecekleri hataları test eden eklenti ailesidir.

SNMP (SNMP) : SNMP servisi vasıtasıyla uzak sistemin tipi, üzerindeki mevcut kullanıcı sorgulaması veya çalışan süreç bilgilerinin elde edilmesinin mümkünlüğünü rapor eden eklentiler yer alır.

Denenmemiş (Untested) : Web, Ftp ve benzer türde hizmet sunan uygulamaların, sahip olduğu eksiklikler ya da çalışma modu düzeylerinin sebep olabileceği zayıflıkların, sistem üzerinde meydana getirebileceği açıkları sorgulayan eklenti ailesidir.

Yararsız hizmetler (Useless services) : Yarasız olarak nitelendirilen bu servisler, genellikle uzaktan erşim ve denetim imkanları sunan ağ uygulamalarıdır. Bu tür servislerin uzak sistem üzerinde oluşturduğu zayıflık derecelerini rapor edebilen eklentiler bu gurubun üyesidir.

Windows (Windows işletim sistemi) : Windows tabanlı bir sistem üzerinde test edilebilecek zayıflıklar bu eklenti ailesinde yer almaktadır. Windows OS ve üzerinde çalışan uygulamaların kullandığı iletişim yapısı, versiyonu gibi çeşitli bilgiler ışığında açıklanmış zayıflıkların varlığı hakkında bilgi toplarlar.

Windows kullanıcı yönetimi (Windows user management) : Uzak sistemin Windows ortamı olması durumunda uygulanabilirliği mümkün olan eklenti ailesidir. Windows OS kullanan sistemler üzerinde oluşturulmuş kullanıcı hesapları üzerinde sorgulama

yapmak maksadıyla geliştirilmiş eklerdir. Kullanıcının dahil olduğu gruplar ve kullanıcı statüleri hakkında bilgiler döndürerek zayıflık taramasına kullanıcı hesabı boyutunu kazandırır.

4.3. Nessus Raporlama Sistemi

Nessus raporlamada html, xml, latex, metin ve kendine özgü nsr formatlarında raporlar üretebilmektedir. Rapor içeriklerinde, bulunan zayıflığın risk derecesi, tanımlaması, CVE (genel zayıflıklar ve korunamayanlar) ve diğer referanslar ayrıntılı olarak bulunmaktadır. Ayrıca zayıflığın nasıl giderilebileceği konusunda bilgiler de bu raporlarda yer almaktadır.

Raporlara, Nessus'un harici olarak kullanabildiği port tarama yazılımlarının ürettiği çıktılarda eklenebilmektedir. Böylece port tarama yazılımlarını tekrar çalıştırmamak ve daha önceden yapılan tarama raporlarını dahil etmek hız ve performans artışı sağlayacaktır.

4.4. Nessus Zayıflık Tarama Uygulaması

Zayıflık tarama yazılımları ile geniş bir ağda, çok kısa süre zarfında güvenlik sorgulamaları yapılabileceğine değinilmişti. Bu aşamada Şekil 4.2'de görülen değişik platformlara sahip geniş bir yerel ağ üzerinde, Nessus 2.0.5 sunucu uygulaması ile zayıflık taraması gerçekleştirilmiş ve üretilen zayıflık raporlarına değinilmiştir. Bu tarama için Nessus'un zayıflık veritabanına dahil edilmiş 1606 eklenti kullanılmıştır.

Nessus sunucu uygulaması Linux Redhat 8.0 üzerine aşağıda belirtilen özellikler dahilinde konfigüre edilmiştir.

Nessus Libraries için :
configure –enable-cipher
make ; make install
--enable-cipher : İstemci ile sunucu arasındaki trafiğin şifrelenmesini sağlar.

Libnasl için :

configure

make ; make install

Nessus-Core için :

configure –enable-syslog –enable-gtk –enable-save-sessions –enable-save-kb

make ; make install

--enable-syslog : Syslog'a bağlantı ile ilgili kayıtların gönderilmesini aktif kılar.

--enable-gtk : İstemci için GTK arayüzünün derlenme seçeneğini aktif hale getirir.

--enable-save-sessions : Yapılan güvenlik taramalarının, sunucu üzerinde kayıt edilme özelliğinin aktif olması için kullanılır.

--enable-save-kb : Yapılan ve kayıt edilen güvenlik taramaları üzerinde karşılaştırma yapılabilmesi, artımlı taramalar yapılabilmesi gibi seçeneklerin aktif olması amacıyla kullanılır.

Nessus-Plugins için :

