

T.C.
Bahçeşehir Üniversitesi

**MOBİL CİHAZLAR YARDIMIYLA DAĞITIK
GÜVENLİK DUVARI YÖNETİMİ**

Yüksek Lisans Tezi

Yafes Murat DUYGULUTUNA

İstanbul, 2010

T.C.
Bahçeşehir Üniversitesi
Fen Bilimleri Enstitüsü
Bilgi Teknolojileri

MOBİL CİHAZLAR YARDIMIYLA DAĞITIK GÜVENLİK DUVARI YÖNETİMİ

Yüksek Lisans Tezi

Yafes Murat DUYGULUTUNA

Danışman: Yrd. Doç. Dr. Yalçın ÇEKİÇ

İstanbul, 2010

T.C.
BAHÇEŞEHİR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
Bilgi Teknolojileri

Tezin Başlığı : MOBİL CİHAZLAR YARDIMIYLA
DAĞITIK GÜVENLİK DUVARI YÖNETİMİ
Öğrencinin Adı Soyadı : Yafes Murat DUYGULUTUNA
Tez Savunma Tarihi : 19.Mart.2010

Bu yüksek lisans tezi Fen Bilimleri Enstitüsü tarafından onaylamıştır.

Yrd. Doç. Dr. F. Tunç BOZBURA
Enstitü Müdürü

Bu tez tarafımızca okunmuş, nitelik ve içerik açısından bir Yüksek Lisans tezi olarak yeterli görülmüş ve kabul edilmiştir.

Tez Sınav Jürisi Üyeleri :

Yrd. Doç. Dr. Yalçın Çekiç :

Doç. Dr. Adem KARAHOCA :

Yrd. Doç. Dr. Mehmet Alper TUNGA :

ÖZET

MOBİL CİHAZLAR YARDIMIYLA DAĞITIK GÜVENLİK DUVARI YÖNETİMİ

Duygulutuna, Yafes Murat

Bilgi Teknolojileri
Tez Danışmanı: Yrd. Doç. Dr. Yalçın Çekiç

Mart, 2010, 75

Hemen hemen günümüzden on yıl önce güvenlik duvarları, orta ve büyük ölçekli firmalarda genellikle İnternet ile iç ağı birbirinden ayırmak için kullanılır, İnternet'ten gelebilecek tehlikelere karşı bir önlem için konumlandırılırdı. Günümüzde ise, bu durum tamamiyle değişerek, orta ölçekli bir şirkette bile iç ağ alanlarını korumak için sayıları onları geçmeye başladı. Hatta, kullanıcı bilgisayarlarına ait güvenlik duvarları da dahil olduğunda, bu sayı yüzlere ulaştı, geçti. Eğer durumu bir de büyük ölçekli şirketler açısından ele alacak olursak, durumun daha da karmaşık hale geldiğini görebiliriz. Bu büyük şirketler için güvenlik duvarlarının yüzlercesinin ağ koruma amaçlı ve binlercesinin de kullanıcı bilgisayarlarını korumak amaçlı kullanıldığını söyleyebiliriz.

Sayıları yüzleri, binleri bulan bu güvenlik duvarı tarlasını yönetebilmek, kural eklemek, silmek gün geçtikçe zorlaşmakta, iç ağa bulaşan bir virüsün yayılmak için kullandığı bir kapıyı kapatmak için ise, zamanında müdahale etmeyi neredeyse imkansız kılmaktadır. Özellikle güvenlik duvarı tarlasını yöneten sistem yöneticilerinin, her an güvenlik sistemlerinin başında olmamaları ya da haftasonları karşılaşacakları bir güvenlik durumunda müdahale etmeleri oldukça güçleşmiştir.

Günümüzde, hayatımızın her aşamasında kullanmaya başladığımız mobil teknolojilerin imkanlarını bu noktada, dağıtık güvenlik tarlalarını yönetmek için kullanmak, en hızlı ve aynı zamanda güvenli bir çözüm olarak karşımızda durmaktadır.

Bu tez ile, birbirinden bağımsız güvenlik duvarlarının mobil cihazlar üzerinden her yerden ve kolayca yönetilmesini sağlamak amaçlanmıştır. Bunun için yapılan çalışmalar ile başarılı bir sonuca ulaşılmıştır.

Bu çalışma, güvenlik duvarı yönetilecek sistem üzerine bir *istemci* uygulamasının yazılması, tüm isteklerin yönetildiği bir *ağ geçidi* uygulaması ve mobil cihazlar tarafından yönetilebilmesi için bir mobil uygulamadan ibarettir.

ABSTRACT

DISTRIBUTED FIREWALL MANAGEMENT VIA MOBILE DEVICE

Duygulutuna, Yafes Murat

Computer Science

Supervisor: Assist. Prof. Dr. Yalçın Çekiç

March, 2010, 75

Almost ten years ago, mid-size and large companies were using a firewall especially for guarding internal network from the Internet.

Nowadays, techniques for securing the internal network has enormously changed both for mid-size and large companies.

For the case of mid-size companies, tens of firewalls are used for securing the internal zones and employees' computers from the Internet and internal threats. On the other hand, for the case of large companies, thousands of firewalls are used.

Management of thousands of firewalls simultaneously, may easily turn out to be a hard work and a real trouble for security/network personnel. For instance, a virus infection spread over the network via a specific port; means vast work loads consist of addition of a rule for that port throughout the firewalls and client systems at the same time.

Instant management of networks became possible with the introduce of mobile technology in recent years. Which allows us to access any system, at anytime easily and reliability. So why not use a mobile device such as an ordinary Java enabled mobile phone for this task.

The purpose of this work, is establishing a distributed firewall management system with the help of a mobile device.

For accomplishing this goal, a client application on the firewall node, a network gateway application on the mediation device, and a mobile application residing in the mobile device is developed.

İÇİNDEKİLER

ÖZET.....	iii
ABSTRACT.....	iv
İÇİNDEKİLER	v
ŞEKİLLER.....	vi
KISALTMALAR	vii
İNGİLİZCE TERİMLER.....	viii
1. GİRİŞ.....	1
2. GENEL BİLGİLER.....	2
2.1. GÜVENLİK DUVARLARI.....	2
2.2. JAVA.....	3
2.2.1. JAVA NEDİR?	3
2.3. JAVA’NIN UYGULAMA GELİŞTİRME DİYAGRAMI.....	3
2.4. MOBİL CİHAZ / TERMİNAL	6
3. UYGULAMA	8
3.1. UYGULAMA ALTYAPISI.....	8
3.2. UYGULAMA DİLİNİN KULLANILDIĞI PLATFORM ve KODLAMA DETAYLARI.....	13
3.2.1. Microsoft Windows 7 Güvenlik Duvarı.....	15
3.2.2. Microsoft Windows XP Güvenlik Duvarı	15
3.2.3. Linux/Unix ve Türevlerinde Kullanılan Güvenlik Duvarı.....	16
3.3. KODLAMAYLA İLGİLİ ÖNEMLİ NOKTALAR.....	17
3.4. KARŞILAŞTIRMA.....	18
4. ÇALIŞMA YÖNTEMİ, TEST PLATFORMLARI ve ÇALIŞMA EKLANLARI.....	19
5. TARTIŞMA.....	27
6. SONUÇ.....	28
KAYNAKÇA.....	29
EKLER.....	30

ŞEKİLLER

Şekil 2.1 : Güvenlik Duvarı	2
Şekil 2.3 : Java Uygulama Geliştirme Aşamaları	4
Şekil 2.4.1 : Mobil Cihazlar	6
Şekil 2.4.2 : Mobil Cihaz	7
Şekil 3.1.1 : Uygulama Altyapısı	8
Şekil 3.1.2 : Mobil İstemci Uygulaması Akış Diagramı	10
Şekil 3.1.3 : Uygulama Ana Ekranı	11
Şekil 3.1.4 : Uygulama Kapı İşlemleri Ekranı	11
Şekil 3.1.5 : Sunucu Aktif Kapılar Listesi Ekranı	12
Şekil 3.2 : Netbeans 6.8	14
Şekil 4.1 : Ağ Geçidi Sunucusu Ekranı	20
Şekil 4.2 : Glassfish Uygulama Sunucusu üzerinde yüklü Ağ Geçidi ve Düğüm uygulamaları	21
Şekil 4.3 : Düğüm sunucu ekranı	22
Şekil 4.4 : Düğüm üzerine yüklenmiş uygulama ekranı	23
Şekil 4.5 : Düğüm sunucuna yapılan bir telnet bağlantı denemesi ekranı	23
Şekil 4.6 : Düğüm'ün 139 nolu kapısına erişim ekranı	24
Şekil 4.7 : Düğüm aktif kapı listesi ekranı	24
Şekil 4.8 : Düğüm'ün 139 nolu kapısının kapatılması ekranı	25
Şekil 4.9 : telnet ile Düğüm kapısına erişmeye çalışma ekranı	25
Şekil 4.10 : Kapıtılan Düğüm kapısına erişilememesi ekranı	26

KISALTMALAR

Internet Protocol	: IP
Windows IP Firewall	: WIPFW
Java Virtual Machine	: JVM
RAM	: Random Access Memory
Personal Digital Assistant	: PDA

İNGİLİZCE TERİMLER

Kapı	: Port
Ağ Geçidi	: Gateway
İstemci	: Client
Mobil İstemci	: Mobile Client
Sanal Makine	: Virtual Machine
Linux/Unix için kural yönetim programı	: iptables
Java	: Uygulama geliştirme derleyicisi
Microsoft'un mobil işletim sistemi	: Microsoft Windows Mobile
İki sistem arasında iletişim için kullanılan bir protokol (21 nolu kapı üzerinde çalışır)	: telnet

1. GİRİŞ

Mobil cihazlar, günümüz ve geleceğin iletişim ihtiyacını yer bağımsız hale getirmiştir. Aynı şekilde bu dinamizm ile birlikte bir çok ihtiyaç da mobil cihazlar sayesinde kolayca gerçekleştirilebilmektedir.

