

T.C.

GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ



NETCHECK GÜVENLİK DUVARI TEST VE
ANALİZ ARACI

140036

Tahsin TÜRKÖZ

YÜKSEK LİSANS TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ

ANABİLİM DALI

TEZ DANIŞMANI

Yrd.Doç.Dr.İbrahim SOĞUKPINAR

GEBZE

2003



Tahsin TÜRKÖZ'ün tez çalışması, G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 06-06-2003 tarih ve .2003/17 sayılı kararıyla oluşturulan jüri tarafından Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS Tezi olarak kabul edilmiştir.

JÜRİ

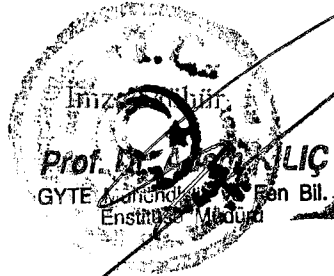
ÜYE : Yrd. Doç. Dr. İbrahim SOĞUKPINAR
(Tez Danışmanı)

ÜYE : Yrd. Doç. Dr. Mehmet GÖKTÜRK

ÜYE : Yrd. Doç. Dr. İlyas KANDEMİR

ONAY

G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
.25.07.2003.. tarih ve ...2003/24... sayılı kararı.



ÖZET

Son yıllarda Internet kullanımının dünya genelinde artması ve daha çok bilginin paylaşılr duruma gelmesi sonucunda güvenlik kavramı önem kazanmıştır. Kurumlar kritik bilgilerini ve ağlara bağı bilgisayarlar, sunumcular, aktif cihazlar gibi varlıklarını korumak için çeşitli güvenlik mekanizmaları kullanmaya başlamışlardır. Bu güvenlik mekanizmalarının başında güvenlik duvarları gelmektedir.

Bir ağı güvenli hale getirmenin maliyetinin fazla olması sebebiyle çoğu kurum, güvenlik mekanizması olarak sadece güvenlik duvarı kullanmayı tercih etmektedir. Bu durum, güvenlik duvarlarının önemini daha da arttırmaktadır. Bu sebeple güvenlik duvarlarının işlevselliğini tam olarak yerine getirip getirmediğinin incelenmesi, doğru konfigürasyon yapılp yapılmadığının ve güvenlik açıklıkları içerip içermediğinin test edilmesi gerekmektedir.

Bu amaçla günümüze kadar bazı güvenlik duvarı test metodolojileri geliştirilmiştir. Fakat bu metodolojiler güvenlik duvarlarını kapsamlı bir şekilde test etme noktasında yeterli değildir. Bu çalışmada son yıllarda geliştirilen saldırı yöntemleri ve daha önceden geliştirilmiş güvenlik duvarı test metodolojilerinin eksiklikleri de göz önünde bulundurarak yeni bir test metodolojisi önerilmiştir. Ayrıca bu test metodolojisinin güvenlik duvarı testlerinde kolaylıkla kullanılabilmesini sağlayacak otomatik bir test aracı (NetCheck) geliştirilmiştir.

Çalışmanın son kısmında NetCheck, diğer güvenlik duvarı test metodolojileri ile karşılaştırılmıştır. Ayrıca iki ayrı ağ topolojisi üzerine entegre edilen iki farklı güvenlik duvarı, NetCheck ile test edilmiştir.

SUMMARY

In recent years, information security concept has got importance as a result of increase in Internet usage and rise in sharing data over it around the world. Associations have begun to integrate security mechanisms into their enterprise networks in order to protect their knowledge and assets such as computers, servers, active equipments etc. Firewalls are one of the mostly used security components in networks.

Because securing a network is a cost effective process, most organizations prefer to use just firewalls as security mechanisms. This situation increases the importance of firewalls. Therefore, it is necessary to examine whether firewalls provide the functionality they claim, and to check whether they contain vulnerabilities and configuration errors which can be used to compromise security.

For this purpose, some firewall testing methodologies have been developed so far. However, these methodologies are not sufficient to test firewalls completely. In this study, a new testing methodology is recommended by considering attack techniques that has been recently found and deficiencies of previously implemented firewall testing methodologies. Also, an automated test tool called NetCheck is developed. NetCheck helps security auditors, testers and system administrators to apply this testing methodology to their firewalls easily.

In the last section, NetCheck is compared with the other testing methodologies. Besides, two firewalls on two different network topologies are tested by using NetCheck and results are examined.

TEŞEKKÜR



Tez çalışmam esnasında her zaman yol gösterici olan ve yardımlarını esirgemeyen tez danışmanım Sayın Yrd. Doç. Dr. İbrahim Soğukpınar'a, TUBITAK-UEKAE bünyesinde bu çalışmayı geliştirme imkanını sunan proje yürütücüm Mert Üneri'ye ve Enstitü müdürüm Önder Yetiş'e, fikirlerine danıştığım "OKTEM-Sınır Güvenliği grubu" üyelerine ve bana manen destekte bulunup her zaman yanımda olan eşim Özen Türköz'e ve oğluma teşekkürlerimi sunarım.



İÇİNDEKİLER DİZİNİ

ÖZET	vii
SUMMARY	iv
TEŞEKKÜR	v
İÇİNDEKİLER DİZİNİ	vi
SİMGELER - KISALTMALAR DİZİNİ	vii
ŞEKİLLER DİZİNİ	x
ÇİZELGELER DİZİNİ	xi
1 GİRİŞ	xiii
1.1 Güvenlik İhtiyaçları ve Eksikleri	1
1.2 Son Yıllardaki Saldırı Trendi	1
1.3 Güvenlik Duvarları ve Roller	3
1.4 Güvenlik Testine Olan İhtiyaç	4
1.5 NetCheck Güvenlik Duvarı Test ve Analiz Metodolojisi	8
2 GÜVENLİK DUVARLARI İÇİN REFERANS MODEL VE TEST METODOLOJİLERİ	10
2.1 Güvenlik Duvarları İçin Referans Model	11
2.1.1 Tanım	11
2.1.2 Tanımların Referans Modele Entegrasyonu	12
2.2 Saldırı Mekanizmaları	14
2.3 Güvenlik Duvarı Test Yöntemleri	18
2.4 Al-Tawil ve Al-Kaltham'ın Metodolojisi	20
2.4.1 Test Yatakları	21
2.4.2 Test Yataklarının Karşılaştırılması	22
2.4.3 Test Metodolojisi	23

2.5	Vigna'nın Test Metodolojisi	28
2.5.1	Ağ Modeli	29
2.5.2	Açıklıkların Modellenmesi	31
2.5.3	Vigna'nın Yaklaşımıyla Güvenlik Duvarlarının Testi	33
3	NETCHECK GÜVENLİK DUVARI TEST VE ANALİZ METODOLOJİSİ	35
3.1	Ağ Modellemesi	35
3.2	Kullanılan Test Modülleri ve Yapıları	37
3.2.1	Topoloji Konum Testi	37
3.2.2	Erişim Testi	40
3.2.3	Bozuk Bağlantı Paketi Testi	42
3.2.4	IP Sahteciliği Testi	43
3.2.5	Sağlaması Hatalı Paket Testi	44
3.2.6	Arazi Saldırısı Testi (LAND attack)	45
3.2.7	Geçersiz Kaynak IP Adresli Paket Testi	45
3.2.8	Bölünmüş Paket Testi	46
3.2.9	Kaynak Yönlendirmeli Paket Testi	49
3.2.10	Bağlantı Seli Testi	50
3.3	NetCheck'e Genel Bir Bakış	51
3.4	Test Metodolojisi ve Ağlara Entegrasyonu	52
3.4.1	Ağa Entegrasyon	52
3.4.2	Topoloji konfigürasyonu yapma	53
3.4.3	Test modüllerini belirleme ve ayarlama	55
3.4.4	Testi başlatma	60
3.5	Diğer Özellikler	61
4	NETCHECK İLE ÖRNEK TESTLER, DİĞER METODOLOJİLER İLE KARŞILAŞTIRMA	64

4.1	NetCheck Örneđi 1	64
4.2	NetCheck Örneđi 2	66
4.3	İki Sonucun Karşılaştırılması	68
4.4	NetCheck İle Diğer Metodolojilerin Karşılaştırılması	70
5	SONUÇLAR VE GELECEK ÇALIŞMALAR	74
	KAYNAKLAR	77
	ÖZGEÇMİŞ	80
	EKLER	81
	Ek-1: NetCheck'i Meydana Getiren Objeler ve Görevleri	81
	Ek-2: NetCheck Test Sonuçları -1	92
	Ek-3: NetCheck Test Sonuçları -2	102

SİMGELER - KISALTMALAR DİZİNİ

CERT	Computer Emergency Response Team
DMZ	De-militarized Zone
DNS	Domain Name Services
FTM	Firewall Test Modülü
FTY	Firewall Test Yatağı
FWTK	Firewall Tool Kit
GAA	Geniş Alan Ağı
GF	Güvenlik Fonksiyonu
ICMP	Internet Control Message Protokol
IP	Internet Protocol
MAC	Media Access Control
MTU	Maximum Transfer Unit
NAT	Network Address Translation
OSI	Open System Interconnection
QOS	Quality of Service
SANS	System Administration and Network Security
STM	IP sahteciliği tespit matrisi
TCP	Transmission Control Protocol
UDP	User Datagram Protokol
VPN	Virtual Private Network

ŞEKİLLER DİZİNİ

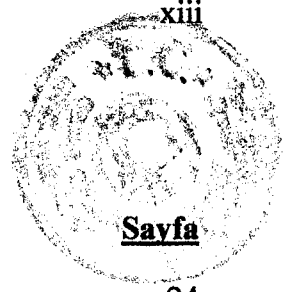
Sekil

1.1 : Güvenlik Duvarının kurumsal ağlarda örnek kullanımı	4
1.2: IPFilter güvenlik duvarı örnek kural tanımı	8
2.1: Referans modele genel bakış [Scuba and Spafford, 1997]	12
2.2: Güvenlik duvarı teknolojisi için referans model	13
2.3: Atak ağaçlarının grafiksel ve metinsel gösterimi	15
2.4: WEB Sunumcu üzerinde yetki kazanma saldırısının atak ağacı ile modellenmesi [Moore et al, 2001]	16
2.5: Saldırıların hedefe uygulanma basamakları	17
2.6: Nmap programı örnek çıktısı	19
2.7:Test yatağı 1 [Al-Tawil and Al-Kaltham, 1999]	21
2.8:Test yatağı 2 [Al-Tawil and Al-Kaltham, 1999]	22
2.9:Test yatağı 3 [Al-Tawil and Al-Kaltham, 1999]	22
2.10: Örnek bir ağ hiper-grafigi [Vigna, 1997]	29
2.11: IP sahteciliği örneği [Vigna, 1997]	32
3.1: Güvenlik Duvarlarının topolojilere entegrasyonu	39
3.2: Güvenlik duvarı örnek kural tablosu -1	40
3.3: Güvenlik duvarı örnek kural tablosu -2	41
3.4: Bölünmüş ICMP paketlerinin yapısı	47
3.5: IP başlığındaki kaynak yönlendirmesi için genel format [Steven, 2001]	49
3.6: Kaynak yönlendirmesi örneği [Steven, 2001]	50
3.7: NetCheck Ana Penceresi	52
3.8: NetCheck'in ağa entegrasyonu	53
3.9: Topoloji Konfigürasyon Penceresi	54
3.10: Modül Konfigürasyonu penceresi	60

3.11: Test Prosesi penceresi	61
3.12: Rapor penceresi	62
4.1: Örnek test yatağı -1	64
4.2: CheckPoint güvenlik duvarı için belirlenen kural tablosu	66
4.3: Örnek test yatağı -2	67
4.4: IPFilter güvenlik duvarı için belirlenen kural tablosu	68



ÇİZELGELER DİZİNİ



Cizelge

2.1: Kurulumların karşılaştırılması [Al-Tawil and Al-Kaltham, 1999]	24
2.2: Konfigürasyon ve çalışmaların karşılaştırılması [Al-Tawil and Al-Kaltham, 1999]	25
2.3: Kullanımın karşılaştırılması [Al-Tawil and Al-Kaltham, 1999]	26
2.4: SOCKS'un avantajlar ve dezavantajlar [Al-Tawil and Al-Kaltham, 1999]	27
2.5: FWTk'nın avantajlar ve dezavantajlar [Al-Tawil and Al-Kaltham, 1999]	28
2.6: H ₄ bilgisayar için IP sahteciliği tespit matrisi [Vigna, 1997]	33
2.7: rlogin servisi için güven vektörü [Vigna, 1997]	33
3.1: Örnek bir açık test modülü	56
3.2: Opsiyon rezerve sözcükleri	58
3.3: İngilizce açık test modülü	63
4.1: İki güvenlik duvarı test sonucunun karşılaştırılması	69
4.2: Güvenlik duvarı test metodolojilerinin karşılaştırılması	73



1. GİRİŞ

1.1. Güvenlik İhtiyaçları ve Eksikleri

Dijital teknolojinin dünya geneline yayılması ile birlikte daha çok bilgi Internet üzerinden paylaşılmaya başlanmıştır. Bu bilgiler kişiye özel olabildiği gibi mali değerler(kredi kartı, banka hesap bilgileri vs.) de içerebilmektedir. Internet'te birçok işlemin kolaylıkla yapılmasına imkan veren bu veri akışı bazı dezavantajları da beraberinde getirmektedir.

Bu dezavantajlardan en önemlisi bilgi güvenliği ihtiyaçlarının karşılanması noktasındaki eksikliklerdir. Yapılan araştırmalar büyük şirketlerin %80'inin Internet üzerinde kritik bilgi bulundurduğunu ortaya koymuştur [Cisco, 1998]. Bilgiye her türlü ortamdan, her türlü cihazla özgürce erişim, kullanıcılar için ne kadar önemli ise bu bilgilerin başkalarının eline geçmesi de o kadar sakıncalıdır. Bu yüzden, Internet üzerinden erişilebilir varlıkların ve verilerin güvenliğinin üst seviyede sağlanması için güvenlik bilinçlenmesine önem verilmeli ayrıca gerekli teknolojiler bütüncül bir güvenlik çözümü içinde ele alınmalıdır. Esnek ve ölçeklenebilir güvenlik stratejileri oluşturulmalı; her türlü bilginin kullanılabilirliği, ulaşılabilirliği, gizliliği ve güvenliği garanti altına alınmalıdır. Bir kurumda maksimum veri güvenliğinin sağlanması ile müşterinin güvenini artırmak, bu kurumun başarısını garantilemekle eşdeğerdir.

Sağlanacak güvenliğin seviyesi belirlenirken öncelikle güvenlik ihtiyaçlarının neler olduğu belirlenmelidir. Internet ortamında ya da geniş alan ağlarında (GAA) üzerinde durulması gereken güvenlik ihtiyaçları dört ana başlıkta toplanabilir:¹

Gizlilik: Bilginin yetkisiz erişime karşı korunması

Süreklilik: Doğru zamanda ve doğru biçimde bilgilere ve sistemlere erişim

Bütünlük: Verilerin yetkisiz düzenlemelere karşı korunması ve yapılan değişikliklerin tanımlanması

Kaynağı Doğrulama: Bilginin özgünlüğünün korunması. Bilginin belirli bir kaynaktan gerçekten çıkıp çıkmadığının kanıtlanması

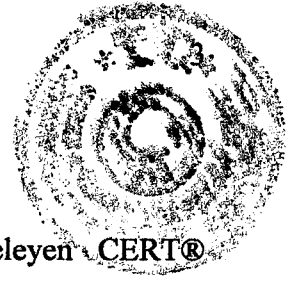
¹ Confidentiality, Availability, Integrity, Non-repudiation

Güvenlik ihtiyaçlarının karşılanamamasının birkaç sebebi vardır. Bunlardan birisi iletişim protokolünden kaynaklanan eksikliklerdir. Bilindiği gibi Internet teknolojisi TCP/IP protokolü kullanılarak gerçekleştirilmektedir. Halbuki TCP/IP protokolü, akademik haberleşme ortamında basit ve açık (üçüncü kişiler tarafından anlaşılabilir) bir iletişim altyapısı sağlamak amacı ile dizayn edilmiştir [Vigna, 1996,]. Bu protokol yapı itibarıyla oldukça kararludur ve bu sebeple ağı bağı bir uçtan çıkan verinin bozulmadan ve hızlı bir şekilde hedef uca erişmesini sağlar. Fakat bu kararlılık hat üzerinde sinyal kaybı, veri çarpışması (data collision) gibi çevresel etkenlerden kaynaklanan problemlere karşı geçerlidir. Yukarıda açıklanan güvenlik ihtiyaçlarını karşılama konusuna yetersizdir. Bu durum, kötü niyetli saldırganların, bu trafik üzerinde güvenlik tehditlerini oluşturabilmelerine imkan tanımaktadır.

Saldırlara karşı açıklıkları bulunan tek yapı TCP/IP değildir. Ayrıca veri trafiğinin üzerinden geçtiği yazılım, donanım, işletim sistemi gibi bileşenlerde de değişik zafiyetler bulunmaktadır. Bu zafiyetler, ürünlerin geliştirilmesi esnasında üreticilerin, güvenliği sektöre uğratacak verilerle² sistemlerini yeterince test etmemelerinden kaynaklanmaktadır. Güvenlik testlerindeki bu yetersizliğin en büyük sebebi ise ürünlere özel test metodlarının ya da verilerinin erişilebilir olmamasıdır [Moore et al, 2001]. Bu tür testlerin çoğu, ürünler üretildikten sonra profesyonel olarak çalışan test laboratuvarlarında gerçekleştirilmektedir. Bu sebeple tasarım aşamasında oluşabilecek birçok açıklık ancak ürün markete sürülüp müşteriler tarafından bir süre kullanıldıktan sonra ortaya çıkmakta ve saldırganların sistemleri kötüye kullanmaları için ortam hazırlamaktadır.

Diğer bir nokta ise sistem yöneticisi ya da sahibi konumundaki insanların güvenlik konusuna yeterince önem vermemeleridir [Arbaugh et. al., 2000]. Son yıllarda yapılan araştırmalar güvenlik açıklıkları içeren sistemlere sahip kurumların, bu açıklıklarına karşılık gelen yamaları yükleme noktasında ilgisiz kaldıklarını ve yayınlanan birçok yamanın kullanılan sistemlerin sadece %40'ında uygulandığını göstermektedir [SANS, 2003].

² Security failure data



1.2. Son Yıllardaki Saldırı Trendi

1988 yılından beri ağ üzerinden yapılan saldırıları inceleyen CERT® Koordinasyon Merkezi, son yıllarda saldırıların görülme oranının oldukça yükseldiğini belirtmektedir. Kurumların işlevselliklerini yerine getirmelerini ve verilerinin korunmalarını ciddi anlamda etkileyen bu artışın sebepleri şu şekilde sıralanabilir [CERT, 2002]:

Saldırıları sırasında kullanılan araçların otomatik olma oranında bir artış olmuştur. Saldırganlar daha önceki yıllarda tarama araçları ile sistemleri tarayıp açıklıkları bulduktan sonra bu açıklığa karşılık gelen saldırıyı gerçekleştirirlerdi. Fakat günümüzde otomatik tarama araçları bu saldırıları bulmakla kalmayıp saldırı kodunu da çalıştırarak sistemleri sömürebilmektedir. Bu durum, saldırı sürecini de kısaltmaktadır. Ayrıca otomatik saldırı araçlarının kullandıkları saldırı yöntemlerinin etkileri eski yıllara nazaran artış göstermiştir. 2000 yılına kadar saldırı araçlarının çalışabilmesi için insanların süreci başlatmaları gerekiyordu. Günümüzde ise bu araçlar saldırı sürecini kendileri başlatabilmektedir. Code Red ve Nimda virüsleri bu tür araçlara örnek gösterilebilir. Bir diğer nokta ise saldırı araçlarının birbirleri ile koordineli çalışmaya başlamasıdır. 1999 yılından beri koordineli araçlar kullanılarak dağıtık saldırılar yapılmaktadır. Bu araçlar en çok servis dışı bırakma saldırılarında kullanılmaktadırlar.

Saldırı araçlarında kullanılan yöntemler daha da gelişmiştir. Günümüzde yapılan saldırıların değişik saldırı imzaları ile tespit edilmesi veya saldırı sonrasında nasıl gerçekleştirildiğinin tespiti zorlaşmıştır. Saldırı yöntemlerinin dinamik yapılar kullanması, saldırı araçlarının bazı kısımlarının güncelleme imkanı saldırırganlara sunması bu gelişmelere verilebilecek örneklerdendir.

Gün geçtikçe daha çok güvenlik açığı keşfedilmektedir. Son yıllarda bulunan açıklık sayısı önceki yıllarda bulunan açıklık sayısını ikiye katlamıştır [CERT, 2002]. Bu sebeple sistem yöneticilerinin sistemlerini yamalara karşı güncel tutabilmeleri zorlaşmış ve daha çok güvenlik açıklığı saldırırganların istismarına sunulmuştur.

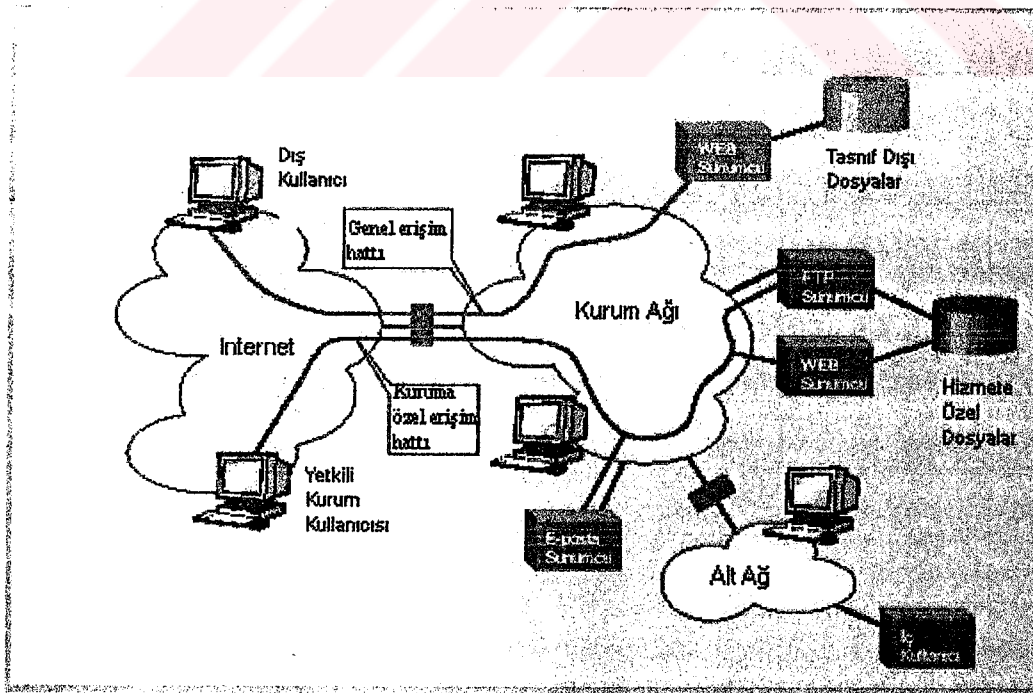
Güvenlik duvarı konfigürasyonlarını atlatmak yeni yöntemler bulunmuştur.

Altyapılara yapılan saldırılar artmıştır. Altyapı saldırıları, Internet üzerinde yer alan anahtar bileşenlere yapılmaktadır. Bu tür saldırılar ilgili bileşeni etkilemekle kalmayıp bu bileşenlere bağımlı diğer yapıları da etkilemektedir. Son yıllarda kök DNS sunucularına ve yönlendiricilere yapılan saldırılar, ve solucanlar kullanılarak yapılan saldırılar bunlara örnek olarak gösterilebilir.

Bu veriler göstermektedir ki, son yıllarda ağ üzerinden yapılan saldırıların sayısı oldukça artmıştır. Bunda kötü niyetli insanların çalışmalarının hız kazanmasının yanında, insanlarda güvenlik bilincinin eksikliğinin ve bu sebeple sistemlerin güvenli hale getirilmemesinin rolü büyüktür.

1.3. Güvenlik Duvarları ve Roller

Ağ güvenliğinde kullanılan en yaygın mekanizmalardan birisi de güvenlik duvarlarıdır. Güvenlik duvarı, sahip olunan ağı, bir ya da birkaç sunumcuyu veya basit bir kişisel bilgisayarı tehdit unsuru olarak görülen ağlardan korumak amacıyla kullanılan, bu bağlamda belirli fonksiyonları yerine getiren öğeler topluluğudur [Scuba and Spafford, 1997]. Diğer bir tanımla, iki ağ arasında belirlenmiş olan erişim kontrol politikasını uygulayan sistem ya da sistemler topluluğudur.



Şekil 1.1 : Güvenlik Duvarının kurumsal ağlarda örnek kullanımı

Şekil 1.1'de güvenlik duvarlarının örnek bir kullanımı gösterilmektedir. Bu örnekte, kurum ağı bir güvenlik duvarı ile dış dünyadan korunmaktadır. Güvenlik duvarı, Internet ağı üzerinden gelen trafiği, erişimde bulunmak isteyen kaynakların kimliklerine göre yetkilendirmektedir. İkinci güvenlik duvarı ise başka bir alt ağı kısıtlı haklarla kurumun ağına erişmesine imkan tanımaktadır.

Özellikle büyük ağların korunması probleminde güvenlik duvarları çok önemli bir rolü üstlenmektedir. Gün boyunca belki terrabyte'lara varan trafik ağı düğüm noktası olan güvenlik duvarlarından geçmektedir. Böyle stratejik noktada duran güvenlik duvarları genel anlamda şu özelliklere sahip olmalıdır [Muffett, 1994].

Güvenlik: Güvenlik duvarları sistem yöneticilerine hangi trafiğin geçeceğini hangi trafiğin ise engelleneceğini belirleme imkanı vermelidir. Bu noktada kullanılan protokol, kaynak ve hedef ağ adresleri, zaman, yetkilendirme seviyesi gibi parametreler geçerli olabilir.

Politika: Kurumda çalışan personelin iç ağdan başka bir ağa geçerken (özellikle Internet ağına) hangi haklara sahip olacağı ya da hangi sitelere bağlanabileceği ve hangi servisleri alabileceği bir politika dahilinde belirlenebilmelidir.

Denetleme: Bütün ağ trafiği üzerinden geçen güvenlik duvarları bu trafiğin istatistiğini tutabilmeli ve bu kayıtları anlaşılır bir biçimde sunabilmelidir.

Güvenlik duvarları bu özellikleri yerine getirirken bazı güvenlik modülleri kullanır. Bu modüllerden her biri saldırganların işlerini bir nebze daha zorlaştırır. Bu sebeple güvenlik duvarı seçimi esnasında sağlanan fonksiyonellikler iyice araştırılmalıdır. Yakın zamana kadar güvenlik duvarları sadece paket filtreleme görevini yerine getirirlerken artık yeni geliştirilen modüllerle birçok saldırıyı engelleyebilmektedir. Bu modüller şöyle sıralanabilir:

Paket Filtreleme: Kaynak ve/veya hedef IP adresi/alan ismi ve/veya port numarası ile herhangi bir ağdaki kaynaklara erişmek amacıyla kullanılır. Bu modül güvenlik duvarlarının sağladığı temel fonksiyonelliktir ve bu özelliğe sahip bir güvenlik duvarı düşük maliyetle edinilebilir.

Paket Etiketleme: Bu fonksiyon, TCP bağlantıların başlaması esnasında SYN bayrağını taşıyan başlangıç paketindeki bilgileri durum tablosunda tutar. Bir paket geldiği zaman öncelikle, bu pakete ait bir bağlantının daha önceden durum tablosu girilip girilmediği kontrol edilir. Şayet durum tablosunda bir girdi bulunursa paket

kabul edilir, eğer bulunamazsa filtreleme modülüne yollanarak sınanır. Bu özellik sayesinde filtreleme kuralları daha doğru tanımlanabilir. Örneğin 1055. porttan başlatılan ve dış ağdaki bir WEB sunumcunun 80. portunu hedefleyen bir bağlantı farzedilsin. Bu bağlantı için sadece dışarı çıkan trafik için 80 numaralı portun açılması yeterlidir. Çünkü bu pakete karşılık gelecek paket, durum tablosuna bakılarak kabul edilecektir. Aksi halde WEB sunumcusunun iç ağdaki istemcinin 1055. portuna bağlanmasına izin verilmesi gerekecekti.

Parçalı Paket İşleme: Birçok atak parçalı paketlerde yapılan oynamalar ile gerçekleşmektedir. Parçalı paketler düzgün bir şekilde işlenirse bu ataklar engellenebilir. Bazı güvenlik duvarları bu fonksiyonelliği sağlayarak bozuk paketleri engelleyebilir.

Ağ Adres Dönüşümü (NAT): Bu özellik, iç IP adreslerini dış dünyadan korumak için kullanılır. İç ağlarda genellikle sanal IP adresleri denilen 192.168.0.0/24, 10.0.0.0/8 gibi adresler kullanılır. NAT özelliği ile bu adres, güvenlik duvarı üzerinden geçerken gerçek değerlerine dönüştürülür. Birçok durumda, iç ağa sadece izin verilen(web, dns, e-posta gibi) yönlendirmelerin yapılmasına izin verilir. Bu şekilde NAT, sanal olarak iç ağı dış dünyadan izole eder. Başlığın tekrar yazılması olarak da nitelendirilen bu işlem port numaraları için de gerçekleştirilebilir [Frantzen, 2001]

Uygulama Filtreleme: Bu fonksiyon genellikle tek bir bilgisayarı koruyan güvenlik duvarlarına adapte edilir ve bağlantı izni verilmeyen uygulamaların istenmeyen bağlantıları kurmalarını engeller. Bu şekilde bilgi açığa çıkarma gibi saldırıları engellenmiş olur. Örneğin bir bilgisayarda e-posta istemcisi olarak Netscape programının kullanıldığı farzedilsin. Bu durumda güvenlik duvarı, 25 numaralı porta sadece Netscape yazılımının bağlanmasına izin verecek şekilde ayarlanır. Bilgisayar üzerinde bir casus yazılım bu porta bağlantı kurmaya çalıştığında ise kullanıcıları uyararak bilgi kaçırılmasını engeller.

Özel Sanal Ağlar: Günümüzde bazı güvenlik duvarları VPN kabiliyetine sahiptir. Basit ifadesiyle VPN, bir veya birkaç bilgisayardan oluşan ağları güvensiz ağlar üzerinden güvenli bir şekilde bağlama amaçlı kullanılır. VPN bağlantılarında standart IP paketleri şifrelenerek kapsüllenir. Böylece ağ üzerinden akan verinin gizliliği, bütünlüğü ve kaynağının inkar edilememesi sağlanmış olur.

7

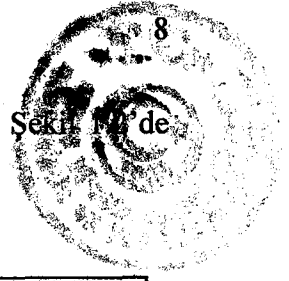
Savunma Modülleri: Geçmişte, güvenlik duvarları pasif güvenlik mekanizmaları olarak bilinirdi. Fakat günümüzde bazı aktif modülleri de içermektedir. Genellikle güvenlik duvarlarına entegre edilen savunma modülleri IP sahteciliği ve bağlantı seli tespit modülleridir. Birinci modül kaynağı inkar edememeyi sağlayıp IP adresi ile MAC adresi uyumsuzluklarını tespit ederken ikincisi, bir kaynaktan gelebilecek anormal sayıdaki bağlantıyı engelleyerek servis dışı bırakma saldırılarını önlemeyi amaçlar.

Vekil Sunumculuk ya da Vekil Sunumcu Yönlendirmesi: Vekil sunumcuları istemci ile sunumcular arasında tampon olarak bulunur [Walder, 2001]. Bir istemci sunumcudan bir servis alacağı zaman öncelikle vekil sunumcuya bağlanır vekil sunumcu ise gerekli bilgiler kaşesinde yoksa sunumcuya bağlanarak bu bilgileri alır ve istemciye iletir. Vekil sunumcu kullanımında gerek iç ağdan dışa gerekse dış dünyadan iç ağa direk bir bağlantı kurulmaz. Bu özellik birçok güvenlik açığını engelleyebilir. Bununla beraber trafik üzerinde bulunabilecek virüs, programcık (java, activex vs.) arka kapı gibi programları da filtreleyebilir. Ayrıca güvenlik duvarlarına üst seviye kimlik doğrulama mekanizması sağlar. Örneğin dış dünyadaki bir kullanıcı iç ağa telnet servisi ile erişmek istediğinde öncelikle vekil sunumcuya kimliğini doğrular daha sonra telnet konsolundan kullanıcı ismi ve şifresini girer. Güvenlik duvarları vekil sunumcu özellikleri taşıyabilecekleri gibi başka vekil sunumcularla birlikte de çalışabilir. Örneğin e-posta trafiği virüs taraması yapan ayrı bir vekil sunumcuya iletebilir. Bu kullanım performans açısından daha uygundur. Çünkü vekil sunumcuda gerçekleşen güvenlik prosesleri sebebiyle oluşabilecek yoğunluk diğer trafikleri yavaşlatmaz.