configure --with-fetchcmd=wget

make ; make install

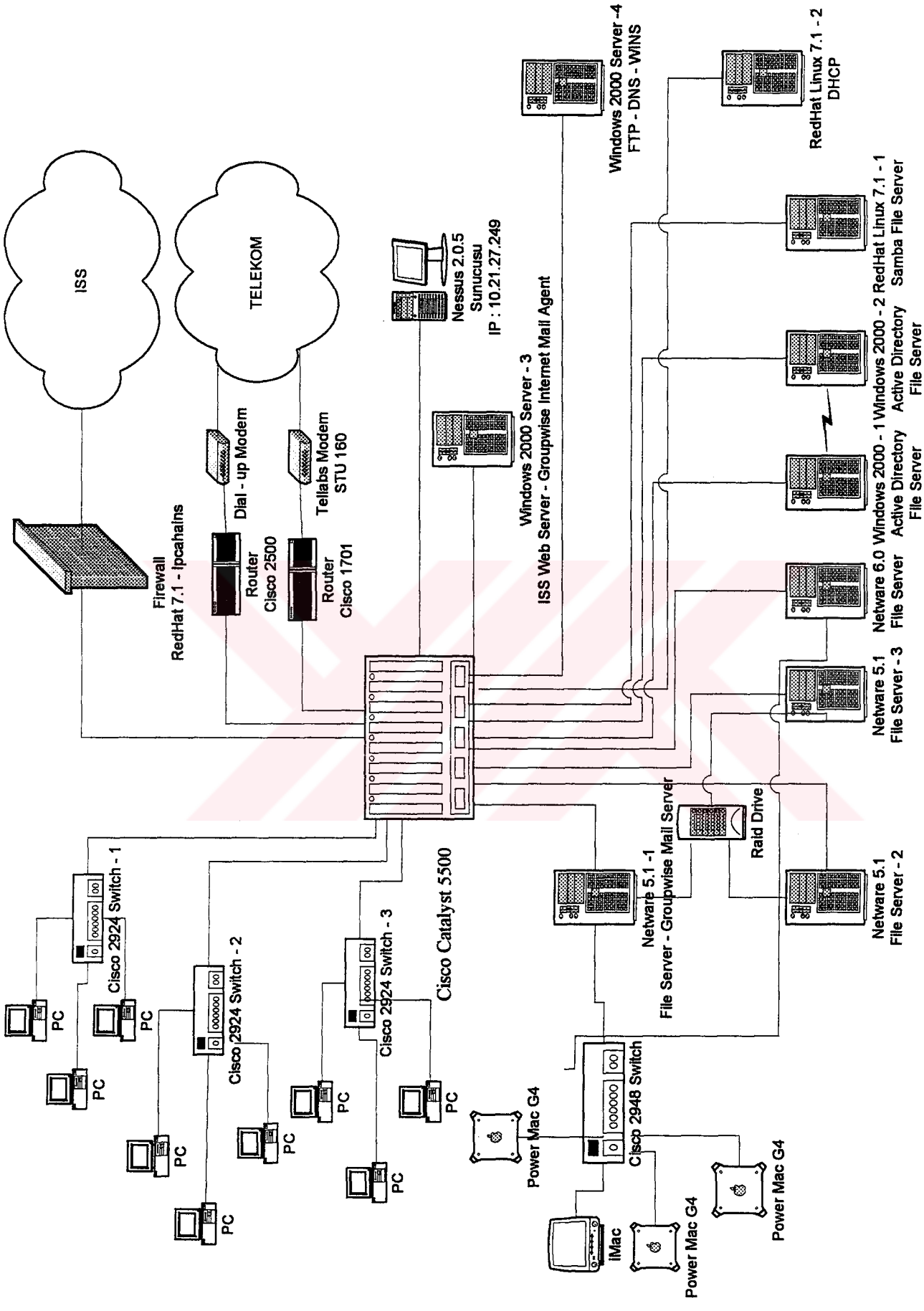
--enable-install=user : *"root"* olmayan bir kullanıcının, hangi taramaları yapabileceğinin izlenmesi için kullanılabilir.

--with-fetchcmd=[wget|lynx|<cmd>] : Zayıflık tanımlamaları ve diğer yazılımların bütünleşik kullanılması için kullanılan eklentilerin, indirilme yönteminin belirlenmesi için kullanılır. Varsayılan yöntem önce wget, eğer bulunamaz ise lynx ve diğer belirtilecek farklı araçlar kullanılabilir.

Nessus sunucusunun zayıflık tarama oturumlarının başlatılması, yönetilmesi ve raporlamanın alınması için kullanılan Nessus istemci yazılımı ise Window platformlarında çalışan Nessus WX 1.4.4'tür. Nessus istemcisinin uygulama için kullanığı ayarlar Tablo 4.1'de verilmiştir.

Tablo 4.1 Nessus istemcisinin tarama ayarları

max_hosts	128
max_checks	128
log_whole_attack	yes
cgi_path	/cgi-bin
port_range	1-1024
optimize_test	yes
language	english
checks_read_timeout	5
non_simult_ports	139, 445
plugins_timeout	320
safe_checks	yes
auto_enable_dependencies	yes
use_mac_addr	no
save_knowledge_base	yes
kb_restore	no
only_test_hosts_whose_kb_we_dont_have	no
only_test_hosts_whose_kb_we_have	no
kb_dont_replay_scanners	no
kb_dont_replay_info_gathering	no
kb_dont_replay_attacks	no
kb_dont_replay_denials	no
kb_max_age	864000
plugin_upload	no
plugin_upload_suffixes	.nasl, .inc
admin_user	root
ntp_save_sessions	yes
ntp_detached_sessions	yes
server_info_nessusd_version	2.0.4
server_info_libnasl_version	2.0.4
server_info_libnessus_version	2.0.4
server_info_thread_manager	fork
server_info_os	Linux
server_info_os_version	2.4.18-14smp
reverse_lookup	yes
ntp_keep_communication_alive	yes
ntp_opt_show_end	yes
save_session	yes
detached_scan	no
continuous_scan	no