Bu tez ile; Microsoft Windows Mobile, Symbian ve Linux bazlı mobil işletim sistemli cihazlar üzerinden, Microsoft Windows, Linux, ve türevlerinde güvenlik duvarı çözümlerini yönetmek (güvenlik duvarı kuralı eklemek, silmek, açık kapıları görmek), bunu aynı zamanda dağıtık olarak da yapmak amaçlanmaktadır.

Basit olarak güvenlik duvarı, internet protokol (IP) adresi almış herhangi bir sistemin, kapılarının (port) açık ya da kapalı olması için kurallar konulan bir uygulamadır. Örneğin bu uygulama, istenilen kapıları kapatarak sistem dışından erişimleri engelleyebilir. IP adresi, herhangi bir sistemin internet dünyasında diğer sistemlerle konuşması için kullanılan bir kimlik numarası olarak düşünülebilir. Mobil cihaz, radyo frekans ile iletişimini sağlayan taşınabilir cihazdır.

Yaptığım araştırmalarda, mobil cihazlar ile dağıtık güvenlik duvarı yönetimi konusunda daha önce yapılmış herhangi bir çalışmaya rastlamadım. Mobil olmayan ve tek sistemi farklı yöntemler kullanarak yöneten çalışmalara rastladım fakat gerek çalışma ve gerekse esneklik anlamında yapılan bu çalışmayla pek ilgileri olmadığını söyleyebilirim.

Mobil cihazlar ile dağıtık güvenlik duvarı yönetimi için kullanılan altyapı ve uygulama yöntemleri şu şekildedir. Uygulama dili olarak her ortamda çalışabilmesi sebebiyle Java seçilmiştir. İşlemlerin hızlı ve esnek olabilmesi için de Web Servisi olarak fonksiyonlar kullanılmıştır. Güvenlik duvarı yönetilecek sistemler, Microsoft Windows, Linux, Solaris, vb işletim sistemleri olabilir. Uygulama altyapısında ağ geçidi görevi görecek sistem, hızlı ve güvenilir olması için Linux Ubuntu Server seçilmiştir.

Devam eden bölümler içerisinde, daha önce tez olarak yapılan ve benzer kabul edilebilecek bir tez ile karşılaştırılması, güvenlik duvarları hakkında bilgi, uygulama altyapısında kullanılan dil ve yöntemler konusunda bilgi, uygulama altyapısının dizayn detayları, kodlamayla ilgili bilgiler, yapılan testler ve sonuçlarını bulacaksınız.

2. GENEL BİLGİLER

Bu bölümde, Güvenlik Duvarının ne olduğunu, Java hakkında bilgi ve uygulama geliştirme aşamalarının neler olduğunu, son olarak da Mobil Terminal hakkında bilgileri bulabilirsiniz.

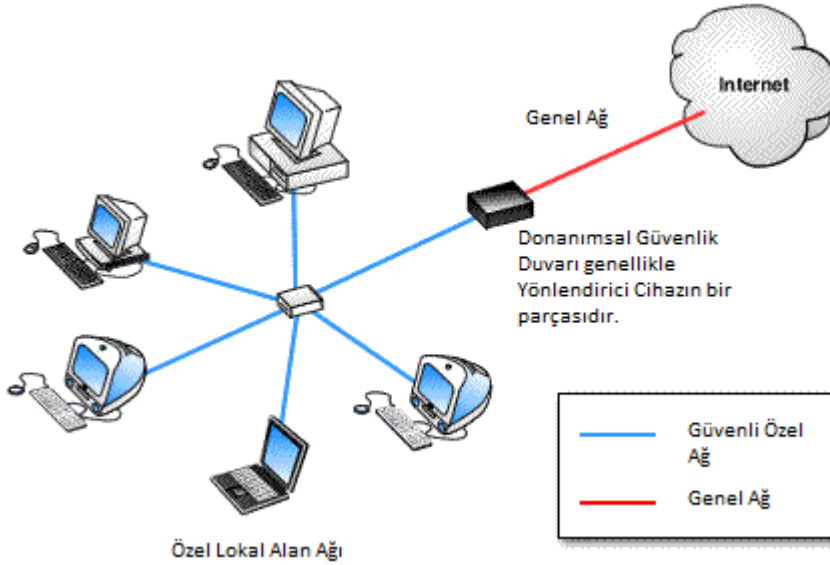
2.1.GÜVENLİK DUVARLARI

Güvenlik Duvarı nedir?

İnternet güvenlik duvarı, İnternet üzerinden bilgisayarınıza erişmeye çalışan bilgisayar korsanlarını, virüsleri ve solucanları engellemeye yardımcı olan yazılım kodu parçası veya donanımdır. Ev kullanıcısı veya küçük işletme kullanıcısıysanız, bilgisayarınızı korumaya yardımcı olmak için uygulayabileceğiniz en etkin ve en önemli adım güvenlik duvarı yüklemektir. Güvenlik duvarının ve virüsten koruma yazılımının İnternet'e bağlanmadan *önce* açılmış olması önemlidir.

Kaynak: Microsoft, 2010,

<http://www.microsoft.com/turkiye/athome/security/protect/firewall.msp>



Şekil 2.1 : Güvenlik Duvarı

Kaynak: GoModem.net, 2010,

http://www.randommart.com/images/firewall_1_images/firewall.diagram2.gif

2.2. JAVA

Yalın olarak ifade etmek gerekirse, Java, yeni nesil nesne tabanlı uygulama geliştirme imkanı sunan ve yazılan uygulamanın farklı platformlarda çalışmasını sağlayan bir derleme aracı, programlama dilidir.

2.2.1. JAVA NEDİR?

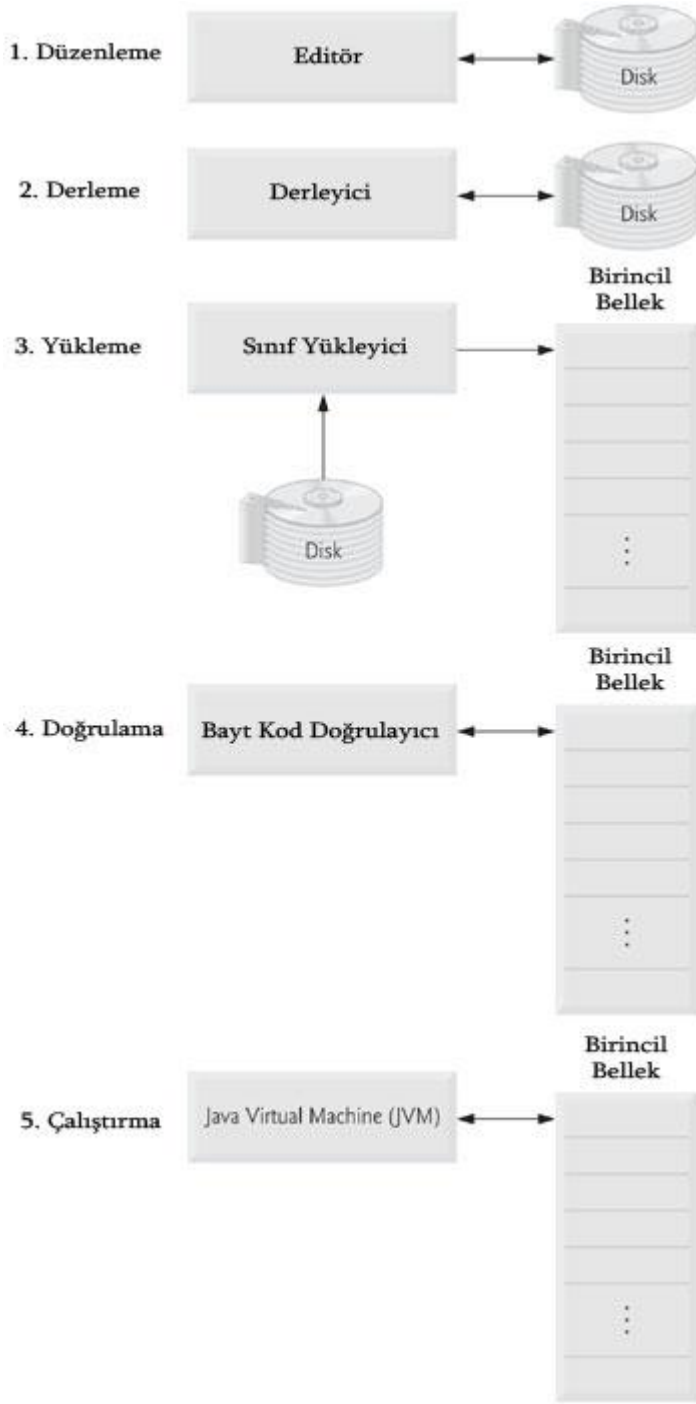
Java Sun Microsystems firması tarafından geliştirilmiş olan üst seviye bir programlama dilidir. İlk adı OAK olan Java aslında bilgisayarlar değil, set üstü araçlar (WebTV gibi) veya avuç içi cihazlar (PDA'ler gibi) için hazırlanmıştı. OAK 1995 yılında başarısız olunca Sun firması bu programlama dilinin adını Java olarak değiştirdi ve Internet üzerinde popüler bir programlama dili haline getirmeye çalıştı. Java, C++ dili gibi nesne yönelimli bir programlama dilidir ancak bu dile göre çok daha sadeleştirilmiştir. Soyadı .java olan Java programlarının çalıştırılabilir hale getirildiğinde soyadları .class olur ve Java desteği olan herhangi bir web tarayıcısı ile çalıştırılabilirler. Java programları çalıştırıldığı işletim sisteminden hemen hemen bağımsızdır çünkü bu programlar için gerekli destek hemen hemen tüm platformlarda (UNIX, Macintosh OS ve Windows gibi) verilmektedir.