Denetleme: Denetleme güvenlik duvarlarının ana özelliklerinden birisidir. Kayıt tutma, yapılan son aktivitelerin izlenebilmesi açısından önemlidir. Güvenlik duvarı seçilirken kayıt tutma kabiliyeti göz önünde bulundurulmalıdır. Kayıt tutma kabiliyeti tanımlanan her bir kural için ayarlanabilmeli böylece gereksiz girdilerin kayıt dosyalarını şişirmeleri engellenmelidir. Ayrıca anlaşılabilir formatta olmalı, bağlantı zamanı, kaynak ve hedef adresler ve port numaraları gibi gerekli bütün bilgileri içermelidir.

Marketteki her güvenlik duvarı bu fonksiyonlardan hepsini ya da alt kümesini gerçekler. Bu özellikler genellikle konfigürasyon dosyasında geçen belirtkeler

tarafından sağlanır. *IPFilter* güvenlik duvarının örnek bir kural satırı Şekil 1.2'de görülmektedir:



pass in log on rll proto tcp from 192.168.88.0/24 to any port 80 flag S keep frags keep state.

Şekil 1.2: *IPFilter* güvenlik duvarı örnek kural tanımı

Bu kuralın yorumu şu şekildedir [Conoboy and Fichtner, 2001]: *rll* güvenlik duvarının ethernet ara birimlerinden birisidir ve kural bu ara birim için geçerlidir. Ethernet ara birimlerinin etiketlenmesi platformlara bağlıdır. *rlx* etiketlemesi genellikle BSD tabanlı işletim sistemlerinde kullanılmaktadır. Kuralda 192.168.88 ile başlayan IP adreslerinden gelen trafikten *SYN* bayrağı set edilmiş olanlara geçirme izni verilmektedir. Gelen paketler parçalanmış ise bu paketler yakalanacak doğruluğu kontrol edilecektir. Ayrıca *SYN* bayrağı set edilmiş paket ile başlatılan bağlantı durum tablosunda tutulacaktır. Bu kuraldan güvenlik duvarının NAT özelliğinin bulunduğu da anlaşılmaktadır. Çünkü 192.168.0.0/12 IP adresi grubu Internet ortamında geçerli değildir. Bu adresler Internet'e çıkarken gerçek IP adreslerine çevrilecektir. Son olarak *log* belirtmesi ile bu kurala karşılık gelen trafiğin kaydının tutulması sağlanmaktadır.

1.4. Güvenlik Testine Olan İhtiyaç

Güvenlik Duvarı, bir kurumsal ağ için en önemli düğüm noktalarından birisidir. Bu düğüm noktasının saldırılarda nasıl kullanılabileceği Bölüm 2.2'de örneklerle anlatılmaktadır. Eğer güvenlik duvarı üzerinde açıklıklar bulunmuyorsa saldırganların bu engeli aşip yetkisiz erişim kazanması mümkün olmayacaktır. Bu sebeple saldırı profilleri iyi anlaşılmalı ve sistemlerin güvenliğinden sorumlu kişiler tarafından tatbik edilerek açıklıklar belirlenmelidir. Bu süreç, el yordamıyla ve sistem yöneticilerinin bu konudaki bilgi birikimleri ile gerçekleştirilebileceği gibi atak profillerini kullanan otomatik araçlar vasıtasıyla da yerine getirilebilir.

Güvenlik Duvarları, saldırıları engellemek için en stratejik noktada bulunmanın yanı sıra çoğu zaman bu işlevi üstlenen tek mekanizma olarak kullanılmaktadır.

Bilindiği gibi güvenlik, çeşitli seviyelerde sağlanan ve maliyetle doğru orantılı olan bir kavramdır. Yerel ağlarda ve geniş alanlarda sağlanabilecek güvenlik de maliyeti nispetinde üst seviyelerde olabilmektedir. Günümüzde kabul edilebilir bir güvenlik birçok güvenlik aracının ağlara entegre edilmesi ve tecrübeli personellerin (güvenlik denetleyicileri ve ağ yöneticileri) önceden hazırlanmış poliçeleri uygulayarak, düzenli denetimler yapmasıyla aktif olarak sağlanabilmektedir. Öte yandan böyle bir güvenliğin ancak yüksek harcamalarla sağlanabilir olması güvenlikten bazı tavizler verilmesine sebep olmaktadır.

Bu sebeple çoğu kurum ağ güvenliklerini sadece güvenlik duvarı kullanarak sağlamaktadır. Güvenlik duvarı üzerindeki herhangi bir eksiklik, fonksiyonel yetersizlik ya da güvenlik açığında yetkisiz erişim, bilgi kaçırma, servis dışı bırakma gibi saldırılarla karşılaşılabilir. Özellikle güvenlik duvarı dışında bir güvenlik önlemi olmadığı düşünülecek olursa bu eksikliklerin ne derece güvenlik riskine sebebiyet verebileceği ortadadır.

Sonuç olarak, güvenlik duvarı testleri önemli ve ihmal edilmemesi gereken bir süreçtir. İhmalinde güvenli kabul edilen ve kritik bilgi taşıyan ağlar tehdit altında kalabilmektedir. Her bir güvenlik duvarı kullanıldığı ağa ve yerine getirmesi beklenen fonksiyonlara göre test edilmeli, düzgün çalıştığı onanmalıdır.

Günümüzde bu amaçla geliştirilmiş bazı yöntemler ve metodolojiler bulunmaktadır. Detayları ve ne derinlikte inceleme yaptıkları değişkenlik göstermektedir. Kullanılan basit yöntemlere port tarama araçları örnek gösterilebilir. Her ne kadar bu araçlarla taramalar filtreleme kuralları hakkında bilgi verse de güvenlik duvarlarını test etmek için yeterli değildir.

Güvenlik duvarı testlerinde kullanılan daha kapsamlı ve detaylı metodolojiler de bulunmaktadır. Bu metodolojiler ayrıntılı bir şekilde Bölüm 2'de ele alınacaktır. Özetle değinilecek olursa, bu metodolojilerden bazılarının test laboratuvarlarında bazılarının ise aktif ağlarda kullanılabildiği görülmektedir. Birçok faktörü göz önünde bulundurarak güvenlik duvarlarını incelemekte ve belirli kriterlere göre karşılaştırma imkanını sunmaktadırlar. Öte yandan güvenlik duvarlarında bulunabilecek açıklıkları tam olarak kapsamadıkları, hızlı ve az bilgi birikimiyle gerçeklemedikleri için kullanışlı değildir. Güvenlik duvarı testlerindeki bu ihtiyacı karşılamak amacıyla NetCheck test metodolojisi geliştirilmiştir.

1.5. NetCheck Güvenlik Duvarı Test ve Analiz Metodolojisi

İlerleyen bölümlerde detaylı bir şekilde değinilecek NetCheck güvenlik duvarı test ve analiz metodolojisi güvenlik duvarları üzerinde bulunan açıklıkları test etmek ve bu açıklıkları kapatmak için geliştirilmiştir.

NetCheck basit ve esnek yapısı ile güvenlik konusunda çok bilgisi olmayan sistem yöneticilerinin bile rahatlıkla kullanabileceği bir yapı sunar. Bu test metodolojisinde öncelikle güvenlik duvarları üzerinden yapılabilecek saldırılar araştırılmıştır. Daha sonra bu saldırılara özel bilgileri temsil edebilecek bir model ortaya konulmuştur. Modelin kullanıcılar tarafından kolaylıkla kullanılabilmesi için örnek test modülleri tanımlanmıştır.

NetCheck ile yapılan testlerde hangi test modüllerinin kullanılacağı, sonuç raporunun yapısı , topoloji bilgileri görsel bir kullanıcı ara yüzü üzerinden ayarlanır. Örnek test yatakları oluşturulup kaydedilebilir, daha sonra tekrar kullanılmak üzere NetCheck'e yüklenebilir.

Test sürecinin başında NetCheck, test yapılacak ağa entegre edilir. Güvenlik duvarını çevreleyen topoloji hakkında bilgi alınır. Bu bilgiler genel olarak tanımlanmış test modüllerinin modifikasyonlarında kullanılır. Test modülleri en son halini aldıktan sonra test başlatılır.

Sonuçlar *html* formatında kullanıcılara sunulur. Sonuç raporu, bulunan açıklıklara karşılık gelen önerileri içerir. Raporun en sonunda bulunan tablo farklı zamanda yapılan testleri karşılaştırabilme imkanını sağlar.

2. GÜVENLİK DUVARLARI İÇİN REFERANS MODEL VE TEST METODOLOJİLERİ

2.1. Güvenlik Duvarları İçin Referans Model

2.1.1. Tanım

Güvenlik Duvarları basit anlamda iki ya da daha fazla ağ arasında kurulan özel ağların korunması amacıyla yerleştirilen öğeler topluluğu olarak tanımlanır [Al-Tawil and Al-Kaltham, 1999]. Öte yandan, bu teknik bir tanımdır ve güvenlik duvarlarının akademik olarak tanımlanmasına ihtiyaç vardır.. Referans modelde göre güvenlik duvarları şu şekilde tanımlanır:

Güvenlik duvarı teknolojisi, bir ağ politika alanına giren ve bu alandan çıkan iletişim trafiği üzerinde ağ alan güvenlik politikasını tatbik eden mekanizmalar topluluğudur. [Scuba and Spafford, 1997]

Bu tanımlama günümüzdeki güvenlik duvarı teknolojisini kapsamaktadır. Bununla beraber, uç noktalar arasındaki kontrol edilebilir veri iletim hattı üzerinde yerleştirilen dağıtık yapıdaki güvenlik duvarı teknolojilerini de kapsar. Tanım içerisinde geçen terimler şu şekilde açıklanabilir [Scuba and Spafford, 1997].

Ağ, bilgisayarların ve diğer elektronik cihazların haberleşmesine imkan veren iletişim altyapısıdır.

İletişim trafiği(T), ağ üzerinden geçen bilgilerdir. Ağ trafiğinin herhangi bir altkümesi *iletişim birimi* olarak tanımlanır. $kayıt(ctrl, veri)=t \in T$ şeklinde gösterilir. *ctrl* kontrol bilgisini *veri* ise gönderilen veriyi ifade eder. İletişim birimi teriminin karşılığı protokole göre değişir. OSI modelinde *datagram* olarak ifade edilir.

t.ctrl alanı, kaynak (*t.ctrl.kynk*) ve hedef (*t.ctrl.hdf*) IP adreslerini, güvenilirlik⁴ (*t.ctrl.gvn*) ve akış kontrol bilgilerini (*t.ctrl.akış*), erişim istek bilgilerini (*t.ctrl.erşm*) ve servis kalitesini(QOS) (*t.ctrl.qos*) içerebilir. *t.veri* alanı ise uygulamaya özel bilgileri ve üst seviye verileri içerebilir. Ağ üzerinden gönderilen iletişim birimlerinde her alan bilgi içermeyebilir. Örneğin, kontrol mesajlarında (ICMP)

⁴ Reliability

$t.veri$ alanına, önceden kurulmuş olan bir bağlantıda ise $t.ctrl.qos$ alanına ihtiyaç yoktur.

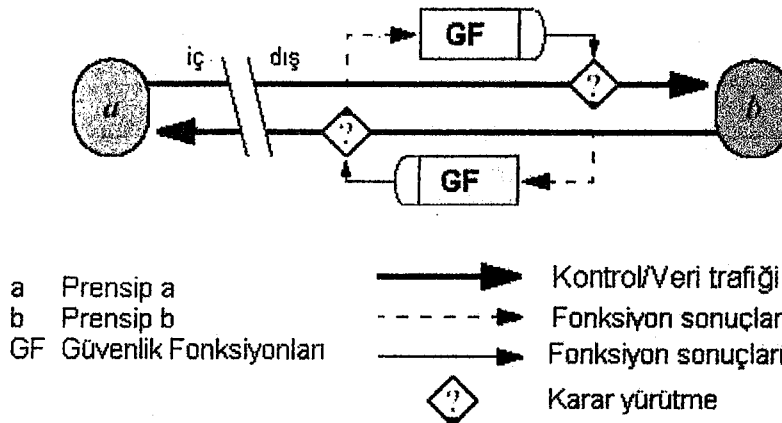
Güvenlik politikası, bir sistemin güvenlik gereksinimlerinin tanımıdır. Bir grup standartlar topluluğu, kurallar ya da pratik uygulamalar şeklinde belirlenebilir. **Ağ alan güvenlik politikası(P)** ise güvenlik politikasının bir alt kümesini oluşturur ve $t \in T$ için kimlik doğrulama gereksinimlerinin ve bütünlüğün sağlanmasını hedef alır. Bir istek için kimlik doğrulama gereksinimleri $istek(t.ctrl.kynk, t.ctrl.hdf, t.ctrl.erşm)$ dır.

Ağ politika alanı (A) ise ağ alan güvenlik politikası(P) ile idare edilen birbirine bağlı ağların, ağ geçitlerinin ve servis sunan ya da alan bilgisayarların topluluğudur.

2.1.2. Tanımların Referans Modele Entegrasyonu

Bölüm 1.3'de açıklandığı gibi güvenlik duvarları birçok fonksiyonelliğe sahiptir. Bu sebeple tam bir referans model oluşturmak kolay değildir. Scuba, günümüz güvenlik duvarı teknolojisini kapsayan ve ağ teknolojilerinde olabilecek gelişmelere açık bir referans modeli sunmuştur [Scuba and Spafford, 1997]. Bu model geliştirilecek ya da doğrulanacak güvenlik duvarı mekanizmaları için ana-çatı olarak kullanılabilir.

Şekil 2.1 referans model için genel bir bakış sunmaktadır. (Daha detaylı bir temsil için bkz. Şekil 2.2)

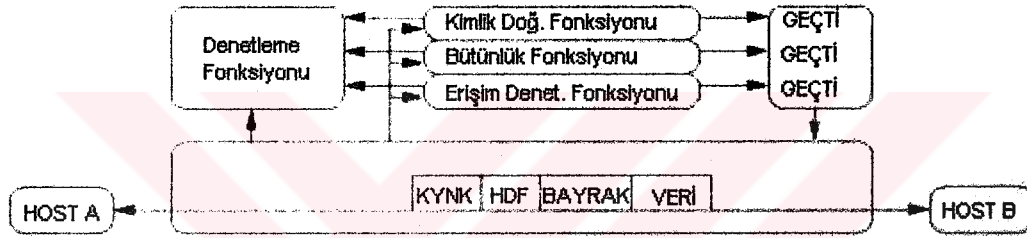


Şekil 2.1: Referans modele genel bakış [Scuba and Spafford, 1997]

Şekil 2.1'de, GF ile etiketlenen kutular, a ve b noktaları arasında gidip gelen iletişim birimlerine uygulanan güvenlik fonksiyonları topluluğunu temsil etmektedir.

Kesik çizgili oklar, bu fonksiyonların aktive edilmesini simgelemektedir. Her bir GF, iletişim birimlerin hepsini ya da bir bölümünü girdi olarak alır. GF'ler her bir iletişim birimi için GEÇTİ yada KALDI şeklinde bir sonuç üretir. “?” işaretli karo ise bu kararı temsil eder. Eğer sonuç KALDI ise kural dışı durum oluşur ve iletişim birimi tercihe göre log dosyasına kayıt alınarak ya da alınmayarak düşürülür.

Şekil 2.2, GF'leri daha detaylı göstermektedir. Burada yapı, denetleme fonksiyonu da içermektedir (DF). DF, iletişim hattında fonksiyon çağırıcısı olarak görev yapmaktadır. Her bir iletişim biriminin kimlik doğrulamasını, bütünlüğünün onaylanması ve erişim kontrol haklarını isteyebilir ve bu fonksiyonlarının sonuçlarının uygulanmasını zorunlu kılabilir.



Şekil 2.2: Güvenlik duvarı teknolojisi için referans model

Kimlik doğrulama, bir girdinin iddia edilen kimliğe sahip olup olmadığını analiz eder. Görünen ve asıl orijinlerin uyumluluğunu kontrol eder. İç ağa doğru gelen trafik için kimlik doğrulama mekanizması uygulanmaması, IP adresi sahteciliği gibi saldırılara imkan verir. Bununla beraber erişim denetimi kontrol mekanizmaları, istekte bulunan kaynakların kimlikleri doğru değilse yanlış sonuçlar üretebilir. Bu sebeple kimlik doğrulama fonksiyonu önemlidir. Bu fonksiyonun üreteceği sonuç KF_GEÇTİ ve KF_KALDI şeklindedir.

Bütünlük fonksiyonu, iletişim trafiğinin yetkisiz ve habersiz bir şekilde değiştirilmesi ile ilgilidir. Bu fonksiyon bu tür kötü niyetli davranışların olmasını engelleyemez ama bütünlüğü bozuk paketlerin tespit edilmesini ve işaretlenmesini sağlar. Üreteceği sonuç BF_GEÇTİ ve BF_KALDI şeklindedir. Bağlantı hırsızlığı gibi saldırılar, iletişim birimlerine bütünlük fonksiyonunu uygulamayan güvenlik duvarlarında yapılabilmektedir. Bu sebeple bütünlük fonksiyonu, modifikasyon tabanlı saldırılara karşı gerekli bir önlemdir.

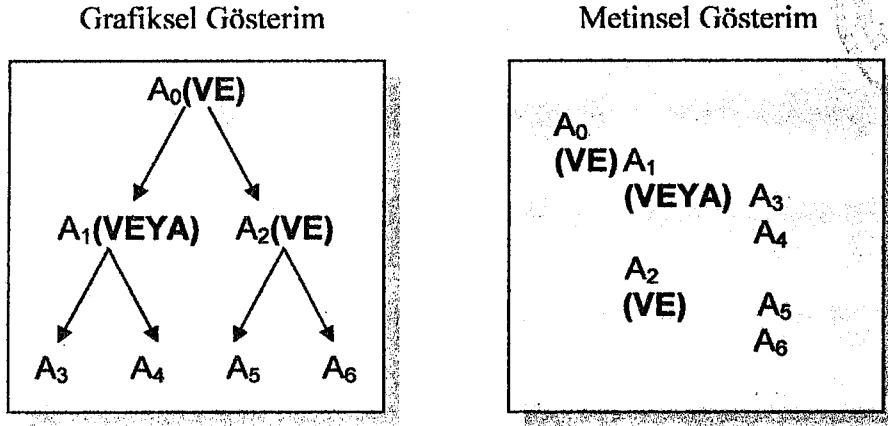
Denetleme fonksiyonu, önemli sistem olaylarının aralıksız ve sıralı bir şekilde kaydedilmesi kabiliyetini sağlar.

Scuba'nın referans modeli, belirli bir güvenlik duvarı anlayışını empoze etmenin yerine temel ve genel fonksiyonları içerir. Bu sebeple güvenlik duvarı test metodolojisi geliştirilmesi esnasında kullanımı oldukça faydalı olabilecek bir yapıdır. Bu sebeple "3.NETCHECK GÜVENLİK DUVARI TEST VE ANALİZ METODOLOJİSİ" bölümünde geliştirilen test metodolojisi için temel teşkil edecektir.

2.2. Saldırı Mekanizmaları

Ağ üzerinde yapılan saldırıların trendi günümüzde oldukça yükselmesine rağmen bunun önüne geçebilmek mümkündür. Bu amaçla geliştirilen birçok güvenlik mekanizması mevcuttur. Bu saldırıların engellenebilmesi amacıyla bazı ülkelerde devlet güvenlik politikaları oluşturulmuş ve saldırılara karşı nasıl önlem alınması gerektiği belirlenmiştir [CERT, 2000]. Öte yandan, maruz kalınabilecek saldırılara engel olabilmek için öncelikle saldırganların hangi sistematığa göre hareket ettiklerinin farkında olunmalıdır.

Modelleme, ağ üzerinden gerçekleşen atakları belirlemede başvurulan yöntemlerden birisidir. En yaygın kullanılan modelleme yöntemi ise atak ağaçlarıdır [Moore et al, 2001]. Atak ağaçları, atak hakkındaki verileri sınıflandırır. Ağaçların köklerinde saldırıların hedefleri yer alırken bu hedefe nasıl ulaşılabileceğini en alt seviyedeki yapraklara giden yollar belirler. Kök ağaçları grafiksel ifade edilebileceği gibi metin olarak da gösterilebilir. Şekil 2.3'de örnek bir atak ağacı grafiksel ve metin olarak örnek verilmiştir.



Şekil 2.3: Atak ağaçlarının grafiksel ve metinsel gösterimi

Atak ağaçlarında *VE* ve *VEYA* işlemleri geçerlidir. Bir saldırının gerçekleşmesi için tanımlı atak basamaklarının hepsinden geçilmesi gerekiyorsa *VE*, sadece birinin tercih edilip uygulanması yeterli ise *VEYA* işlemi kullanılır. Bir atak ağacında her iki operatör de yer alabilir. Örneğin Şekil 2.3’de gösterilen atak ağacında saldırının başarılı sonuçlanması, yani ağacın kökü olan A_0 ’a ulaşılabilmesi için (A_3, A_5, A_6) veya (A_4, A_5, A_6) yolları izlenmelidir.

Atak ağacı modellemesi, Internet’e bağlı ağların maruz kalabilecekleri tehditleri üst seviyede belirlemede oldukça faydalıdır. Örneğin Şekil 1.1’deki kurumsal ağı ele alındığında, bu ağda bulunan ve Internet’e hizmet veren WEB sunumcu üzerinde kötü niyetli kişilerin yetkisi yüksek erişim hakkı kazanmaları atak ağaçları ile temsil edilebilir. Şekil 2.4 bu temsili göstermektedir. Saldırgan öncelikle kurum ağını ve bu ağı koruyan güvenlik duvarını tespit etmekte daha sonra güvenlik duvarının açık portlarını kullanarak web sunumcunun ve üzerinde koştugu işletim sisteminin versiyonunu öğrenmeye çalışmaktadır. Son basamakta ise bilinen açıklıkları deneyerek saldırının amacına ulaşmaktadır.

Kurum WEB Sunucusuna Yetkisiz Erişim Kazanılması

- (VE) 1.Kurum alan ismini belirle
 2.Güvenlik Duvarının IP adresini belirle
 (VEYA)1.Alan isim sunucusunu sorgula
 2.Güv. Duvarını bulmak için ağı tara.
 3.Web sunucusuna güvenlik duvarı üzerinden traceroute çek.
 3.Güv. duvarının erişim kontrol listesi belirle.
 (VEYA)1.Dinleyen belirli portları araştır.
 2.Bütün portları tarayarak dinleyenleri belirle.
 4.WEB sunumcunun ve üzerinde koştuğu işletim sisteminin çeşidini belirle.
 (VEYA)1.İşl. Sis.'de koşan servislerin karşılama mesajlarını tara
 2.TCP/IP yığını İşl. Sis.'ne ait karakteristik bilgileri bulmak için gözle
 5.WEB sunumcunun açıklıklarını sömür.
 (VEYA)1.Hassas lokal ağ kaynaklarına direk eriş
 2.Hassas verilere WEB sunumcusu üzerindeki yetkisi yüksek hesaplardan ulaş

Şekil 2.4: WEB Sunumcu üzerinde yetki kazanma saldırısının atak ağacı ile modellenmesi [Moore et al, 2001]

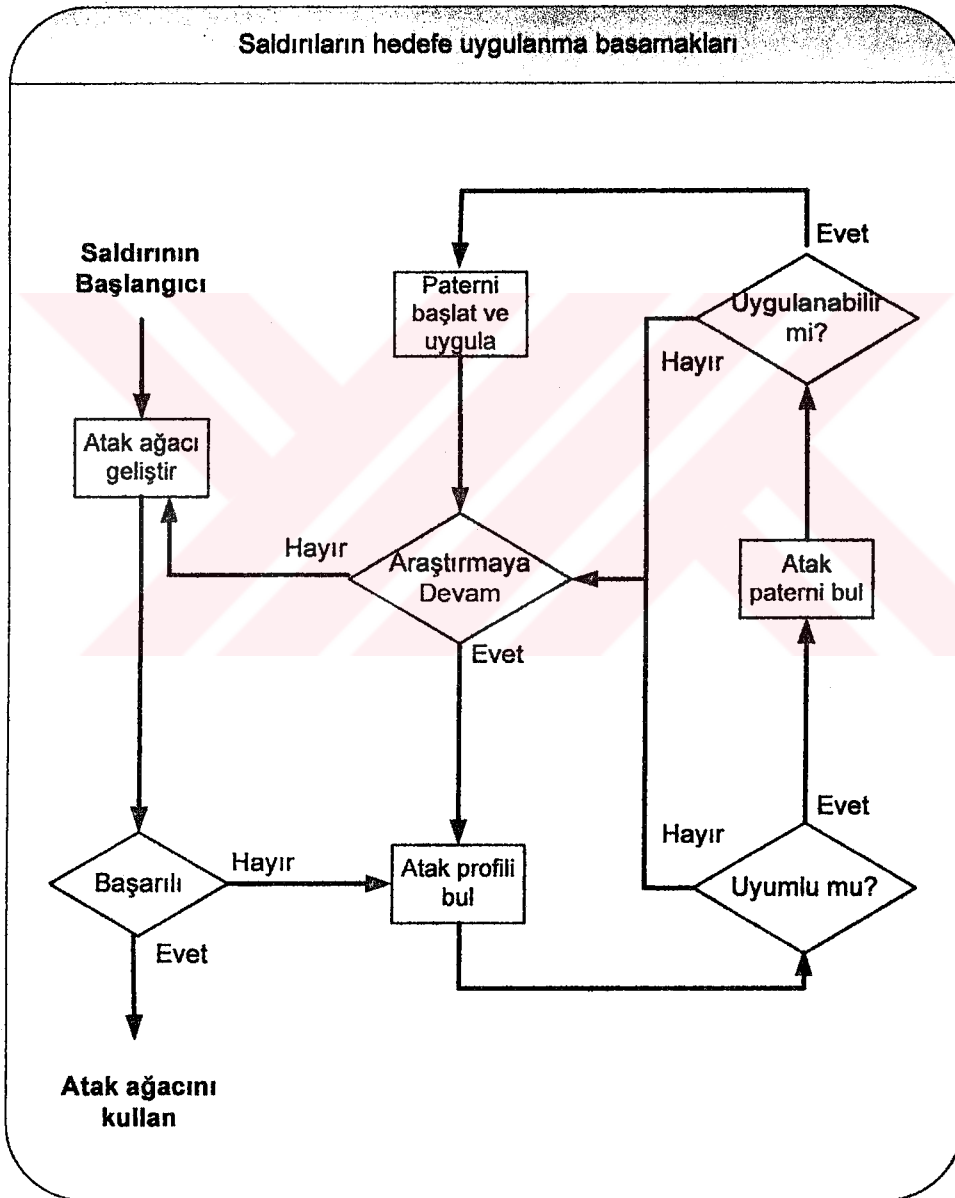
Belirlenen atak ağaçları ya da bir diğer deyişle atak şablonları ortak paydalar altında sınıflandırılarak *atak profillerini* oluşturur. Atak profilleri

- Genel bir referans model (Saldırının uygulanabileceği örnek mimariler)
- Değişkenler dizisi
- Atak şablonları
- Tanımlanan terimler ve deyimlerin sözlüğü

içerir. Referans model mimari şablonları temsil eder ve sistemlere özel bazı değişkenleri de içerebilir. Ayrıca atak şablonları da bazı değişkenler içerebilir. Bu değişkenler güvenlik duvarının güvenilirlik seviyesi, sistemin yapısı vb. gibi değerlerdir. Atak profilleri belirlenirken saldırıların çeşitleri, sistemler üzerinde elde edilen hakların seviyeleri gibi değişik etkenler göz önünde bulundurulabilir.

Bölüm 1.2'de bahsedildiği gibi günümüzde ağ üzerinden yapılan saldırıların trendinde ciddi bir artış gözlenmektedir. Bunda saldırganların yüksek seviyede bilgiye sahip olmalarının ve yukarıda tanımlanan saldırı modellerini uygulamalarının

rolü büyüktür. Bir saldırgan daha önce tanımlanmış olan saldırı modellerini, ki bu saldırı modellerine Internet'ten erişmek mümkündür, kullanabileceği gibi kendi geliştirmiş olduğu atak ağaçlarını kullanarak da emeline ulaşabilir. Şekil 2.5 saldırganların kullanmış oldukları metodolojiyi özetlemektedir. Saldırı başlamadan önce en büyük çaba araştırma yönünde harcanmaktadır. Bu araştırmalar atak profillerini tespit etme noktasında olabileceği gibi bu profillerde geçen değişkenlerin hedef ağa uyumlu olup olmadığını gözlemleme şeklinde de olabilir.



Şekil 2.5: Saldırıların hedefe uygulanma basamakları

Atak ağaçlarının yapılarından da anlaşılabilceği gibi *VE* operatörünün

görüldüğü noktalarda saldırganların ilerleme kat edebilmeleri, operatörün kapsamında olan adımları gerçekleştirmeleri ile mümkündür. Diğer bir deyişle bu noktalarda yapılabilecek güvenlik ayarlamaları, saldırganları durdurma noktasında etkili olacaktır. Güvenlik duvarları VE operatörünün kullanıldığı en kritik noktalardan birisidir. Bu sebeple güvenlik duvarlarının ne kadar güvenli olduğu, doğru ayarlanıp ayarlanmadığı ve üzerinde kullanıldığı topolojik yapı incelenmeli ve bulunan açıklıklar kapatılmalıdır.

2.3. Güvenlik Duvarı Test Yöntemleri

Güvenlik testlerinin amacı sistemde bulunan açıklıkları tespit edebilmek ve kısa sürede bu açıklıkları kapatmaktır [Wack and Trasey, 2002]. Bu açıklıklar bulunurken bir önceki bölümde açıklanan temel fonksiyonların düzgün çalışıp çalışmadığı ya da doğru ayarlanıp ayarlanmadığı kontrol edilir.

Günümüzde güvenlik duvarlarını test etmek için değişik yöntemler geliştirilmiştir. Bu yöntemlerin çoğu güvenlik duvarlarının sadece tanımlanan filtreleme politikasında eksiklik olup olmadığını inceler. Bu amaçla port tarama araçları kullanılır. Port tarama araçları hedef makineye bağlantı kurma denemelerinde bulunarak hangi portların açık olduğunu tespit ederler. Gelişmiş tarama araçları hedef bilgisayarda açık olan servisleri hangi programların çalıştırdığını, bu programların sistemdeki sahiplerini, hatta gönderilen paketlere verilen cevaplar değerlendirilerek işletim sistemin ne olduğu bulunabilir.

Nmap programı port tarama araçlarına örnek gösterilebilir. Nmap programı komut satırından çalıştırılır. Yapılacak taramanın hangi derinlikte olacağı verilen parametrelerle belirlenir. Örnek bir port taraması Şekil 2.6'de görülmektedir. Tarama nmap'in üzerinde çalıştığı bilgisayar için yapılmıştır. Taranan port aralığı 1 ile 1024. olarak belirlenmiştir. İşletim sistemi tahmininin yapılması istenmiş fakat nmap bir sonuç döndürememiştir. Tarama sonuçlarında açık olan servislerin sahibi olan kullanıcıların isimleri listelenmiştir.

```
# Nmap (V. nmap) scan initiated 2.53 as: nmap -p1-1024 -I -O -sR -oN log.txt localhost
Interesting ports on localhost (127.0.0.1):
(The 1014 ports scanned but not shown below are in state: closed)
Port      State  Service (RPC)          Owner
21/tcp    open   ftp                    root
22/tcp    open   ssh                    root
25/tcp    open   smtp                    root
53/tcp    open   domain                  named
80/tcp    open   http                    wwwrun
110/tcp   open   pop-3                  root
111/tcp   open   sunrpc (rpcbind V2)    bin
113/tcp   open   auth                    root
515/tcp   open   printer                 root
901/tcp   open   samba-swat              root

TCP Sequence Prediction: Class=random positive increments
Difficulty=1019830 (Good luck!)
No OS matches for host (If you know what OS is running on it, see
http://www.insecure.org/cgi-bin/nmap-submit.cgi).
# Nmap run completed at Wed Aug 22 16:19:21 2001 -- 1 IP address (1 host up) scanned
in 13 seconds
```

Şekil 2.6: Nmap programı örnek çıktısı

Güvenlik duvarlarının arkasındaki bilgisayarlar, port tarama araçları ile taranırken bazen yanlış sonuçlar elde edilebilir. Çünkü port tarama araçları, bağlantı isteklerine hedef bilgisayardan gelen cevaplara göre portun durumuna karar verebilirler. Eğer bağlantı isteğine gelen cevap güvenlik duvarı tarafından engelleniyorsa bu durumda servis açık olsa da bunu tespit edemezler.

Port tarama araçları ile güvenlik duvarlarının filtreleme politikalarını belirlerken de durum böyledir. Bu sebeple güvenlik duvarları bağlantı izni verilen portları ve protokolleri tespit etmek için taranırken hedef ağ diliminde paketleri yakalayan bir araç çalıştırılmalıdır. Tarama işlemini başlatan kaynağın IP adresinden gelen paketler incelenmelidir [Pallar, 2001]. Böylelikle izin verilen bağlantılar doğru bir şekilde belirlenebilir.

Bazı metodolojilerde, port tarama araçları güvenlik duvarına özel bir yapıya büründürülerek ek fonksiyonlar eklenir. [Kurt ve Soğukpınar, 2002] [Mayer et. al, 2000]. Bu araçlarda geniş bir IP aralığı taranırken kurulan bağlantılarda servislerin açılış pankartları yakalanır ve tarama sonuçları içerisinde sunulur. Böylece, servisleri çalıştıran programların ismi, versiyon numaraları hakkında bilgi edinilmiş olur.