Şekil 4.2 Nessus zayıflık taraması gerçekleştirilen bilgisayar ağı

Tablo 4.2 Aktif ağ cihazları alt ağ bağlantı eşleşimi

Alt Ağlar / ***Aktif Ağ Cihazları***	*10.21.10.0/24*	*10.21.11.0/24*	*10.21.13.0/24*	*10.21.27.0/24*	*10.21.160.0/24*
IBM Uyumlu PC	X IP: 10.21.10.1-200	X IP: 10.21.11.1-200	X IP: 10.21.13.1-200	-	-
Macintosh	-	-	-	-	X IP: 10.21.160.1-200
Netware5.1 - 1	X IP:10.21.10.241	X IP:10.21.11.235	-	-	-
Netware5.1 – 2	X IP:10.21.10.223	X IP:10.21.11.223	-	-	X IP:10.21.160.254
Netware5.1 - 3	X IP:10.21.10.227	X IP:10.21.11.221	-	-	X IP:10.21.160.253
Netware 6	X IP:10.21.10.229	X IP:10.21.11.229	-	-	-
Windows 2000 -1	X IP:10.21.10.225	X IP:10.21.11.226	-	-	-
Windows 2000 -2	X IP:10.21.10.205	-	-	-	-
Windows 2000 -3	X IP:10.21.10.233	-	-	X IP:10.21.27.200	-
Windows 2000 -4	-	X IP:10.21.11.239	-	-	-
RedHat Linux 7.1 – 1	-	-	-	X IP:10.21.27.236	-
RedHat Linux 7.1 – 2	-	-	-	X IP:10.21.27.230	-
Cisco Switch 2924 – 1	X IP:10.21.10.252	-	-	-	-
Cisco Switch 2924 – 2	-	X IP:10.21.11.252	-	-	-
Cisco Switch 2924 – 3	-	-	X IP:10.21.13.251	-	-
Cisco Switch 2948	-	-	-	-	X IP:10.21.160.252
Cisco Router 2500 – 1	-	X IP:10.21.11.238	-	-	-
Cisco Router 1701 – 2	-	X IP:10.21.11.241	-	-	-
Cisco Catalyst	X IP:10.21.10.254	X IP:10.21.11.254	X IP:10.21.13.254	X IP:10.21.27.254	-

Uygulama Sisteminin Genel Yapısı :

Nessus zayıflık taraması gerçekleştirilen Şekil 4.2'de izlenilen sistem, bir backbone switch merkezli, değişik sunucu ve istemci platformlarına sahip, VLAN (Virtual LAN) mantığını ile çalışan bir yapılanmaya sahiptir. Kiralık ve çevirmeli hatlar ile gerçekleştirilen WAN bağlantıları yönlendirici cihazlar üzerinden kotarılmaktadır. Internet bağlantısı ise çift arayüzlü bir konak firewall üzerinden ISS-backbone switch arasında tesis edilmiştir. Buna göre aktif ağ cihazlarının bağlı oldukları alt ağ bağlantı ve yerleşimleri Tablo 4.2 üzerinde görülmektedir.

Gerçekleştirilen zayıflık taraması yerel ağ üzerinden ve backbone switch'e doğrudan bağlı Nessus sunucusu ile yapılmıştır. Harici ağlardan ya da Internet üzerinden yapılacak zayıflık taraması ise farklı sonuçların üretilmesini sağlayacaktır. Çünkü bu sistemin harici ağ bağlantısı çift ağ arayüzlü konak türünden bir ateş duvarı ile denetim altında tutulmaktadır. Bunun yanı sıra ISS tarafında da saldırı tespit sistemleri ve ateş duvarlarıyla desteklenen güvenlik söz konusudur.

Ağdaki mevcut sunucular Windows, Linux ve Novell platformlarındadır. Sunulan hizmetler yerel ağa yönelik ya da harici ağa yönelik olarak gruplandırılabilir. Yerel ağ bazındaki kullanıcılara yönelik PC ve Macintosh istemcileri için dosya paylaşımı, Mail, Web, Ftp hzimetleri sunulurken, ağın çalışabilirliğinin sağlanması içinde WINS, DNS, DHCP türünde hizmetler sunulmaktadır. Bu hizmetlerin yanı sıra uzaktan erişim türündeki hizmetler ilede yönetimsel kolaylık sağlayan programlarda kullanılmaktadır.

Raporlama :

Zayıflık taraması her altağ için ayrı ayrı gerçekleştirilmiştir. Tablo 4.3'de Nessus'un ürettiği zayıflık raporlarında yer alan bilgilendirme alanları görülmektedir. Bu alanlarda zayıflık sorgulaması yapılan alt ağlar için üretilen kayıt sayısı ve üretilen kayıtların zayıflık bulguları doğrultusunda sıralandığı belirtilmiştir. Tüm altağlar için toplamda 6547 kayıt üretilmiş ve bunlardan 574 tanesi yüksek seviye, 2588 tanesi düşük seviye güvenlik açığını belirtirken 3385 tanede bilgilendirme mesajı içermektedir.

Tablo 4.3 Nessus zayıflık taraması raporları bilgilendirme alanları

a) 10.21.10.0 altağı zayıflık rapor bilgileri

Network Vulnerability Assessment Report **Sorted by vulnerabilities** **27.05.2003**		
Session name: LAN_10_Taraması	**Start Time:**	**27.05.2003 15:29:49**
	Finish Time:	**27.05.2003 20:37:28**
	Elapsed:	**0 day(s) 05:07:39**
Total records generated : 2428		
High severity : 280		
Low severity : 1245		
Informational : 903		

b) 10.21.11.0 altağı zayıflık rapor bilgileri

Network Vulnerability Assessment Report **Sorted by vulnerabilities** **27.05.2003**		
Session name: LAN_11_Taraması	**Start Time:**	**27.05.2003 21:35:17**
	Finish Time:	**28.05.2003 09:45:25**
	Elapsed:	**0 day(s) 12:10:07**
Total records generated : 2724		
High severity : 121		
Low severity : 662		
Informational : 1941		

c) 10.21.13.0 altağı zayıflık rapor bilgileri

Network Vulnerability Assessment Report **Sorted by vulnerabilities** **27.05.2003**		
Session name: LAN_13_Taraması	**Start Time:**	**28.05.2003 09:11:33**
	Finish Time:	**28.05.2003 11:43:03**
	Elapsed:	**0 day(s) 02:31:30**
Total records generated : 515		
High severity : 58		
Low severity : 259		
Informational : 198		

d) 10.21.27.0 altağı zayıflık rapor bilgileri

Network Vulnerability Assessment Report **Sorted by vulnerabilities** **27.05.2003**		
Session name: LAN_27_Taraması	**Start Time:**	**27.05.2003 19:59:27**
	Finish Time:	**27.05.2003 20:20:33**
	Elapsed:	**0 day(s) 00:51:06**
Total records generated : 632		
High severity : 101		
Low severity : 346		
Informational : 185		

e) 10.21.160.0/24 altağı zayıflık rapor bilgileri

Network Vulnerability Assessment Report **Sorted by vulnerabilities** **27.05.2003**		
Session name: **LAN_160_Taraması**	**Start Time:**	**27.05.2003 16:51:20**
	Finish Time:	**27.05.2003 19:58:34**
	Elapsed:	**0 day(s) 03:07:14**
Total records generated : 248		
High severity : 14		
Low severity : 76		
Informational : 158		