Kaynak: Bilişim Terimleri, 2010,
http://www.bilisimterimleri.com/bilgisayar_bilgisi/bilgi/18.html

2.3. JAVA’NIN UYGULAMA GELİŞTİRME DİYAGRAMI

Java, kendinden önceki dillerden farklı olarak, her platformda çalıştırılabilecek uygulamalar yapmak üzere hazırlanmıştı. Bunun yapabilmek için de platform ile uygulama arasında köprü olacak **Virtual Machine**, yani sanal makine katmanını geliştirdi. Bu sayede, platform, yani örneğin bilgisayarın işletim sistemi ne olursa olsun, bu işletim sisteminin üzerinde çalıştırılması istenen uygulamalar aslında bu sanal makine üzerinde çalıştırılacaktı. Dolayısıyla platformdan bağımsız bir programlama dili oluşturulacaktı.

Bu rüya tam anlamıyla sağlanamamış olsa da Java uygulamalarının çalışma prensibi bu şekilde işliyor. Aşağıdaki diyagramda Java uygulamalarının geliştirilme aşamalarını görebiliriz.



Şekil 2.3 : Java Uygulama Geliştirme Aşamaları

Bu safhaları kısaca açıklamak gerekirse;

1. Safha: Düzenleme

Program, bir editörde oluşturulur ve diskte .java uzantılı bir dosya olarak kaydedilir.

2. Safha: Derleme

Derleyici bayt kodlarını (bytecodes) oluşturur ve bunları diskte .class uzantılı bir dosyada saklı tutar.

3. Safha: Yükleme

Sınıf yükleyici diskten içinde bayt kodları bulunan .class dosyalarını okur ve bu bayt kodlarını belleğe (RAM ya da birinil bellek olarak ayarlanmış bellek hangisiyse) koyar.

4. Safha: Doğrulama

Bayt kod doğrulayıcı, bütün bayt kodların geçerli olduğunu ve bunların Java güvenlik kısıtlamalarını ihlal etmediğini onaylar

5. Safha: Çalıştırma

Programı çalıştırmak için Java Sanal Makinesi (Java Virtual Machine - JVM), bayt kodlarını okur ve hemen o anda (Just in time -JIT) çalıştırılan bilgisayarın anlayacağı dile çevirir yani uygun dile derler. Program çalıştırdığından, birincil belleğe veri kaydedebilir.

Kaynak: Hosaf.org, 2010, <http://hosaf.org/yazi/javanin-uygulama-gelistirme-diyagrami>

2.4. MOBİL CİHAZ / TERMİNAL

Mobil cihazlar, günümüz ve geleceğin iletişim ihtiyacını bir ağı kablosuz olarak radyo frekans üzerinden dahil olarak, birbirine bağlı kablosuz iletişim ağlarının yaygın olduğu her yerde kullanılabilen taşınabilir cihazlardır.



Şekil 2.4.1 : Mobil Cihazlar

Kaynak: AdamSelwood, 2009,
http://farm4.static.flickr.com/3105/3158195615_3263e1cc6f.jpg



Şekil 2.4.2 : Mobil Cihaz

Kaynak: Somewhat Frankö, 2009,
<http://www.flickr.com/photos/somewhatfrank/3596303489/>

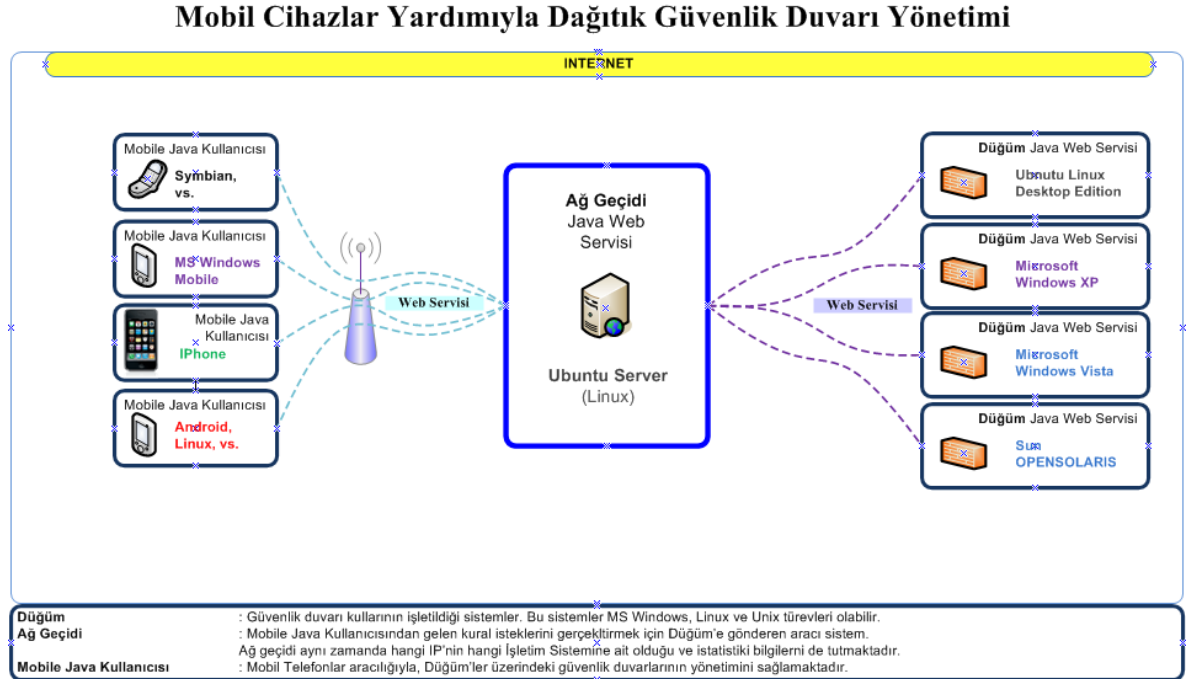
3. UYGULAMA

Mobil Cihazlar Yardımıyla Güvenlik Duvarı Yönetimi için geliştirilen uygulamalar, platform bağımsız olarak yazılmış olduğundan, üzerinde Java bulunan hemen her mobil cihazda çalışabilirler.

3.1. UYGULAMA ALTYAPISI

Uygulama altyapısı tasarlanırken, her ortamda çalışabilmesi için Java seçildi. Yönetilecek güvenlik duvarı düğümü ile, ağ geçidi görevi gören sunucunun gerekli fonksiyonları kolayca yerine getirebilmesi için, web servisi olarak tasarlandı. Kullanıcı tarafı ise her tür telefonda çalışabilmesi için Mobil Java uygulaması olarak geliştirildi.

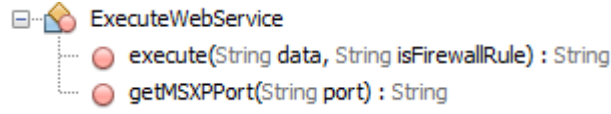
Aşağıda uygulama altyapısına ait diagramı bulabilirsiniz:



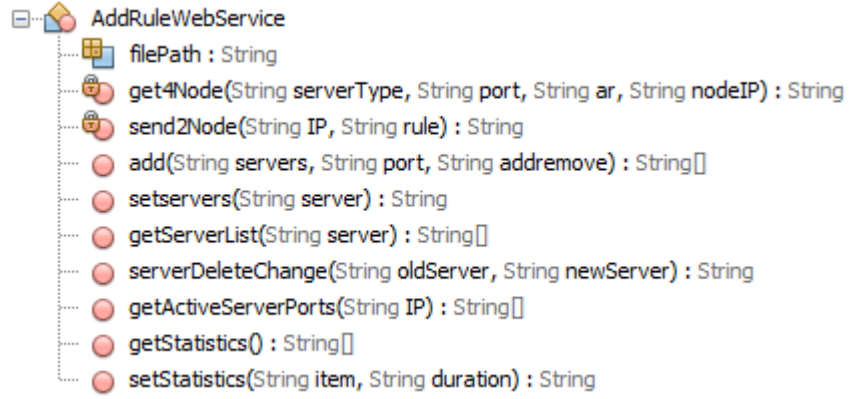
Şekil 3.1.1 : Uygulama Altyapısı

Uygulamaların web servisleri ise:

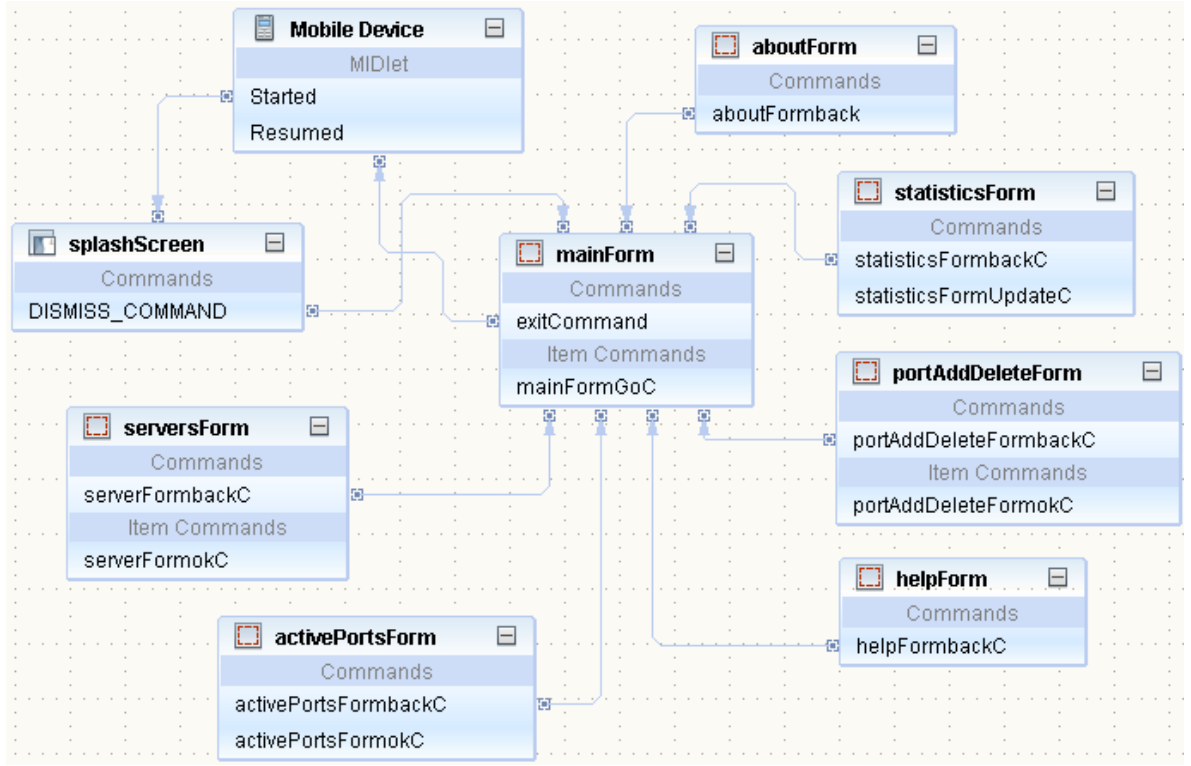
Düğüm Web Servisleri:



Ağ Geçidi Web Servisler:



Mobil Kullanıcı Akış Diagramı:



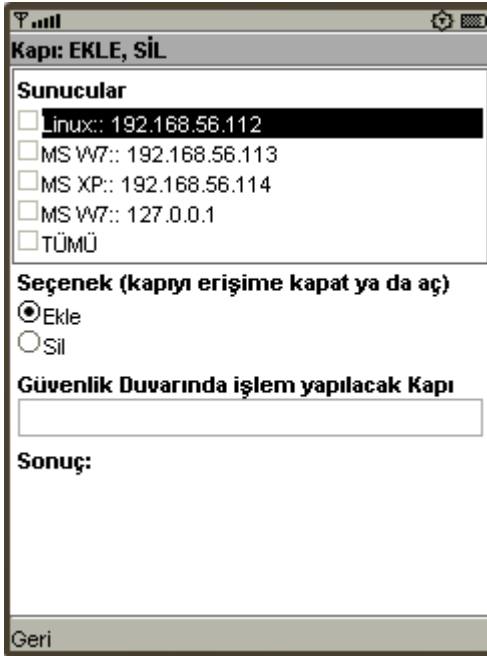
Şekil 3.1.2 : Mobil İstemci Uygulaması Akış Diagramı

Mobil Kullanıcı Ekranlarından bazıları: Ana ekran:



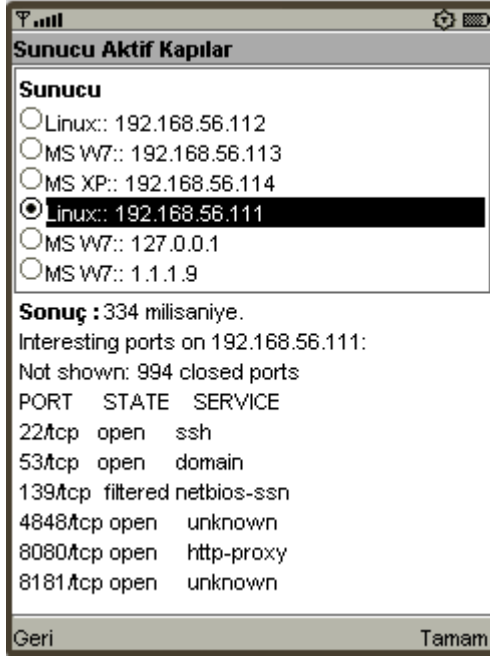
Şekil 3.1.3 : Uygulama Ana Ekranı

Sunucu listesi ve kapı bloklama ekranı:



Şekil 3.1.4 : Uygulama Kapı İşlemleri Ekranı

Port Listesi Ekranı:



Şekil 3.1.5 : Sunucu Aktif Kapılar Listesi Ekranı

3.2. UYGULAMA DİLİNİN KULLANILDIĞI PLATFORM ve KODLAMA DETAYLARI

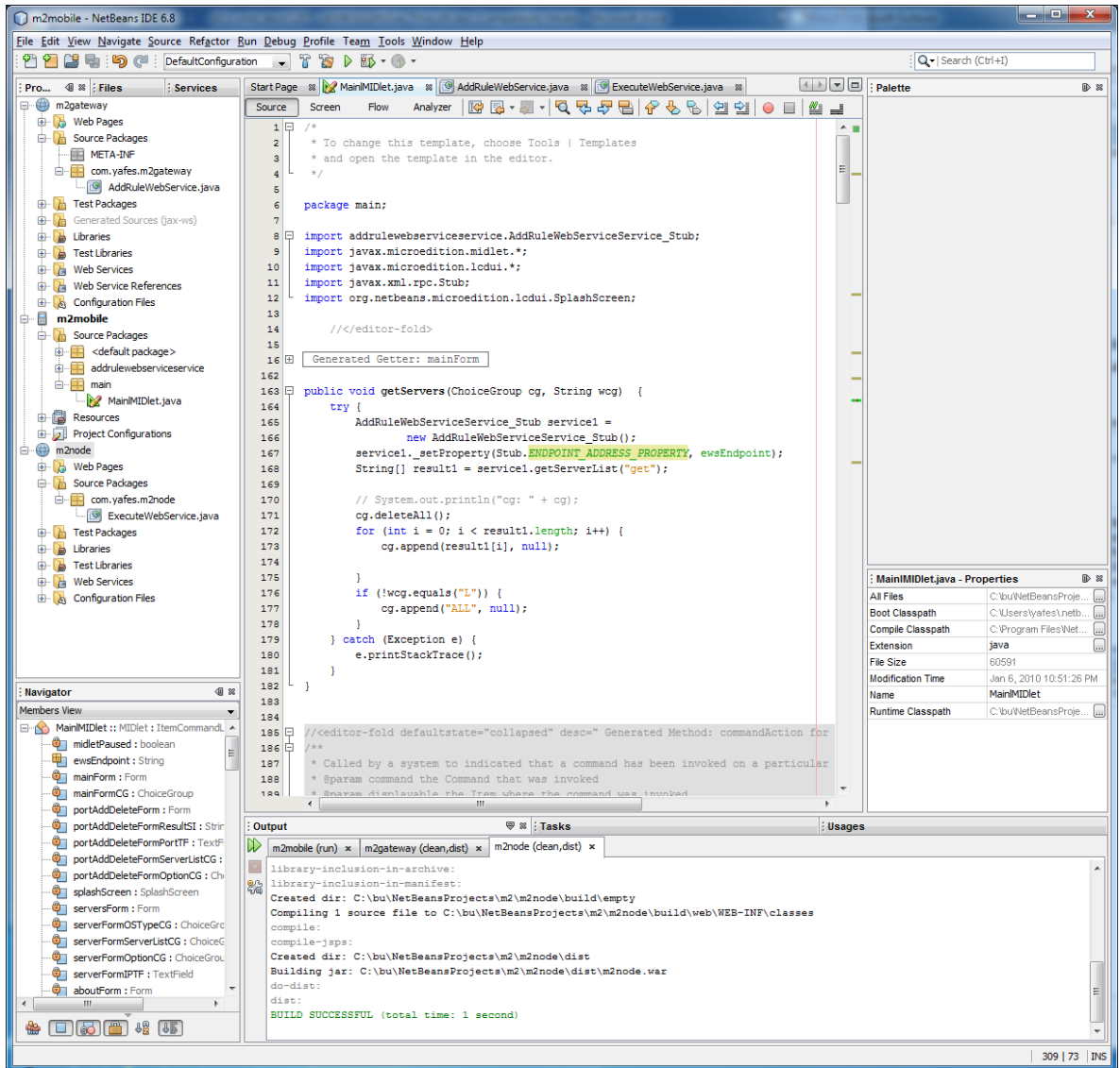
Java ile kodlar geliştirilirken Netbeans platformu kullanılmıştır (Şekil 3.2 : Netbeans 6.8). Kodlar nesne tabanlı uygulama mimarisine uygun olacak şekilde yazılmış olup, değişken isimlerinde mümkün olduğunca anlaşılır ve kullanım alışlanlıklarına uygun şekilde geliştirilmeye çalışılmıştır.