Port tarama araçları her ne kadar kullanımı kolay ve hızlı sonuç veren programlar olsa da güvenlik duvarlarını test etmek için yeterli değildir. Çünkü:

- Port tarama araçları güvenlik duvarlarının içerebileceği açıklıkları tespit

edemezler. Bu açıklıklar güvenlik duvarının işlevselliğini etkileyeceği gibi güvenlik duvarı üzerinde haklar edinmeye, yetkisiz erişimlere olanak tanıyabilir. Özellikle güvenlik duvarlarının üstlendikleri kritik görev düşünüldüğünde incelenmesi gerek en önemli konunun bu açıklıkların bulunması olduğu ortadadır.

- Güvenlik duvarlarının kullanımında güvenliği etkileyen faktörlerden birisi de topolojik konumdur. Ağ dilimlerinin güvenlik seviyelerine göre belirlenmesi birbirlerinden izolasyonu gibi konular topolojik yapı içerisinde ele alınır. Güvenlik duvarı test metodolojileri topolojik konumu incelemelidir.
- Port tarama araçları standart bağlantı paketleri yollarlar. Örneğin IP başlığında bulunan sağlama değerinin yanlış olması durumunda güvenlik duvarının nasıl davrandığı bu tür araçlarla tespit edilemez.

Port tarama tabanlı test yöntemlerinin yukarıda geçen eksiklikleri düşünüldüğünde kapsamlı metodolojiler tanımlanması ihtiyacı ortaya çıkmıştır. Bu amaçla bazı metodolojiler geliştirilmiştir. Bunlardan bazıları güvenlik duvarına özel olmayan, genel bir metodolojiyle süreci birkaç adımda özetler [Herzog, 2001]. Bazıları ise detaylı bir yöntem sunar.

Bu bölümün ilerleyen kısımlarında günümüze kadar geliştirilmiş olan test metodolojilerinden ikisi üzerinde durulacaktır. Test metodolojileri incelendiğinde bazılarının test ortamlarında bazılarının ise gerçek ortamlarında uygulanabilir olduğu görülmektedir [CERT, 2000]. Açıklanacak metodolojiler de bu iki çeşide birer örnek teşkil etmektedir. Bölüm 4’de ise bu metodolojiler, NetCheck güvenlik duvarı test metodolojisi ile karşılaştırılacaktır.

2.4. Al-Tawil ve Al-Kaltham’ın Metodolojisi

Bu metodoloji, güvenlik duvarlarının, aktif olarak kullanıldığı ağlardan çok test ortamlarında test edilmesi için kullanışlıdır. Güvenlik duvarı üzerinde bazı konfigürasyon değişiklikleri yapılarak test adımları çalıştırılmakta, bu sebeple güvenlik duvarına devamlı çalışma ortamı sağlanamamaktadır [Al-Tawil and Al-Kaltham, 1999]. Test metodolojisi uygun bir test yatağı seçimi ile başlar. Kurulabilecek birkaç test yatağı çeşidi mevcuttur. Örneğin bir test ve değerlendirme

laboratuvarında kurulacak test yatağı ile bir firmanın güvenlik duvarı kurulumu aşamasında kullandığı test yatağı arasında farklılıklar vardır.

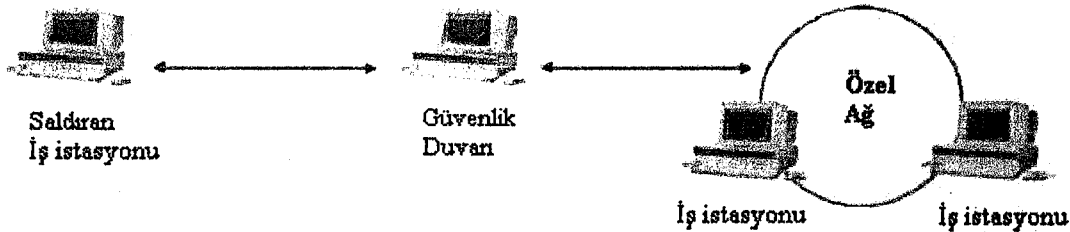
2.4.1. Test Yatakları

Güvenlik duvarlarının testi ve değerlendirilmesi için kullanılabilecek üç farklı test yatağı Şekil 2.7, Şekil 2.8 ve Şekil 2.9’de çizilmiştir. Bu yapılar, güvenlik duvarlarını çalıştıran bilgisayarlar ile ağ analizi ve atak simülasyonları yapan bilgisayarlar arasında bağlanabilirliği sağlayan omurga mimarisi sağlamaktadır. Aşağıda bu test yatakları tanımlanmaktadır.

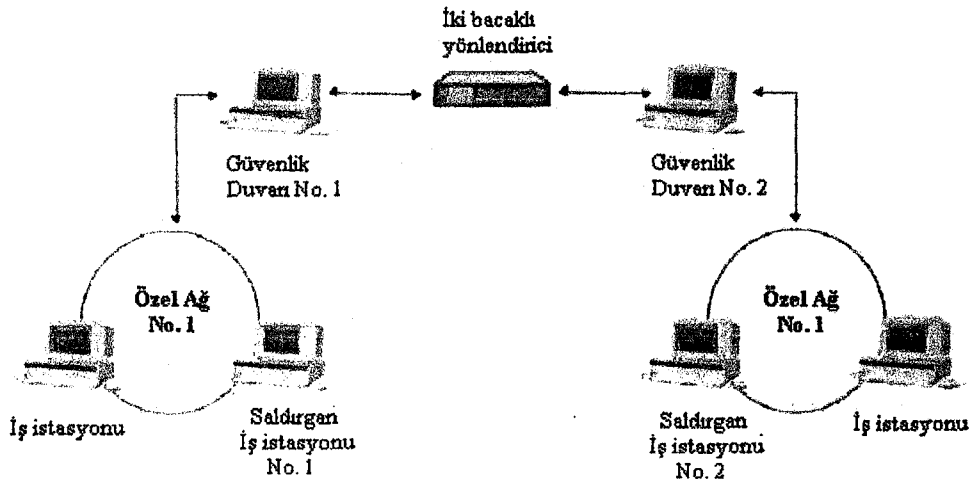
Birinci ve en basit test yatağı, Şekil 2.7’de gösterildiği gibi, bir atak bilgisayarı, güvenlik duvarını çalıştıran bir güvenlik duvarı makinesi ve güvenlik duvarının koruduğu özel ağda bulunan iki makineden oluşmaktadır. Güvenlik analiz araçları, saldırgan makine üzerinde özel ağa saldırma amaçlı çalışmaktadır. Bu test ortamına herhangi bir güvenlik duvarı entegre edilerek testleri gerçekleştirilebilir.

İkinci test yatağında, her biri güvenlik duvarları ile korunan iki özel ağ birbirine bağlayan bir yönlendirici bulunmaktadır (Şekil 2.8). Bu ortamda saldırgan yazılımlar özel ağlarda çalıştırılır ve diğer ağın güvenlik duvarına saldırma amaçlı kullanılır.

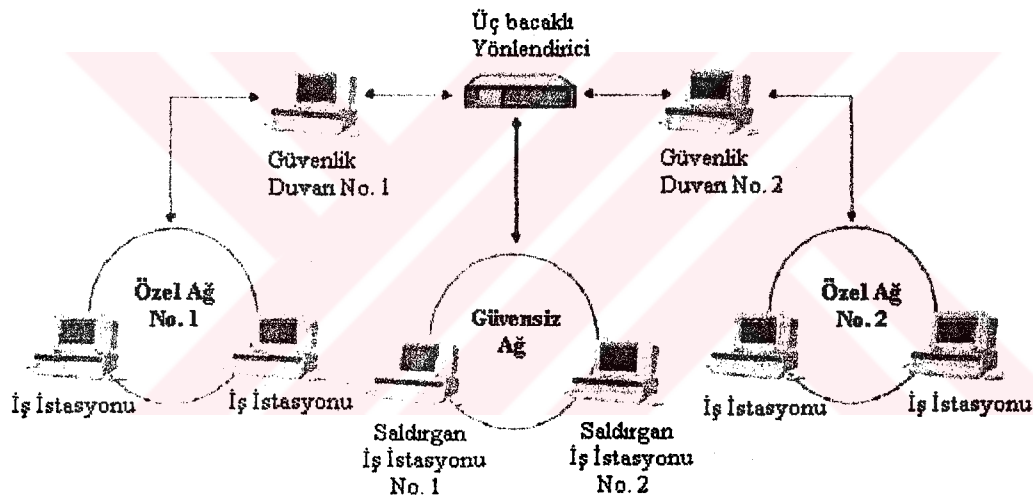
Son olarak, üçüncü test yatağı Şekil 2.9’deki gibi iki özel ağ ve bir güvensiz ağdan meydana gelir. Bütün bağlantılar üç bacaklı bir yönlendirici üzerinden sağlanır. Saldırı araçları güvensiz ağdan çalıştırılarak özel ağları koruyan güvenlik duvarları test edilir.



Şekil 2.7: Test yatağı 1 [Al-Tawil and Al-Kaltham, 1999]



Şekil 2.8: Test yatağı 2 [Al-Tawil and Al-Kaltham, 1999]



Şekil 2.9: Test yatağı 3 [Al-Tawil and Al-Kaltham, 1999]

2.4.2. Test Yataklarının Karşılaştırılması

(1. Test Yatağı) Donanım ihtiyacı düşünülecek olursa en ucuz test yatağıdır. Test süresi göz önüne alındığında en pahalı çözümdür.

(2. Test Yatağı) Birinci test yatağına göre daha fazla donanım gereksinimi duyar. Öte yandan iki güvenlik duvarı aynı anda test edilebileceği için en daha hızlıdır. Fakat, test koşulları tamamıyla aynı olmadığı için sonuçların %100 karşılaştırılabilir olduğunu söylemek güçtür.

(3. Test Yatağı) En pahalı çözümdür. Fakat ikinci test yatağındaki avantajlara ek olarak testler aynı koşullarda gerçekleştiği için karşılaştırılabilir sonuçlar üretir. Ayrıca ağlar birbirlerinden bağımsız oldukları için aynı anda birkaç güvenlik analiz aracı ya da saldırı modülü çalıştırılabilir.

2.4.3. Test Metodolojisi

Doğru test yatağı seçilip gerekli yazılımlar ve güvenlik duvarı kurulduktan sonra test metodolojisi iki aşamadan oluşur:

Otomatik testler: Dış ağdan güvenlik analiz araçları çalıştırılarak güvenlik duvarı ve iç ağdaki bilgisayarlar taranarak güvenlik açıkları araştırılır.

Gözlem tabanlı testler: Güvenlik duvarı gözleme dayalı test kriterlerine göre incelenerek karşılaştırılabilir tablolar hazırlanır.

Otomatik Testler

Günümüzde Nessus, ISS Internet Scanner, Sara, Saint gibi bazı bilgisayar ve ağ tabanlı otomatik açıklık tarama araçları vardır. Bu araçların karşılaştırılması ve değerlendirilmesi bu çalışmanın kapsamı dışındadır. Fakat özetle şöyle denilebilir ki; bir açıklık tarama aracının işlevselliğini doğru bir şekilde yerine getirebilmesi için en son saldırı imzalarını içermesi ve gelecekte keşfedilecek yeni saldırıları kapsayacak şekilde güncellenebilmesi gereklidir.

Bu aşamada otomatik tarama araçları ile tarama işlemi üç farklı durumda gerçekleştirilecektir:

(1) *IP iletimi aktifken:* Bu durumda güvenlik duvarının koştugu bilgisayar üzerinde IP iletimi aktif hale getirilir. Güvenlik duvarı yazılımı ise çalıştırılmaz. Böylece gelen her trafiğin iç ağa girmesine izin verilir.

(2) *IP iletimi pasifken:* Güvenlik duvarının koştugu bilgisayar üzerinde IP iletimi pasif hale getirilir. Güvenlik duvarı yazılımı yine çalıştırılmaz. Bu test durumu, güvenlik duvarı aktif iken alınan sonuçlarla karşılaştırma amaçlı gerçekleştirilmektedir.

(3) *Güvenlik Duvarı aktifken:* IP iletimi pasif, güvenlik duvarı ise aktif iken iç

ağdaki bilgisayarlardaki açıklıkları bulacak güvenlik analiz araçları çalıştırılır.

Gözlem Tabanlı Testler

Gözlem tabanlı testler kullanım kolaylığı, esneklik, kurulum ve çalışma gereksinimleri, kullanıcı ile etkileşim ihtiyacı, desteklenen fonksiyonlar(Bölüm 1.3 de açıklanmıştır) gibi bir grup kriteri içerir. Sonuçta test edilen güvenlik duvarının avantajları ve dezavantajları listelenir. Sonuçlar karşılaştırılabilir formatta olup doğru güvenlik duvarı mekanizmasını seçmede kullanılabilir. Bu yöntemle test edilmiş Socks v5 ve TIS 2.0 güvenlik duvarlarının sonuç raporları Çizelge 2.1, Çizelge 2.2,Çizelge 2.3, Çizelge 2.4 ve Çizelge 2.5’de görülmektedir.

Çizelge 2.1. Kurulumların karşılaştırılması [Al-Tawil and Al-Kaltham, 1999]

Karşılaştırma Faktörü	Socks v5	TIS FWTK 2.0
Kurulun öncesi konfigürasyon	Birçok opsiyon desteği ile otomatik	El yordamıyla. Sistem hakkında bilgi sahibi olmaya ihtiyaç var.
Kurulum	Otomatik	Otomatik fakat bazı paketlerin ayrı kurulması gerekiyor.
Programlar	Tek program halinde	Ayrı ayrı programlardan oluşuyor.
Etkileşimde olan bilgisayarlar	Sunumcu ve iç ve dıştaki istemci bilgisayarlar	Sadece sunumcu makine
Desteklenen mimariler	Tek ve çok ağ dilimli yapı	Tek ve çok ağ dilimli yapı

Çizelge 2.2. Konfigürasyon ve çalışmaların karşılaştırılması [Al-Tawil and Al-Kaltham, 1999]

Karşılaştırma Faktörü	Socks v5	TIS FWTK 2.0
Kurulum sonrası konfigürasyon	Basit, konfigürasyon dosyaları değiştirilerek yapılır. Sunumcu tarafından bir konfigürasyon dosyası, her bir istemci için ise bir başka dosya kullanılır.	1. Basit konfigürasyon ile oldukça kolay 2. Daha gelişmiş konfigürasyon yaparken bileşenlerin birlikte ve düzgün bir şekilde çalışmasının sağlanması zordur.
bilgisayar güvenliği	Gelişmiş değil	Gelişmiş
İstemciler	Statik ve dinamik olarak eklenebilir.	Özel bir proses gerektirmez.
Kimlik doğrulama opsiyonları	Herhangi bir kimlik doğrulama metodu geçerli	Herhangi bir kimlik doğrulama metodu geçerli
Erişim denetimi opsiyonları	Her bir kural için kaynak ve hedef adresler, servis portları, kullanıcı ismi ve şifre ve ara birim adreslerine göre tanımlanabilir.	Her bir servis için kaynak bilgisayar, kullanıcı ismi ve şifre, güçlü kimlik doğrulama ve DNS sorgularına göre tanımlanabilir.
Ara birim tespiti	Ara birimleri tespit edebilir.	Ara birimleri tespit edemez.
DNS Servisi	Kontrol edilmez	IP Sahteciliğine karşı kontrol edilebilir.
TCP tabanlı servisler	Eklenebilir ve kullanılabilir.	ftp, http, rlogin, telnet, X11 ve plug-gw için vekil sunumcuları bulunmaktadır. Netacl ise diğer servisleri kontrol altında tutabilir.
UDP tabanlı servisler	Uygun değil	UDP servislerine izin verilmez. Trafik yönlendirmesi yapılarak sağlanabilir.

Çizelge 2.3. Kullanımın karşılaştırılması [Al-Tawil and Al-Kaltham, 1999]

Karşılaştırma Faktörü	Socks v5	TIS FWTK 2.0
Güvenlik Duvarı Çeşidi	Devre(circuit) seviyesi	Uygulama seviyesi
Durdurma, başlatma, tekrar başlatma	SOCK komutları kullanarak basit	Başlatırken <i>inetd.conf</i> dosyası isteğe göre değiştirilir. <i>inetd</i> çalıştırılabilir dosyası başlatılır.
İstemciler	Socks uygulamaları kullanılmak zorunda	Hedef siteye bağlanmadan önce güvenlik duvarına bağlanır
Saydamlık	Kullanıcılara %100 saydam	Saydam değil. Kullanıcılar FWTK sunumcuya bağlanmak zorunda.
İçe doğru trafik	Sadece Sock istemcileri ve sunumcularından	Herhangi bir TCP/IP istasyonundan
Diğer güvenlik duvarına bağlantı	Socks istemcileri FWTK üzerinden bağlanabilirler.	FWTK istemcileri Socks sunumcusu üzerinden bağlanabilirler.
Güvenlik duvarına erişim	İzin verilmez	Birkaç yolla izin verilebilir ya da engellenebilir.
İzin verilen servisler	Herhangi bir Socks uygulaması ya da servisi	Sadece standart FWTK vekil servisleri ya da FWTK kontrolünde çalıştırılan diğer servisler.

Çizelge 2.4. SOCKS'un avantajlar ve dezavantajlar [Al-Tawil and Al-Kaltham, 1999]

Avantajları	Dezavantajları
Saydam	Devre seviyesi. Kontroller sadece bağlantı anında yapılır.
Socks istemcileri TCP tabanlı diğer güvenlik duvarları üzerinden bağlanabilir.	Sadece Socks güvenlik duvarı sunumcuları ve istemcilerinden gelen bağlantılar kabul edilir.
Kaynak/hedef bilgisayar/port kombinasyonlarına göre erişim denetimi sağlanır.	Lokal Socks güvenlik duvarının adresi istemciler tarafından bilinmeli ve konfigürasyon dosyasına girilmeli.
Ara birim tespiti yapılır.	Güvenlik duvarı üzerinden sadece Socks uygulamaları ve servislerine ait trafik geçebilir.
Güvenlik duvarlarına bağlantı yapılamıyor.	Üzerinde çalıştığı sunumcunun güvenliğini arttırmaz.
	UDP desteği yok

Çizelge 2.5. FWTK'nın avantajlar ve dezavantajlar [Al-Tawil and Al-Kaltham, 1999]

Avantajları	Dezavantajları
Uygulama seviyesi	Saydam değil
Kimlik doğrulama, güçlü kimlik doğrulama ve şifreleme desteği vardır.	UDP desteği yok, fakat bazı ek modüller üzerine tünelleme yapılırsa UDP servisleri çalıştırılabilir.
Kaynak/hedef bilgisayar/port kombinasyonlarına göre erişim denetimi sağlanır.	Ara birim tespiti yok.
Herhangi bir TCP/IP iş istasyonu tarafından gelen bağlantılar kabul edilir.	
Herhangi bir servis güvenlik duvarı üzerinden bağlanabilir.	
DNS kontrolü yapılır.	
Üzerinde çalıştığı bilgisayarın güvenliğini artırır.	

2.5. Vigna'nın Test Metodolojisi

Vigna'nın yaklaşımı TCP/IP ağlarının ve bu ağlardaki açıklıkların modellenmesi üzerine kuruludur. Bu model, test mühendislerinin ve sistem yöneticilerinin ağ ortamlarını ve korunacak servislerini tanımlamalarına imkan verir. Bu modeli kullanarak güvenlik duvarlarının topolojide yerleştirildikleri konumlarının kurum

ağlarını korumak için yeterli imkanları sağlayıp sağlamadığını belirlemek ve ispatlamak mümkündür [Vigna, 1997]. Bununla beraber, modellenen saldırılara karşı güvenlik duvarının işlevini tam olarak yerine getirip getirmediğinin sınavacak test metotları hazırlanabilir.

2.5.1. Ağ Modeli

Metodoloji öncelikle TCP/IP protokolünün katmanlarını(fiziksel, veri hattı, taşıma ve uygulama katmanları) modeller. Böylece, uygulama protokollerinin tanımlanmasına imkan veren bir altyapı sağlar.

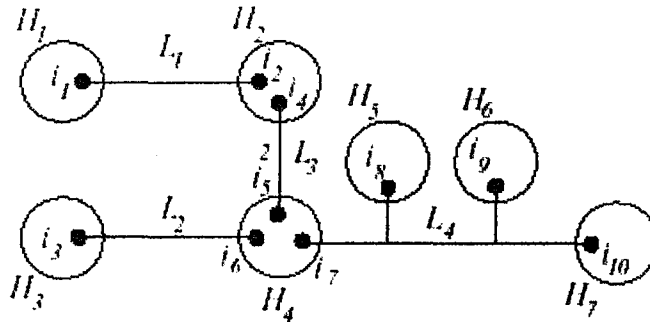
Bir ağ, birbirine bağlı ethernet ara birimlerinden $I = \{i_1, i_2, \dots, i_n\}$ oluşan hiper-grafik olarak modellenir. Bu hiper-grafik bir grup bilgisayar(H) ve bir grup linkten(L) oluşur. bilgisayarlar en az bir ethernet ara birimine sahiptir, linkler ise en az iki ethernet ara birimi içerir. Şekil 2.10 bir ağ hiper-grafiğini göstermektedir.

$$\bigcup_{i=1}^p H_i = I \quad (2.1)$$

$$H_i \cap H_j = \emptyset; i, j \in \{1, 2, \dots, p\} \text{ ve } i \neq j$$

$$\min |L_j| \geq 2; \bigcup_{i=1}^4 L_i = I; L_i \cap L_j = \emptyset; \quad (2.2)$$

$$i, j \in \{1, 2, \dots, q\} \text{ ve } i \neq j$$



Şekil 2.10: Örnek bir ağ hiper-grafiği [Vigna, 1997]

Şekil 2.10 ile temsil edilen ağda yedi bilgisayar, dört link ve on ethernet ara birimi bulunmaktadır. Ağ (2.3),(2.4),(2.5),(2.6) denklemleri formüle edilmiştir

$$I = \{i_1, i_2, i_3, i_4, i_5, i_6, i_7, i_8, i_9, i_{10}\} \quad (2.3)$$

$$H = \{\{i_1\}, \{i_2, i_4\}, \{i_3\}, \{i_5, i_6, i_7\}, \{i_8\}, \{i_9\}, \{i_{10}\}\} \quad (2.4)$$

$$L = \{\{i_1, i_2\}, \{i_3, i_6\}, \{i_4, i_5\}, \{i_7, i_8, i_9, i_{10}\}\} \quad (2.5)$$

$$N = \{H_1, \dots, H_7, L_1, \dots, L_4\} \quad (2.6)$$

Ağ katmanlarından veri hattı seviyesinde, bir mesaj ($m \in M$) kaynak ara birim(i_s), hedef ara birim(i_d) ve paket verisinden(p) oluşmaktadır. Kaynak ara birim aynı linke ait olmalıdır.

$$m = (i_s, i_d, p) \text{ öyle ki} \quad (2.7)$$

$$i_s \in L_i \wedge i_d \in L_i \text{ ve } L_i \in L.$$

Model ayrıca, bir grup kaynak/hedef ara birim, kaynak/hedef bilgisayar ve paket verisi değerlerini veren bir grup fonksiyonu da sunar. Bu fonksiyonlar, si , sh , di , dh , pl ve al 'dir.

Modelin IP seviyesinde, bağlantı katmanı üzerinde IP datagramı olarak adlandırılan genel bir adresleme mekanizması tanımlanır. Bir IP ağında her bilgisayar IP datagramları yoluyla birbiriyle iletişim kurabilir. Eğer hedef, aynı alt-ağda değilse, IP datagramı bir veya birkaç ağ geçidi üzerinden iletilir. Bu işlem yönlendirme olarak adlandırılır. Örnek olarak, Şekil 2.10'da, H_1 , H_7 'ye sadece H_2 ve H_4 üzerinden mesaj gönderebilir. Bir IP datagramı kaynaktan yola çıkıp hedef bilgisayara ulaşana kadar her bilgisayar aralığında yeni bir mesaj üretilir. Bu mesaj dizisinin($m_1, m_2, \dots, m_n$) tümü IP datagramını oluşturur. Her bir mesajın verisinde (a_s, a_d, p') bulunur. a_s mesajı hazırlayan kaynağın IP adresi, a_d varılacak en son hedefin IP adresi, (yönlendirmedeki ara ağ geçitleri değil) ve p' ise kaynak bilgisayar tarafından oluşturulan datagram verisini simgeler. Her mesajdaki i_s o an mesajın çıkacağı bilgisayarın ya da ağ geçidinin ara birim adresini, i_d ise bir sonraki ara birim adresini içerir. Diğer bir deyişle, m_i 'nin hedef ara birim adresi m_{i+1} 'nin kaynak ara birim adresine eşittir.

Üst katmanlarda, IP datagramları daha detaylı bilgiler içerir. Örneğin UDP protokolünde, p' , (p_s, p_d, p'') dan oluşur. p_s ve p_d , kaynak ve hedef port bilgilerini simgeler. Bir UDP isteği bu istek karşılığında bir cevap bekler. İstek ve cevap mesajlarının verileri şu şekilde ifade edilir.

$$\begin{aligned} pl(m_i^1) &= (a_i, a_j, (p_i, p_j, p_i^{*1})) \text{ ve} \\ pl(m_i^2) &= (a_j, a_i, (p_j, p_i, p_j^{*2})) \end{aligned} \quad (2.8)$$

2.5.2. Açıklıkların Modellenmesi

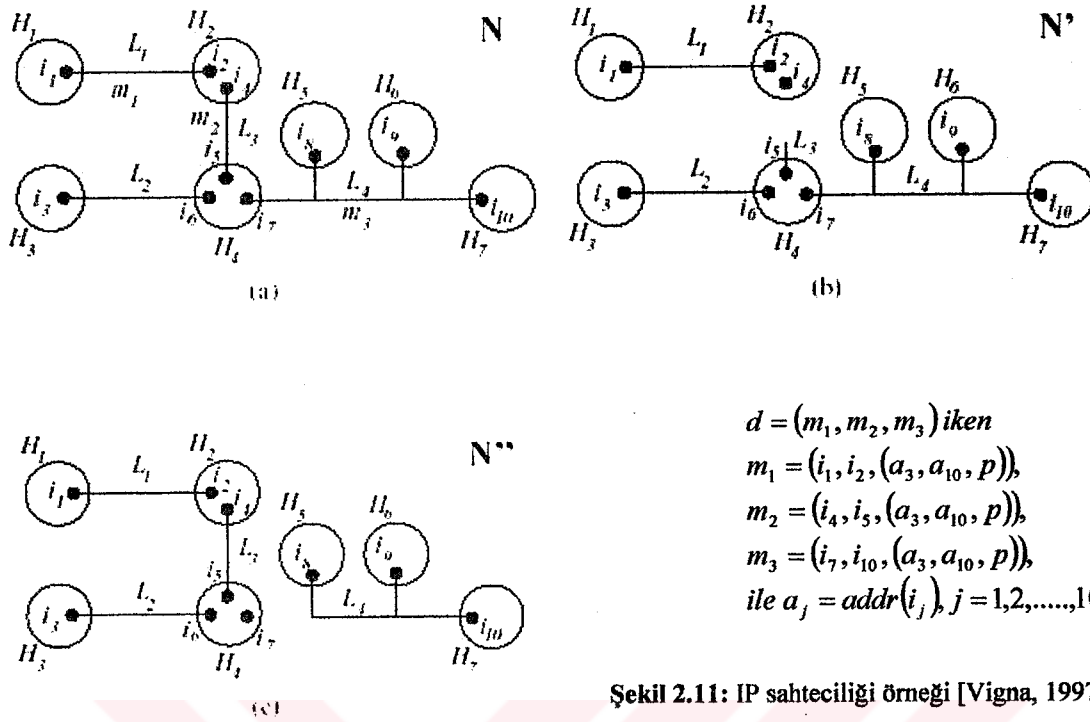
Vigna tarafından yukarıda sunulan ağ modeli ağ güvenlik açıklıklarını modellemek için kullanılır. Burada örnek olarak bir IP sahteciliği saldırısının örneklenmesi üzerinde durulacaktır. Modellemeler hakkında daha detaylı bilgi [Vigna, 1996]'da mevcuttur.

Bir bilgisayarın başka bir bilgisayarın yerine geçmeye çalışması IP sahteciliğinin varlığını gösterir. Genellikle bu saldırı, yerine geçilmeye çalışılan bilgisayarın haklarını elde etmek amacıyla yapılır.

$$\begin{aligned} d &= (m_1, m_2, \dots, m_n) \text{ sahtedir şayet :} \\ \forall m_i \in d, dsa(m_i) &= a_s \text{ ve } addr^{-1}(a_s) \in H_j \\ \text{ve } si(m_i) &= i_s \in H_i \neq H_j \end{aligned} \quad (2.9)$$

Bir H_j bilgisayar H_k 'ya bağlanmaya çalıştığı zaman, H_i bağlantıya sahte bir datagram üreterek müdahale eder. Bu datagram H_j 'nin adresini kaynak adres olarak ama H_i 'nin ara birimlerinden birini de kaynak ara birim olarak içerir.

H_k , m_n mesajını aldığı zaman, hazırlanan IP datagramının H_j tarafından gönderildiğini farzeder. IP sahteciliği sadece belirli topolojilerde tespit edilebilir. Eğer mesajın kaynak ara birimini bulunduğu linkten çıkarıldığında, datagramın kaynak adresine ve mesaj kaynak adresine karşılık gelen ara birimler arasında bir yol yoksa bir datagramın sahte olup olmadığı anlaşılabilir.



Örnek olarak Şekil 2.11 üzerinde durulsun. Bilgisayar H_1 , sahte datagram $d=(m_1, m_2, m_3)$ ile H_3 'ün yerine geçmeye çalışarak H_7 'ye doğru bağlantı kurmaya çalışmaktadır. m_2 mesajını gören H_4 bilgisayarı, bu saldırının meydana gelişini Şekil 2.11-b'de sunulan N' ağından üzerinden tespit edebilir: Çünkü $i_3 = \text{addr}^{-1}(\text{dsa}(m_2))$ ve $i_4 = \text{si}(m_2)$ arasında bir yol yoktur. Öte yandan, H_5 , m_3 mesajını aldığı anda IP sahteciliği saldırısını tespit edemez. Çünkü N'' ağında (Şekil 2.11-c) i_3 ve i_7 arasında bir yol vardır.

Belirlenen her bir bilgisayar için bir sahtecilik tespit etme matrisi $\text{STM}(H_i)$ hazırlanabilir. Bu matriste satırlar bilgisayarın bağlantıda olduğu linklerle tanımlanır. Sütunlar da IP adreslerinin listesini içerir. Bir link üzerindeki bir IP sahteciliği tespit edilebiliyorsa ilgili link ve IP adresi çifti 1 değerini alır. Diğerleri ise 0 değerini alır. olur. Şekil4-a'daki H_4 bilgisayarı için IP sahteciliği tespit matrisi Çizelge 2.6'teki gibidir.

Çizelge 2.6. H₄ bilgisayarı için IP sahteciliği tespit matrisi [Vigna, 1997]

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
L_2	1	1	0	1	1	1	1	1	1	1
L_3	0	0	1	0	1	1	1	1	1	1
L_4	1	1	1	1	1	1	1	1	1	1

Ayrıca herhangi bir uygulama veya servis için güven vektörü oluşturulabilir. Örnek olarak, H₇ üzerinde rlogin servisi olduğu farzedilsin. Bu servis H₅, H₆ ve H₁'den gelen bağlantıları sorgulamasız kabul etmektedir. Güven vektörü Çizelge 2.7'deki gibidir.

Çizelge 2.7. rlogin servisi için güven vektörü [Vigna, 1997]

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
1	0	0	0	0	0	0	1	1	0

Her bir bilgisayar için sahtecilik tespit matrisi ve güven vektörü oluşturulabilir ve bu bilgilere dayanarak ağda IP sahteciliği saldırısının yapılabileceği noktalar belirlenebilir. Yukarıdaki örnek için, H₄'ün H₅ ve H₆'ya yapılan sahtecilik saldırılarını hangi linkten gelirse gelsin tespit edebildiği ama L₃' linkindeki a₁ adresi için başarısız olduğu gözlenmektedir. Sonuç olarak, muhtemel bütün saldırıları gözlemlemek ve servis sağlayıcıyı korumak için H₂ ve H₄'ün her ikisini de kullanılması gerektiği sonucu bu çalışmadan anlaşılmaktadır.

2.5.3. Vigna'nın Yaklaşımıyla Güvenlik Duvarlarının Testi

Vigna'nın test metodolojisi yukarıda bir örneği verilen açıklıkları ve ağ modellemesini kullanır. Amaçlanan metodoloji güvenlik duvarlarını aktif olarak kullanıldıkları ağ girişlerinde test etmek için kullanışlıdır. Metodoloji beş adımdan oluşur:

Ağ modellemesi: Öncelikle, ağ topolojisi ağ hiper-grafikleri kullanılarak tanımlanır. Servisler ve güvenli kabul edilen ilişkiler, güven vektörleri ile gösterilir. Çok büyük ağları tek bir hiper-grafik ile tanımlamak mümkün olmayabilir. Bundan dolayı, hiper-grafiklerde her bilgisayarı göstermek yerine bilgisayar toplulukları kullanmak makul olabilir. (Örneğin, Internet ağını temsil etmek için).