Nessus'un yüksek seviye güvenlik açıkları nitelemesi, zararlı sonuçlar doğuracak erişim ya da engellemelerin gerçekleşmesini olanaklı kılan zayıflık tespitleridir. Özellikle bu türdeki zayıflıklar, bu iş için otomatize edilmiş saldırı programları ile gerçekleştirilebilmekte ve elde edilen sonuçlarda çoğu zaman saldırgan açısından başarı ile sonuçlanmaktadır. Düşük seviye güvenlik açıkları ise genelde hizmet engellemesi ya da bilgi kaybına sebep olabilecek türdeki zayıflık tespitleri için kullanılmakadır. Bilgilendirme mesajları ise yapılan port taraması sonucunda tespit edilen açık portlardır. Tespit edilen açık port üzerinden hizmet veren uygulama Nessus kayıtlarında yer alıyorsa uygulama katman protokolüyle birlikte, bilinmemesi durumunda ise "unknown" olarak kullanılan taşıma katman protokolüyle birlikte bilgi olarak sunulur.

Tablo 4.4 Sunucu ve aktif ağ cihazları için bilgilendirme raporu

Konak	IP	Zayıflık	Uyarı	Açık Port	Durum
Netware 5.1 – 1	10.21.10.241	6	28	28	Finished
Netware 5.1 - 1	10.21.11.235	6	25	29	Finished
Netware 5.1 – 2	10.21.10.223	5	31	27	Finished
Netware 5.1 - 2	10.21.11.223	5	27	24	Finished
Netware 5.1 – 2	10.21.160.254	3	15	25	Finished
Netware 5.1 - 3	10.21.10.227	2	10	21	Finished
Netware 5.1 – 3	10.21.11.221	2	16	1289	Finished
Netware 5.1 - 3	10.21.160.253	5	15	22	Finished
Netware 6	10.21.10.229	5	17	21	Finished
Netware 6	10.21.11.229	3	14	23	Finished
Windows 2000 – 1	10.21.10.225	8	63	67	Finished
Windows 2000 – 1	10.21.11.226	6	57	65	Finished
Windows 2000 – 2	10.21.10.205	7	59	48	Finished
Windows 2000 – 3	10.21.10.233	12	46	30	Finished
Windows 2000 – 4	10.21.11.239	2	42	30	Finished
Linux RedHat 7.1 – 1	10.21.27.236	28	53	20	Aborted
Linux RedHat 7.1 - 2	10.21.27.230	7	12	2	Finished
Cisco Switch 2924 - 1	10.21.10.252	0	0	3	Aborted
Cisco Switch 2924 - 2	10.21.11.252	0	0	3	Aborted
Cisco Switch 2924 - 3	10.21.13.251	0	0	3	Aborted
Cisco Switch 2948	10.21.160.252	0	0	3	Finished
Cisco Router 2500	10.21.11.238	0	0	29	Aborted
Cisco Router 1701	10.21.11.241	0	0	2	Aborted
Cisco Catalyst 5500	10.21.10.254	0	0	2	Aborted
Cisco Catalyst 5500	10.21.11.254	0	0	2	Aborted
Cisco Catalyst 5500	10.21.13.254	0	0	2	Aborted
Cisco Catalyst 5500	10.21.27.254	3	13	8	Finished

Şekil 4.2'deki sistemde yer alan sunucu ve aktif ağ cihazları için Nessus'un ürettiği zayıflık ve bilgilendirme raporu Tablo 4.4'de düzenlenerek verilmiştir. Bu raporda görüldüğü üzere Nessus, sunucu yazılımları üzerinde mevcut tüm zayıflık sorgulamalarını yürütüp normal olarak sonlandırabilirken, tüm ağ cihazları üzerinde

aynı başarıyı elde edememiştir. Bir kısım ağ cihazı için yapılan zayıflık sorgulamaları cevapsız kalmış ve bu nedenle kesilerek sonlandırılabilmiştir.

Tehlikeli olarak bilinen ve çalıştırılmalarının hedef sistemin hizmet akışını bozabildiği zayıflık eklentileri mevcutttur. Bu uygulama için Nessus'un tüm eklentileri kullanılmıştır. Bunun sonucu olarak sunucularda hizmet aksamalarıda gözlenmiştir. Özellikle Novell OS'ların kısıtlı hizmet sunduğu ABEND seviyelerine geçtiği gözlemlenmiştir. Windows tabanlı sistemlerde ise sunucunun Windows mavi hata ekranına geçtiği ya da bağlantılarda kesilme olduğuda izlenmiştir. Linux tabanlı sistemler ise tehlikeli zayıflıklardan gözlemlenebildiği kadarıyla bir zarar görmemiştir.

Zayıflık tarama sonuçları aktif ağ cihazları ya da tespit edilen zayıflık bulguları baz alınarak raporlandırılabilir. Burada verilen örnek, FTP hizmeti zayıflık bulgularını baz almıştır.