Ağ Geçidi sunucusu ile, Düğüm'ler üzerindeki istekler için web servisleri yöntemi seçilmiş ve kodlama bu şekilde mümkün olduğunca düzenli ve anlaşılır hale getirilmiştir.

Java'nın 1.6.x.x sürümü kullanılmıştır.
Uygulama sunucusu olarak da Glassfish kullanılmıştır.

Microsoft Windows 7 üzerinde Microsoft'a ait aktif güvenlik duvarı kullanılmıştır.
Microsoft Windows XP üzerinde ücretsiz yazılım olan WIPFW uygulaması kullanılmıştır.
Linux işletim sistemlerinde de iptables kullanılmıştır. Aynı zamanda iptables ile Unix ve Unix ailesinden gelen sistemlerin de güvenlik duvarlarını yönetmek mümkündür.

Netbeans 6.8



Şekil 3.2 : Netbeans 6.8

3.2.1. Microsoft Windows 7 Güvenlik Duvarı

Microsoft Windows Vista ile Microsoft İşletim Sistemi Ailesine katılan güvenlik duvarı Windows 7 üzerinden biraz daha geliştirilmiştir.

Bir kapının kapatılması için aşağıdaki komut kullanılabilir:

```
"netsh advfirewall firewall add rule name=blockKAPINO protocol=TCP dir=in  
localport=KAPINO action=block"
```

Kapalı bir kapının açılması için ise:

```
"netsh advfirewall firewall delete rule name=blockKAPINO"
```

komutunu kullanmak yeterli olacaktır.

KAPINO (Port) : Anlaşılması kolay olması amacıyla bu şekilde ifade edilmiştir.

Kaynak: Microsoft, 2010,

<http://windows.microsoft.com/tr-TR/windows7/products/features/windows-firewall>

3.2.2. Microsoft Windows XP Güvenlik Duvarı

Microsoft Windows XP üzerinde gelen aktif bir güvenlik duvarı bulunmamaktadır. Sistem kaynaklarına erişerek güvenlik duvarı yönetimi için WIPFW uygulaması kullanılmıştır. Bu güvenlik duvarına kural eklemek için:

```
"ipfw add deny tcp from any to me KAPINO"
```

demek yeterli olacaktır.

Eğer kapıyı açmak isterseniz:

```
"ipfw delete KAPINOiçinKURALNUMARASI"
```

demek yeterli olacaktır.

Kaynak: WIPFW, 2010, <http://wipfw.sourceforge.net/>

3.2.3. Linux/Unix ve Türevlerinde Kullanılan Güvenlik Duvarı

Linux/Unix ve türevleri işletim sistemlerinde kapıları yönetebilmek için iptables programı kullanılmıştır.

Bu tür sistemlerde kapı kapatmak için:

```
"iptables -A INPUT -p TCP --dport KAPINO -j REJECT"
```

Kapı açmak için:

```
"iptables -D INPUT -p TCP --dport KAPINO -j REJECT"
```

komutlarını kullanmak yeterli olacaktır.

Kaynak: Netfilter, 2010, <http://www.netfilter.org/>

3.3. KODLAMAYLA İLGİLİ ÖNEMLİ NOKTALAR

Uygulama kodları geliştirilirken, düğümlerin ana görevi olan güvenlik duvarı yönetimi göz önünde bulundurularak sistem kaynaklarıyla çakışacak ya da sistemi sağlam olmayan duruma sokacak bir kancalamadan uzak durulmuş ve kodun özellikle minimum kaynak tüketmesi için oldukça yalın ve kısa yazılmıştır. Düğüm üzerindeki web servisi toplam iki web servisi ve yaklaşık 50 kod satırından oluşmaktadır.

Gerek Ağ Geçitlerinin farklı farklı olabilmesi ve özellikle de düğüm sayısının çokluğuna göre elle ayar yapılmaması için, Web Servisi için ulaşım noktası kodu yazılmıştır. İlgili kod parçacığı, Mobil Java ve standart Java yazım kodlamasında farklılık göstermesinden dolayı aşağıda her iki kodu da ekliyorum.

Standart Java ulaşım noktası kodlaması:

```
private String get4Node(String serverType, String port, String ar, String nodeIP) {  
    ...  
    String ewsEndpoint = "http://" + nodeIP.trim()  
        + ":8080/m2node/ExecuteWebServiceService";  
    ExecuteWebServiceService ewsService = new ExecuteWebServiceService();  
    ExecuteWebService ruleNumber = ewsService.getExecuteWebServicePort();  
    ((BindingProvider)ruleNumber).getRequestContext().put(  
        BindingProvider.ENDPOINT_ADDRESS_PROPERTY, ewsEndpoint);  
    ...  
}
```

Mobil Java Kullanıcısı ulaşım noktası kodlaması:

```
...  
String ewsEndpoint =  
    "http://192.168.56.111:8080/m2gateway/AddRuleWebServiceService";  
...  
public void getServers(ChoiceGroup cg, String wcg) {  
    ...  
    AddRuleWebServiceService_Stub service1 =  
        new AddRuleWebServiceService_Stub();  
    service1._setProperty(Stub.ENDPOINT_ADDRESS_PROPERTY, ewsEndpoint);  
    ...  
}
```

3.4. KARŞILAŞTIRMA

Yapılan arařtırmalar sonrasında mobil cihazlarla dađıtık güvenlik duvarı yönetimine dair herhangi bir çalıřmaya rastlanmamıřtır.

Yapılan mobil cihazlarla dađıtık güvenlik duvarı yönetimi tezine benzerlik tespit edilen bir çalıřma, Development of a Distributed Firewall Administrator Tool çalıřmadır.

Kaynak: Yunus Erdoğan, 2008,

<http://library.iyte.edu.tr/tezler/master/bilgisayaryazilimi/T000226.pdf>

İzmir Institute of Technology'ye verilen bu tez, Microsoft Windows işletim sistemine C/C++'da yazılan uygulama ile kanca atarak ađ paketlerinin kontrolünü aktif olarak yönetmesi amaçlanmıřtır.

Mobil cihazlar ile yaptığım bu tez çalıřmasının, sadece Microsoft Windows işletim sistemi deđil, Microsoft'a ait 2000 yılı ve sonrası tüm işletim sistemlerini, tüm Linux ve türevlerini, tüm Unix ve türevlerini yönetebilirlik olarak kapsaması ve hemen hemen tüm cep telefonlarından kullanılabilmesi en büyük farkıdır.

4. ÇALIŞMA YÖNTEMİ, TEST PLATFORMLARI ve ÇALIŞMA EKRANLARI

Uygulama altyapısının koşması için, öncelikle Ağ Geçidinin ayağa kalkması gerekmektedir. Mobil Kullanıcının herhangi bir anda Ağ Geçidine web servisleri aracılığıyla bağlanıp yeni düğümler eklemesi ve düğümler üzerine de güvenlik duvarının yönetimi için web servisinin yüklenmesi yeterli olacaktır.

Yeni düğümlerin eklenmesi için gerek Ağ Geçidinin veya herhangi bir düğüm ya da mobil kullanıcı sisteminin kapatılmasına ya da uygulamanın tekrardan başlatılmasına gerek yoktur. Web Servisi olarak uygulama mimarisinin geliştirilmiş olmasının en önemli avantajlarından birisi de budur.

Koşmaya başlayan mobil kullanıcı uygulaması, Ağ Geçidine bağlanarak web servisleri aracılığıyla aktif sunucu listesini çekerek kullanıcıya sunar. Yeni sunucu eklenmesi, veya sunucu silinmesi, sunucu IP'si düzeltmesi yapılabilir. Bunların dışında, kullanıcı herhangi bir sunucuya ait aktif kapı listesini Ağ Geçidinden talep edebilir, herhangi bir düğüm ya da tüm düğümlere bir anda kapı kapatma ya da kapanmış kapının açılması görevini atayabilir. Atanan tüm görevlerin sürelerine uygulama ana ekranından erişilen İstatistikler penceresinden ulaşılabilir.