İhtiyaç modellemesi: İhtiyaçlar, tespit edilmesi gereken açıklıkları içerir. İhtiyaç modellemesi, korunmuş ağ bağlantılarında görülen/ortaya çıkan ağ mesajları üzerindeki kısıtlamaları tanımlayarak yapılabilir. Her bir saldırının mesajlar üzerinde imzası vardır. Örnek olarak, parçalı paketlerle yapılan saldırılarda, bu saldırının meydana gelmesini sağlayan şablon, mesaja ya da mesaj serilerine eklenir. Bu imzalar test modülü için gerekli kısıtlamaları meydana getirir.

Ortam doğrulaması: Bu adım tanımlanan hedef ağın, belirlenen güvenlik duvarı ihtiyaçları için gerekli koşulların sağlanıp sağlanmadığını doğrular.

Test modüllerinin oluşturulması: Bu adım sürecinde her bir ihtiyaç için tanımlanan şablonlara karşılık gelen bir grup mesaj hazırlanır. Örneğin, iç ağa telnet bağlantısı için izin verilmediği durumda, bir telnet bağlantısı açmak için bir grup test modeli üretilir. Ayrıca her bir model için, *takipçi*⁵ adı verilen uygun bir kaynak ve monitörleyen uygun bir *gözlemci* atanır.

Testlerin çalıştırılması: Sonuç olarak, üretilen test modellerine karşılık gelen mesajlar güvenlik duvarının kullanıldığı ağa, ilgili takipçileri tarafından gönderilir. Tanınacak mesaj şablonlarını bilen gözlemciler güvenlik duvarı mekanizmasının beklendiği gibi çalışıp çalışmadığını inceler.

⁵ Probe

3. NETCHECK GÜVENLİK DUVARI TEST VE ANALİZ METODOLOJİSİ

NetCheck güvenlik duvarı test ve analiz metodolojisi ve bu metodolojiyi otomatik olarak uygulayan NetCheck güvenlik duvarı analiz aracı daha önceden geliştirilen ve örnekleri bir önceki bölümde anlatılan metodolojilerin eksikliklerini kapatmak ve yeni özellikler ve kolaylıklar ekleme amaçlı geliştirilmiştir.

Geliştirilen metodolojide diğerlerine kıyasla en önemli fark, güvenlik duvarları üzerinde oluşabilecek açıklıkları kapsamlı bir şekilde ele alması ve günümüzde bilinen saldırı tekniklerinin hemen hemen hepsini içeriyor olmasıdır. Aşağıda bu saldırı yöntemleri üzerinde detaylı olarak durulacaktır.

3.1. Ağ Modellemesi

NetCheck güvenlik duvarı test metodolojisinde, test modüllerinin belirlenmesi ve belirli bir sistematikte uygulanabilmesi amacıyla bir ağ modeli ortaya atılmıştır. Bu model, Internet ortamında kullanılan TCP/IP altyapısının güvenlik duvarları ile ilgili alt kümesini içerir. Modelde geçen terimler şöyledir:

IP Adres Alanı (IP_Alan), güvenlik duvarının kullanıldığı topolojide test edilecek ağ dilimlerinin bileşkesidir.

$$IP_Alan = \{AD_0, AD_1, \dots, AD_n\} \quad (3.1)$$

Ağ Dilimi (AD), ağ maskesine ait IP adresleri arasında direk bağlantı kurma imkanını veren altyapıdır. Bir ağ dilimi başka ağ dilimlerine bağlı olabileceği gibi izole bir yapıda da olabilir.

IP Adresi(IP), 0 ile 255 arasındaki dört adet tamsayının üç nokta ile ayrıldığı ve her bir ağ ara birimi için üst seviye kimlik bilgisini ifade eden gösterimdir.

Güvenlik Seviyesi (GüvSev), bir ağ diliminin erişilebilirliğine göre belirlenen 0 ile 2 arasında bir değerdir. 0 diğer ağ dilimlerinden erişimi engellenen, 1 kısıtlı erişim hakkı verilen, 2 ise genel olarak erişime açık ağ dilimlerinin güvenlik seviyelerini simgelemek.

Bir ağ dilimini mantıksal bir isim, güvenlik duvarının bu ağ dilimine bağlı ara

biriminin IP adresi, NetCheck yazılımının IP adresi ve güvenlik seviyesi tanımlar. Bu bağlamda bir ağ dilimi şu şekilde modellenir.

$$\forall AD \in IP_Alan \text{ için}$$

$$AD = \begin{cases} Isim \\ GdlP \\ AgMas \\ NetCheckIP \\ GüvSev \end{cases} \quad (3.2)$$

öyle ki

$$\begin{aligned} Isim &\in \{A..Z, a..z, 0..9\}^n \wedge \\ GdlP, AgMas, NetCheckIP &\in IP \wedge \\ GdlP \& AgMas &= NetCheck \& AgMas \wedge \\ GüvSev &\in \{0,1,2\} \end{aligned}$$

Test modülü(TM), güvenlik duvarına gönderilen ağ paketleri(P) topluluğudur. Bir paket, kaynak ve hedef ağ dilimleri ile bu pakete ait veri ile tanımlanır. Her bir test modülü için tanımlı kaynak ağ dilimi (AD_{Kaynak}) ve hedef ağ dilimi (AD_{Hedef}) bu modülde yer alan her pakette için de geçerlidir.

$$TM = \{P_1, P_1, \dots, P_n\} \quad (3.3)$$

$$P = (AD_{Kaynak}, AD_{Hedef}, veri) \quad (3.4)$$

$$\begin{aligned} \forall P_x, P_y \in TM \text{ için} \\ F_AD_{Kaynak}(P_x) &= F_AD_{Kaynak}(P_y) \wedge \\ F_AD_{Hedef}(P_x) &= F_AD_{Hedef}(P_y) \end{aligned} \quad (3.5)$$

Erişim(E), kaynak ve hedef ağ dilimleri farklı olan bir test modülüne(TM) ait her paketin(P) kaynaktan çıkıp hedef ağ dilimine geçmesidir. Bu paketin hedef bilgisayarda açık olan bir porta karşılık gelmesi gerekmez. Güvenlik duvarı testi kapsamında, erişimin sağlanabilmesi için paketlerin güvenlik duvarından geçip hedef ağ diliminde görülebilmesi yeterlidir. Bu amaçla $F_Gon(P,AD)$ ve $F_Yak(P,AD)$ fonksiyonları tanımlanmıştır. Fonksiyonlar sırasıyla paket gönderildiğinde ve yakalandığında 1 değerini alırlar. Bu tanımların ışığında erişim modellemesi şu şekildedir.

$$\begin{aligned}
&\forall P \in TM \text{ için} \\
&AD_x = F_AD_{Hedef}(P) \wedge \\
&AD_y = F_AD_{Kaynak}(P) \text{ iken} \\
&F_Gön(P, AD_x) \wedge F_Yak(P, AD_y) \equiv 1 \\
&\text{ise} \\
&TM \text{ E' dir}
\end{aligned} \tag{3.6}$$

3.2. Kullanılan Test Modülleri ve Yapıları

Her bir test modülü güvenlik duvarlarına yapılabilecek saldırının belirli bir formatını içerir. Yani bir saldırıya karşılık gelen birkaç test modülü olabilir. Aşağıda öncelikle saldırının nasıl yapıldığı ve bu saldırıların etkileri açıklanacak, daha sonra bu saldırıların güvenlik duvarı testlerinde nasıl kullanılabileceği anlatılacaktır.

3.2.1. Topoloji Konum Testi

3.2.1.1. Açıklama

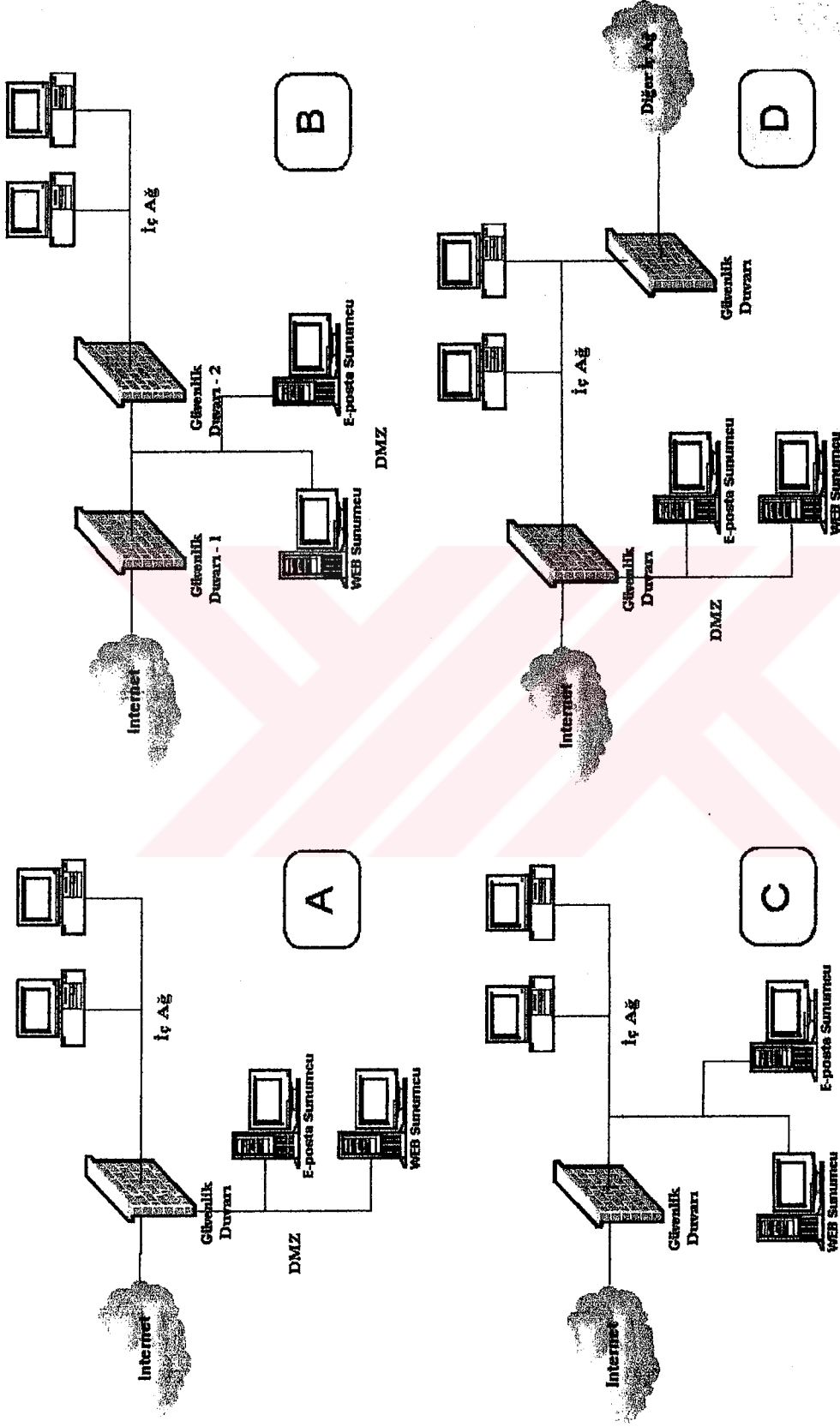
Güvenlik duvarı kurulumunda ilk ve önemli adım doğru topolojinin seçimidir. Güvenlik duvarları üzerindeki bazı açıklıklar, ufak güncelleme ve kural ayarlamaları ile çözülebilmesine karşın topolojik açıklıkların düzeltilmesinde IP adres değişimi, sunumcu ve istemciler arasındaki güven ilişkilerin yeniden düzenlenmesi, DNS kayıtlarının güncellenmesi gibi birçok güçlüklerle karşılaşılabilir. Bu sebeple güvenlik duvarının ilk kurulması safhasında topoloji belirlenirken dikkatli olunmalıdır.

Dikkat edilecek ilk husus yarı korunmalı bölge olarak nitelendirilen DMZ ağ diliminin topolojide yer almasıdır. DMZ ağ dilimi kuruma ait sunumcuların bulunduğu dış dünyadan belirli haklarla erişilebilen bir ağıdır. DMZ kurulumunun güvenlik açısından bir çok faydası vardır. Öncelikle dış dünyadan erişilen sunumcuların hepsi bu ağ dilimine taşınacağı için iç ağa dış dünyadan erişme ihtiyacı ortadan kalkar. Yani güvenlik duvarına Internet ağından başlatılan ve iç ağı hedef alan bütün bağlantıları blokla kuralı eklenebilir. Böylece ayrı güvenlik seviyesinde alanlar oluşturulup güvenlik seviyesi artırılır. Şekil 3.1-C de DMZ ağ dilimi olmayan bir güvenlik duvarı kurulumu görülmektedir. Bu kurulum en basit anlamıyla Şekil 3.1-A'ya dönüştürülerek DMZ ağ dilimine sahip olunabilir.

İkinci faydası ise kritik sunumcuları iç ağdaki kullanıcılardan korumaktır. Bilindiği gibi ağ üzerinden yapılan saldırıların büyük bir çoğunluğu iç ağdan kaynaklanmaktadır. Bunda sunumcularla kullanıcıların aynı ağda olmasının ve hiçbir güvenlik önlemi ile karşılaşmadan sunumculara erişebilmelerinin rolü büyüktür. DMZ kurulumu sonucunda kritik sunumcular ayrı bir ağ dilimine alınacağından kullanıcılardan kaynaklanan tehditler azaltılmış olur.

Bununla beraber, Şekil 3.1-B'deki gibi oluşturulmuş bir topolojide, saldırganların birinci güvenlik duvarını ele geçirmeleri durumunda sadece DMZ ağ dilimindeki sunumculara direk erişim haklarının olduğu, iç ağdaki kullanıcıların ise hala diğer güvenlik duvarı tarafından korunduğu görülmektedir. Böyle bir yapı, Şekil 3.1-C'deki yapıya göre çok daha fazla güvenlik sağlar. Bu sebeple bilgi güvenliğinin ön planda olduğu ağlarda iki güvenlik duvarı kullanımı tavsiye edilir. Fakat bu iki güvenlik duvarı farklı yazılımlar olmalıdır. Çünkü aynı yazılımların kullanıldığı ağlarda birinci güvenlik duvarını geçen saldırgan, ikinciyi de aynı şekilde saf dışı bırakabilir.

Kurum ağlarında karşılaşılan bir diğer problem ise kuruma ait farklı güvenlik seviyelerindeki alt ağların aynı ağ dilimi içerisinde bulunmalarıdır. Bu durum da başka bir güvenlik açığı teşkil etmektedir. Gerek duyuluyorsa alt ağlar arasında da güvenlik duvarı kullanılabilir. Şekil 3.1-D'deki topoloji buna bir örnektir.



Şekil 3.1: Güvenlik Duvarlarının topolojilere entegrasyonu

3.2.1.2. Güvenlik Duvarlarına Uygulanması

Topoloji yapıları çok çeşitli olduğu için bütün ağları kapsayacak bir test modülü tanımlamak güçtür. Bu sebeple genel bir test modülü tanımlanmıştır.

Topoloji Konum Testi: Güvenlik Duvarı üzerinde güvensiz ağ ve iç ağ yapıları tanımlı ise yarı korunmalı ağın bulunup bulunmadığı incelenecektir.

3.2.2. Erişim Testi

3.2.2.1. Açıklama

Günümüzde güvenlik duvarı üzerinden yapılan saldırıların büyük bir kısmı güvenlik duvarında yalın yanlış konfigürasyondan kaynaklanmaktadır. Yanlış konfigürasyon sonucunda korunamayan servisler saldırılara ortam hazırlamaktadır [Schultz, 1997]. Bununla beraber, kurum ağı içindeki ve dışındaki ihtiyaçların çeşitli olması, işlerin aksamasına tahammülsüzlük gibi sebeplerle sistem yöneticileri ya da güvenlik sorumluları kurum bilgisayarlarına geniş erişim hakları vermektedirler. Bu sebeple güvenlik duvarı üzerinde birçok erişim açıklığı oluşabilmektedir. Bu açıklar iki grupta ele alınabilir. Birincisi kurum ağına yapılan bağlantılar, bir diğeri ise kurum ağından dış dünyaya açılan bağlantılar.

Kurum ağına doğru yapılan bağlantılarda genellikle düşülen hata, filtreleme politikasında sadece IP adresinin kullanılmasıdır. Örneğin bir kurum ağında Internet üzerinden erişilebilen bir e-posta sunumcu olduğu (IP adresi 139.179.11.111) ve kurumun güvenlik duvarı üzerinde şöyle bir kural⁶ tanımlandığı farzedilsin.

pass in quick on rl0 from any to 139.179.11.111

block in all

Şekil 3.2: Güvenlik duvarı örnek kural tablosu -1

Görüntüde bu kural dizisi sadece web erişimine izin veren güvenli bir konfigürasyon gibi gözükmetedir. Halbuki birinci kural dikkatlice incelendiğinde genellikle güvenlik duvarının arkası olduğu için varsayılan kurulumda olan ve hiçbir güvenlik yaması içermeyen web sunumcusuna Internet ağından her türlü trafiğe izin verildiği görülecektir. Örneğin IIS Web sunumcusu varsayılan kurulumda ftp servisi

⁶ Kural sentaksı için IPFİLTERR güvenlik duvarının komutları kullanılmıştır.

açıktır. IIS kullanan kurumlar, dış dünyaya ftp servisi sunmak istemese bile bu kural hatasından dolayı dış dünyadan ftp servisine erişilebilmekte ve bu durum bir güvenlik açığı oluşturmaktadır. Doğru tanımlanan kural şu şekilde olmalıdır.

```
pass in quick on rl0 proto tcp from any to 139.179.11.111 port 80
block in all
```

Şekil 3.3: Güvenlik duvarı örnek kural tablosu -2

Böylelikle sadece 80 numaralı porta tcp bağlantıları için izni verilmiş, diğer bağlantılar engellenmiş olur. Düşülen bir diğer hata ise kurum içi bilgisayarlara çok fazla bağlantı hakkı verilmesidir. Hatta dış dünyaya doğru her türlü bağlantı açımına izin verilmesidir. Çoğu zaman dış dünyadan gelen bağlantıları minimize etmenin güvenlik problemini çözdüğü düşünülür. Halbuki, birçok ağda içeriye doğru girmesine izin verilen web, e-posta gibi trafikler ile virüs, truva atı, arka kapı gibi kötü niyetli programcıklar gelmekte ve bu programcıklar, kullanıcı bilgisayarlarına yerleştikten sonra dış dünyaya doğru bağlantı kurmaktadır. Eğer güvenlik duvarı dış dünyaya doğru olan bağlantıları sadece ihtiyaç olunan servislerle kısıtlamazsa kurum ağında bilgi sızdıran saldırıların önünün alınması mümkün olmayacaktır.

3.2.2.2. Güvenlik Duvarlarına Uygulanması

Bu testin güvenlik duvarlarına uygulanması birkaç test modülü ile sağlanacaktır. Bunlar şöyle sıralanabilir:

İç ağ erişim testi: Dış dünyadan ve DMZ ağ diliminden iç ağa tcp, udp ve icmp bağlantıları kurulacaktır. Port seçimi rasgele yapılacaktır. Bu bağlantılardan birinin kurulması güvenlik duvarının kural tablosunun güvensiz olduğunu gösterecektir.

DMZ'e erişim testi: Dış dünyadan ve iç ağdan DMZ ağ dilimine tcp, udp ve icmp bağlantıları kurulacaktır. Port seçimi rasgele yapılacaktır. Bu bağlantılardan birinin kurulması güvenlik duvarının kural tablosunun güvensiz olduğunu gösterecektir.

Güvensiz ağ erişim testi: İç ağdan ve DMZ ağ diliminden dış dünyaya tcp, udp ve icmp bağlantıları kurulacaktır. Port seçimi rasgele yapılacaktır. Bu

bağlantılardan birinin kurulması güvenlik duvarının kural tablosunun güvensiz olduğunu gösterecektir.

WEB sunumcusu erişim testi -1: Kurumun Web sunumcusuna 80 no'lu port dışında bir porttan tcp bağlantısı kurulacaktır.

WEB sunumcusu erişim testi -2: Kurumun Web sunumcusuna 80 no'lu porttan udp bağlantısı kurulacaktır.

DNS sunumcusu erişim testi -1: Kurumun DNS sunumcusuna 53 no'lu port dışında bir porttan udp bağlantısı kurulacaktır.

DNS sunumcusu erişim testi -2: Kurumun Web sunumcusuna 53 no'lu porttan tcp bağlantısı kurulacaktır.

Bu testler kurumun dış dünyaya açık sunumcuları sayısınca çoğaltılabilir.

3.2.3. Bozuk Bağlantı Paketi Testi

3.2.3.1. Açıklama

TCP/IP protokolü üzerinden yapılan bağlantılı haberleşmeler el sıkışma ile başlar. Öncelikle bağlantıyı kuracak olan istemci bir sıra numarası seçip SYN bayrağı aktive edilmiş bir paketle bunu hedefe yollar. Hedef, bu paketi aldıktan sonra sıra no üretir. İstemcinin sıra numarasını bir attırıp kendi sıra numarası ile birlikte SYN ve ACK bayrakları aktive edilmiş bir şekilde geri yollar. Daha sonra istemci ACK bayraklı paket ile hedefin sıra numarasını bir arttırıp geri yollar. Aşağıda el sıkışma süreci modellenmiştir [Bellovin, 1989].

$$I \rightarrow H: \text{SYN}(SN_i)$$

$$H \rightarrow I: \text{SYN}(SN_H), \text{ACK}(SN_i + 1)$$

$$I \rightarrow H: \text{ACK}(SN_H + 1)$$

(3.7)

Görüldüğü gibi TCP bağlantıları SYN bayrağı aktif paket tarafından başlatılmaktadır. Başka bayrakları aktifleştirilmiş paketlerle bağlantı kurma isteği normal bir davranış değildir. Genellikle bu standart dışı paketler port tarayıcıları gibi saldırı başlanıcı araştırmalarını yapan araçlar tarafından gönderilmektedir [Blossom, 2000]. Bu sebeple, sadece SYN bayrağı aktif hale getirilmiş paketler dışındaki el sıkışma süreci başlangıç paketleri güvenlik duvarı tarafından düşürülmelidir.

3.2.3.2. Güvenlik Duvarlarına Uygulanması

Güvenlik duvarı üzerinden , izin verilen bir porttan standart dışı paketlerle bağlantı kurulmaya çalışılacaktır.

SYN ve ACK bayraklı paket yollanması: SYN ve ACK bayrakları aktifleştirilmiş bir paket, güvenlik duvarının durum tablosuna bulunmayan yani ilk defa kurulacak bir bağlantıyı başlatmak için güvenlik duvarına gönderilecektir.

FIN bayraklı paket yollanması: FIN bayrağı aktifleştirilmiş bir paket, güvenlik duvarının durum tablosuna bulunmayan yani ilk defa kurulacak bir bağlantıyı başlatmak için güvenlik duvarına gönderilecektir.

3.2.4. IP Sahteciliği Testi

3.2.4.1. Açıklama

IP sahteciliği en çok bilinen saldırı yöntemlerinden biridir [Tanese, 2003]. TCP/IP formatındaki bazı açıklıkları kullanarak gerçekleşir. IP sahteciliğinde paketlerin IP başlığında yer alan kaynak adres alanı değiştirilir. Bazen TCP başlığında yer alan sıra numaraları da değiştirilebilir. IP sahteciliği tekniği diğer saldırıların yapılmasına da temel teşkil eder. Günümüzde bu saldırıların bazılarının yapılabilirliği azalmasına karşın hala aktif olarak yapılan saldırılar da bulunmaktadır. IP sahteciliğini kullanan saldırılar öyle sıralanabilir [Tanese, 2003].

Kör Sahtecilik: Bu saldırı saldırgan ile kurbanın aynı alt ağda olduğu durumlarda yapılır. Sıra ve bilgi numaraları hat üzerinden dinlenir. Böylelikle bu numaraları tahmin etmek için bir çaba sarfetmeye gerek kalmaz. Bu tür saldırıların en tehlikelisi oturum hırsızlığıdır. Bu saldırı, kurulmuş bir oturumdaki veri akışını bölerek ve doğru sıra ve bilgi numaraları ile hedef bilgisayara bağlanarak gerçekleştirilir. Böylece kimlik doğrulama mekanizmaları atlatılır.

Kör Olmayan Sahtecilik: Daha karmaşık bir saldırıdır. Çünkü sıra ve bilgi numaraları erişilebilir değildir. Bu durumu atlatabilmek için hedef bilgisayara bir kaç paket gönderip sıra numaraları örnekleri toplanır. Bu örnekler değerlendirilerek sıra numara oluşturma algoritmasına ulaşılabilir [Haeni, 1997]. Bu teknik ile başarılı olmak günümüzde oldukça zordur. Çünkü işletim sistemlerinin çoğu rasgele sayılar oluşturmakta ve tahmin edilmesini güçleştirmektedir. Bunun yerine TCP oturumlarını ve paketleri inceleyerek başka açıklıklar bulma yönünde çalışmak daha

kolaydır. Örneğin günümüzde birçok servisin çalışma mekanizması IP adresine güvenme tabanlıdır. (rlogin, web erişim vs) Güvenilen IP adreslerinin tespit edilmesi sistemler üzerinde üst seviyeli yetki kazanımıyla sonuçlanan saldırılara imkan tanıyabilir.

Araya Girme Saldırısı: Her iki saldırı yöntemi araya girme (Man in the middle attack) saldırısının değişik formlarıdır. Bu saldırılarda, kötü niyetli saldırgan yasal bir bağlantının arasına girerek kaynak bilgisayarın gönderdiği bilgiyi kaynak ve hedef hakkında bilgi sahibi olmaya gerek duymadan, kendi isteğine göre değiştirebilir.

Servis Dışı Bırakma Saldırısı: Günümüzde önlenmesi en zor ataklardan biri olan servis dışı bırakma saldırılarının hemen hemen hepsinde IP sahteciliği saldırısı kullanılır. Bu kullanım servis dışı bırakma saldırılarının tespit edilebilmesini ve önlenmesini daha da zorlaştırır. Çünkü saldırının kaynaklandığı IP adresi aralığını filtrelemek oldukça güçleşir.

IP sahteciliği yöntemi Bölüm 2.5.2'de yer alan örnekte de anlatıldığı gibi sadece aynı alt ağa yapıldığı zaman farkedilebilir. Bu amaçla güvenlik duvarı bulunduğu özel ve tasnif dışı ağ dilimlerinde yapılabilecek IP sahteciliği denemelerini tespit edebilmelidir.

3.2.4.2. Güvenlik Duvarlarına Uygulanması

IP Sahteciliği Testi: Sahte kaynak IP adresli paket güvenlik duvarına gönderilecektir.

3.2.5. Sağlaması⁷ Hatalı Paket Testi

3.2.5.1. Açıklama

Bir IP datagramı ağda ilerlerken bütünlüğünün bozulup bozulmadığı TCP ve IP başlığında geçen sağlama değeri ile anlaşılır. Paket hatta basılmadan önceki hesaplanan değer ile hedef bilgisayarda hesaplanan değer aynı olması beklenir. Bir paketin değiştirilmiş olma ihtimali göz önünde bulundurularak sağlama değeri güvenlik duvarı tarafından kontrol edilmelidir. Eğer yanlışlık varsa paket düşürülmelidir.

⁷ Checksum

3.2.5.2. Güvenlik Duvarlarına Uygulanması

Sağlaması hatalı paket testi: Güvenlik duvarına sağlaması hatalı paketi gönderilecektir.

3.2.6. Arazi Saldırısı Testi (LAND attack)

3.2.6.1. Açıklama

Arazi Saldırısı, IP sahteciliği saldırısını ve açık TCP portlarını kullanarak gerçekleştirilir. Saldırgan SYN bayrağı aktive edilmiş bir tcp paketini hedefe gönderir. Fakat bu paketin kaynak IP adresini hedef bilgisayarın IP adresi ile aynı ayarlar. Bu paket hedef bilgisayara ulaştığında SYN paketine cevap verir. Cevap verirken ACK bayrağını aktive eder ve paketi hedefe yani kendisine geri gönderir. Fakat bu ACK bayrağı aktive edilmiş paket, gönderilen sıra no'su ile aynı no'da yani bir arttırılmamış olarak geldiği için paketin hedefe ulaşmadığını düşünür ve tekrar ACK bayraklı paket tekrar yollar. Bu durum sonsuz bir döngüye girer ve bu saldırının yaması yoksa hedef servis dışı kalabilir [Huovinen and Hursti, 2003].

Güvenlik duvarı bu tür saldırıları engelleyecek şekilde ayarlanmalıdır. Bu da kurum ağında bulunan IP adresleri kaynak IP adresi olarak ayarlanmış paketleri, dış ağdan kurum ağına geçirme kuralı tanımlanarak yapılabilir.

3.2.6.2. Güvenlik Duvarlarına Uygulanması

Aynı kaynak ve hedef IP adresli paket testi: Kaynak IP adresi hedef IP adresi ile aynı olan paket güvenlik duvarına gönderilecektir.

3.2.7. Geçersiz Kaynak IP Adresli Paket Testi

3.2.7.1. Açıklama

Internet üzerinden, özellikle tek paket ile yapılan saldırılarda iz kaybettirmenin en kolay yolu yanlış kaynak IP adresi vermektir. Bu amaçla genellikle 192.168.0.0/12, 10.0.0.0/8, 127.0.0.1 gibi gerçek ağlarda geçerli olmayan IP adresleri kullanılır. Bu sebeple geçersiz IP adreslerinden gelen paketler güvenlik duvarı tarafından engellenmelidir.

3.2.7.2. Güvenlik Duvarlarına Uygulanması

Geçersiz Kaynak IP Adresli Paket Testi: Kaynak IP adresi 127.0.0.1. olan

bir paket güvenlik duvarına gönderilecektir.

3.2.8. Bölünmüş Paket Testi

3.2.8.1. Açıklama

IP katmanına bir IP datagramı ulaştığında öncelikle bu datagramın hangi ethernet ara biriminden gönderileceği belirlenir. Daha sonra bu ara birim üzerinden gönderilecek maksimum transfer ünitesi (MTU) öğrenilir. IP datagramının boyutu MTU değerinden büyük ise datagram bölünerek gönderilir. Bölünme işlemi datagramın kaynağında olabileceği gibi, ara yönlendiricilerde de olabilir.

Bir IP datagramı bölündüğü zaman hedefe ulaşmaya kadar birleştirilmez. Bu paketler hedefin IP katmanına geldiğinde birleştirilir. Ayrıca bir kere bölünmüş bir IP datagramına ait paket kaynaktan hedefe doğru yolculuğunda bir kere daha bölünebilir. IP bölünmesi, IP başlığında geçen bazı alanlara belirli değerler atanması ile anlaşılır ve bu bilgiler kullanılarak IP datagramı hedefte tekrardan oluşturulabilir. Bu alanlar ve kullanımları şu şekildedir [Steven, 2001]:

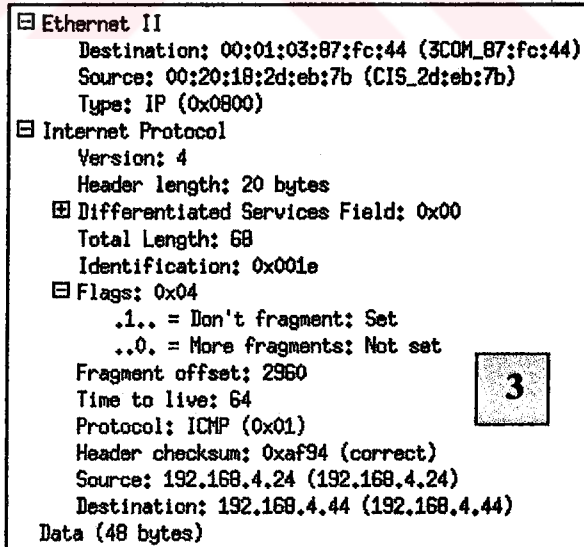
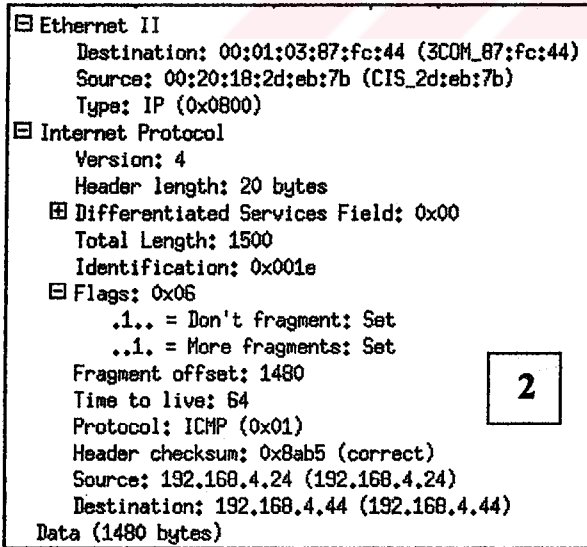
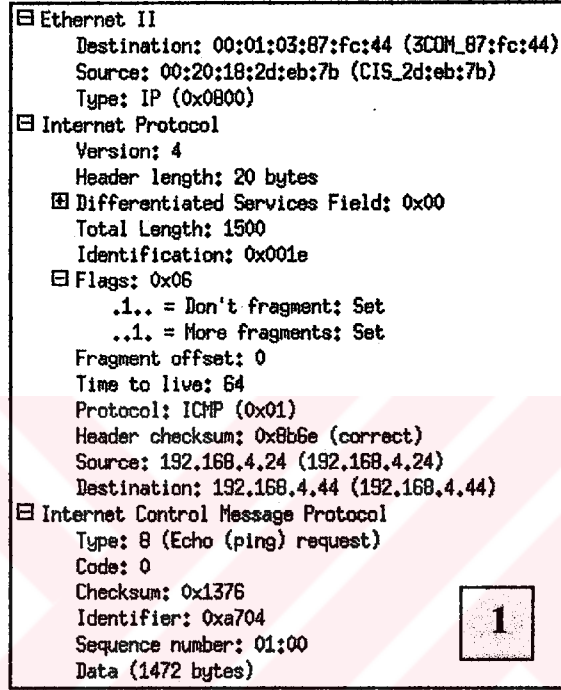
Kimlik(identification): Her IP datagramı için belirleyicidir ve bu datagrama ait bütün paketlerde aynı değeri alır.