Nessus zayıflık tarama rapor sonuçları tümüyle incelendiğinde bu sistem için yüksek düzeyli güvenlik açıklarının; ajp13 (8009/tcp), font-service (7100/tcp), ftp (21/tcp), general/tcp, ldap (389/tcp), http (80/tcp), netbios-ssn (139/tcp), smtp (25/tcp), snmp (161/udp), unknown (1214/tcp), unknown (32789/udp), unknown (8008/tcp), ms-sql-s (1433/tcp), unknown (1311/tcp) protokolleri üzerinde odaklandığı görülür.

FTP hizmeti sunan sunucular Nessus tarafından Tablo 4.5'de izlendiği gibi toplu olarak sunulmuştur. Tablo 4.6 ise FTP hizmetinde yüksek seviye güvenlik açığı tespit edilen sunucular ve güvenlik açığı hakkında bilgi vermektedir. Nessus ürettiği raporlarda ayrıca güvenlik açığının nasıl giderilebileceği hakkında bilgide sunmaktadır. FTP hizmetinde düşük seviye güvenlik açıkları tespit edilen sunucular ve bu güvenlik açıkları karşısında yapılması gereken çözümler ise Tablo 4.7'de verilmiştir.

Tablo 4.5 Zayıflık taraması ile tespit edilen FTP hizmeti sunan konak bilgisi

ftp (21/tcp)	**Info**	10.21.11.226 10.21.10.205 10.21.10.233 10.21.11.219 10.21.11.237 10.21.11.239 10.21.10.225	Port is open

Tablo 4.6 Zayıflık taraması ile tespit edilen FTP yüksek seviye güvenlik açıkları

ftp (21/tcp)	**High**	10.21.11.237 10.21.11.219 10.21.10.233 10.21.11.226 10.21.10.205 10.21.10.225 10.21.11.239	It may be possible to make the remote FTP server crash by sending the command 'STAT *?AAA...AAA. An attacker may use this flaw to prevent your site from distributing files *** Warning : we could not verify this vulnerability. *** Nessus solely relied on the banner of this server Solution : Apply the relevant hotfix from Microsoft See:http://www.microsoft.com/technet/security/bulletin/ms02-018.asp Risk factor : High CVE : CAN-2002-0073 BID : 4482

Tablo 4.7 Zayıflık taraması ile tespit edilen FTP düşük seviye güvenlik açıkları

ftp (21/tcp)	**Low**	10.21.10.205 10.21.11.239 10.21.11.219 10.21.11.237 10.21.10.233 10.21.10.225 10.21.11.226	An FTP server is running on this port. Here is its banner : 220 DagitimServer Microsoft FTP Service (Version 5.0).
ftp (21/tcp)	**Low**	10.21.10.205 10.21.11.239 10.21.11.237 10.21.10.233 10.21.10.225 10.21.11.226	An FTP server is running on this port. Here is its banner : 220 yanyayinlar Microsoft FTP Service (Version 5.0).
ftp (21/tcp)	**Low**	10.21.11.237 10.21.10.205 10.21.10.233 10.21.11.219 10.21.11.226 10.21.10.225 10.21.11.239	This FTP service allows anonymous logins. If you do not want to share data with anyone you do not know, then you should deactivate the anonymous account, since it can only cause troubles. Under most Unix system, doing : echo ftp >> /etc/ftpusers will correct this. Risk factor : Low CVE : CAN-1999-0497

Bu örneği diğer yüksek düzeyde güvenlik açıklığına sahip protokoller için çoğaltmamız mümkündür. Aynı şekilde düşük düzeyde güvenlik açığına sahip protokoller içinde benzer bir raporlama Nessus tarafından hazırlanmıştır. Nessus zayıflık taraması sonucunda üretilen tüm rapor çıktları EK-CDROM halinde sunulmuştur. Bu CDROM içerisinde Nessus zayıflık raporları, desteklenen text, html, pdf, nsr formatlarında mevcuttur.

Nessus'un tanımlayamadığı ama yüksek seviyede güvenlik açıklığı oluşturabilecek hizmetlerde bulunmaktadır. Nessus olsun diğer zayıflık tarama sistemlerinin olsun tüm zayıflıkları fark edebilmesi, ancak zayıflık veritabanlarında mevcut bulunması ile mümkündür. Nessus bu türden tanımlayamadığı fakat açık olarak tespit ettiği portlar için düşük seviye güvenlik açığı bildirimi yapar. Bu sayede sistem

yöneticilerinin dikkatini çekmesi ve açık bu porta yoğunlaşmaları için zemin hazırlamış olur. Uygulamada bu türden bir zayıflık Web sunucu hizmeti sunan Windows 2000 – 3 bilgisayarında tespit edilmiştir. Tablo 4.8'de izlenen TCP protokolü 6129 portunu kullanan bu hizmet, Nessus tarafından tanımlanamadığı için "unknown" olarak belirtilmiştir. Düşük seviye zayıflık açığı olarak raporlanmış ve Nessus takımına bu port üzerinde tanımlanan hizmet için bilgi gönderilmesi rica edilmiştir.

TCP 6129 portu üzerine tanımlı bu uygulamanın bir uzaktan kontol programı olduğu tespit edilmiştir. (DameWare Mini Remote Control). Buda aslında çok tehlikeli bir zayıflık olduğunu ortaya koymaktadır. Bu şekilde tespit edilen zayıflığın, sistemin tümünde sorgulanması ancak onun bir eklenti haline getirilip Nessus zayıflık veritabanına eklenmesi ile mümkün olur. Bunun içinde ya Nessus takımının bunu yapması beklenecek ya da bir eklenti halinde sistem yönetcileri tarafından Nessus zayıflık veritabanına ilave edilecektir. Ek A'da TCP 6129 portunu kullanan DameWare Mini Remote Control hizmetinin tespit edilmesini sağlayan eklenti görülmektedir.