Bayrak(flag): *More Fragments* ve *Don't Fragment* bayrakları IP bölünmesinde kullanılır. *More Fragments* alanı son paket IP datagramına ait bütün paketlerde aktive edilir. Aktif olduğu durumda kendisinin IP datagramına ait son paket olmadığını başka paketlerin geleceğini bildirir. *Don't Fragment* bayrağı ise IP datagramının hedefe ulaşana kadar IP bölünmesine uğramamasını şart koşar. Eğer datagram bir yönlendiricide MTU değerini aşıyor ve hedefe başka yollarla iletilemiyorsa "fragmentation needed but don't fragment bit set" ICMP hatası döndürülür.

Bölünme ofseti: IP datagramının başından bu paketin içerdiği verinin başladığı noktaya kadar olan veri boyutunu içerir. Bu değer her 8 byte bir birim olarak kabul edilir. IP başlığında bölünme ofseti için ayrılmış değer 13 bittir. Yani bir IP datagramının bölünebilmesi için boyunun maksimum $2^{13} \times 8 = 65536$ byte olması gerekmektedir.

Toplam uzunluk: Bu alanın değeri bölünmüş parçanın boyutuna eşittir.

Şekil 3.4'de boyutu 3000 byte olan bir IP-ICMP datagramının MTU değeri 1500 byte olan ethernet ara biriminden gönderilmesi esnasında yukarıdan açıklanan alanların alacağı değerler gösterilmektedir. Ekran kopyaları "ethereal" programından alınmıştır. Her pakette IP başlığı bulunduğu için maksimum transfer edilebilecek verinin boyutu 1480 byte'tır. Ayrıca ilk pakette giden ICMP başlığının da boyutu toplam boyuta eklenmelidir ($8 + 3000 = 3008$) [Novak, 2001].



Şekil 3.4: Bölünmüş ICMP paketlerinin yapısı

3008 byte verinin ilk 1480 byte'ı (ICMP Başlığı+ 1472 byte veri) ilk pakette, ikinci 1480 byte'ı ikinci pakette, geri kalan 48 byte'ı ise üçüncü pakette

gönderilmektedir. Birinci ve ikinci paketlerde *More Fragment* bayrağı 1'dir. Üçüncü pakette ise 0'dır. *Fragmentation offset* ise birinci pakette 0, ikinci pakette 1480, üçüncü pakette ise 2960'dır.

IP datagramlarının bölünme sistemi bu şekildedir. Güvenlik duvarları ağ üzerindeki trafiği filtrelerken bölünmüş paketleri doğru şekilde kotarmalı, bu paketleri kullanarak yapılan saldırıları engellemelidir.

3.2.8.2. Güvenlik Duvarlarına Uygulanması

Bölünmüş paketler güvenlik duvarlarına değişik formatta gönderilerek düzgün kotarılıp kotarılmadıkları kontrol edilir.

İzin verilmeyen bölünmüş paket testi: İzin verilmeyen kural için test modülünde tanımlı paket sayısı kadar bölünmüş paket güvenlik duvarına gönderilir.

TCP bölünmüş paket testi: İzin verilen kural için test modülünde tanımlı paket sayısı kadar tcp bölünmüş paketi güvenlik duvarına gönderilir.

UDP bölünmüş paket testi: İzin verilen kural için test modülünde tanımlı paket sayısı kadar udp bölünmüş paketi güvenlik duvarına gönderilir.

Sahte IP adresli bölünmüş paket testi: Kaynak IP adresi sahte olan bölünmüş paket güvenlik duvarına gönderilir.

TCP karışık bölünmüş paket testi: Hazırlanan tcp bölünmüş paketleri karışık sırada gönderilir.

UDP karışık bölünmüş paket testi: Hazırlanan udp bölünmüş paketleri karışık sırada gönderilir.

Maksimum datagram uzunluğunu geçen bölünmüş paket testi: Maksimum IP datagram uzunluğu 65536 byte'tan daha büyük bir veri bölünerek güvenlik duvarına gönderilir.

Ofset değerleri çakışan TCP bölünmüş paket testi: Ofset değeri olması gerekenden küçük olan bölünmüş tcp paketleri gönderilir. Güvenlik duvarı du paketleri birleştirilirken hata vermelidir.

Ofset değerleri çakışan UDP bölünmüş paket testi: Ofset değeri olması gerekenden küçük olan bölünmüş udp paketleri gönderilir. Güvenlik duvarı du paketleri birleştirilirken hata vermelidir.



3.2.9. Kaynak Yönlendirmeli Paket Testi

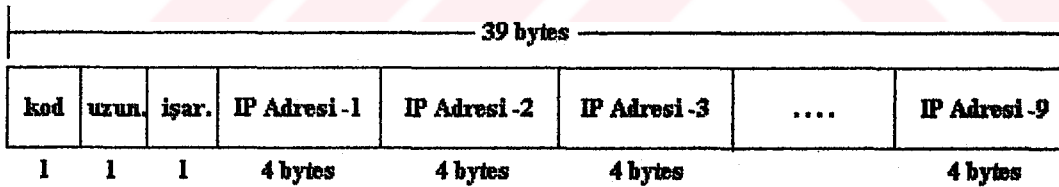
3.2.9.1. Açıklama

Internet üzerinde kötüye kullanılacak en kolay yollardan birisi de kaynak yönlendirmesidir. Normal şartlarda yönlendirme her yönlendiricinin vermiş olduğu karara göre yapılır. Kaynak yönlendirmesi, otomatik yönlendirmenin çalışmadığı durumlarda bağlantı kurabilmek amaçlı kullanılır. İki çeşit kaynak yönlendirmesi vardır [STD-5, 1985].

Sıkı(Strict) kaynak yönlendirmesi: Gönderici IP datagramının ilerleyeceği yolu tamamıyla belirler. Datagramın ulaştığı yönlendirici eğer paketin içerisinde bulunan bir sonraki yönlendiriciye ulaşamazsa ICMP “source route failed” hatasını döndürür.

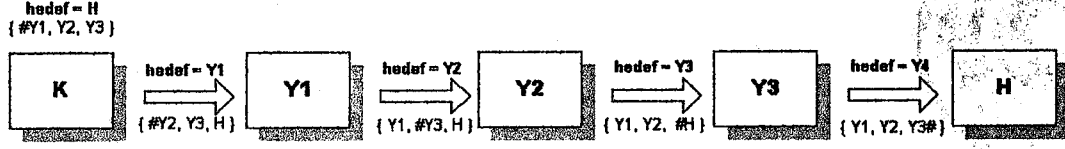
Gevşek(Loose) kaynak yönlendirmesi: Gönderici, bir IP datagramının geçmesi gereken bir grup adresi belirler. Datagram listede geçen iki IP adresi arasında başka yönlendiricilere de gidebilir.

IP başlığındaki kaynak yönlendirme opsiyonunun yapısı Şekil 3.5’deki gibidir. Listedeki en fazla dokuz IP adresi yer alabilir. Kod bölümü, kaynak yönlendirmesinin sıkı ya da gevşek olmasını belirler.



Şekil 3.5: IP başlığındaki kaynak yönlendirmesi için genel format [Steven, 2001]

Bir istemci paketi gönderileceği zaman kaynak yönlendirme listesindeki ilk IP adresini alarak paketin hedef IP adresi yapar. Gerçek hedef IP adresi ise yönlendirme listesinin en sonuna eklenir. En son olarak da en baştaki girdi silinerek liste sola doğru bir 4 byte kaydırılır. Aradaki yönlendiriciler ise işaretçinin gösterdiği noktadaki IP Adresini alıp hedef IP adresi yaparlar, bu alana kendi IP adreslerini yazarlar. İşaretçiyi ise listede bir sonra gelen IP adresine yönlendirirler [Steven, 2001]. Örnek bir kaynak yönlendirme Şekil 3.6’de görülmektedir.



Şekil 3.6: Kaynak yönlendirmesi örneği [Steven, 2001]

Kaynak yönlendirmesi TCP/IP protokolünün faydalı özelliklerinden biri olmasına karşın saldırganlar tarafından kötüye kullanılabilir. Örneğin Şekil 1.1'deki genel ağ örneği ele alınsın. Dış kullanıcı isimli yetkisiz kullanıcının, kaynak IP adresi yine dış ağda bulunan yetkili kurum kullanıcısının IP adresini ile aynı ayarlanmış bir paketi kurum ağına yolladığı farzedilsin. Normal şartlarda bu pakete verilecek cevap dış ağda bulunan yetkili kurum kullanıcısına gidecektir. Fakat bu IP sahteciliği saldırısı kaynak yönlendirme ile beraber kullanılırsa yetkisiz kullanıcı bu pakete gelen cevabı kendi ağına doğru yönlendirebilecektir. Bu şekilde kurum ağında yetkili elde edebilecektir.

Örnekten de anlaşılacağı gibi, kaynak yönlendirmesi özellikle IP sahteciliği ile birlikte kullanıldığında etkili saldırılara imkan vermektedir. Bu sebeple güvenlik duvarları kaynak yönlendirmesi içeren paketleri geçirmemelidir.

3.2.9.2. Güvenlik Duvarlarına Uygulanması

Kaynak yönlendirmesi içeren paketler güvenlik duvarına gönderilecektir.

İzin verilmeyen bağlantı için kaynak yönlendirme testi: Test izin verilmeyen bir bağlantı üzerinden denenecektir.

İzin verilen bağlantı için kaynak yönlendirme testi: Test izin verilen bir bağlantı üzerinden denenecektir.

3.2.10. Bağlantı Seli Testi

3.2.10.1. Açıklama

İnternet üzerinden son yıllarda en sık yapılan saldırı türü servis dışı bırakma saldırılarıdır. Özellikle otomatik araçların da geliştirilmesiyle bu saldırı daha da kolaylaşmıştır. Bu saldırıda bir ya da birkaç bilgisayar, hedef bilgisayara çok sayıda bağlantı açıp ya da bu bilgisayarın maksimum bağlanılabilme sayısını doldurarak başkalarının bağlanmasını engellemekte ya da açılan çok sayıda bağlantı ile bilgisayarın kaynaklarını tüketmekte ve diğer isteklere geç cevap verilmesine sebep

olmaktadır. Bu saldırı güvenlik duvarları için de geçerlidir. Bu sebeple güvenlik duvarlarının ne kadar bağlantıyı destekledikleri test edilmelidir.

3.2.10.2. Güvenlik Duvarlarına Uygulanması

Bağlantı seli saldırıları birkaç formatta güvenlik duvarına uygulanır.

İzin verilen kural için TCP bağlantı seli testi: İzin verilen bir bağlantı için test modülünde tanımlı paket sayısı kadar tcp bağlantısı kurulmaya çalışılır.

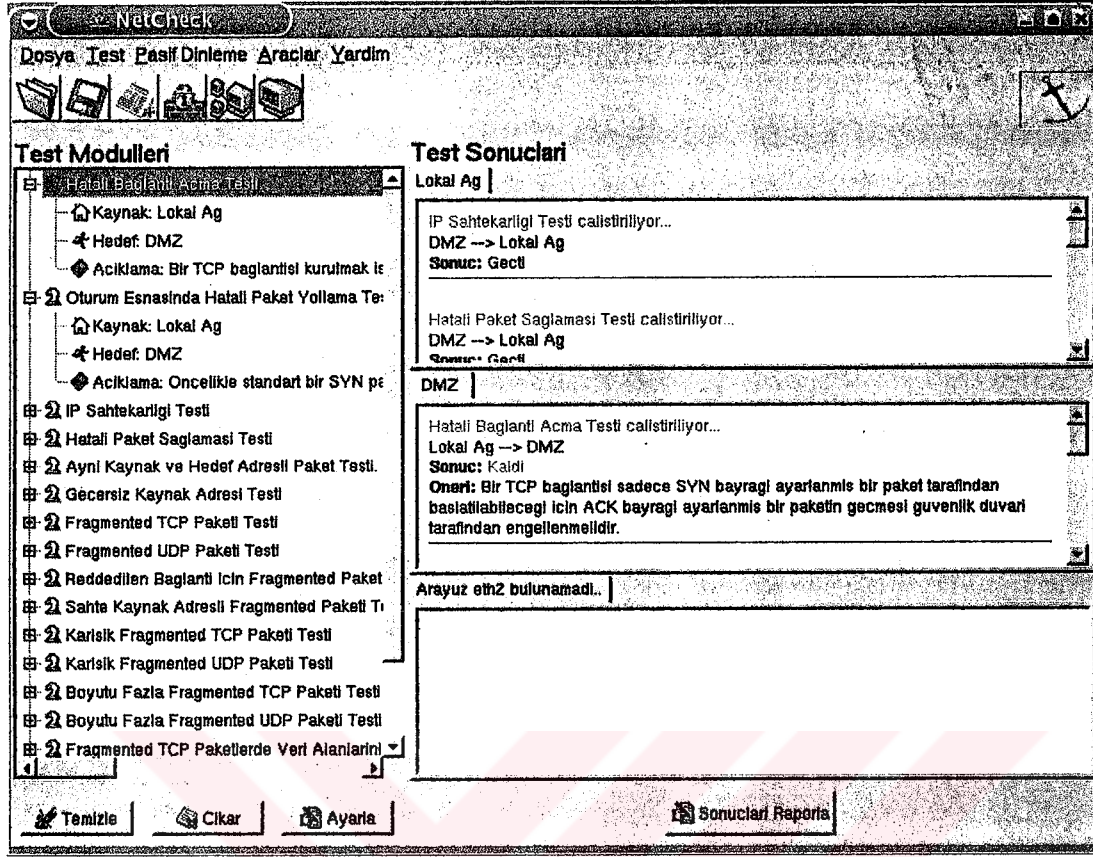
İzin verilmeyen kural için TCP bağlantı seli testi: İzin verilmeyen bir bağlantı için test modülünde tanımlı paket sayısı kadar tcp bağlantısı kurulmaya çalışılır.

İzin verilen kural için UDP bağlantı seli testi: İzin verilen bir bağlantı için test modülünde tanımlı paket sayısı kadar udp bağlantısı kurulmaya çalışılır.

İzin verilmeyen kural için UDP bağlantı seli testi: İzin verilmeyen bir bağlantı için test modülünde tanımlı paket sayısı kadar udp bağlantısı kurulmaya çalışılır.

3.3. NetCheck'e Genel Bir Bakış

NetCheck kullanımı oldukça kolay bir kullanıcı ara birimi ile detaylı güvenlik duvarı testleri gerçekler. Şekil 3.7'de NetCheck'in ana penceresi görülmektedir. Sol taraftaki panel'de test modülleri listelenmektedir. Sağ panelde ise yapılan test sonuçların hedef ağ dilimi isimlerine göre sınıflandırılarak gösterilmektedir. Kullanıcılar bu pencere üzerinde test modülleri ekleyebilir, çıkarabilir, topoloji konfigürasyonunu ayarlayabilir, hazırlanacak sonuç raporu hakkında bazı bilgileri girebilir. Ayrıca sonuç raporları, *html* formatında kaydedilerek daha sonra herhangi bir Web tarayıcısı ile açılabilir.



Şekil 3.7: NetCheck Ana Penceresi

3.4. Test Metodolojisi ve Ağlara Entegrasyonu

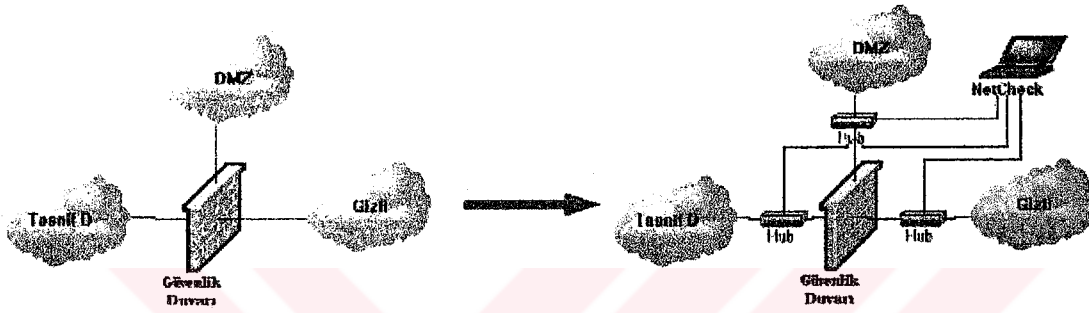
NetCheck Bölüm 2’de sınıflandırılan güvenlik duvarı test metodolojilerini birleştirerek ara katman bir metodoloji sunar. Diğer bir deyişle testleri halihazırda işleyen bir ağı koruyan güvenlik duvarları üzerinde gerçeklerken test laboratuvarlarında alınabilecek sağlıklı sonuçları alır. Testler otomatik olarak gerçekleşir. Bu sebeple çok fazla kullanıcı etkileşimine ihtiyaç duymaz. Test metodolojisi belirli adımlardan oluşur. Bu adımlar şöyledir:

3.4.1. Ağa Entegrasyon

NetCheck’in ağlara entegrasyonu Şekil 3.8’de görüldüğü gibi oldukça kolaydır. Sadece güvenlik duvarının ayırdığı ağ dilimleri sayısınca “hub” ve NetCheck’in üzerinde koştuğu bilgisayar kullanılarak bu entegrasyon yapılabilir. Hub’lara, segmentlere ait anahtar ya da hubların güvenlik duvarından uzak olması durumunda ihtiyaç vardır. Eğer bu aktif cihazlar, güvenlik duvarına yakınsa ve üzerlerinde boş

port varsa kullanılabilir. NetCheck'in kurulu olduğu bilgisayarın diz üstü olması taşınabilirlik ve aynı konfigürasyonla birkaç ağ topolojisinin test edilip sonuçların karşılaştırılabilmesi açısından kolaylık sağlar. Ayrıca NetCheck'in üzerinde çalıştığı bilgisayarda en az güvenlik duvarının ayıldığı ağ dilimi sayısı kadar ethernet ara birimi bulunmalıdır.

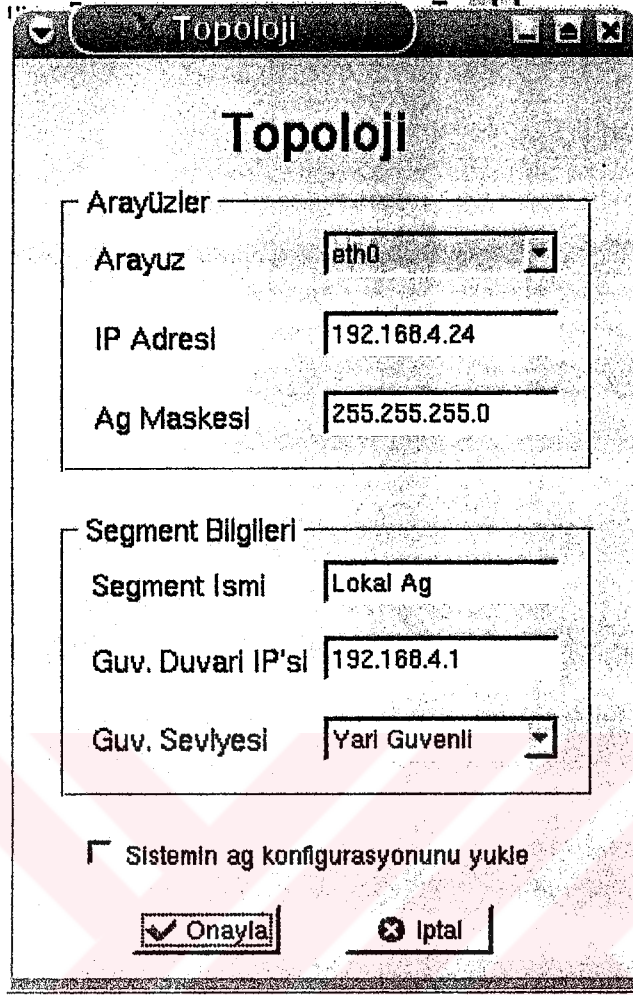
Bu basit ve esnek yapı, güvenlik duvarı testlerinin maliyetini oldukça düşürür. Ayrıca test ortamı kurma aşamasında gereken ağ ya da güvenlik testi uzmanlığına da ihtiyaç bırakmaz.



Şekil 3.8: NetCheck'in ağa entegrasyonu

3.4.2. Topoloji konfigürasyonu yapma

NetCheck, güvenlik duvarının ethernet ara birimleri ile ayrılan alt ağları analiz eder. Bunun için, testten sorumlu kişiden her alt ağ için bazı bilgiler alır. Bu bilgiler, güvenlik duvarının ilgili alt ağa bakan IP adresi, alt ağ maskesi, NetCheck tarafından kullanılacak ve başka bir bilgisayar tarafından kullanılmayan IP adresi, bu ağ dilimi için verilen mantıksal bir isim ve bu ağ biriminin güvenlik seviyesidir. Şekil 3.9'de topoloji konfigürasyonunun yapıldığı pencerenin ekran kopyası görülmektedir.



Topoloji

Arayüzler

Arayuz: eth0

IP Adresi: 192.168.4.24

Ag Maskesi: 255.255.255.0

Segment Bilgileri

Segment Ismi: Lokal Ag

Guv. Duvarı IP'si: 192.168.4.1

Guv. Seviyesi: Yarı Güvenli

☐ Sistemin ag konfigürasyonunu yükle

Şekil 3.9: Topoloji Konfigürasyon Penceresi

Konfigürasyon esnasında NetCheck'in kullandığı IP adresi, güvenlik duvarının IP'si ile aynı alt ağda verilmelidir. Aksi durumda NetCheck bu konfigürasyon için hata verecektir. Çünkü testin yapılması esnasında güvenlik duvarının MAC adresinin öğrenmek için NetCheck, kendi bulunduğu ağa ARP paketi yayını yapacaktır. Eğer NetCheck ile güvenlik duvarı aynı alt ağda değil ise bu ARP isteğine cevap alamayacağı için test devam edemeyecektir.

. Her alt ağ için bir mantıksal isim tanımlamak zorunludur. Bu mantıksal isim testi yapacak kişiye kolaylık sağlamak amacıyla test modüllerinde kullanılacaktır. Ayrıca her alt-ağ için güvenlik seviyesi tanımlanması yapılacak testler için gereklidir. Çünkü testlerin çoğu belirli bir güvenlik seviyesinden bir diğerine doğru gerçekleştirilmektedir. Güvenlik seviyeleri güvenli, yarı güvenli ve güvensiz olarak tanımlıdır. Güvensiz ağ, kontrolü yapılamayan bir başka değişle sahip olunan ağın

dışı olarak kabul edilen bölümdür. Yarı güvenli ağ(DMZ) hem güvenli hem de güvenli ağlardan erişilebilen genellikle kurum sunumcularının bulunduğu bölümdür. Güvenli ağ ise sadece bu ağdan başlatılan bağlantıların cevaplarını alabilen diğer ağlardan gelen bağlantı isteklerine kapalı olan ağdır. Testin sorumlusu olan kişi bu kriterlere göre her ağın güvenlik seviyesini belirlemelidir.

3.4.3. Test modüllerini belirleme ve ayarlama

Bir sonraki adım ise test modüllerinin belirlenmesidir. NetCheck test modülleri iki gruba ayrılır. Bunlar gizli ve açık test modüleridir. Gizli testler topolojiyi sınar ve güvenlik duvarının filtreleme politikasını inceler. Bu test modüllerinin yapısını kullanıcılar göremezler. Sadece Ayarlar -> Tercihler -> Testler menüsünde, test süreci başladığında gizli testlerin yapılıp yapılmayacağı tercihinde bulunulabilir.

Açık testler, test modüllerinin büyük bir kısmını oluşturur. Bu tür testler güvenlik duvarının bir ağ diliminden diğerine paketler yollayarak güvenlik duvarını geçmeye çalışır. Bu modüllerin hangi tür saldırıları yapabildiği Bölüm 3.2’de anlatılmıştı. Açık test modülleri, ana pencerede sol taraftaki panelde yer alır. Ağaç yapısında girdiler içeren bu panelde herhangi bir test modülünün kenarına tıklandığında bu modüle ait kaynak ve hedef ağ dilimi isimleri, test modülü hakkında açıklama gibi bilgilere erişilebilir. (Şekil 3.7)

Açık test modülleri, *f1m*(firewall test modülü) uzantılı dosyalarla yüklenebileceği gibi *f1y*(firewall test yatağı) uzantılı dosyalarla da yüklenebilir. Bu dosyalar metin tipinde olup herhangi bir editörde görülebilir, değiştirilebilir. Örnek bir açık test modülü Çizelge 3.1’de görülmektedir.

Çizelge 3.1. Örnek bir açık test modülü

Parametre İsmi	Değer
İsim	Hatalı Bağlantı Acma Testi
Kaynak	Internet
Hedef	DMZ
Protokol	Tcp
Hedef Port	53
Paket Sayısı	4
Opsiyon	ACK
Beklenen Sonuç	Dusur
Acıklama	Bir TCP bağlantısı kurulmak istendiğinde gönderilecek pakette ACK bayrağı etkin hale getirilir ve bu paketin güvenlik duvarından geçip geçmediği gözlenir.
Uyari	Bir TCP bağlantısı sadece SYN bayrağı ayarlanmış bir paket tarafından başlatılabileceği için ACK bayrağı ayarlanmış bir paketin geçmesi güvenlik duvarı tarafından engellenmelidir.

Açık test modüllerinde belirleyici değerler parametre isimlerinin karşısına yazılır. Kullanılan parametre isimleri şunlardır.

İsim: Testin ismini içerir.

Kaynak: Testin hangi kaynaktan paket yollayacağını belirler. Bu alanın karşısına topoloji ayarlarında belirlenen mantıksal isimler kullanılır.

Hedef: Paketlerin hedefinin ve dinlemenin hangi ağ dilimi üzerinde yapılacağını belirler. Kaynak parametresi gibi değer olarak mantıksal isimler alır.

Acıklama: Test hakkında detaylı açıklama içerir.

Protokol: Yollanacak paketin protokolünü belirler. Desteklenen protokoller *tcp*, *udp* ve *icmp* dir.

Beklenen Sonuç: Yollanan paketlere karşı güvenlik duvarının davranışının ne olması gerektiğini belirler. *Düşür* ya da *Geçir* değerlerini alır.

Uyari: Testin sonucu olumsuz olursa güvenlik duvarının sahibi olan kişi ya da kuruma verilecek tavsiyeyi içerir.

Kaynak IP: Kaynak IP parametresi tanımlandığında standart kaynak IP değeri olan NetCheck'in IP adresi yerine bu değer kullanılır.

Hedef IP: Hedef IP parametresi tanımlandığında standart hedef IP değeri olan NetCheck'in IP adresi yerine bu değer kullanılır.

Hedef Port: Paketlerin hedef aldığı port no'sunu belirler. Sadece *tcp* ve *udp* bağlantılarında kullanılır.

Kaynak Port: Paketlerin kaynak port no'sunu belirler. Sadece *tcp* ve *udp* bağlantılarında kullanılır.

Paket Sayisi: Kaç tane paket yollanacağını belirler

Opsiyon: Test modüllerinin karakteristiğini belirleyen parametredir. Bir değer alacağı gibi birkaç değer de alabilir. Bu değerler rezerve edilmiş sözcükler ile ifade edilir. Bu sözcükler ve anlamları **Çizelge 3.2**'de gösterilmektedir.

Test modülü parametrelerinden ilk yedisinin(*Isim, Kaynak, Hedef, Açıklama, Protokol, Beklenen Sonuc, Uyari*) karşısında bir değer olması zorunludur. Ayrıca *Protokol* değeri "tcp" veya "udp" olan modüllerde hedef port belirlenmesi gereklidir.

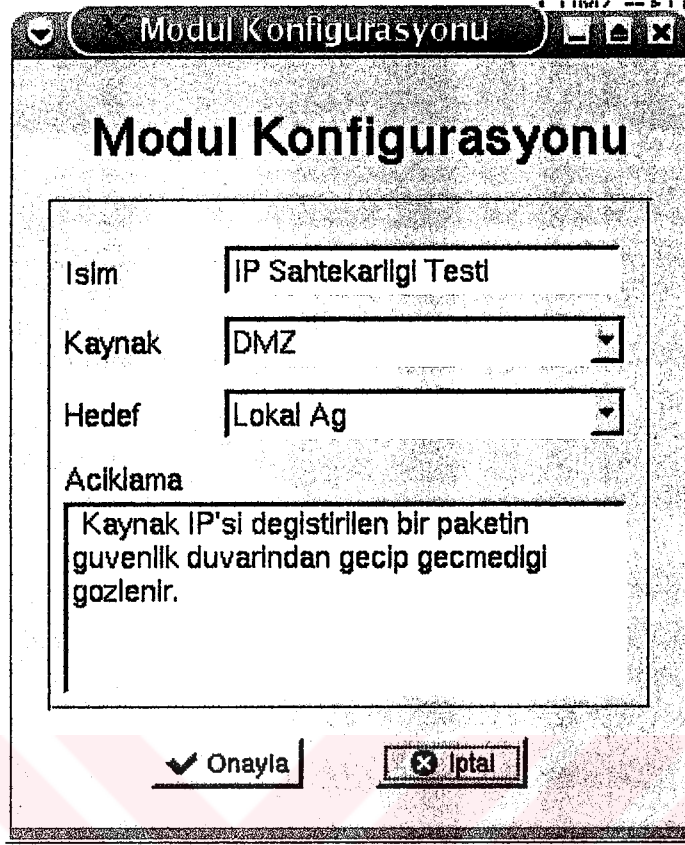
Çizelge 3.2. Opsiyon rezerve sözcükleri

Rezerve Sözcükler	Anlamı
Spoofed	Sahte IP adresi üretir.
Fragmented	Parçalı IP paketi üretir. Eğer <i>Paket Sayısı</i> tanımlı ise bu sayı kadar paket oluşturulur. Aksi durumda dört paket yollanır.
W_Fragmented	Bozuk parçalı IP paketi üretir. Eğer <i>Paket Sayısı</i> tanımlı ise bu sayı kadar paket oluşturulur. Aksi durumda dört paket yollanır. Parçalı paketlerin "Fragmentation Offset" alanına yanlış değer girilerek bozuk paketler oluşturulur.
M_Fragmented	Karışık parçalı IP paketi üretir. Eğer <i>Paket Sayısı</i> tanımlı ise bu sayı kadar paket oluşturulur. Aksi durumda dört paket gönderilir. Yollanan paketler şu sıradadır. 4-3-2-1-8-7-6-5-9...
O_Fragmented	Maks. paket boyutu olan 65536 boyundan büyük parçalı paket gönderilir. Gönderilen paket sayısı $65536/sizeof(payload)+2$ dir.
W_CheckSum	IP başlığındaki sağlama bilgisini yanlış bir değere atar.
After_Conn	Bu opsiyon genellikle protokolü tcp olan test modüllerinde diğer opsiyonlarla beraber kullanılır. Öncelikle standart bir bağlantı kurulmaya çalışılır. Daha sonra diğer opsiyon girdileri kullanılarak gerekli trafik oluşturulur.
Land	Kaynak IP adresini hedef IP adresine eşitler.
InvalidIP	Kaynak IP adresine 127.0.0.1 gibi geçersiz bir değer atar.
Source_Route	Yönlendirmesinin kaynak tarafından belirlendiği paketler hazırlar.
SYN, FIN, RST, PUSH, ACK, URG	TCP başlığında kullanılan ilgili bayrak değerlerini aktive eder. Diğer opsiyonlarda olduğu gibi bu rezerve sözcüklerin bir kaçı aynı anda kullanılabilir.

Opsiyon parametresine karşılık gelen rezerve sözcükler birlikte kullanıldıklarında bazı çelişkilerin olması olasıdır. Örneğin *Invalid_IP* opsiyonu ile

Spoofed opsiyonu birbiri ile çakışmaktadır. Çünkü iki opsiyon da kaynak IP adresi üzerinde bazı değişiklikler yapmaktadır. Bu sebeple opsiyonlar arasında bir öncelik tanımlaması yapılmıştır. Bu sıralama önceliği çok olandan az olana doğru *Kaynak IP*, *Source_Route*, *Spoofed*, *Land*, *Invalid_IP* şeklindedir.