Tablo 4.8 Tanımlanamayan zayıflık kaydı

unknown (6129/tcp)	**Info**	Port is open
unknown (6129/tcp)	**Low**	An unknown server is running on this port. If you know what it is, please send this banner to the Nessus team: 00: 30 11 0.

BÖLÜM 5. SONUÇLAR

TCP/IP protokol ailesinin gerek Internet, gerekse Intranet ağları için sunduğu hizmet çeşitliliği ve bilgiyi paylaşıma dönük yapısı onun çok fazla kullanılır bir protokol ailesi olmasını sağlamıştır.

TCP/IP'nin bu kadar ön plana çıkması, açık kaynak kodlu olması, üzerinde yeni hizmetlerin geliştirilmesine imkan sağlaması, kötü niyetli kişilerinde ilgisini çekmiştir. Bu kişilerin, TCP/IP protokol ailesinin sahip olduğu eksiklikleri ve hataları bulmaları, bunlar üzerine geliştirdikleri değişik saldırı şekilleri bilgi güvenliği açısından tehdit unsuru oluşturmuşlardır.

Saldırganlar sadece TCP/IP'nin zayıf yanlarından yararlanmaz. TCP/IP protokol ailesini kullanan sunucu OS ve yazılımlarıda güvenlik zayıflıklarına sahip olabilmektedir. Saldırganlar bu OS ve yazlımları tespit ettikten sonra hedef sistemlere daha bilinçli teknikler ile saldırabilmektedir. Bu nedenle saldırganların hedef aldığı sistemler hakkında edinebilecekleri bilgilerin asgariye düşürülmesi, hem de aldatıcı düzenlemeler yapmak saldırganların işlerini zorlaştıracaktır.

Peki bunlar göz önünde tutulduğunda saldırganlara karşı neler yapılabilir. İşte bu noktada bir güvenlik politikası belirlenmesi gerekliliği ortaya çıkmaktadır. Çok yönlü düşünülmesi gereken güvenlik politikası hem sistemlerin zayıflıklarını gidermeye yönelik çalışmalar içermeli hemde çalışanların güvenlik kavramları konusunda bilinçlendirilmesi ve eğitilmesinide kapsamalıdır.

Bizim ilgilendiğimiz yönüyle yani sistem zayıflıklarının giderilmesi için ateş duvarları, saldırı tespit sistemleri ve zayıflık tarama sistemlerinin kullanılması bir gerekliliktir.

Güvenlik politikasının belirlediği çizgiler dahilinde, güvenliği tesis edilecek ağa yönelik ve bu ağlardan harici ağlara yönelik yetkisiz erişimler üzerinde kısıtlamalar getirecek ateş duvarları kurulmalıdır. Bu sistemler sayesinde harici ağ bağlantısı olan ağ, tamamı saldırıya açık risk alanı olmaktan çıkacaktır, hakkında istenmeyen bilgi toplanılması, port taramaları yapılması engellenecektir. Ateş duvarları paket filtrelemeye yönelik, vekil özelliğinde ya da uygulama düzeyli olmaktadır. Bunların tek başlarına ya da seçilen mimari dahilinde birlikte kullanılması güvenlik seviyesini arttıracaktır.

Ateş duvarları bir şekilde devre dışı bırakılmış ve ağa yetkisiz erişim imkanı kazanılmış olabilir. Bu tip istenmeyen durumların tespit edilmeside önemlidir. Bu amaçla geliştirilen saldırı tespit sistemleri, güvenliği tesis edilecek ağlarda olabilecek anormallikleri ve sızmaları yakalayarak, yayınladıkları uyarılar ve tuttukları kayıtlar ile sistem yöneticilerinin bundan haberdar olmasını sağlayacaktır.

Saldırı tespit sistemleri veritabanlarında yer alan saldırı imzalarını, ağ trafiğinde gözlemledikleri taktirde karşı kural yürütebilir, bağlantıların sonlandırılmasını sağlayarak istenmeyen erişimlere engel olunması gibi işlevleride yerine getirebilirler

Zayıflık tarama sistemleri bilgisayar ağlarını saldırganlar gözüyle inceleyen sistemlerdir. Gerçekleştirilecek zayıflık taraması, saldırılara karşı ne kadar güvenilir bir sistem oluşturulmuş olduğunu ortaya koyacaktır. Saldırganların izlediği aşamaları kaydederek öncelikle hedef sistem üzerindeki açık portların tespitini, daha sonrasında ise bu portlar üzerinden hizmet veren uygulamaları çözmeye çalışırlar. Veritabanlarındaki zayıflık eklentilerinin hedef sistem üzerinde ne kadar başarılı olduğunu ürettiği sonuçlar ile sistem yöneticilerine rapor ederler.

Hedef sistem üzerinde uygulaması gerçekleştirilen Nessus zayıflık tarama sistemi, orta seviye bir saldırgan kadar başarı sergilemektedir. Nessus kullandığı zayıflık veritabanının izin verdiği ölçülerde zayıflık taraması gerçekleştirebilmektedir. Zayıflıklar belirli ana başlıklar altında gruplandırılmış ve her geçen gün yenilerinin eklenmesiyle serbest dağıtılabilir programlar arasında ciddi bir büyüklüğe ulaşmıştır.

Nessus'un artımlı tarama yapabilmesi yani daha önceden yapılan testleri uygulamayıp sadece yeni zayıflık testlerini uygulaması önemli artısıdır.

Güvenlik testinin gerçekleştirimi sırasında bazı zayıflık testlerinin hedef sisteme zarar vereceği düşünülerek tehlikeli zayıflıklar olarak belirlenmiş ve seçimlilik olarak sunulmuştur. Uygulama esnasında tehlikeli olarak nitelenen zayıflıkların testinin gerçekleştirildiği ağ cihazlarında, cihazların hizmet veremez duruma geldiği ve kendini kapattıkları görülmüştür. Bazı sunucu türleri ise özellikle Novell OS çok hassas yanıtlar vererek kendisini kısıtlı hizmet sunduğu ABEND seviyelerine geçirdiği gözlemlenmiştir. Windows tabanlı sistemlerde ise sunucunun Windows mavi hata ekranına geçtiği, kapatılma ihtiyacı hissetiği pratik olarak gerçekleştirilmiştir. Linux tabanlı sistemler ise tehlikeli zayıflıklardan gözlemlenebildiği kadarıyla bir zarar görmemiştir.

NASL script dilinin yapısı gereği hedef sistem üzerinde kesinlikle kod çalıştırılmasına veya dosya kopyalanmasına izin vermeyecektir. Böylece testler daha güvenli yapılmaktadır.

Nessus zayıflık tarama sisteminin vereceği güvenlik zayıflık bilgisi ve çözüm yolu, saldırılara karşı ne yapılması gerektiği, zayıflığın nasıl giderilebileceği konusunda da sistem yöneticilerine büyük katkı sağlar.

Gerek saldırı tespit sistemleri, gerekse zayıflık tarama sistemleri daha önce yapılmış saldırı imzalarını ve zayıflık bilgilerini baz alarak değerlendirme yaparlar. Bu doğrultuda kullanılan ürünler için geliştirilen son veritabanlarının sağlanması çok önemlidir.

Ağ güvenliği için tesis edilecek bu sistemler kadar, ağlarda mevcut kullanılan diğer ürünler için geliştirilen son güvenli versiyonların ya da zayıflıklarını kapatmak için çıkarılan yamaların takip edilerek sistemlere yüklenmeside bilgi güvenliği açısından çok önemlidir.

TARTIŞMA ve ÖNERİLER

Bilgisayar ağlarındaki güvenlik politikası, kullanılan yazılım ya da protokolün yapısının sağladığı güvenlik desteğinden yola çıkarak aşama aşama kat ettiği yol ile günümüzde ateş duvarları, saldırı tespit sistemleri ve zayıflık tarama sistemleri gibi güvenlik araçlarını içerecek hale gelmiştir. Bu sistemleri kullanmak artık saldırıları tespit ve güvenlik zayıflıklarını gidermek için şart durumuna gelmiştir.

Bilgisayar ağları ne kadar güvenli oluşturulursa oluşturulsun belli bir süre sonra yapılan çalışmalar ve değişiklikler yeteri kadar takip edilemezse sistem üzerinde güvenlik açıklarının doğmasına neden olabilmektedir. Bu nedenle güvenlik tedbirlerinin uygulanması ve etkinliğinin sürdürülebilmesi için bilgisayar sistemlerine saldırgan gözüyle bakmak güvenlik zayıflıklarının yakalanmasına yardımcı olacaktır. Periyodik olarak bu şekilde yapılan gözlemler ve tespit edilen zayıflıkların giderilmesi birçok saldırganın düşüncelerini gerçekleştirmesini engellenmiş olacaktır

Zayıflık tarama sistemlerinin kullandığı zayıflık veritabanları şu zaman için yeterli değildir. Yani gerçekleştirilen zayıflık tespiti sistemimizin gerçekten güvende olduğunu ya da zayıflığa sahip olduğunu belirtemez. Bu yüzden yapılacak zayıflık taramaları mümkün olduğunca farklı uygulamalarca ve uygulamaların sahip olduğu üstünlükler göz önünde bulundurularak yapılmalıdır.

Zayıflık taraması yapılmak istenen sistem için uygun zaman dilimlerinin seçilerek gerçekleştirilmesi bilgisayar sistemlerinin iş akışını aksatmamış ve sunulan hizmetleri engellememiş olacaktır. Ayrıca tehlikeli olduğu vurgulanan zayıflık tespit eklentilerinin uygulanmadan önce hedef sistemde kalıcı problem yapıp yapmadığı araştırılmalıdır. Dikkat edilmemesi sistem yöneticilerinin saldırganların yaptığı işlevi üstlenmesini sağlayacaktır.

Zayıflık taramalarının değişik konumlar üzerinden uygulanması hedef sistem açısından daha sağlıklı bilgi edinilmesini sağlayacaktır. Bu yüzden hedef sistem zayıflık taramaları için yerel ağ üzerinden varsa harici ağ üzerinden uygun zayıflık tarayıcı bağlantıları kurularak gerçeleştirilmelidir. Harici ağ üzerinden yapılan gözlemleme, sistem yöneticilerine saldırganların bakış açısını kısmende olsa kazandıracaktır.

KAYNAKLAR

[1] ÇÖLKESEN, R., ÖRENCİK, B., "Bilgisayar Haberleşmesi ve Ağ Teknolojileri", Papatya Yayıncılık, İstanbul, 2000.

[2] DAYIOĞLU, B., ÖZGİT, A., "Internet'te Saldırı Tespit Teknolojileri", İletişim Teknolojileri 1. Ulusal Sempozyumu Ve Fuarı, Ankara, Ekim 2001.

[3] CERT Advisory, "Vulnerabilities", www.cert.org, 2002.

[4] KARAHMETOĞLU, O., "Internet Güvenliği Kavramları ve Teknolojileri", Yüksek Lisans Tezi, ITU Fen Bilimleri Enstitüsü, 2001, İstanbul.

[5] AY, Y.,"İnternet'te Firewall Güvenlik Kavramı ve Hizmetlere Erişim Denetimi", Yüksek Lisans Tezi, ITÜ Fen Bilimleri Enstitüsü, 1996, İstanbul.

[6] BARRON, B., ELLSWORTH, J.H., SAVETZ, K.M., "Internet Unleashed", SARIHAN, T.D., Sistem Yayıncılık, Istanbul, 1998.