NetCheck'in kullanmış olduğu modüller yapı yeni test modülleri geliştirilmesine imkan tanır. Bu tanımlama işlemi, metin editörlerde halihazırda tanımlı açık test modüllerini açıp bazı değişiklikler yaparak gerçekleştirilebilir. Bu modüller NetCheck'in kurulu olduğu dizinde bulunan *methods* izini altındadır. Bununla beraber NetCheck yazılımı üzerinden de bazı ufak değişiklikler yapılabilmektedir. Şekil 3.7'de görülen NetCheck ana penceresinin solundaki Test Modülleri paneline eklenen bir modül seçilip sol altta yer alan Ayarla düğmesine basıldığında açılan pencere bu amaçlı kullanılır. Şekil 3.10'de bu pencere görülmektedir. "Modül Konfigürasyonu" penceresinde modülün ismi kaynak ve hedef ağ dilimlerinin mantıksal isimleri ve açıklama değiştirilebilir. Bir modülün kaynak ve hedefi için seçilen ağ dilimlerinin farklı olmasına dikkat edilmelidir.



Modul Konfigurasyonu

Isim: IP Sahtekarligi Testi

Kaynak: DMZ

Hedef: Lokal Ag

Aciklama: Kaynak IP'si degistirilen bir paketin guvenlik duvarindan gecip gecmedigi gozlenir.

Onayla Iptal

Şekil 3.10: Modül Konfigürasyonu penceresi

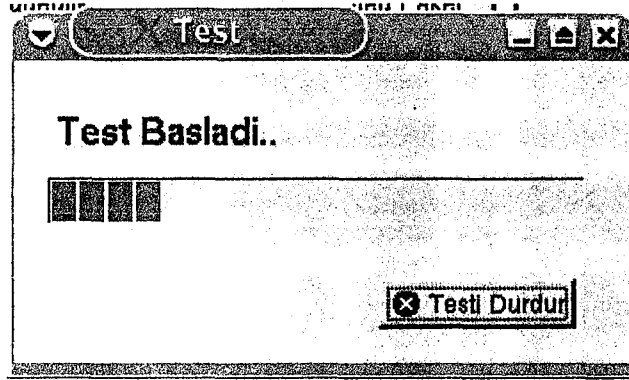
3.4.4. Testi başlatma

Test modülleri de belirlendikten sonra testi başlatma aşamasına geçilir. NetCheck yazılımı öncelikle ağ ve test modül konfigürasyonun doğru olup olmadığı kontrol eder. Testin yapılabilmesi için en azından bir açık test modülü yüklü olmalıdır ya da Ayarlar-> Tercihler penceresinde “Gizli testleri Gerçekle” seçeneği seçili olmalıdır. Ayrıca her test modülünün kaynak ve hedef ağ dilimleri için tanımlanmış mantıksal isimler topoloji ayarları ile aynı olmalıdır.

Eğer seçili ise öncelikle gizli testler gerçekleştirilir. Daha sonra açık testlere geçilir. Her bir testin sonucu sağ tarafta yer alan test sonuçları panelinde başlığı modülün hedef ağ dilimi ile aynı olan metin alanına yazılır (Şekil 3.7). Eğer bir testin sonucu başarısız ise ilgili test modülünde geçen Uyarı parametresi karşısında anlatılan öneri de metnin sonuna eklenir.

Testlerin genel yapısı, belirli formatta paket yollama ve bu paketi hedef ağ diliminde dinleme şeklindedir. NetCheck’in gönderici ve dinleyici kısımları ayrı prosesler olarak çalışır ve ortak kullandıkları bellek alanı üzerinden haberleşirler.

Ayrıca test süreci boyunca testin ne kadarının tamamlandığı hakkında bilgi veren pencere de gönderici bölümün bir parçası(thread) olarak çalışır. Bu pencere Şekil 3.11’de gösterilmektedir. Kullanıcı istediği zaman testi bu pencere üzerinden durdurabilir. Bu durumda pencere yakalayıcı bölüme bir sinyal yollar. Yakalayıcı kısım da belirli aralıklarla kontrol ettiği sinyali alarak dinleyici prosesi öldürür, o ana kadar yapılan testlerin sonuçlarını sağdaki panele yazar.



Şekil 3.11: Test Prosesi penceresi

3.5. Diğer Özellikler

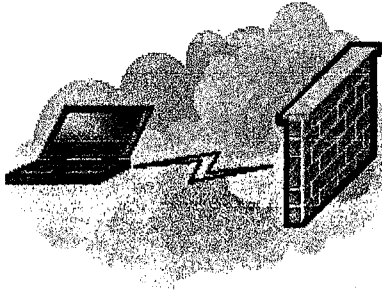
NetCheck ile yapılan testlerin sonuçları, daha sonra değerlendirmek üzere *html* formatında kaydedilebilir. Bunun için test bitiminde aktif hale gelen “Sonuçları Raporla” tuşuna basmak yeterlidir. Öncelikle kullanıcının karşısına, raporun hazırlanması esnasında gerek duyulan bazı bilgileri alan bir pencere çıkar. Bu pencerede müşteri firmanın ismi, testi yapan firmanın ve test sorumlusunun isimleri istenir. Ayrıca bütün test modüllerinin mi yoksa sadece başarısız test sonuçlarının mı raporlanacağı tercihi test sorumlusundan alınır. Bu bilgiler girildikten sonra rapor bir pencere içerisinde *html* formatında sunulur. Örnek bir rapor penceresi de görülmektedir.

Report





Guvenlik Duvari Testi Sonuclari



Yapilan Test Sayisi: 11	Testi Yapan Firma: GYTE
Bulunan Aciklik Sayisi: 5	Musteri Firma: TRUST INF. SYSTEM
Tarih: Thu May 22 09:29:59 2003	Test Sorumlusu: Tahsin Turkoz

Topoloji Bilgileri

Lokal Ag

Guv. Duvar Arayuzunun IP'si: 192.168.4.1

Alt-ag Maskesi: 255.255.255.0

Test aracinin IP'si: 192.168.4.24

 Kaydet
 Kapat

Şekil 3.12: Rapor penceresi

Güvenlik duvarı testlerinde, test hazırlığı olarak da nitelendirilebilecek “Topoloji Konfigürasyonu Yapma” ve “Test Modüllerini Belirleme” aşamaları hazırlanan test yataklarının kaydedilmesi ile kısaltılabilir. NetCheck kullanıcılarına bu imkanı sağlar. Test yatakları *.fity* uzantılı dosyalara kaydedilebilir. Bu dosyalarda “---METHODS---” başlığı altında test modülleri “---NETWORK DEVICES---”

başlığı altında ise her bir ağ dilimi için tanımlı ağ konfigürasyonları bulunmaktadır. Test modülleri birbirlerinden “Method” kelimesi ile ayrılmaktadır.

NetCheck kullanıcılarına farklı dillerde test modülü tanımlama ve kullanıcı arayüzlerinde farklı dillere ait etiketleri kullanma imkanı da vermektedir. Bunun için NetCheck kurulumunun kök dizininin altındaki *conf* dizinine *dil* uzantılı dosyalar koymak gereklidir. Standart kurulumda *English.dil* dosyası program ile beraber gelmektedir. Bu dosya değiştirilip [dil_ismi].dil ismi ile aynı dizine kaydedilirse NetCheck programı bu dosyayı tanıyacak ve Araçlar->Tercihler->Dil penceresinde kullanıcıların seçimine sunacaktır. Bu dosyada sadece program içerisinde kullanılan etiketler değil aynı zamanda test modülleri ve test yatakları dosyalarında parametre isimleri de belirlenebilir. Örneğin kullanım dili İngilizce olarak seçildiğinde Çizelge 3.3’deki test modülü NetCheck tarafından tanınacaktır.

Çizelge 3.3. İngilizce açık test modülü

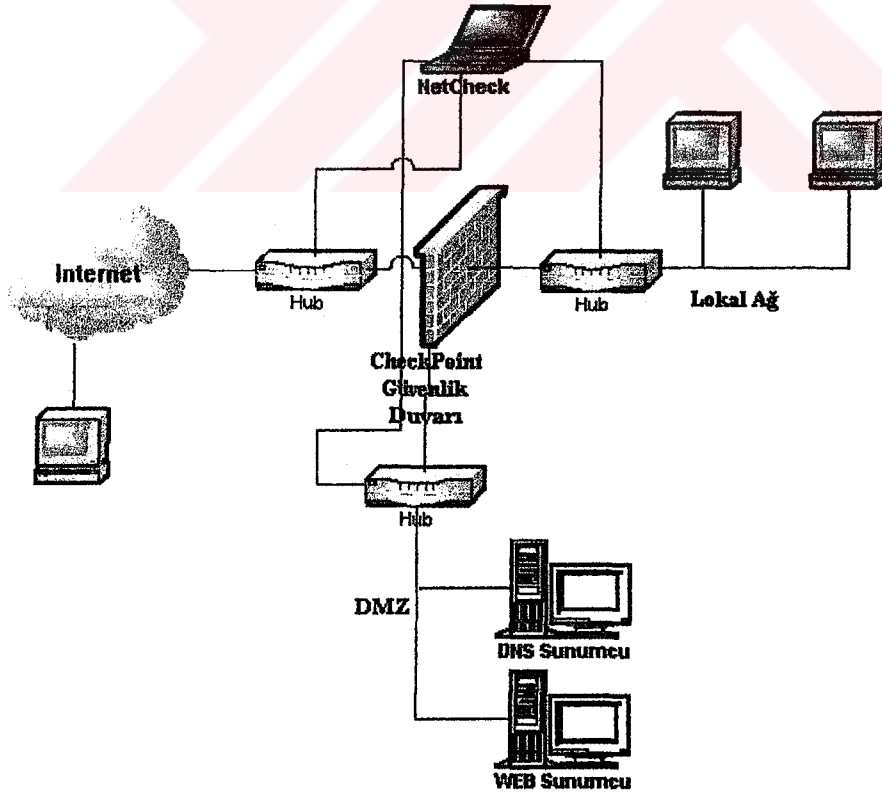
Parametre İsmi	Değer
Name	IP Spoofing test
Source	Internet
Destination	DMZ
Protocol	Tcp
Destination Port	53
Packet Number	4
Option	Spoofed
Expected Result	Drop
Description	Will be test whether Firewall prevents IP spoofing.
Suggestion	IP spoofing is not regular connection method which is generally used by attackers to receive the packets belong to others or try to establish connection with the rights of granted hosts. Therefore, IP Spoofing should be prevented by Firewall.

4. NETCHECK İLE ÖRNEK TESTLER, DİĞER METODOLOJİLER İLE KARŞILAŞTIRMA

Bu bölümde, NetCheck güvenlik duvarı test metodolojisinin uygulanması hakkında daha detaylı bilgi verebilmek amacıyla iki örnek ağ üzerinde testler gerçekleştirilmiştir. Daha sonra iki test sonucu karşılaştırılıp hangi ağın daha güvenli olduğu hakkında fikir yürütülmüştür. Son bölümde ise NetCheck, Bölüm 2.4 ve 2.5’de detaylı bir şekilde anlatılan güvenlik duvarı test metodolojileri ile karşılaştırılmıştır.

4.1. NetCheck Örneği 1

Birinci örnekte güvenlik duvarı testi yapılacak ağ Şekil 4.1’deki gibi belirlenmiştir. Güvenlik duvarı testlerinde güvenlik duvarının bulunduğu konum, ağ dilimlerinin sayısı ve güvenlik seviyesi testlerin sonucunu doğrudan etkileyen bir faktördür. Bu konu Bölüm 3.2.1’de anlatılmıştır.



Şekil 4.1: Örnek test yatağı -1

Örnek topolojide kurumun ağı Internet ortamına bağlıdır. Internet ağı ile kurum

ağı arasında bir güvenlik duvarı bulunmaktadır. Güvenlik duvarı üzerinde üç ağ ara birimi tanımlıdır. Bu ara birimlerden birisi dış dünya ikisi ise kurum ağı ile irtibatlıdır. Kurum ağı kendi içerisinde güvenlik seviyeleri farklı olan iki alt ağa ayrılmıştır. Bunlardan birisine diğer alt ağdan ve Internet ağından erişilen sunumcular diğerinde ise kuruma ait bilgisayarlar bağlıdır. Bu alt diliminin güvenlik seviyesi yarı güvenli'dir. Diğer alt ağda ise kurum çalışanlarının bilgisayarları bulunmaktadır ve güvenlik seviyesi güvenli'dir. NetCheck test aracı üç ağ dilimine birer hub cihazı ile bağlıdır. Ağ dilimleri şu şekilde modellenmiştir:

$$\text{Ağ_Dilimi}_0 = \left\{ \begin{array}{l} \text{İsim:Lokal Ağ} \\ \text{GdIP:10.10.1.1} \\ \text{AğMas:255.255.255.0} \\ \text{NCIP:10.10.1.9} \\ \text{GüvSev:0} \end{array} \right\} \quad (4.1)$$

$$\text{Ağ_Dilimi}_1 = \left\{ \begin{array}{l} \text{İsim:DMZ} \\ \text{GdIP:10.10.2.1} \\ \text{AğMas:255.255.255.0} \\ \text{NCIP:10.10.2.9} \\ \text{GüvSev:1} \end{array} \right\} \quad (4.2)$$

$$\text{Ağ_Dilimi}_2 = \left\{ \begin{array}{l} \text{İsim:Internet} \\ \text{GdIP:10.10.3.1} \\ \text{AğMas:255.255.255.0} \\ \text{NCIP:10.10.3.9} \\ \text{GüvSev:2} \end{array} \right\} \quad (4.3)$$

Kurum ağını Internet'ten ayıran güvenlik duvarı Windows 2000 işletim sistemi üzerinde çalışan CheckPoint güvenlik duvarı olarak belirlenmiştir. Filtreleme politikası ise Şekil 4.2'deki gibi tanımlanmıştır. Kurum iç ağa ve dış dünyaya http ve dns servisleri sunmaktadır. Bu sebeple DMZ ağında 80. ve 53. portlar Internet ağından erişime açılmıştır. Lokal ağdan diğer ağlara erişim kısıtlanmamıştır. DMZ, Internet ağına sınırsız haklarla erişebilmektedir. Lokal Ağa ise hiçbir ağdan erişime izin yoktur.

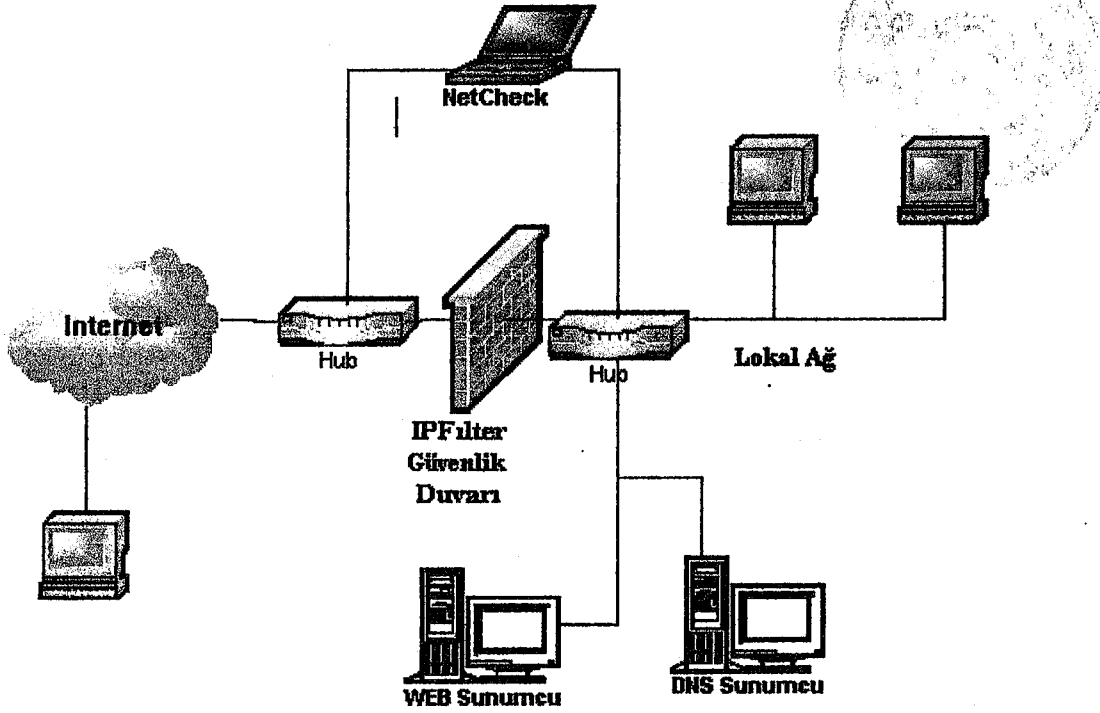
NO	SOURCE	DESTINATION	IP VIA	SERVICE	ACTION	TRACK	INSTALL ON
1	Internet	DMZ	* Any	* PORT_23	accept	- None	* Policy Targets
2	Internet	DMZ	* Any	* PORT_25	accept	- None	* Policy Targets
3	Internet	DMZ	* Any	* TCP_ANY	drop	- None	* Policy Targets
4	Internet	DMZ	* Any	* UDP_ANY	drop	- None	* Policy Targets
5	Internet	local_ag	* Any	* Any	drop	- None	* Policy Targets
6	local_ag	DMZ	* Any	* Any	accept	- None	* Policy Targets
7	local_ag	Internet	* Any	* Any	accept	- None	* Policy Targets
8	DMZ	Internet	* Any	* Any	accept	- None	* Policy Targets
9	DMZ	local_ag	* Any	* Any	drop	- None	* Policy Targets

Şekil 4.2: CheckPoint güvenlik duvarı için belirlenen kural tablosu

Topoloji ayarlaması yapıldıktan sonra NetCheck tarafından sunulan açık test modülleri yüklenmiştir. Ağ dilimleri için tanımlanmış mantıksal ağ isimleri açık test modüllerinin kaynak ve hedef ağ dilimlerine karşılık gelen değerler olarak girilmiştir. Bazı test modülleri, sunulan servis sayısınınca çoğaltılmış ve hedef IP adresleri ilgili sunumcuya göre belirlenmiştir. Gizli test modüllerinin de çalıştırılması ayarı yapılmıştır. Bu ayarların sonucunda test başlatılmıştır. Test sonuçları Ek -2'de sunulmuştur.

4.2. NetCheck Örneği 2

İkinci örnekte güvenlik duvarı testi yapılacak ağ Şekil 4.3'deki gibi belirlenmiştir. Örnek topolojide kurumun ağı Internet ortamına bağlıdır. Internet ağı ile kurum ağı arasında bir güvenlik duvarı bulunmaktadır. Güvenlik duvarı üzerinde iki ağ ara birimi tanımlıdır. Bu ara birimlerden birisi dış dünya ikincisi ise kurum ağı ile irtibatlıdır. Kurum ağının güvenlik seviyesi yarı güvenlidir.



Şekil 4.3: Örnek test yatağı -2

Ağ dilimleri şu şekilde modellenmiştir:

$$\text{Ağ_Dilimi}_0 = \left\{ \begin{array}{l} \text{İsim:Lokal Ağ} \\ \text{GdIP:10.10.2.1} \\ \text{AğMas:255.255.255.0} \\ \text{NCIP:10.10.2.9} \\ \text{GüvSev:1} \end{array} \right\} \quad (4.4)$$

$$\text{Ağ_Dilimi}_1 = \left\{ \begin{array}{l} \text{İsim:Internet} \\ \text{GdIP:10.10.3.1} \\ \text{AğMas:255.255.255.0} \\ \text{NCIP:10.10.3.9} \\ \text{GüvSev:2} \end{array} \right\} \quad (4.5)$$

Kurum ağını Internet'ten ayıran güvenlik duvarı OpenBSD işletim sistemi üzerinde çalışan IPFilter güvenlik duvarı olarak belirlenmiştir. Filtreleme politikası ise Şekil 4.4'deki gibi tanımlanmıştır. Kurum dış dünyaya http ve dns servisleri sunmaktadır. Bu sebeple 10.10.2.15 IP adresli sunumcunun tcp/80. portu ve 10.10.2.20 IP adresli sunumcunun udp/53. portu Internet ağından erişime açılmıştır. Diğer erişimler engellenmiştir. Lokal ağdan Internet ağına erişim ise kısıtlanmamıştır.

```

scrub in all

pass in quick on rl2 proto tcp from any to 10.10.2.15 port 80
flags S keep state
pass in quick on rl2 proto udp from any to 10.10.2.20 port 53
keep state

pass out quick on rl2 from any to any

block in on rl2 from any to any

```

Şekil 4.4: IPFilter güvenlik duvarı için belirlenen kural tablosu

Topoloji ayarlaması yapıldıktan sonra NetCheck tarafından sunulan açık test modülleri yüklenmiştir. Ağ dilimleri için tanımlanmış mantıksal ağ isimleri açık test modüllerinin kaynak ve hedef ağ dilimlerine karşılık gelen değerler olarak girilmiştir. Bazı test modülleri sunulan servis sayısının çoğaltılmış ve hedef IP adresleri ilgili sunumcuya göre belirlenmiştir. Gizli test modüllerinin de çalıştırılması ayarı yapılmıştır. Bu ayarların sonucunda test başlatılmıştır. Test sonuçları Ek -3'de sunulmuştur.

4.3. İki Sonucun Karşılaştırılması

NetCheck'in sunmuş olduğu sonuç raporu test sonuçlarını karşılaştırılabilme imkanını vermektedir. Çizelge 2.1'de sonuç raporlarının son bölümünde yer alan tablolar birleştirilmiş ve her modülde güvenlik duvarlarının nasıl davranış sergiledikleri karşılaştırılmıştır. "+" işareti testin başarılı olduğunu, "-" başarısız olduğunu, "?" ise testin güvenlik duvarına uygulanmadığını göstermektedir.

Test sonuçları incelendiğinde ikinci test yatağının kurulumunun güvenli olmadığı görülmektedir. Her iki ağda da kurum kullanıcılarının dış dünyaya erişimleri kısıtlanmamış böylelikle bilgi açığa çıkarma saldırılarına imkan tanınmıştır. Birinci ağda icmp ping istekleri ile kurum lokal ağının yapısı ve aktif bilgisayarlar hakkında bilgi edinmek mümkündür. Ayrıca dış dünyaya hizmet veren sunumculara erişimler yeterince kısıtlanmamış, geniş servis bir aralığı erişime açık bırakılmıştır. Checkpoint güvenlik duvarı oturum esnasında gönderilen hatalı paketleri geçirirken IPFilter IP sahteciliği saldırısına karşı savunmasızdır. Bununla beraber Checkpoint güvenlik duvarı, sağlama değeri hatalı ve bölünmüş paketlerde başarısızdır. Ayrıca bağlantı sellerinde ve kaynak yönlendirmesi yapılan paketlerde de etkisizdir.

Sonuç olarak birinci test yatağında filtreleme kurallarında bazı hatalar bulunmasının yanında güvenlik duvarının birçok saldırıda başarısız olduğu görülmektedir. Öte yandan IPFilter güvenlik duvarı genel anlamda saldırıları engellerken kullanıldığı topolojik yapıda eksiklikler bulunmaktadır.

Çizelge 4.1. İki güvenlik duvarı test sonucunun karşılaştırılması

Testin Adı	Checkpoint 3.0 Güv. Duvarı Test Sonuçları	IPFilter Güv. Duvarı Test Sonuçları
Topoloji Konum Testi	+	-
Guvenli Segmente Erisim Testi 1	+	+
Guvenli Segmente Erisim Testi 2	+	?
Guvensiz Segmente Erisim Testi 1	-	-
Guvensiz Segmente Erisim Testi 2	-	?
Yari Guvenli Segmente Erisim Testi 1	+	?
Yari Guvenli Segmente Erisim Testi 1	+	?
ICMP Echo Istek Testi	-	+
WEB Sunumcusu Erisim Testi -1	-	+
WEB Sunumcusu Erisim Testi -2	-	+
DNS Sunumcusu Erisim Testi -1	-	+
DNS Sunumcusu Erisim Testi -2	-	+
SYN ve ACK bayrakli paket yollanmasi	+	+
FIN bayrakli paket yollanmasi	+	+
Oturum Esnasinda Hatali Paket Yollama Testi	-	+
IP Sahtekarligi Testi	+	-
Hatali Paket Saglamasi Testi	-	+
Ayni Kaynak ve Hedef Adresli Paket Testi.	+	-
Gecersiz Kaynak Adresi Testi	-	+
Izin Verilmeyen Bolunmus Paket Testi	-	+
TCP Bolunmus Paket Testi	-	+
UDP Bolunmus Paket Testi	-	+
Sahte Kaynak Adresli Bolunmus Paket Testi	+	-
TCP Karisik Bolunmus Paket Testi	-	+
UDP Karisik Bolunmus Paket Testi	-	+
Boyutu Fazla Bolunmus TCP Paketi Testi	+	+
Bolunmus TCP Paketlerinde Veri Alanlarinin Cakismasi Testi	+	-
Bolunmus UDP Paketlerinde Veri Alanlarinin Cakismasi Testi	+	-
Izin Verilen Baglanti Icin Kaynak Yonlendirme Testi	-	+
Izin Verilen Kural Icin TCP Baglanti Seli Testi	-	+
Izin Verilmeyen Kural Icin TCP Baglanti Seli Testi	+	+
Izin Verilen Kural Icin UDP Baglanti Seli Testi	-	+
Izin Verilmeyen Kural Icin UDP Baglanti Seli Testi	+	+

4.4. NetCheck İle Diğer Metodolojilerin Karşılaştırılması

Bölüm 2 ve 3’de günümüze kadar geliştirilmiş güvenli duvarı test metodolojileri üzerinde durulmuştur. Bu metodolojiler birbirleri ile kıyaslandığında bazı avantajları ve dezavantajları içerdikleri görülmektedir.

Adı geçen güvenlik duvarı test metodolojilerinin karşılaştırılabilmesi için bazı kriterler belirlenmiştir. Bu kriterler belirlenirken risk analiz metotlarını karşılaştıran bir çalışma [Karabacak, 2003] incelenmiş ve kriterler güvenlik duvarlarına adapte edilip genişletilmiştir. Kriterler şunlardır:

Güvenilirlik: Metodolojinin sunduğu test yönteminin sonuçları güvenilir olmalıdır. Başka metodolojilerle desteklenmeye ihtiyaç bırakmamalıdır.

Hız: Test süreci kısa bir zamanda tamamlanmalıdır.

Maliyet: Testler, maliyet etkin bir şekilde gerçekleştirilmelidir. Her test için maliyetli test yatakları kurmaya ihtiyaç bırakmamalıdır.

Kapsam: Test metodolojisi gerek test laboratuvarlarında kurulan gerekse aktif kullanımda olan güvenlik duvarlarına uygulanabilir olmalıdır.

Bütünlük: Güncel saldırıları kapsamlı bir şekilde içeriyor olmalıdır.

Karşılaştırılabilirlik: Sunduğu sonuçlar karşılaştırılabilir formatta olmalıdır.

Kısıtlamalar: Kullanılan güvenlik duvarı ya da kurulu ağ dilimi sayısı metodolojinin uygulanabilirliğini etkilememelidir.

Kullanım Kolaylığı: Test sorumluları tarafından kolaylıkla kullanılabilmeli, ağ güvenliği veya TCP/IP altyapısı gibi konularda derinlemesine bilgi sahibi olmaya ihtiyaç bırakmamalıdır.

Yönlendirme: Bulunan açıklıklar ve bu açıklıkların kapatılma yolları hakkında anlaşılabilir tavsiyelerde bulunmalıdır.

Otomatize Yapı: Testler otomatik olarak gerçekleştirilebilmelidir.

Bu kriterler ile açıklanan üç güvenlik duvarı kıyaslandığında şu sonuçlar ortaya çıkmaktadır:

Her üç metodolojinin sonuçları güvenilirirdir. Öte yandan bir test metodolojisinin verdiği sonuçların güvenilirliği kendi kapsamında

değerlendirilmelidir. Diğer bir deyişle bir testin sonucunun güvenilir olması test edilen ögenin güvenilir olduğu anlamına gelmez.

Metodolojiler testlerin tamamlanma süresine göre kıyaslandığında Al-Tawil'in metodolojisinin oldukça süre aldığı görülmektedir. Çünkü öncelikle test yatağı hazırlanmalı daha sonra testte göz önünde bulundurulan kriterler belirlenmelidir. Vigna'nın metodoloji de kısa sürede sonuç vermez. Çünkü günümüzde kullanılan test yöntemlerinin hepsi bu modelde tanımlı değildir. Metodolojide yer alan beş adım kapsamlı bir şekilde ele alınmalıdır. NetCheck ise test modüllerini genel formatta içermekte ve ufak değişikliklerle farklı yapıdaki topolojilere uygulanabilmektedir. Bu sebeple testler kısa sürede tamamlanır.

Birinci metodolojide test yatağı kurulması zorunlu olduğu için ekstra maliyet gerekir. Diğer metodolojilerde ise maliyet düşüktür.

Al-Tawil'in metodolojisi test laboratuvarlarında kullanılmaya elverişli iken Vigna'nın metodolojisi aktif olarak kullanılan ağlarda oluşabilecek açıklıklar üzerinde durur. NetCheck ise orta katman bir metodoloji sunarak hem test laboratuvarlarında hem de gerçek ağlarda etkin testler yapar.

Güvenlik duvarları güvenlik kriterleri göz önünde bulundurularak incelendiğinde öncelikle yapılması gereken çalışma, saldırı metotlarının etrafıca incelenerek sağlam bir altyapı oluşturulmasıdır. Al-Tawil metodolojisi bu basamağı atmış ve otomatik açıklık analiz araçlarını bu amaçla kullanmayı tavsiye etmiştir. Öte yandan bu araçlar, işletim sistemlerinde olabilecek açıklıkları bulmaktan öte bir işlev görmemekte, güvenlik duvarına özel testleri gerçekleştirememektedir. Vigna'nın metodolojisinde ise öncelikle bu konuda çalışma yapılmıştır [Vigna, 1996]. NetCheck de böyledir (Bkz. Bölüm 3.1).

Al-Tawil'in test sonuçları karşılaştırılabilir formattadır. Sonuçlar alt gruplara ayrılarak tablolar halinde test sürecinin sahibine sunulur. Vigna'nın metodolojisinde ise raporlama bölümü yer almamaktadır. NetCheck, yapmış olduğu testlerin sonuçlarını, sonuç raporunun sonunda yer alan bir tabloda karşılaştırılabilir formatta sunmaktadır.

Al-Tawil'in metodolojisinde, uygun topoloji kurulursa açıklanan test metodolojisi uygulanabilir. Öte yandan kompleks topolojilerin kurulması testin maliyetini yükseltecektir. Ağ dilimi sayısında da bir sınırlama yoktur. Vigna'nın test

metodolojisi de durum böyledir. Fakat topolojinin karışıklığı testin maliyetini yükseltmez. Sadece test tamamlanma süresini arttırır. NetCheck de ise güvenlik duvarı sayısı testleri etkilemezken ağ dilimi sayısının üçü geçtiği durumlarda ağın aynı anda test edilmesi mümkün değildir. Bu durumda üçlü ağ dilimi kombinasyonları seçilip testler yapılır. Sonuç raporları da birleştirilerek testler tamamlanır. Bu çözüm, NetCheck'in sunmuş olduğu kolaylıklar ile çeliştiği için bu konuda başarısız addedilebilir.

Al-Tawil'in ve Vigna'nın test metodolojileri derin ağ ve bilgi güvenliği tecrübesi gerektirir. Test yatağı kurulumu, saldırıların modellenmesi gibi konular sistemlerinin güvenlik seviyeleri hakkında bilgi edinmek isteyen sıradan ağ yöneticilerinin ağ güvenliği alanındaki bilgilerini aşabilecek konulardır. NetCheck ise basit anlaşılabilir bir metodoloji ile önceden hazırlanıp kullanıma sunulmuş test modüllerini içerir.

Vigna'nın metodolojisi ve NetCheck, testler sonucunda bulunan açıklıkların kapatılması hakkında öneriler içerir. Al-Tawil'in metodolojisi de ise sadece sonuçlar raporlanır. Öneri içermez.

İncelenen metodolojilerden Al-Tawil'inki kısmen otomatik, NetCheck tam otomatik yapıda çalışır. Vigna'nın metodolojisi ise tamamıyla el yordamıyla uygulanır.

Çizelge 4.2'de yapılan karşılaştırmaların özeti bulunmaktadır. Karşılaştırmalar genel bir incelenecek olursa NetCheck'in diğer iki metodolojiye kıyasla daha esnek, kullanılabilir bir yapı sunduğunu, birçok yönlerden diğer metodolojilerin yaptığı işleri yapabildiği görülmektedir.

Çizelge 4.2. Güvenlik duvarı test metodolojilerinin karşılaştırılması

Karşılaştırma Kriteri	Al-Tawil'in metodolojisi	Vigna'nın metodolojisi	NetCheck metodolojisi
Güvenilir mi?	Evet. Yaptığını iddia ettiği testler gözlemlere ve ölçülebilir bilimsel değerlere dayanır.	Evet. Test metotları tcp/ip tabanlı bilinen açıklıklara dayanır.	Evet. Test metotları tcp/ip tabanlı bilinen açıklıklara ve güvenlik duvarlarında oluşabilen zayıflıklara dayanır.
Kısa sürede sonuçlanabiliyor mu?	Hayır. Test yatağının seçimi ve kurulumu, güvenlik duvarının kurulumu, test adımlarının uygulanması iki-üç ay gibi bir süre alır.	Hayır. İhtiyaçların belirlenmesi, ortam doğrulaması ve tcp/ip tabanlı açıklıkların modellenmesi, iki ay gibi bir süre alır.	Evet. Entegrasyon, sisteme uygun test adımlarının seçimi ve uygulanması bir günde tamamlanabilir.
Maliyeti yüksek mi?	Evet. Test yatağı kurulum maliyeti, test sırasında kullanılan yazılımlar ve sarfedilen iş gücü maliyeti yükseltir.	Hayır. Maliyet, testi uygulayan personelin maliyetine eşdeğerdir.	Hayır. Hem testin kısa sürede sonuçlanması hem de sadece personel maliyeti olması bu metodolojiyi maliyet etkin kılar.
Test laboratuvarlarında uygulanabiliyor mu?	Evet. Jenerik yapılarda uygulanan testlerle farklı güvenlik duvarları test edilebilir.	Hayır. Sadece aktif ağlarda uygulanır. Çünkü testler ağırlıklı olarak güvenlik duvarının topolojik konumu üzerinedir. Test ortamında başarılı bulunan bir güvenlik duvarı aktif olarak farklı topolojik konumlarda kullanılırken başarısız görülebilir.	Evet. Jenerik yapılarda uygulanan testlerle farklı güvenlik duvarları test edilebilir.
Aktif ağlardaki güvenlik duvarlarına uygulanabiliyor mu?	Hayır. Sadece test laboratuvarlarında uygulanır. Çünkü güvenlik duvarının açılıp kapatılması, konfigürasyonunun değiştirilmesini gerektirir.	Evet. Metodoloji sadece aktif ağlarda sağlıklı sonuç verebilir.	Evet. Güvenlik duvarının işlevselliğini bozmadığı ve konfigürasyon değişikliği gerektirmediği için uygulanabilir.

Karşılaştırma Kriteri	Al-Tawil'in metodolojisi	Vigna'nın metodolojisi	NetCheck metodolojisi
Güvenlik Duvarına özel testleri kapsamlı bir şekilde içeriyor mu?	Hayır. Yapılan testler kullanıcı etkileşimi, konfigürasyon kolaylığı gibi kriterleri içerirken, kullanılan tarama araçları sadece işletim sistemi tabanlı açıklıkları bulma kabiliyetine sahiptir.	Evet. Hazırlanan test modülleri güvenlik duvarı açıklıklarına ve zafiyetlerine yöneliktir.	Evet. Hazırlanan test modülleri güvenlik duvarı açıklıklarına ve zafiyetlerine yöneliktir.
Karşılaştırılabilir formatta sonuç sunabiliyor mu?	Evet. Sonuçlar tablolar halinde sunulur. Farklı güvenlik duvarları için yapılan testler sonucunda hazırlanan tablolar kullanarak karşılaştırma yapılabilir.	Hayır. Test sonucunda formatı belirli, standart bir çıktı sunulmaz.	Evet. Sonuçlar raporun son bölümünde bir tabloda karşılaştırılabilir formatta sunulur.
Birden fazla güvenlik duvarının olduğu ağları test edebiliyor mu?	Evet. Kullanılabilecek güvenlik duvarı sayısında sınırlama yoktur.	Evet. Kullanılabilecek güvenlik duvarı sayısında sınırlama yoktur.	Evet. Kullanılabilecek güvenlik duvarı sayısında sınırlama yoktur.
Test yapılacak ortamdaki ağ dilimi sayısında sınırlandırma var mı?	Hayır.	Hayır.	Evet. Üçten fazla ağ dilimini bulunuyorsa tek basamakta testi tamamlamak mümkün değildir.
Test aşamasında yüksek derecede ağ güvenliği bilgisine ihtiyaç duyuluyor mu?	Evet. Testler testi yapan personelin bilgi seviyesi ve tecrübesine dayanır.	Evet. Testler testi yapan personelin bilgi seviyesi ve tecrübesine dayanır.	Hayır. Jenerik test modülleri kullanarak az bilgi birikimi ile testler gerçekleştirilebilir.
Bulunan açıklıklar için öneri içeriyor mu?	Hayır. Sonuçlar hakkında yorum yapılmaz.	Evet. Sonuçlar güvenlik duvarı için kullanılabilecek en uygun topolojiyi ve konfigürasyonu içerir.	Evet. Raporda başarısız test sonuçlarını öneriler izler.
Otomatik yapıda mı?	Yarı Otomatik. Testlerin bir kısmı otomatik bir kısmı ise gözlem tabanlıdır.	Hayır. Testlerin hepsi manuel metotlarla gerçekleşir.	Evet. Tamamıyla otomatik yapıda çalışan bir araç sunar.

5. SONUÇLAR VE GELECEK ÇALIŞMALAR

Bu çalışmada günümüze kadar geliştirilmiş güvenlik duvarı test metodolojileri incelenmiş, daha sonra yeni bir metodoloji geliştirilmiştir. Bu metodolojinin geliştirilmesindeki amaç bilinen test metodolojilerinin eksikliklerini kapatmak ve güvenlik duvarı testlerinin daha hızlı ve kolay yapılmasını sağlamaktır.

İncelenen metodolojilerden birincisi Al-Tawil ve Al-Kaltham tarafından geliştirilmiştir. Bu metodolojinin test laboratuvarında kullanımı daha uygundur. Metodoloji ilk olarak doğru test ortamının belirlenmesi ile başlar. Daha sonra testler gerçekleşir. Testler otomatik ve gözlem tabanlı testler olmak üzere ikiye ayrılır. Otomatik testler, güvenlik duvarı değişik formlarda çalışırken açıklık tarama araçları ile yapılan analizlerden oluşur. Öte yandan bu araçlar, güvenlik duvarları için geliştirilmemiş, işletim sistemlerinin açıklıklarını bulmayı amaçlayan yazılımlardır. Gözlem tabanlı testlerde ise kurulum, konfigürasyon, kullanım gibi alanlarda güvenlik duvarının kabiliyetleri, avantaj ve dezavantajları incelenip karşılaştırılabilir formatta sunulur.

İkinci metodoloji ise Vigna tarafından geliştirilmiştir. Bu metodoloji test laboratuvarlarından daha çok aktif ağlarda kullanılan güvenlik duvarlarını test etmede kullanışlıdır. TCP/IP ağlarının ve bu ağlardaki açıklıkların modellenmesi üzerine kuruludur. Öncelikle test edilecek ağ modellenir, ağ üzerindeki güvenlik ihtiyaçları ve ortamın bu ihtiyaçları karşılayacak yapıya sahip olup olmadığı belirlenir. Daha sonra test modülleri oluşturulup çalıştırılır.

Her iki test metodolojisi de güvenlik duvarlarının sağladıkları güvenlik seviyeleri hakkında belirli bir fikir verir. Fakat gerek kullanım alanları gerekse testlerin detayı noktalarında bazı eksiklikleri içerirler.

Bir güvenlik duvarının sağlamış olduğu güvenlik seviyesi, yapılan konfigürasyonun ve kurumlara özel topolojik kurulumun doğruluğuna bağlıdır. Ayrıca güvenlik duvarında güncel saldırılara karşı savunma mekanizmalarının bulunup bulunmadığı da etkilidir. Bütün bu faktörler göz önünde bulundurulduğunda güvenlik duvarı test metodolojilerinin hem test laboratuvarlarında hem de aktif ağda kullanılabilir olması, güncel saldırıları içeriyor olması ve kısa sürede ve az bilgi birikimiyle sonuca ulaşması gerekmediği ortaya çıkmaktadır.



Bu amaçla yeni bir güvenlik duvarı test metodolojisi ve bu metodolojiyi otomatik olarak gerçekleyen NetCheck yazılımı geliştirilmiştir. Saldırı yöntemleri hakkında geniş bir araştırma yapılmış, bu yöntemlerin uygulanmasını sağlayacak bir metodoloji sunulmuştur.

Metodolojinin dört aşamada uygulanır. Birinci aşamasında NetCheck yazılımı testin yapılacağı ağa entegre edilir, ikinci kullanıcıdan topoloji hakkında bilgi alınır. Ağ dilimleri, fonksiyonlarına bağlı olarak belirli güvenlik seviyeleri ile etiketlenir. Daha sonra, önceden tanımlı olan açık ve gizli test modüllerinden hangilerinin kullanılacağı ihtiyaca göre belirlenir. Son aşama ise test çalıştırılır.

Önerilen yöntemin gerçekleşmesi ve yapılan testler ile aşağıdaki sonuçlar elde edilmiştir.

1. Önerilen yöntem (Netcheck) düşük maliyetlidir. Testlerin gerçekleştirilebilmesi için bir bilgisayar ve ağ dilimi sayısınca çoklayıcı yeterlidir.
2. Kolay uygulanabilir bir yöntemdir. Ağ güvenliği konusunda ekspertiz ihtiyacı yoktur.
3. Gerekli konfigürasyon yapıldıktan sonra otomatik olarak çalışır. Bu sebeple testler kısa sürede tamamlanır.
4. Dinamik yapıya sahiptir. Yeni test modülleri eklenebilir. Tanımlı olanlar üzerinde değişiklikler yapılabilir.
5. Test laboratuvarlarında ve aktif olarak kullanılan ağlarda uygulanabilir. Güvenlik duvarının durdurulması ya da filtreleme politikasının değiştirilmesine ihtiyaç yoktur.
6. Sonuçlar karşılaştırılabilir formatta sunulur. Başarısız testler için güvenliği artırıcı yönde tavsiyeler içerir.

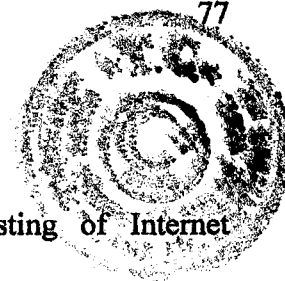
NetCheck yazılımı, güvenlik duvarı testlerinde birçok kolaylık sağlamanın yanında geliştirilmeye ve yeni opsiyonlar eklemeye elverişli bir altyapıya sahiptir. Bu altyapı kullanılarak aşağıda belirtilen çalışmalar yapılabilir:

- Daha fazla ağ diliminin yer aldığı topolojilerde testler, üçlü ağ dilimi kombinasyonları üzerinde tekrarlanır. Bu sınırı ortadan kaldıracak eklemeler.

- Uygulama seviyesinde vekil sunumculuk yapan güvenlik duvarlarına özgü testler (virüs içeren e-posta, html trafiği gibi) ilave edilmesi.
- NetCheck'in yapmış olduğu testlere uygun formattaki saldırıları yakalayabilecek saldırı tespit sistemi olarak tasarımı.



KAYNAKLAR



1. Al-Tawil K., Al-Kaltham A., 1999, "Evaluation and Testing of Internet Firewalls", International Journal of Network Management
2. Arbaugh W.A., Fithen W.L., and McHugh J., Dec. 2000, "Windows of Vulnerability: A Case Study Analysis," IEEE Computer, Vol. 33, No. 12
3. Bellovin S. M., 1989, "Security Problems in the TCP/IP Protocol Suite", Computer Communication Review, Vol. 19, No. 2, pp. 32-48
4. Blossom M. A., 2000, "Firewall Penetration Testing", <http://www.wittys.com/files/mab/fwpen testing.html>
5. CERT Coordination Center Report, 2000, "Internet Denial of Service Attacks and the Federal Response", <http://www.cert.org>
6. CERT Coordination Center Report, 2001, "Test The Firewall System", <http://www.cert.org>
7. CERT Coordination Center Report, 2002, "Overview of Attack Trends", <http://www.cert.org>
8. Checkley R., 1992, "Building secure systems. In Jackson, KM & Hruska, J (Eds.)", Computer Security Reference Book. Oxford, Butterworth-Heinemann
9. Cisco, 1998, "ICSA Firewall Buyer's", ICSA Inc. Report
10. Conoboy B., Fichtner E., 2001, "IP Filter Firewalls How-To", <http://www.nwo.net/ipf/ipf-howto.pdf>
11. Frantzen M., Kerschbaum F., Schultz E., 2001, "A Framework for Understanding Vulnerabilities in Firewalls Using a Dataflow Model of Firewall Internals". Computers & Security Vol.20,No:3,pp.263-270
12. Haeni R. E., 1997, "Firewall Penetration Testing", George Washington University Report

13. Herzog P., 2001, "Open-Source Security Testing Methodology Manual",
<http://www.ideahamster.org/>
14. Huovinen L., Hursti J., 2003, "Denial of Service Attacks: Teardrop and Land",
Helsinki University of Technology, Technical Report
15. Karabacak B., 2003, "Bilgi Güvenliği Risk Analizi (BIGRA) Yöntemi", Yüksek
Lisans Tezi, Gebze Yüksek Teknoloji Enstitüsü
16. Kurt S., Soğukpınar İ., 2002, "Developing Testing Methodology for Firewalls",
Istanbul University Journal of Electrical & Electronics, V.2, No:1, pg: 345-358
17. Mayer A., Wool A., Ziskind E., 2000, "Fang: A Firewall Analysis Engine", IEEE
Symposium on Security and Privacy
18. Moore A. P., 2001, R. J. Ellison, R. C. Linger, "Attack Modeling for Information
Security and Survivability", CMU/SEI-2001-TN-001
19. Muffett, A. 1994, "Proper Care and Feeding of Firewalls", SUN Microsystems
Report
20. Novak J., 2001, "Network Traffic Analysis Using Tcpdump", Proceeding of
SANS Intrusion Detection In-Depth
21. Pallar A., 2001, "Choosing and Justifying Right Intrusion Detection and
Vulnerability Analysis Tools", Proceeding of SANS Auditing Networks,
Perimeters and Systems Conference
22. SANS Organization, 2003, "SANS Critical Vulnerability Analysis" mailing list
23. Schultz E., Şubat 1997, "When Firewalls Fail: Lessons Learned From Firewall
Testing", Network Security Journal, Elsevier Science Ltd.
24. Scuba C., Spafford E., 1997, "A Reference Model for Firewall Technology",
Proceedings of 13th Annual Computer Security Applications Conference
(ACSAC)
25. STD-5, 1985, "Internet Protocol Specification", Information Sciences Institute
University of Southern California

- 
26. Steven W. R., Nov 2001, "TCP/IP Illustrated Volume 1", Addison-Wesley Professional Computing Series
 27. Tanese M., 2003, "Introduction to IP Spoofing", www.securityfocus.com
 28. Vigna G., 1996, "A Topological Characterization of TCP/IP Security", Politecnico di Milano Technical Report TR-96.156
 29. Vigna G., 1997, "A Formal Model for Firewall Testing", Politecnico di Milano Internal Report
 30. Wack J., Tracey M., Şubat 2002, "Guideline on Network Security Testing", NIST Special Publication 800-42
 31. Walder B., 2001, "Firewalls", NSS Labs, www.nss.co.uk
- 

ÖZGEÇMİŞ



Tahsin Türköz, 3 Ocak 1977 tarihinde İzmir’de doğdu. İlk, orta ve lise eğitimini İzmir’de tamamladı. Bilkent Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği bölümünden 2000 senesinde mezun oldu. 2000 senesi haziran ayından itibaren TÜBİTAK Ulusal Elektronik ve Kriptoloji Araştırma Enstitüsü’nde araştırmacı olarak görev yapmaktadır. Uzmanlık alanları, ağ güvenliği, güvenlik denetlemeleri ve testleri, Unix tabanlı işletim sistemlerinin güvenliği ve uygulama güvenliğidir.



EKLER



Ek-1: NetCheck'i Meydana Getiren Objeler ve Görevleri⁸

NetCheck Bileşenlerinin Listesi

<u>Method</u>	Metot String listesinde hata olup olmadigini kontrol eder
<u>MyListen</u>	Test metodu tarafından cagilir. Paket gonderildikten sonra bu paketin guvenlik duvarindan gecip gecmedigini gozlemler
<u>net</u>	Ag degerlerinin dogrulugunu kontrol eder
<u>TDconfirm</u>	Programin kullanimi esnasinda kullaniciya sorulmasi gereken sorulari belirli bir formatta kullaniciya sunar
<u>TestMethod</u>	Test modullerini calistiran metotlari iceren siniftir
<u>TFAbout</u>	Programin yapimcisi ve kullanim haklari hakkında bilgi veren pencereyi olusturur
<u>TFReport</u>	Raporu gosteren pencereyi olusturur
<u>TFTercihler</u>	Programin kullanimi ve testlerin yapilmasi esnasinda kullanicilarin tercihlerini ayarladigi bir diyalog kutusu olusturur
<u>TFTopoloji</u>	Ag topolojisi uzerinde ayarlamalarin yapildigi pencereyi olusturur
<u>TFwTest</u>	Program calistiginda ekrana gelen ana monudur. diger siniflara bu sinif uzerinden erisilir
<u>ThStart</u>	Test sureci basladiginda ekrana cikan StartBox'i ayri bir thread olarak acar
<u>TMethotConf</u>	Test modullerinin konfigurasyonunun yapildigi diyalog kutusunu olusturur
<u>TReportConf</u>	Test sonucunda hazirlanacak raporun ayarlarinin yapildigi bir

⁸ Ek-1 program uzerindeki yorumlar kullanilarak otomatik olusturulduđu ve programın geliřtirildiđi editör Tůrkçe karakterleri desteklemediđi iin Latin karakter seti (ISO-8859-1) kullanılmıřtır.

dialog kutusu olusturur

TStartBox

Test prosesi basladiginda bir dialog kutusu olustur. Progres bar ile testin ne kadarinin tamamlandigi hakkında bilgi verir

TUyari

Programin kullanimi esnasinda verilmesi gereken uyari mesajlarini belirli bir formatta kullaniciya sunar

Method Sınıfı Referansı

__fastcall Method (char *filename, TString * _MethodParams)

Sinifin kurucu metodu -1: Metotlari dosyadan alir.

__fastcall Method (TString * _method, TString * _MethodParams)

Sinifin kurucu metodu -1: Metodu cagiran metotdan alir.

void __fastcall init ()

Bazi degiskenlere baslangic degerlerini atar. Bazilari icin bellekte yer ayirir.

bool __fastcall ErrorCheck () throw ()

Hata kontrolu bu metotta yapilir. Sonucu dondurur.

bool __fastcall ParamsWork (TString *Param)

Metot String'inde kullanilacak parametre isimlerini orijinal(Turkce) isimlere ceviri.

bool __fastcall CheckOpt ()

Metot Stringinde Opsiyon parametresinin karsisinda yazan degerlerin dogrulukunu kontrol eder. Sonucu dondurur.

MyListen Sınıfı Referansı

__fastcall MyListen (AnsiString _name, int *_shcount, AnsiString _inter, AnsiString _filter)

Sinif kuruldugu esnada dinlemenin yapilacagi ethernet arayuzunun ismini, dinleme esnasinda kullanilacak filtreyi ve dinlenen paket sayisinin

yazilacagi paylasilan bellek alaninin isaretcisini alir.

void __fastcall **StartListen** () throw ()

Dinleme islemini baslatir.

void **my_callback** (u_char *useless, const struct
pcap_pkthdr *pkthdr, const u_char *packet)

Belirlenen paterne uyan bir paket yakalandiginda paket sayacini
bir arttirip paylasilan bellek alanina yazar.

net Sınıfı Referansı

virtual bool **netConfOk** (TStrings *SgList, int netdev, char *error)

virtual bool **isValidIP** (char *_IP, uchar *bitIP) throw ()

IP adresinin gecerli olup olmadigini kontrol eder. Bit formatinda
IP adresini ve sonucu dondurur.

bool **isValidIP** (char *_IP) throw ()

IP adresinin gecerli olup olmadigini kontrol eder. Sonucu
dondurur.

virtual bool **inSameSubnet** (uchar *bit_ip, uchar *bit_fwip, uchar
*bit_subnet)

bit formatinda verilen iki IP'nin subnet degerine gore ayni agda
olup olmadigini kontrol eder. Sonucu dondurur.

TDconfirm Sınıfı Referansı

void __fastcall **iptalClick** (TObject *Sender)

Kullanici sorulan soruya hayir cevabini verdiginde cagrilir.

void __fastcall **tamamClick** (TObject *Sender)

Kullanici sorulan soruya evet cevabini verdiginde cagrilir.

__fastcall **TDconfirm** (TComponent *Owner)

TConfirm sinifinin kurucu metodu.



TestMethod Sınıfı Referansı

int __cdecl Execute (TStrings *method, int _kaynak, int _hedef,
int *fw_pass) throw ()

Test modulunu calistirir. Sonuc "fw_pass" degiskeni ile verilir.

__fastcall TestMethod (TStrings *SgList, int netdev, bool
*_stopped)

Sinifin kurucu metodu. Ag topolojisi bilgilerini, ethernet arayuz sayisini alir.

TStrings __fastcall SetOpt (char *_method)

Metot Stringi icerisindeki Opsiyon parametresi karsisinda gecen degerleri ayristirir.

int __fastcall Send And Listen (TStrings *method, TStrings
*Options, int *fw_passh)

Paket yollama ve dinleme islemlerini baslatir.

void __fastcall Send Packets (TStrings *method, TStrings
*Options)

Metota uygun paket ya da paketleri yollar.

void get_hw_addr (char *buf, char *str)

str degiskenindeki hardware adresini bit formatinda buf degiskenine atar.

void cp_hw_addr (char *buf, char *str)

Hardware adresini str degiskeninden buf degiskenine kopyalar.

u_long SpoofIP (ulong SrcIP, ulong DstIP)

Sahte IP adresi uretir. Sonuc u_long formatindadir.

AnsiString SpoofIP (AnsiString src_ip, AnsiString dst_ip)

Sahte IP adresi uretir. Sonuc AnsiString formatindadir.

TFAbout Sınıfı Referansı

void __fastcall **iptalClick** (TObject *Sender)

Dialog kutusunu kapatir.

__fastcall **TFAbout** (TComponent *Owner)

About sinifinin kurucusu.

TFReport Sınıfı Referansı

void __fastcall **BKapatClick** (TObject *Sender)

Raporu gosteren pencereyi kapatir.

void __fastcall **BKaydetClick** (TObject *Sender)

Raporun html formatinda kaydedilebilmesi icin "save Dialog Box" i
acar.

__fastcall **TFReport** (TComponent *Owner)

Sinifin kurucu sinifi.

TFTercihler Sınıfı Referansı

void __fastcall **iptalClick** (TObject *Sender)

Degisiklikleri kaydetmeden dialog kutusunu kapatir.

void __fastcall **tamamClick** (TObject *Sender)

Degisiklikleri kaydetme cagrisinda bulunup dialog kutusunu
kapatir.

__fastcall **TFTercihler** (TComponent *Owner)

Sinifin kurucusu.

TFTopoloji Sınıfı Referansı

void __fastcall **iptalClick** (TObject *Sender)

Yapilan degisiklikler kaydedilmeden pencere kapatilir.

void __fastcall **tamamClick** (TObject *Sender)

Yapılan degisiklikler kaydedilip pencere kapatilir.

void __fastcall **FormShow** (TObject *Sender)

Penceredeki tekst alanlarinin degerlerini atar.

void __fastcall **CArayuzSelect** (TObject *Sender)

CArayuz Combo Kutusundan bir ethernet arayuzu secildiginde
diğer tekst alanlarini atar.

void __fastcall **EIPAdresChange** (TObject *Sender)

IP adresi degistiginde bu degeri tampon stringine koyar.

void __fastcall **ESegNameChange** (TObject *Sender)

ağ dilimi ismi degistiginde bu degeri tampon stringine koyar.

void __fastcall **EFwIPChange** (TObject *Sender)

Guvenlik duvarinin IP adresi degistiginde bu degeri tampon
stringine koyar.

void __fastcall **CDefaultClick** (TObject *Sender)

Ag konfigurasyonunu sistemde kullanilan degerlere degistirir.

void __fastcall **EAgMaskesiChange** (TObject *Sender)

Ag maskesi degistiginde bu degeri tampon stringine koyar.

void __fastcall **CSecLevelChange** (TObject *Sender)

Guvenlik seviyesi degistiginde bu degeri tampon stringine koyar.

__fastcall **TFTopoloji** (TComponent *Owner)

Sinifin kurucu metodu. Bazi degiskenler icin bellek ayirimi ve
deger atamaları yapar.

TFwTest Sınıfı Referansı

void __fastcall **TemizleClick** (TObject *Sender)

Metotların bulunduğu paneli temizler.

void __fastcall **CikarClick** (TObject *Sender)

Metotlardan secili olani listeden cikarir.

void __fastcall **BaslatClick** (TObject *Sender)

Test prosesini baslatir. Ag konfigurasyonunu kontrol etme, statu cubugu iceren bir diyalog kutusu acma, TCP/IP ayarlarini yapma, test modullerini calistirma cagrilarinda bulunur. Sonuclari sagdaki test sonuclari paneline yazar.

void __fastcall **Cikis1Click** (TObject *Sender)

Programdan cikma prosesini baslatir. Genel ayarlari konfigurasyon dosyasina kaydeder.

void __fastcall **AcClick** (TObject *Sender)

Daha onceden hazirlanmis test yatagi dosyasini yukleme islemini baslatir.

void __fastcall **Test Yatagi Ac** (AnsiString FileName)

Filename isimli test yatagi dosyasini acar. Hatalar varsa kullaniciyi uyarir.

void __fastcall **KaydetClick** (TObject *Sender)

Olusturulan test yatagini kaydeder.

void __fastcall **EkleClick** (TObject *Sender)

Test modul dosyalarini test yatagina ekler.

void __fastcall **FormResize** (TObject *Sender)

Ana cercevenin boyutu degistirilirken sagdaki test sonuclari panellerinin esit ebatta kalmasini saglar.

void __fastcall **AyarlaClick** (TObject *Sender)

Test modullerini ayarlama penceresindeki degerleri atayarak bu pencereyi acar.

void __fastcall **FormShow** (TObject *Sender)

NetCheck ayarlarinda son yuklenen test yatagini yukle tercihi secili ise bu dosyanin yuklenme prosesini baslatir.

void __fastcall **Topoloji1Click** (TObject *Sender)

Topoloji ayarlari dugmesine basildiginda ilgili pencerenin acilma prosesini baslatir.

void __fastcall **FormClose** (TObject *Sender, TCloseAction &Action)

Cikis1Click metodunu cagirir.

void __fastcall **RaporClick** (TObject *Sender)

Sonuclari Raporla dugmesine basildiginda rapor hazirlar.

void __fastcall **Rapor AyarlariClick** (TObject *Sender)

Raporlama ayarlarininin yapildigi pencereyi acar.

void __fastcall **Tercihler1Click** (TObject *Sender) .

NetCheck'in Ayarlarinin yapildigi pencereyi acar.

void __fastcall **TreeView1KeyDown** (TObject *Sender, WORD &Key, TShiftState Shift)

Test modulleri listesindeki bir modulun uzerinde "delete" tusuna basildiginda bu modulu siler.

void __fastcall **TreeView1ItemDoubleClick** (TObject *Sender, TTreeNode *Node)

Test modulleri listesinde bir modulun uzerine tiklandiginda bu modulun ayarlarini iceren pencereyi acar.

void __fastcall **Hakkinda1Click** (TObject *Sender)

Program hakkında bilgileri iceren pencereyi acar.

void __fastcall **ApplySetTopology** ()

Ayarlanan ag konfigurasyonunu sisteme uygular.

void __fastcall **SetDefaultTopology** ()

Hali hazirda isletim sisteminde kullanilan ag ayarlarini alir.

void __fastcall **SetTreeEntry** (int entry)



Test modulleri agacina yeni bir girdi ekler.

void __fastcall **SetMemoCaptions ()**

Test sonuclari panellerinin basliklarini ayarlar.

void __fastcall **AskforSave ()**

Test yataginin kaydedilmesi icin kullaniciya soru sorar. Cevap olumlu ise **KaydetClick()** metodunu calistirir.

bool __fastcall **CheckTestConfiguration ()**

Test prosesi calistirilmadan once ag ayarlarinin ve test modullerinin dogru ayarlanip ayarlanmadigini kontrol eder.

int __fastcall **IdofHedef (AnsiString StHedef)**

Hedef segmentin numarasini dondurur.

int __fastcall **IdofKaynak (AnsiString StKaynak)**

Kaynak segmentin numarasini dondurur.

bool __fastcall **RAyar Menu ()**

Rapor Ayarlari Monusunu hazirlar.

void __fastcall **ChangeLang (AnsiString CurrentLang)**

Konfigurasyon dosyasindaki dil secimine gore programlarin ve metod dosyalarinin dilini ayarlar.

__fastcall **TFwTest (TComponent *Owner)**

Sinifin kurucu metodu. Bazi degiskenlere baslangic degerleri verilir.

int __fastcall **Perform Imp Tests (TestMethod *tmethod, ThStart *startbox)**

Gizli testler calistirilir.

int __fastcall **SecLevelExist (TStrings *SgList, int seclev, int *SegNo)**

ag dilimleri guvenlik seviyelerine gore siniflandirir.

void __fastcall **Output Int Test (TStrings *int_method, int result,**

ThStart *startbox)

Gizli test sonuclarini test sonuclari panellerine yazar.

void __fastcall **Try To Conn** (int mcount, **TestMethod** *tmethod, AnsiString conn_type, int kaynak, int hedef)

Gizli metotlar calistirilirken rastgele bir porttan hedef ađ dilimine baglanti kurmaya calisir.

void __fastcall **Ent Output** (TStrings *amethod, int sghedef)

Test modulunun sonucunun giris ekran ciktisi.

void __fastcall **Fin Output** (TStrings *amethod, int sghedef, int res)

Test modulunun sonucunun sonuc ekran ciktisi.

void __fastcall **Report Method** (TStrings *amethod, int res, int int_count)

Test modulunun sonucunu raporlar.

void __fastcall **Report As Table** (TStrings *amethod, int res)

Test modulunun sonucunu tablo formatinda raporlar.

ThStart Sınıfı Referansı

__fastcall **ThStart** (bool CreateSuspended, bool *_stopped, bool *_sfinished)

Sınıfın kurucu metodu. Testin devam ettiğini gösteren değişken değerlerini atar.

void __fastcall **StepBy** (int stepsize)

Progress Cubugunun değerini "stepsize" boyutunca artırır.

void __fastcall **Stop** ()

StartBox'i kapatır.

TMethotConf Sınıfı Referansı

void __fastcall **iptalClick** (TObject *Sender)

Yapılan degisiklikleri kaydetmeden diyalog kutusunu kapatir.

void __fastcall **FormShow** (TObject *Sender)

Diyalog kutusu acilirken tekst alanlarini kayitli olan bilgilerle doldurur.

void __fastcall **tamamClick** (TObject *Sender)

Yapılan degisiklikleri kaydedip diyalog kutusunu kapatir.

__fastcall **TMethodConf** (TComponent *Owner)

TMethodConf sinifinin kurucusu.

TReportConf Sinifi Referansı

void __fastcall **tamamClick** (TObject *Sender)

Degisiklikleri kaydetme cagrisinda bulunup diyalog kutusunu kapatir.

__fastcall **TReportConf** (TComponent *Owner)

Sinifin kurucusu.

TStartBox Sinifi Referansı

void __fastcall **DurdurClick** (TObject *Sender)

Test prosesini durdurma dugmesine basildiginda calisir. diger siniflar tarafından erisilen "stopped" degerini "true" yapar.

__fastcall **TStartBox** ()

Sinifin kurucu metodu.

TUyari Sinifi Referansı

void __fastcall **tamamClick** (TObject *Sender)

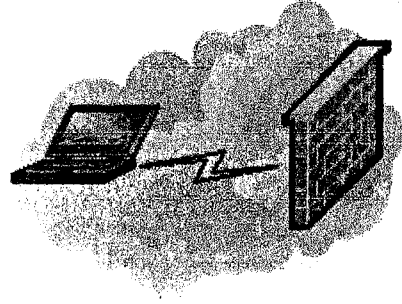
Kullanici, diyalog kutusunu kapatmak istediginde calisir.

__fastcall **TUyari** (TComponent *Owner)

TUyari sinifinin kurucu metodu.

Ek-2: NetCheck Test Sonuçları -1

Guvenlik Duvari Testi Sonuclari



Yapilan Test Sayisi: 29	Testi Yapan Firma: GYTE
Bulunan Aciklik Sayisi: 7	Musteri Firma: TRUST INF. SYSTEM
Tarih: Wed May 28 17:07:37 2003	Test Sorumlusu: Tahsin Turkoz

Topoloji Bilgileri

Lokal Ag

Guv. Duv. Arayuzunun IP'si: 10.10.1.1

Alt-ag Maskesi: 255.255.255.0

Test aracının IP'si: 10.10.1.9

Guvenlik Seviyesi: Guvenli

DMZ

Guv. Duv. Arayuzunun IP'si: 10.10.2.1

Alt-ag Maskesi: 255.255.255.0

Test aracının IP'si: 10.10.2.9

Guvenlik Seviyesi: Yari Guvenli

Internet

Guv. Duv. Arayuzunun IP'si: 10.10.3.1

Alt-ag Maskesi: 255.255.255.0

Test aracının IP'si: 10.10.3.9

Guvenlik Seviyesi: Guvensiz

Detayli Test Sonuclari

Topoloji Konum Testi

Aciklama: Guvenlik Duvarinin kurulum mimarisi incelenecektir.

Sonuc: Gecti

Guvenli ađ dilimine Erisim Testi 1

Aciklama: Guvenli ađ dilimine erisim izinleri incelenecektir.

Kaynak ađ dilimi: Internet

Hedef ađ dilimi: Lokal Ag

Sonuc: Gecti

Kurulmaya calisilan baglantilar:

Baglanti: 983(tcp) Sonuc: Kurulamadi.