[7] ÇETİN, G., ÇELİK, K. G., "Linux Ağ Yönetimi", Seçkin, p159, Ankara, 2000.

[8] ÖZAVCI, F., "Saldırı Tespit Sistemleri Giriş", www.siyahsapka.com, Kasım 2001.

[9] ÖZAVCI, F., "NESSUS ve Zayıflık Tarama Sistemleri v.1", www.siyahsapka.com, Ocak 2002.

[10] Chapman, D.B. And Zwicky, E.D., "Building Internet Firewalls", O'Reilly, s. 1-496, Cambridge, Mass., 2000.

[11] Anonymous, "Maximum security : A Hacker's Guide To Protecting Your Internet Site And Network", Sams.net, Indianapolis, 1998.

[12] DEREGÖZÜ, R., "Bilgisayar Ağlarında Güvenlik Sorunu 'Firewall' Kullanarak Ağ Güvenliğini Sağlama", Yüksek Lisans Tezi, ITÜ Fen Bilimler Enstitüsü, İstanbul, 1999.

[13] CANAVAN, J.E., "Fundamentals of Network Security", Artech House, Boston, 2001.

[14] TANENBAUM, A.S., "Computer Networks", Prentice Hall PTR, Upper Saddle River, N.J., 1996.

[15] PABRAI, U.O., GURBAI, K.V., "Internet and TCP/IP Network Security", McGraw-Hill, Washington, 1996.

[16] SIYAN, K. And HARE, C., "Internet Firewalls And Network Security", Net Riders Publishing.

[17] WILDER, F., "A Guide to the TCP/IP Protocol Suite", Artech House, Boston, 1993.

[18] COMER, D.E., STEVENS, D.L., "Internetworking with TCP/IP", Prentice Hall, Englewood Cliffs, c1991-1994.

[19] KIRCH, O., DAWSON T., "Linux network administrator's guide", O'Reilly, Cambridge, Mass, 2000.

[20] DERFLER, F.J., "Network Sistemleri ve Bilgisayar Bağlantı Kılavuzu", CERİT, B., Sistem Yayıncılık, İstanbul, 1996.

[21] SUBRAMANIAN, M., "Network Management", Addision-Wesley, California, 2000.

[22] KURT, E. "Internet Güvenliği", www.olympos.org, 2002.

[23] Güvenlik Haber , "Bilgi Güvenliği", www.guvenlikhaber.com, 2002.

[24] Snort Advisory, "How to Write Snort Rules", www.snort.sourcefire.com, 2002.

[25] DERAISON, R., "Nessus Documentation", http://www.nessus.org/documentation.html, 2003.

[26] ERKAN, H.O., "Ipchains MAN", www.linux.org.tr/belgeler/ipchains/ipchains-man.txt, Eylül 1999.

[27] ViSolve Advisory, "Squid Configuration Manual", www.squid.visolve.com , May 2002.

EK – A

TCP 6129 portunu kullanan DameWare Mini Remote Control hizmetinin tespit edilmesini sağlayan NASL eklenti kodu:

```
#
# This script was written by Umit Ersoz
# This is version 2.0 of this script.
#
#

if(description)
{
 script_id(97531);
 script_version ("$Revision: 1.0 $");
 name["english"] = "Check for DameWare Mini Remote Control";
 name["francais"] = "";
 script_name(english:name["english"], francais:name["francais"]);

 desc["english"] = "
The remote server is running DameWare Mini Remote Control.
DameWare Mini Remote Control permits a console to be displayed remotely.

Solution: Disable DameWare Mini Remote Control access from the network by
using a firewall, or stop VNC service if not needed.

Risk factor : Medium";
```

```
 desc["francais"] = "";

 script_description(english:desc["english"], francais:desc["francais"]);

 summary["english"] = "Checks for DameWare Mini Remote Control";
 summary["francais"] = "";

 script_summary(english:summary["english"],
francais:summary["francais"]);

 script_category(ACT_GATHER_INFO);


 script_copyright(english:"This script is Copyright (C) 2000 Umit Ersoz",
            francais:"");
 family["english"] = "Backdoors";
 family["francais"] = "Backdoors";
 script_family(english:family["english"], francais:family["francais"]);
 script_dependencie("find_service.nes");
 script_require_ports("Services/vnc", 6129);
 exit(0);
}

#
# The script code starts here
#

function probe(port)
{
 if(get_port_state(port))
 {
  soc = open_sock_tcp(port);
  if(soc)
```

```
  {
   r = recv(socket:soc, length:3072);
   security_warning(port);
   security_warning(port:port, data:string("DameWare Mini Remote Control Protocol
: ",r));

   close(soc);
  }
 }
}

port = get_kb_item("Services/DWRC");
if(port)probe(port:port);
else
{
 port=6129;
 probe(port:port);
}
```

ÖZGEÇMİŞ

Ümit Ersöz, 05.03.1978 yılında Karabük'te doğdu. İlkokulu Karabük Şirinevler İlkokulu'nda, orta öğrenimini Atatürk Merkez Orta Okulu'nda ve liseyi 1995'te Karabük Teknik Lisesi'nde tamamladı. Aynı yıl Kocaeli Üniversitesi Bilgisayar Mühendisliği Bölümünü'ne girmeye hak kazandı. Temmuz 1999'da bölüm birincisi olarak mezun oldu.

Ekim 2000'de Sakarya Ünivertisesi Bilgisayar ve Bilişim Mühendisliği yüksek lisans programına kayıt oldu.

Lisans programı bitirme tezinde IBM DB2 veritabanını kullanarak Turizm Otomasyon Programı hazırladı. Halen ağ sistem yönetimi ve geliştirilmesi üzerine özel bir şirkette çalışmakta.