Baglanti: 113(udp) Sonuc: Kurulamadi.

Baglanti: icmp Sonuc: Kurulamadi.

Guvenli ađ dilimine Erisim Testi 2

Aciklama: Guvenli ađ dilimine erisim izinleri incelenecektir.

Kaynak ađ dilimi: DMZ

Hedef ađ dilimi: Lokal Ag

Sonuc: Gecti

Kurulmaya calisilan baglantilar:

Baglanti: 885(tcp) Sonuc: Kurulamadi.

Baglanti: 746(udp) Sonuc: Kurulamadi.

Baglanti: icmp Sonuc: Kurulamadi.

Guvensiz ađ dilimine Erisim Testi 1

Aciklama: Guvensiz ađ dilimine erisim izinleri incelenecektir.

Kaynak ađ dilimi: Lokal Ag

Hedef ađ dilimi: Internet

Sonuc: Kaldi

Oneri: Guvensiz aglara dogru verilen baglanti izinleri, truva atlari virusler gibi arka planda calisan zararli programlarin guvenli ya da yari guvenli segmentlerden bilgi kacirmasina mahal verebilir. Bu sebeple sadece guvensiz segmentlerden alinan servisler icin gerekli baglantilara izni verilmeli, diđer baglantilar guvenlik duvari tarafından bloklanmalidir.

Kurulmaya calisilan baglantilar:

Baglanti: 240(tcp) Sonuc: Kuruldu.

Baglanti: 597(udp) Sonuc: Kuruldu.

Baglanti: icmp Sonuc: Kuruldu.

Guvensiz ağ dilimine Erisim Testi 2

Aciklama: Guvensiz ağ dilimine erişim izinleri incelenecektir.

Kaynak ağ dilimi: DMZ

Hedef ağ dilimi: Internet

Sonuc: Kaldi

Oneri: Guvensiz ağılara doğru verilen bağlantı izinleri, truva atları virusler gibi arka planda çalışan zararlı programların güvenli ya da yarı güvenli segmentlerden bilgi kacırmalarına mahal verebilir. Bu sebeple sadece güvenli segmentlerden alınan servisler için gerekli bağlantılara izin verilmeli, diğer bağlantılar güvenlik duvarı tarafından bloklanmalıdır.

Kurulmaya çalışılan bağlantılar:

Baglanti: 289(tcp) Sonuc: Kuruldu.

Baglanti: 693(udp) Sonuc: Kuruldu.

Baglanti: icmp Sonuc: Kuruldu.

Yarı Güvenli ağ dilimine Erisim Testi 1

Aciklama: Yarı güvenli ağ dilimine erişim izinleri incelenecektir.

Kaynak ağ dilimi: Lokal Ağ

Hedef ağ dilimi: DMZ

Sonuc: Gecti

Kurulmaya çalışılan bağlantılar:

Baglanti: 840(tcp) Sonuc: Kurulamadi.

Baglanti: 893(udp) Sonuc: Kurulamadi.

Baglanti: icmp Sonuc: Kurulamadi.

Yarı Güvenli ağ dilimine Erisim Testi 2

Aciklama: Yarı güvenli ağ dilimine erişim izinleri incelenecektir.

Kaynak ağ dilimi: Internet

Hedef ağ dilimi: DMZ

Sonuc: Gecti

Kurulmaya çalışılan bağlantılar:

Baglanti: 336(tcp) Sonuc: Kurulamadi.

Baglanti: 231(udp) Sonuc: Kurulamadi.

Baglanti: icmp Sonuc: Kurulamadi.

ICMP Echo İstek testi

Aciklama: Yari guvenli ađ dilimine guvenlik duvari uzerinden ping paketleri yollanacaktır.

Kaynak ađ dilimi: Internet

Hedef ađ dilimi: DMZ

Sonuc: Kaldi

Oneri: Sistem icin kritik sunucularin bulundugu yari guvenli ađ dilimine ping istemlerini ulasmamalidir.

WEB Sunumcusu Erisim Testi -1

Aciklama: Kurumun Web sunumcusuna 80 no'lu port disinda bir porttan tcp baglantisi kurulacaktır.

Kaynak ađ dilimi: Internet

Hedef ađ dilimi: DMZ

Sonuc: Kaldi

Oneri: Web sunumcusu sadece web hizmeti sunuyorsa 80 no'lu port disinda bir porttan erisim olmamalidir. Erisim izni verilen her servis guvenlik riskini arttirir.

WEB Sunumcusu Erisim Testi -2

Aciklama: Kurumun Web sunumcusuna 80 no'lu porttan udp baglantisi kurulacaktır.

Kaynak ađ dilimi: Internet

Hedef ađ dilimi: DMZ

Sonuc: Kaldi

Oneri: Web sunumcusu sadece web hizmeti sunuyorsa udp servisleri icin erisime izin verilmemelidir. Erisim izni verilen her servis guvenlik riskini arttirir.

DNS Sunumcusu Erisim Testi -1

Aciklama: Kurumun Web sunumcusuna 53 no'lu port disinda bir porttan udp baglantisi kurulacaktır.

Kaynak ađ dilimi: Internet

Hedef ađ dilimi: DMZ

Sonuc: Kaldi

Oneri: DNS sunumcusu sadece dns hizmeti sunuyorsa 53 no'lu port disinda bir porttan erisim olmamalidir. Erisim izni verilen her servis guvenlik riskini arttirir.

DNS Sunumcusu Erisim Testi -2

Aciklama: Kurumun Web sunumcusuna 53 no'lu porttan tcp baglantisi kurulacaktır.

Kaynak ađ dilimi: Internet

Hedef ađ dilimi: DMZ

Sonuc: Kaldi

Oneri: DNS sunumcusu sadece dns hizmeti sunuyorsa tcp servisleri icin erisime izin verilmemelidir. Eger alan transferi yapiliyorsa o zaman sadece ikincil dns sunumcuya bu izin verilmelidir. Erisim izni verilen her servis guvenlik riskini arttirir.

SYN ve ACK bayrakli paket yollanmasi

Aciklama: Bir TCP baglantisi kurulmak istendiginde gonderilecek pakette ACK bayragi etkin hale getirilir ve bu paketin guvenlik duvarindan gecip gecmedigi gozlenir.

Kaynak ađ dilimi: Internet

Hedef ađ dilimi: DMZ

Sonuc: Gecti

FIN bayrakli paket yollanmasi

Aciklama: Bir baglanti kurmak icin FIN bayragi etkin hale getirmis paket gonderilir. Bu paketin guvenlik duvarindan gecip gecmedigi gozlenir.

Kaynak ađ dilimi: Internet

Hedef ađ dilimi: DMZ

Sonuc: Gecti

Oturum Esnasinda Hatali Paket Yollama Testi

Aciklama: Oncelikle standart bir SYN paketi ile oturum acilir. Pesisira, oturumda yollanmaması gereken SYN-ACK bayrakli paket yollanir. Bu paketin guvenlik duvarindan gecip gecmedigi gozlenir.

Kaynak ađ dilimi: Internet

Hedef ađ dilimi: DMZ

Sonuc: Kaldi

Oneri: Guvenlik duvarinin baglanti tablosuna eklenmis bir TCP oturumunda, baglanti istegini baslatanin karsi taraftan cevap beklemeden baska paket ya da paketler gondermesi normal degildir. Bu tur trafikler guvenlik duvari tarafından engellenmelidir.

IP Sahtekarligi Testi

Aciklama: Kaynak IP'si degistirilen bir paketin guvenlik duvarindan gecip gecmedigi gozlenir.

Kaynak ađ dilimi: Lokal Ag

Hedef ađ dilimi: Internet

Sonuc: Gecti

Hatali Paket Saglamasi Testi

Aciklama: IP basliginda gecen paket saglamasi(checksum) yanlis olan paketlere karsi guvenlik duvarinin davranisi gozlenir.

Kaynak ađ dilimi: Lokal Ag

Hedef ađ dilimi: Internet

Sonuc: Kaldi

Oneri: Bazi saldırılar hatalı paketler göndererek yapılabilir. Bu sebeple sağlaması hatalı olan paketler güvenlik duvarı tarafından düşürülmelidir.

Aynı Kaynak ve Hedef Adresli Paket Testi.

Acıklama: Kaynak IP Adresi, hedef IP adresi ile aynı olan paketlere karşı güvenlik duvarının davranışı gözlenir.

Kaynak ağ dilimi: Internet

Hedef ağ dilimi: DMZ

Sonuc: Gecti

Gecersiz Kaynak Adresi Testi

Acıklama: Kaynak Adresi 127.0.0.1 olan paketlere karşı güvenlik duvarının davranışı gözlenir.

Kaynak ağ dilimi: Internet

Hedef ağ dilimi: DMZ

Sonuc: Kaldi

Oneri: Bazi saldırılarda gerçek dünyada geçersiz olan IP'ler kullanılarak saldırıların kaynağının bulunması engellenebilmektedir. Güvenlik duvarı geçersiz IP adresleri içeren paketleri düşürmelidir.

İzin Verilmeyen Bolunmuş Paket Testi

Acıklama: İzin verilmeyen bir bağlantı, bolunmuş tcp paketleri ile kurulmaya çalışıldığında güvenlik duvarının davranışı gözlenir.

Kaynak ağ dilimi: Internet

Hedef ağ dilimi: DMZ

Sonuc: Kaldi

Oneri: Güvenlik duvarı izin verilmeyen kural için bolunmuş tcp paketlerini düşürmelidir.

TCP Bolunmuş Paket Testi

Acıklama: İzin verilen bir bağlantı, bolunmuş tcp paketleri ile kurulmaya çalışıldığında güvenlik duvarının davranışı gözlenir.

Kaynak ağ dilimi: Internet

Hedef ağ dilimi: DMZ

Sonuc: Kaldi

Oneri: Güvenlik duvarı izin verilen kural için bolunmuş tcp paketlerini düzgün bir şekilde geçirmelidir.

UDP Bolunmuş Paket Testi

Acıklama: İzin verilen bir bağlantı, bolunmuş udp paketleri ile kurulmaya



calisildiginda guvenlik duvarinin davranisi gozlenir.

Kaynak ağ dilimi: Internet

Hedef ağ dilimi: DMZ

Sonuc: Kaldi

Oneri: Guvenlik duvari izin verilen kural icin bolunmus udp paketlerini duzgun bir sekilde gecirmelidir.

Sahte Kaynak Adresli Bolunmus Paket Testi

Aciklama: Sahte kaynak adresli bolunmus UDP paketlerine karsi guvenlik duvarinin davranisi gozlenir.

Kaynak ağ dilimi: Lokal Ag

Hedef ağ dilimi: Internet

Sonuc: Gecti

TCP Karisik Bolunmus Paket Testi

Aciklama: Karisik sirada gonderilen bolunmus TCP paketlerine karsi guvenlik duvarinin davranisi gozlenir.

Kaynak ağ dilimi: Lokal Ag

Hedef ağ dilimi: Internet

Sonuc: Kaldi

Oneri: Guvenlik duvari karisik sirada gonderilen bolunmus TCP paketlerini kotarabilmelidir.

UDP Karisik Bolunmus Paket Testi

Aciklama: Karisik sirada gonderilen bolunmus UDP paketlerine karsi guvenlik duvarinin davranisi gozlenir.

Kaynak ağ dilimi: Lokal Ag

Hedef ağ dilimi: Internet

Sonuc: Kaldi

Oneri: Guvenlik duvari karisik sirada gonderilen bolunmus UDP paketlerini kotarabilmelidir.

Boyutu Fazla Bolunmus TCP Paketi Testi

Aciklama: Maksimum bolunmus paket uzunlugu olan 65536 byte'i gecen bolunmus TCP paketlerine karsi guvenlik duvarinin davranisi gozlenir.

Kaynak ağ dilimi: Lokal Ag

Hedef ağ dilimi: Internet

Sonuc: Gecti

Bolunmus TCP Paketlerinde Veri Alanlarinin Cakismasi Testi

Aciklama: Bolunmus TCP paketlerinde giden verinin baslangic ve bitis noktalarini belirleyen ofset numaralarinin cakismasi durumunda guvenlik duvarinin davranisi gozlenir.

Kaynak ağ dilimi: Lokal Ağ

Hedef ağ dilimi: Internet

Sonuc: Gecti

Bolunmus UDP Paketlerinde Veri Alanlarinin Cakismasi Testi

Aciklama: Bolunmus UDP paketlerinde giden verinin baslangic ve bitis noktalarini belirleyen ofset numaralarinin cakismasi durumunda guvenlik duvarinin davranisi gozlenir.

Kaynak ağ dilimi: Lokal Ağ

Hedef ağ dilimi: Internet

Sonuc: Gecti

Izin Verilen Baglanti Icin Kaynak Yonlendirme Testi.

Aciklama: Yonlendirmenin kaynak tarafından yapildigi paketler izin verilen baglantiyi kurmaya calisiyorsa bu durumda guvenlik duvarinin davranisi gozlenir.

Kaynak ağ dilimi: Internet

Hedef ağ dilimi: DMZ

Sonuc: Kaldi

Oneri: Paketlerin IP basliklarinda hangi yonlendiriciler uzerinden hedefe ulasilmasi gerektiği bilgisi bulunabilmektedir. Bu ozellik IP sahtekarligi ile beraber kullanildiginda yetkisiz erisimle sonuclanabilecek saldirilara neden olabilir. Bu sebeple kaynak yonlendirme yapilan paketler guvenlik duvari tarafından bloklanmalidir.

Izin Verilen Kural Icin TCP Baglanti Seli Testi

Aciklama: Izin verilen bir servise 3500 tane tcp baglantisi acilmaya calisilir.

Guvenlik duvarinin bu baglanti isteklerine cevap verip veremedigi gozlenir.

Kaynak ağ dilimi: Internet

Hedef ağ dilimi: DMZ

Sonuc: Kaldi

Oneri: Guvenlik Duvari yogun trafikte kilitlenmemeli, duzgun calismaya devam etmelidir. Aksi durumda basit bir saldiri ile servis disi birakilabilir.

Izin Verilmeyen Kural Icin TCP Baglanti Seli Testi

Aciklama: Izin verilmeyen bir servise 3500 tane tcp baglantisi acilmaya calisilir.

Guvenlik duvarinin bu istekleri kısa usrede dusurup calismaya devam edip edemedigi gozlenir.

Kaynak ağ dilimi: Internet

Hedef ağ dilimi: DMZ

Sonuc: Gecti

Izin Verilen Kural Icin UDP Baglanti Seli Testi

Aciklama: Izin verilen bir servise 3500 tane udp baglantisi acilmaya calisilir. Guvenlik duvarinin bu baglanti isteklerine cevap verip veremedigi gozlenir.

Kaynak ağ dilimi: Internet

Hedef ağ dilimi: DMZ

Sonuc: Kaldi

Oneri: Guvenlik Duvari yogun trafikte kilitlenmemeli, duzgun calismaya devam etmelidir. Aksi durumda basit bir saldiri ile servis disi birakilabilir.

Izin Verilmeyen Kural Icin UDP Baglanti Seli Testi

Aciklama: Izin verilmeyen bir servise 3500 tane udp baglantisi acilmaya calisilir. Guvenlik duvarinin bu istekleri kısa usrede dusurup calismaya devam edip edemedigi gozlenir.

Kaynak ağ dilimi: Internet

Hedef ağ dilimi: DMZ

Sonuc: Gecti

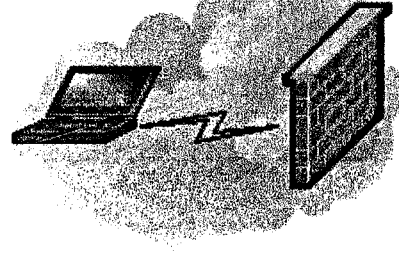
Sonuc Tablosu

Testin Adi	Kaynak ağ dilimi	Hedef ağ dilimi	Sonuc
Topoloji Konum Testi	-	-	+
Guvenli ağ dilimine Erisim Testi 1	Internet	Lokal Ag	+
Guvenli ağ dilimine Erisim Testi 2	DMZ	Lokal Ag	+
Guvensiz ağ dilimine Erisim Testi 1	Lokal Ag	Internet	-
Guvensiz ağ dilimine Erisim Testi 2	DMZ	Internet	-
Yari Guvenli ağ dilimine Erisim Testi 1	Lokal Ag	DMZ	+
Yari Guvenli ağ dilimine Erisim Testi 2	Internet	DMZ	+
ICMP Echo Istek testi	Internet	DMZ	-
WEB Sunumcusu Erisim Testi -1	Internet	DMZ	-
WEB Sunumcusu Erisim Testi -2	Internet	DMZ	-
DNS Sunumcusu Erisim Testi -1	Internet	DMZ	-
DNS Sunumcusu Erisim Testi -2	Internet	DMZ	-
SYN ve ACK bayrakli paket yollanmasi	Internet	DMZ	+
FIN bayrakli paket yollanmasi	Internet	DMZ	+
Oturum Esnasinda Hatali Paket Yollama Testi	Internet	DMZ	-
IP Sahtekarligi Testi	Lokal Ag	Internet	+

Hatalı Paket Sağlaması Testi	Lokal Ag	Internet	-
Aynı Kaynak ve Hedef Adresli Paket Testi.	Internet	DMZ	+
Gecersiz Kaynak Adresi Testi	Internet	DMZ	-
İzin Verilmeyen Bolunmuş Paket Testi	Internet	DMZ	-
TCP Bolunmuş Paket Testi	Internet	DMZ	-
UDP Bolunmuş Paket Testi	Internet	DMZ	-
Sahte Kaynak Adresli Bolunmuş Paket Testi	Lokal Ag	Internet	+
TCP Karışık Bolunmuş Paket Testi	Lokal Ag	Internet	-
UDP Karışık Bolunmuş Paket Testi	Lokal Ag	Internet	-
Boyutu Fazla Bolunmuş TCP Paketi Testi	Lokal Ag	Internet	+
Bolunmuş TCP Paketlerinde Veri Alanlarının Çakışması Testi	Lokal Ag	Internet	+
Bolunmuş UDP Paketlerinde Veri Alanlarının Çakışması Testi	Lokal Ag	Internet	+
İzin Verilen Bağlantı İçin Kaynak Yönlendirme Testi.	Internet	DMZ	-
İzin Verilen Kural İçin TCP Bağlantı Seli Testi	Internet	DMZ	-
İzin Verilmeyen Kural İçin TCP Bağlantı Seli Testi	Internet	DMZ	+
İzin Verilen Kural İçin UDP Bağlantı Seli Testi	Internet	DMZ	-
İzin Verilmeyen Kural İçin UDP Bağlantı Seli Testi	Internet	DMZ	+

Ek-3: NetCheck Test Sonuçları -2

Guvenlik Duvari Testi Sonuclari



Yapılan Test Sayısı: 29	Testi Yapan Firma: GYTE
Bulunan Aciklik Sayısı: 9	Musteri Firma: TRUST INF. SYSTEM
Tarih: Fri May 30 14:25:33 2003	Test Sorumlusu: Tahsin Turkoz

Topoloji Bilgileri

Lokal Ag

Guv. Duv. Arayuzunun IP'si: 10.10.2.1

Alt-ag Maskesi: 255.255.255.0

Test aracının IP'si: 10.10.2.9

Guvenlik Seviyesi: Guvenli

Internet

Guv. Duv. Arayuzunun IP'si: 10.10.3.1

Alt-ag Maskesi: 255.255.255.0

Test aracının IP'si: 10.10.3.9

Guvenlik Seviyesi: Guvensiz

Detayli Test Sonuclari

Topoloji Konum Testi

Aciklama: Guvenlik Duvarinin kurulum mimarisi incelenecektir.

Sonuc: Kaldi

Oneri: Guvenlik Duvari korunmasiz bir ag ile lokal bir agi ayirmak amaciyla kullanilmaktadir. Bu tur kullanimlarda guvenlik duvarinda bir segment yari korunmali olarak tanimlanmalidir vesegmente korunmasiz agdan erisilmesi gereken sunumcular yerlestirilmelidir. Korunmasiz agdanlokal aga erisim engellenmelidir.

Guvenli Segmente Erisim Testi 1

Aciklama: Guvenli segmente erisim izinleri incelenecektir.

Kaynak Segment: Internet

Hedef Segment: Lokal Ag

Sonuc: Gecti

Kurulmaya calisilan baglantilar:

Baglanti: 655(tcp) Sonuc: Kurulamadi.

Baglanti: 570(udp) Sonuc: Kurulamadi.

Baglanti: icmp Sonuc: Kurulamadi.

Guvensiz Segmente Erisim Testi 1

Aciklama: Guvensiz segmente erisim izinleri incelenecektir.

Kaynak Segment: Lokal Ag

Hedef Segment: Internet

Sonuc: Kaldi

Oneri: Guvensiz aglara dogru verilen baglanti izinleri, truva atlari virusler gibi arka planda calisan zararli programlarin guvenli ya da yari guvenli segmentlerden bilgi kacirmasina mahal verebilir. Bu sebeple sadece guvensiz segmentlerden alinan servisler icin gerekli baglantilara izni verilmeli, diger baglantilar guvenlik duvari tarafından bloklanmalidir.

Kurulmaya calisilan baglantilar:

Baglanti: 559(tcp) Sonuc: Kuruldu.

Baglanti: 543(udp) Sonuc: Kuruldu.

Baglanti: icmp Sonuc: Kuruldu.

ICMP Echo Istek testi

Aciklama: Yari guvenli segmente guvenlik duvari uzerinden ping paketleri yollanacaktır.

Kaynak Segment: Internet

Hedef Segment: Lokal Ag

Sonuc: Gecti

WEB Sunumcusu Erisim Testi -1

Aciklama: Kurumun Web sunumcusuna 80 no'lu port disinda bir porttan tcp baglantisi kurulacaktır.

Kaynak Segment: Internet
Hedef Segment: Lokal Ag
Sonuc: Gecti

WEB Sunumcusu Erisim Testi -2

Aciklama: Kurumun Web sunumcusuna 80 no'lu porttan udp baglantisi kurulacaktır.
Kaynak Segment: Internet
Hedef Segment: Lokal Ag
Sonuc: Gecti

DNS Sunumcusu Erisim Testi -1

Aciklama: Kurumun Web sunumcusuna 53 no'lu port disinda bir porttan udp baglantisi kurulacaktır.
Kaynak Segment: Internet
Hedef Segment: Lokal Ag
Sonuc: Gecti

DNS Sunumcusu Erisim Testi -2

Aciklama: Kurumun Web sunumcusuna 53 no'lu porttan tcp baglantisi kurulacaktır.
Kaynak Segment: Internet
Hedef Segment: Lokal Ag
Sonuc: Gecti

SYN ve ACK bayrakli paket yollanmasi

Aciklama: Bir TCP baglantisi kurulmak istendiginde gonderilecek pakette ACK bayragi etkin hale getirilir ve bu paketin guvenlik duvarindan gecip gecmedigi gozlenir.
Kaynak Segment: Internet
Hedef Segment: Lokal Ag
Sonuc: Gecti

FIN bayrakli paket yollanmasi

Aciklama: Bir baglanti kurmak icin FIN bayragi etkin hale getirmis paket gonderilir. Bu paketin guvenlik duvarindan gecip gecmedigi gozlenir.
Kaynak Segment: Internet
Hedef Segment: Lokal Ag
Sonuc: Gecti

Oturum Esnasinda Hatali Paket Yollama Testi

Aciklama: Oncelikle standart bir SYN paketi ile oturum acilir. Pesisira, oturumda yollanmaması gereken SYN-ACK bayrakli paket yollanir. Bu paketin guvenlik duvarindan gecip gecmedigi gozlenir.

Kaynak Segment: Internet

Hedef Segment: Lokal Ag

Sonuc: Gecti

IP Sahtekarligi Testi

Aciklama: Kaynak IP'si degistirilen bir paketin guvenlik duvarindan gecip gecmedigi gozlenir.

Kaynak Segment: Lokal Ag

Hedef Segment: Internet

Sonuc: Kaldi

Oneri: IP Sahtekarligi yetkisiz erisim icin kullanılan en yaygın saldırılardan birisidir. Bu sebeple guvenlik duvarilari IP sahtekarligini tespit edebilmelidir.

Hatali Paket Saglamasi Testi

Aciklama: IP basliginda gecen paket saglamasi(checksum) yanlis olan paketlere karsi guvenlik duvarinin davranisi gozlenir.

Kaynak Segment: Lokal Ag

Hedef Segment: Internet

Sonuc: Gecti

Ayni Kaynak ve Hedef Adresli Paket Testi.

Aciklama: Kaynak IP Adresi, hedef IP adresi ile ayni olan paketlere karsi guvenlik duvarinin davranisi gozlenir.

Kaynak Segment: Internet

Hedef Segment: Lokal Ag

Sonuc: Kaldi

Oneri: Kaynak adresi hedef ile ayni verip hedef sunumcudaki sonsuz paket dongusune sebep olan saldırılar yapılabilir. Bu saldırıyı engellemek için guvenlik duvari hangi IP araliklarin hangi segmente ait oldugunu bilmeli ve kaynak IP si kaynak segmente ait olmayan paketleri dusurmeli.

Gecersiz Kaynak Adresi Testi

Aciklama: Kaynak Adresi 127.0.0.1 olan paketlere karsi guvenlik duvarinin davranisi gozlenir.

Kaynak Segment: Internet

Hedef Segment: Lokal Ag

Sonuc: Gecti

Izin Verilmeyen Bolunmus Paket Testi

Aciklama: Izin verilmeyen bir baglanti, bolunmus tcp paketleri ile kurulmaya calisildiginda guvenlik duvarinin davranisi gozlenir.

Kaynak Segment: Internet

Hedef Segment: Lokal Ag

Sonuc: Gecti

TCP Bolunmus Paket Testi

Aciklama: Izin verilen bir baglanti, bolunmus tcp paketleri ile kurulmaya calisildiginda guvenlik duvarinin davranisi gozlenir.

Kaynak Segment: Internet

Hedef Segment: Lokal Ag

Sonuc: Gecti

UDP Bolunmus Paket Testi

Aciklama: Izin verilen bir baglanti, bolunmus udp paketleri ile kurulmaya calisildiginda guvenlik duvarinin davranisi gozlenir.

Kaynak Segment: Internet

Hedef Segment: Lokal Ag

Sonuc: Gecti

Sahte Kaynak Adresli Bolunmus Paket Testi

Aciklama: Sahte kaynak adresli bolunmus UDP paketlerine karsi guvenlik duvarinin davranisi gozlenir.

Kaynak Segment: Lokal Ag

Hedef Segment: Internet

Sonuc: Kaldi

Oneri: Guvenlik duvari bolunmus paketleri direk gecirmemeli bu paketlerde IP sahteciligi yapilip yapilmadigini kontrol etmelidir.

TCP Karisik Bolunmus Paket Testi

Aciklama: Karisik sirada gonderilen bolunmus TCP paketlerine karsi guvenlik duvarinin davranisi gozlenir.

Kaynak Segment: Lokal Ag

Hedef Segment: Internet

Sonuc: Gecti

UDP Karisik Bolunmus Paket Testi

Aciklama: Karisik sirada gonderilen bolunmus UDP paketlerine karsi guvenlik duvarinin davranisi gozlenir.
Kaynak Segment: Lokal Ag
Hedef Segment: Internet
Sonuc: Gecti



Boyutu Fazla Bolunmus TCP Paketi Testi

Aciklama: Maksimum bolunmus paket uzunlugu olan 65536 byte'i gecen bolunmus TCP paketlerine karsi guvenlik duvarinin davranisi gozlenir.
Kaynak Segment: Lokal Ag
Hedef Segment: Internet
Sonuc: Gecti

Bolunmus TCP Paketlerinde Veri Alanlarinin Cakismasi Testi

Aciklama: Bolunmus TCP paketlerinde giden verinin baslangic ve bitis noktalarini belirleyen ofset numaralarinin cakismasi durumunda guvenlik duvarinin davranisi gozlenir.
Kaynak Segment: Lokal Ag
Hedef Segment: Internet
Sonuc: Kaldi
Oneri: Guvenlik duvari gonderilen bolunmus TCP paketlerinin ofsetlerini kontrol etmeli, hatali ise ilgili paketleri dusurmeli.

Bolunmus UDP Paketlerinde Veri Alanlarinin Cakismasi Testi

Aciklama: Bolunmus UDP paketlerinde giden verinin baslangic ve bitis noktalarini belirleyen ofset numaralarinin cakismasi durumunda guvenlik duvarinin davranisi gozlenir.
Kaynak Segment: Lokal Ag
Hedef Segment: Internet
Sonuc: Kaldi
Oneri: Guvenlik duvari gonderilen Bolunmus UDP paketlerinin ofsetlerini kontrol etmeli, hatali ise ilgili paketleri dusurmeli.

Izin Verilen Baglanti Icin Kaynak Yonlendirme Testi

Aciklama: Yonlendirmenin kaynak tarafından yapildigi paketler izin verilen baglantiyi kurmaya calisiyorsa bu durumda guvenlik duvarinin davranisi gozlenir.
Kaynak Segment: Internet
Hedef Segment: Lokal Ag
Sonuc: Gecti

Izin Verilen Kural Icin TCP Baglanti Seli Testi

Aciklama: Izin verilen bir servise 3500 tane tcp baglantisi acilmaya calisilir. Guvenlik duvarinin bu baglanti isteklerine cevap verip veremedigi gozlenir.

Kaynak Segment: Internet

Hedef Segment: Lokal Ag

Sonuc: Geçti

Izin Verilmeyen Kural Icin TCP Baglanti Seli Testi

Aciklama: Izin verilmeyen bir servise 3500 tane tcp baglantisi acilmaya calisilir. Guvenlik duvarinin bu istekleri kısa usrede dusurup calismaya devam edip edemedigi gozlenir.

Kaynak Segment: Internet

Hedef Segment: Lokal Ag

Sonuc: Gecti

Izin Verilen Kural Icin UDP Baglanti Seli Testi

Aciklama: Izin verilen bir servise 3500 tane udp baglantisi acilmaya calisilir. Guvenlik duvarinin bu baglanti isteklerine cevap verip veremedigi gozlenir.

Kaynak Segment: Internet

Hedef Segment: Lokal Ag

Sonuc: Gecti

Izin Verilmeyen Kural Icin UDP Baglanti Seli Testi

Aciklama: Izin verilmeyen bir servise 3500 tane udp baglantisi acilmaya calisilir. Guvenlik duvarinin bu istekleri kısa usrede dusurup calismaya devam edip edemedigi gozlenir.

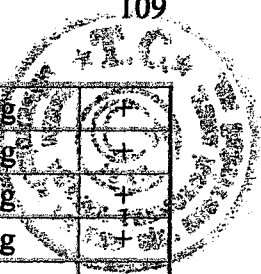
Kaynak Segment: Internet

Hedef Segment: Lokal Ag

Sonuc: Gecti

Sonuc Tablosu

Testin Adi	Kaynak Segment	Hedef Segment	Sonuc
Topoloji Konum Testi	-	-	-
Guvenli Segmente Erisim Testi 1	Internet	Lokal Ag	+
Guvensiz Segmente Erisim Testi 1	Lokal Ag	Internet	-
ICMP Echo Istek testi	Internet	Lokal Ag	+
WEB Sunumcusu Erisim Testi -1	Internet	Lokal Ag	+
WEB Sunumcusu Erisim Testi -2	Internet	Lokal Ag	+



DNS Sunumcusu Erisim Testi -1	Internet	Lokal Ag	+
DNS Sunumcusu Erisim Testi -2	Internet	Lokal Ag	+
SYN ve ACK bayrakli paket yollanmasi	Internet	Lokal Ag	+
FIN bayrakli paket yollanmasi	Internet	Lokal Ag	+
Oturum Esnasinda Hatali Paket Yollama Testi	Internet	Lokal Ag	+
IP Sahtekarligi Testi	Lokal Ag	Internet	-
Hatali Paket Saglamasi Testi	Lokal Ag	Internet	+
Ayni Kaynak ve Hedef Adresli Paket Testi.	Internet	Lokal Ag	-
Gecersiz Kaynak Adresi Testi	Internet	Lokal Ag	+
Izin Verilmeyen Bolunmus Paket Testi	Internet	Lokal Ag	+
TCP Bolunmus Paket Testi	Internet	Lokal Ag	+
UDP Bolunmus Paket Testi	Internet	Lokal Ag	+
Sahte Kaynak Adresli Bolunmus Paket Testi	Lokal Ag	Internet	-
TCP Karisik Bolunmus Paket Testi	Lokal Ag	Internet	+
UDP Karisik Bolunmus Paket Testi	Lokal Ag	Internet	+
Boyutu Fazla Bolunmus TCP Paketi Testi	Lokal Ag	Internet	+
Bolunmus TCP Paketlerinde Veri Alanlarinin Cakismasi Testi	Lokal Ag	Internet	-
Bolunmus UDP Paketlerinde Veri Alanlarinin Cakismasi Testi	Lokal Ag	Internet	-
Izin Verilen Baglanti Icin Kaynak Yonlendirme Testi	Internet	Lokal Ag	+
Izin Verilen Kural Icin TCP Baglanti Seli Testi	Internet	Lokal Ag	+
Izin Verilmeyen Kural Icin TCP Baglanti Seli Testi	Internet	Lokal Ag	+
Izin Verilen Kural Icin UDP Baglanti Seli Testi	Internet	Lokal Ag	+
Izin Verilmeyen Kural Icin UDP Baglanti Seli Testi	Internet	Lokal Ag	+