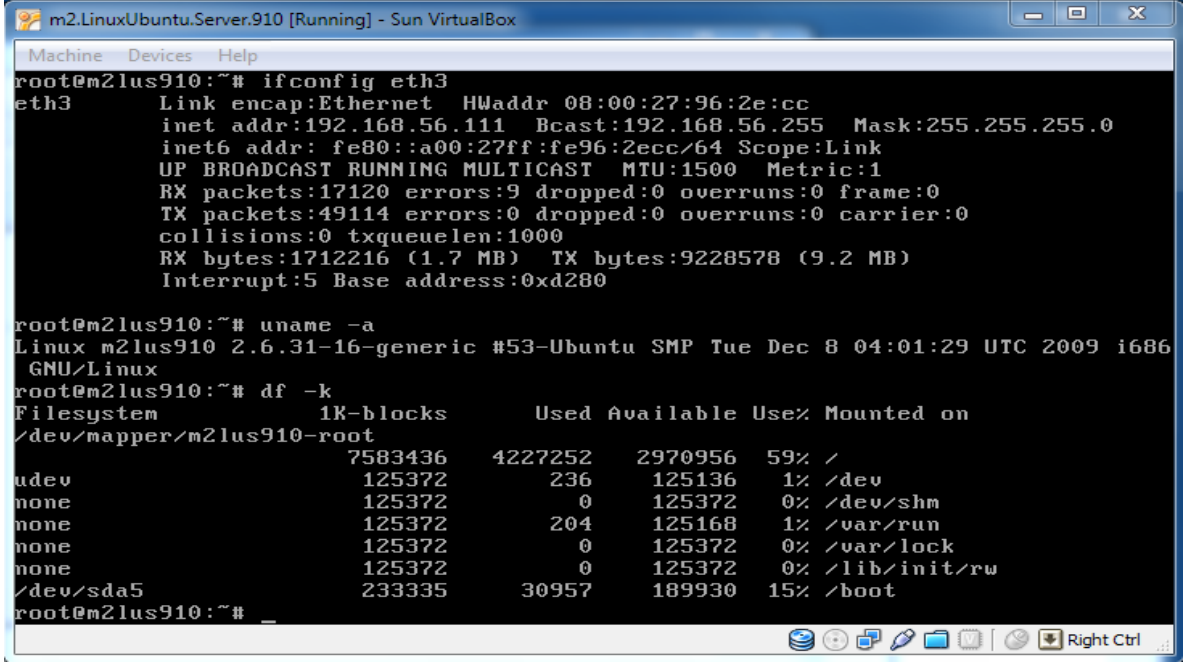
Uygulamanın gerçeğe yakın olarak çalıştırılabilmesi için dört adet sanal sistem kurulmuştur. Sanal sistem yönetimi için VirtualBox tercih edilmiştir. Bu sanal sistemler Ağ Geçidi ve aynı zamanda düğüm olarak kullanılan Ubuntu Server 9.10, düğüm olarak kullanılan Ubuntu Desktop 9.10, ve yine düğüm olarak kullanılan Microsoft Windows XP ve Microsoft Windows 7 işletim sistemlerinden oluşmaktadır.

Ağ Geçidi üzerinde tüm düğümlerin IP ve tip bilgileriyle, istatistik verileri tutulmakta olup, düğümlerin kapı durumlarının incelemesi de yine Ağ Geçidi tarafından yapılmaktadır.

Ağ Geçidi sanal sistemi, 256 Mb RAM ve 10Gb disk ile oluşturulup, uygulamanın bir çok farklı ağ ortamlarında çalışabilmesi için dört adet ağ adaptörü atanmış ve ayarları yapılmıştır.

Düğümler için Linux Desktop Ağ Geçidi ile aynı özelliklere sahip olup, Microsoft Windows XP, 384 Mb RAM ve 10Gb disk aynı özellikleri ağ bağdaştırıcıları ve Microsoft Windows 7 ise 768 Mb RAM ve 20Gb disk ve yine aynı özelliklerde ağ bağdaştırıcıları ayarları yapılmıştır.

Ağ Geçidi sunucusu ekranı:

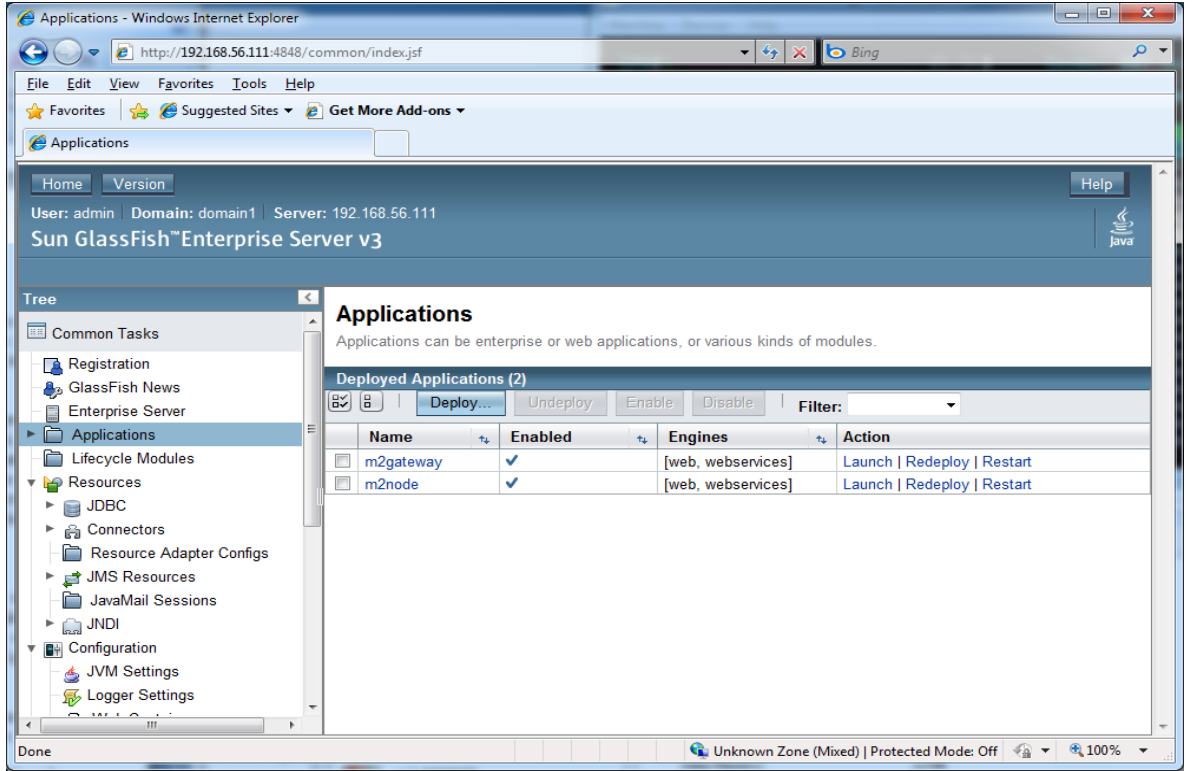


```
m2.LinuxUbuntu.Server.910 [Running] - Sun VirtualBox
Machine  Devices  Help
root@m2lus910:~# ifconfig eth3
eth3      Link encap:Ethernet  HWaddr 08:00:27:96:2e:cc
          inet addr:192.168.56.111  Bcast:192.168.56.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe96:2ecc/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:17120 errors:9 dropped:0 overruns:0 frame:0
          TX packets:49114 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1712216 (1.7 MB)  TX bytes:9228578 (9.2 MB)
          Interrupt:5 Base address:0xd280

root@m2lus910:~# uname -a
Linux m2lus910 2.6.31-16-generic #53-Ubuntu SMP Tue Dec 8 04:01:29 UTC 2009 i686
GNU/Linux
root@m2lus910:~# df -k
Filesystem            1K-blocks      Used Available Use% Mounted on
/dev/mapper/m2lus910-root
7583436    4227252    2970956    59% /
udev                125372         236    125136     1% /dev
none                125372          0    125372     0% /dev/shm
none                125372         204    125168     1% /var/run
none                125372          0    125372     0% /var/lock
none                125372          0    125372     0% /lib/init/rw
/dev/sda5            233335         30957    189930    15% /boot
root@m2lus910:~#
```

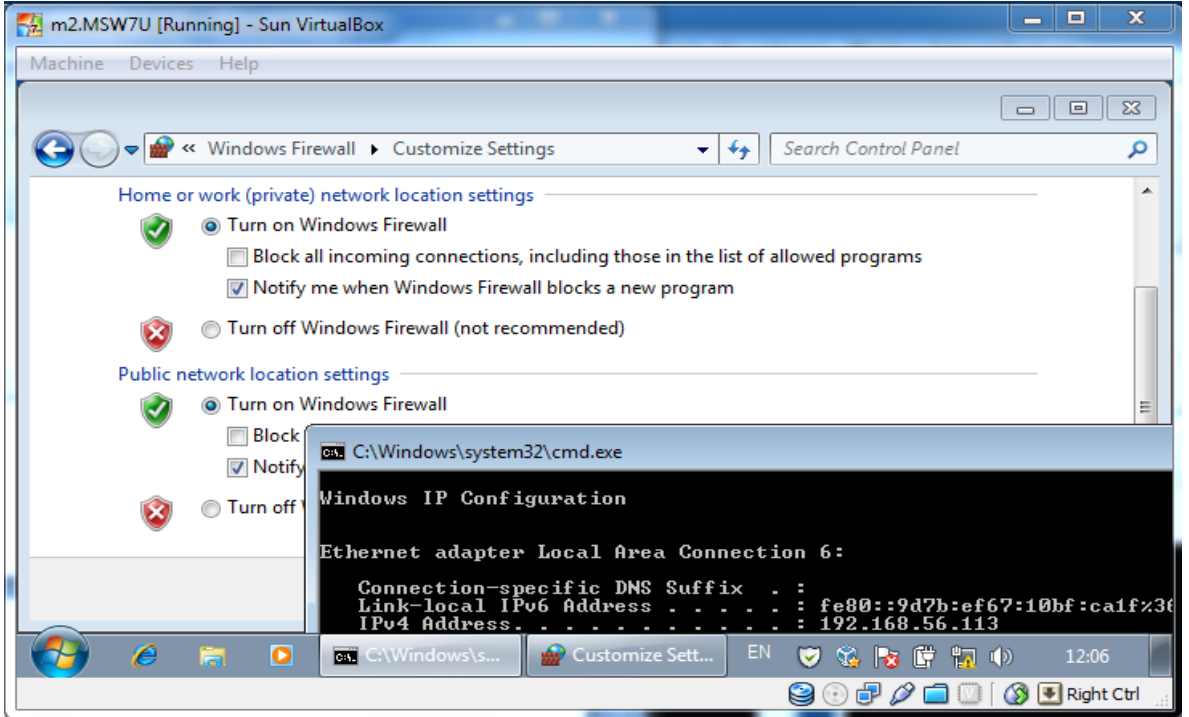
Şekil 4.1 : Ağ Geçidi Sunucusu Ekranı

Üzerine Glassfish Uygulama Server'ı ve yüklenmiş Ağ Geçidi ve Düğüm uygulamaları ekranı:



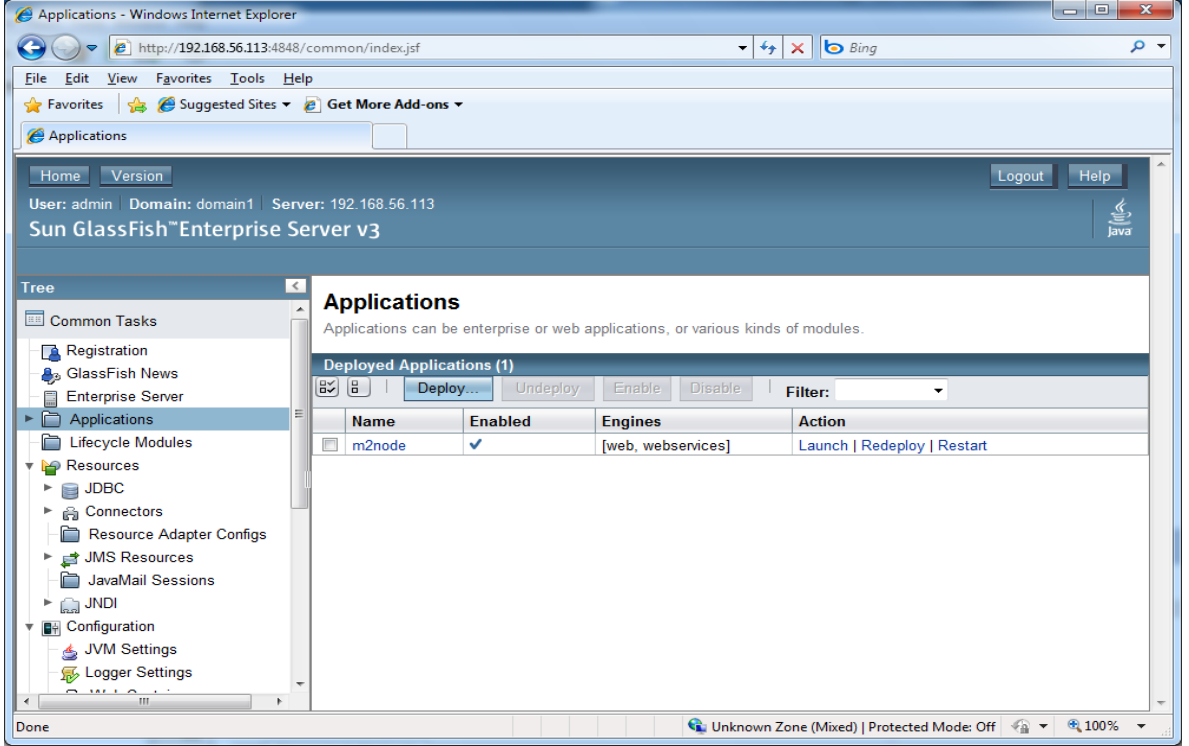
Şekil 4.2 : Glassfish Uygulama Sunucusu üzerinde yüklü Ağ Geçidi ve Düğüm uygulamaları

Bir düğüm sunucu ekranı:



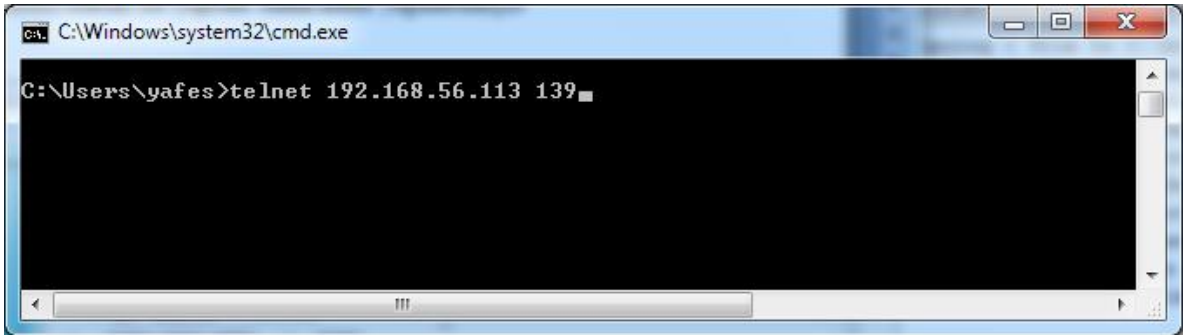
Şekil 4.3 : Düğüm sunucu ekranı

Üzerinde Glassfish Uygulama Server'ı ve yüklenmiş Düğüm uygulamaları ekranı:



Şekil 4.4 : Düğüm üzerine yüklenmiş uygulama ekranı

Test amaçlı Microsoft Windows 7 üzerindeki 139 nolu kapının:
Kapıyı kapatmadan önce ulaşılabilirliğini kontrol etmek:



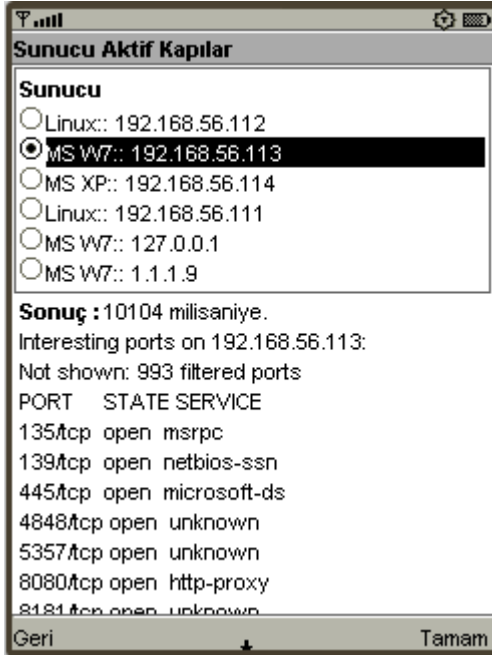
Şekil 4.5 : Düğüm sunucuna yapılan bir telnet bağlantı denemesi ekranı

Erişilen kapıda, kapı komut bekliyor:



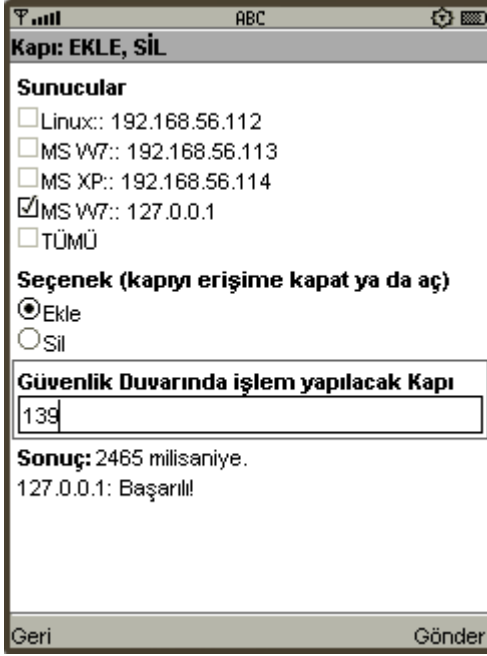
Şekil 4.6 : Düğüm'ün 139 nolu kapısına erişim ekranı

Mobil Kullanıcısı tarafından Ağ Geçidi aracılığıyla ilgili düğümün kapı durumu alınıyor:



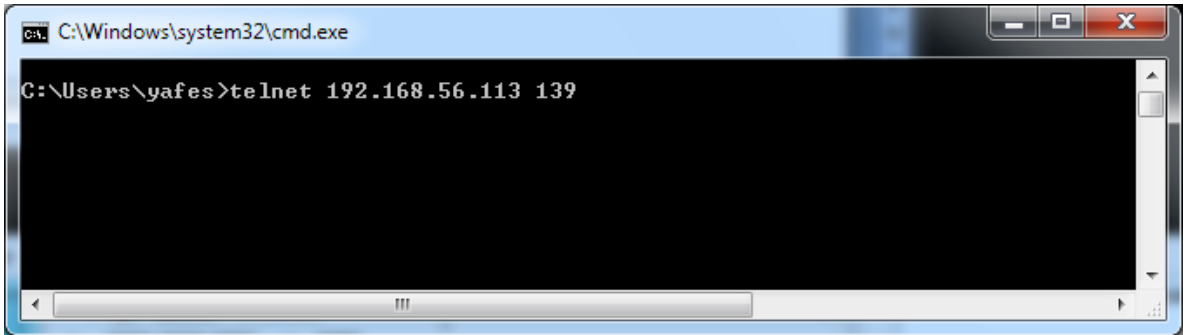
Şekil 4.7 : Düğüm aktif kapı listesi ekranı

Kapı numarası 139 mobil kullanıcı tarafından kapatılıyor:



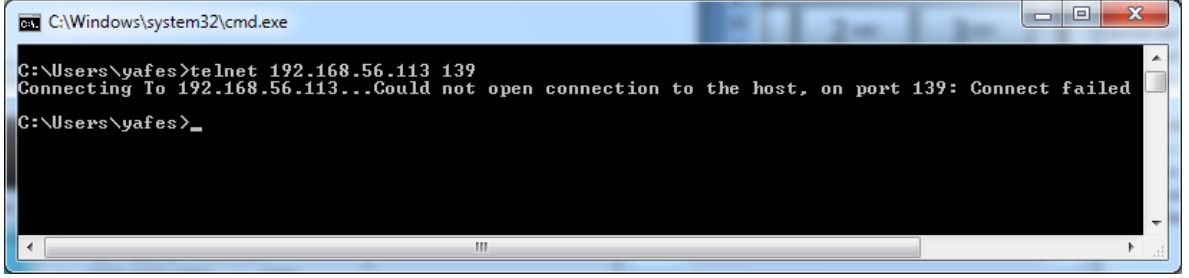
Şekil 4.8 : Düğüm'ün 139 nolu kapısının kapatılması ekranı

Kapıya tekrar telnet ile ulaşılmaya çalışılıyor:



Şekil 4.9 : telnet ile Düğüm kapısına erişmeye çalışma ekranı

Ve sonuç olarak kapattığımız kapıya erişim engelleniyor:



```
C:\Windows\system32\cmd.exe
C:\Users\yafes>telnet 192.168.56.113 139
Connecting To 192.168.56.113...Could not open connection to the host, on port 139: Connect failed
C:\Users\yafes>_
```

Şekil 4.10 : Kapıtılan Düğüm kapısına erişilememesi ekranı

5. TARTIŞMA

Tezin hedefi, yalın olarak mobil cihazlarla dağıtık güvenlik duvarlarını kural ekleyebilmek ve çıkarabilmektir. Başarılı bir şekilde sonuçlanan bu tez, uygulama altyapısına güvenlik duvarlarını yönetebilmek için daha fazla fonksiyonun eklenmesi, güvenli yollar üzerinden verinin ulaştırılması, güvenli doğrulama yöntemleriyle uygulamaların ve kullanıcıların iletişim kurması şeklinde genişletilebilir.

Tezde olduğu gibi mobil cihazlar yardımıyla, sadece güvenlik duvarı yönetimi değil, bununla birlikte hemen her tür sistem altyapılarını da yönetebileceğimizi düşünebiliriz.

6. SONUÇ

Bir kaç yıl öncesine kadar, uygulama geliştirmeden bahsedilince aklımıza masaüstü ya da taşınabilir bilgisayar ve sunucular üzerinde uygulama geliştirmek anlaşılıyordu. Fakat günümüzde önemli seviyelere gelen Mobil Uygulama Geliştirme kıpırdanmaları, önümüzdeki yıllarda hem uygulama teknolojileri anlamında, hem de uygulama pazar payı anlamında ciddi boyutlara ulaşp, hayatımızın kaçınılmaz ihtiyaçları arasına gireceği ve hatta belki en önemlileri arasında kendisine yer bulabileceğini söyleyebiliriz.

Bu tez ile, mobil uygulama imkanlarının işyerlerimizde başta olmak üzere, eğitim sektöründe, evlerimizde, IP dünyasına bağlı sistemleri yönetmek için ve bu şekilde sıralamanın uzayıp gideceği hayatımızın hemen her aşamasında kullanılabileceğini görmüş oluyoruz.

Mobil uygulamalar, önümüzdeki yıllarda, uygulama dünyasının kalbine doğru aldığı yolda oldukça hızlanacaktır öngörüsündeyim.

KAYNAKÇA

AdamSelwood, 2009, http://farm4.static.flickr.com/3105/3158195615_3263e1cc6f.jpg

Antsilevich, Ugen J. S., Kamp, Poul-Henning, Nash, Alex, Cobbs, Archie and Rizzo, Luigi WIPFW, 2010, <http://wipfw.sourceforge.net/index.html>

Bilişim Terimleri, 2010,
http://www.bilisimterimleri.com/bilgisayar_bilgisi/bilgi/18.html

Java, 2010, <http://java.sun.com/>

Glassfish, 2010, <https://glassfish.dev.java.net/>

GoModem.net, 2010,
http://www.randommart.com/images/firewall_1_images/firewall.diagram2.gif

Hosaf.org, 2010, <http://hosaf.org/yazi/javanin-uygulama-gelistirme-diyagrami>

Mernin, James, 2010, <http://www.mernin.com/blog/?p=21>

Microsoft, 2010,
<http://www.microsoft.com>
<http://www.microsoft.com/turkiye/athome/security/protect/firewall.msp>
<http://windows.microsoft.com/tr-TR/windows7/products/features/windows-firewall>

Netbeans, 2010, <http://netbeans.org/>

Netfilter, 2010, <http://www.netfilter.org/>

Rubio, Daniel, 2010,
http://searchsoa.techtarget.com/tip/0,289483,sid26_gci1188572,00.html

Somewhat Frankö, 2009,
<http://www.flickr.com/photos/somewhatfrank/3596303489/>

Sun Java, 2010, <http://java.sun.com/javase/technologies/desktop/javahelp/>

Ubuntu, 2010, <http://www.ubuntu.com>

Virtualbox, 2010, <http://www.virtualbox.org/>

EKLER

- Güvenlik Duvarı Web Servis Uygulaması Kodları (Ek: 1)**
- Ağ Geçidi Sunucu Web Servis Uygulaması Kodları (Ek: 2)**
- Mobil İstemci Uygulaması Kodları (Ek: 3)**

EK 1

Güvenlik Duvarları üzerinde çalışması için geliştirilen web servis kodu:

```
/**
 *
 * @author Yafes Duygulutuna
 * Bahcesehir University, Computer Science, Master Education, 2007-2010
 */

package com.yafes.m2node;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import javax.ws.WebMethod;
import javax.ws.WebParam;
import javax.ws.WebService;

@WebService()
public class ExecuteWebService {

    /**
     * Web service operation
     */
    @WebMethod(operationName = "execute")
    public String execute(@WebParam(name = "data")
        String data, @WebParam(name = "isFirewallRule")
        String isFirewallRule) {
        //TODO write your implementation code here:
        String OperationResult = "Başarılı!";
        Runtime ProcessCommand = Runtime.getRuntime();
        try {
            Process ProcessProcess = ProcessCommand.exec(data);
            InputStream inStd = ProcessProcess.getInputStream();
            InputStreamReader inStdR = new InputStreamReader(inStd);
            BufferedReader bStd = new BufferedReader(inStdR);
            String line = null;
            while((line = bStd.readLine()) != null && (!isFirewallRule.equals("FW"))) {
                OperationResult = OperationResult + "\n" + line;
            }
        } catch (IOException ioe) {
            ioe.printStackTrace();
            OperationResult = "Başarısız!";
        }
        return OperationResult;
    }

    @WebMethod(operationName = "getMSXPPort")
    public String getMSXPPort(@WebParam(name = "port")
        String port) {
```



```

//TODO write your implementation code here:
boolean findOK= false;
String OperationResult = null;
Runtime ProcessCommand = Runtime.getRuntime();
try {
    Process ProcessProcess = ProcessCommand.exec("ipfw -at list");
    InputStream inStd = ProcessProcess.getInputStream();
    InputStreamReader inStdR = new InputStreamReader(inStd);
    BufferedReader bStd = new BufferedReader(inStdR);
    String line = null;
    while(((line = bStd.readLine()) != null) && (!findOK)) {
        if (line.indexOf("deny tcp from any to me " + port) > -1) {
            findOK= true;
            OperationResult= line.substring(0, 5);
            // System.out.println("rule number: " + OperationResult);
        }
    }
} catch (IOException ioe) {
    ioe.printStackTrace();
}
return OperationResult;
}
}

```

EK 2

Ağ Geçidi olarak göre yapan sunucu için yazılan web servis kodu:

```
/**
 *
 * @author Yafes Duygulutuna
 * Bahcesehir University, Computer Science, Master Education, 2007-2010
 */

package com.yafes.m2gateway;

import com.yafes.m2node.ExecuteWebService;
import com.yafes.m2node.ExecuteWebServiceService;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.util.ArrayList;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.WebService;
import javax.xml.ws.BindingProvider;

@WebService()
public class AddRuleWebService {

    String filePath=
        //"c:/bu/NetBeansProjects/m2/m2gateway/";
        "/usr/local/sges-v3/glassfish/domains/domain1/m2gateway.data/";

    private String get4Node(String serverType, String port, String ar,
        String nodeIP) {
        String s1= null;
        if (serverType.equals("MS W7")) {
            if (ar.equals("A")) {
                s1= "netsh advfirewall firewall add rule name=block" + port
                    + " protocol=TCP dir=in localport=" + port
                    + " action=block";
            } else {
                s1= "netsh advfirewall firewall delete rule name=block" + port;
            }
        } else if (serverType.equals("MS XP")) {
            if (ar.equals("A")) {
                s1= "ipfw add deny tcp from any to me " + port;
            } else {
                String ewsEndpoint = "http://" + nodeIP.trim()

```

```

        + ":8080/m2node/ExecuteWebServiceService";
ExecuteWebServiceService ewsService =
    new ExecuteWebServiceService();
ExecuteWebService ruleNumber =
    ewsService.getExecuteWebServicePort();
    ((BindingProvider)ruleNumber).getRequestContext().put(
        BindingProvider.ENDPOINT_ADDRESS_PROPERTY,
        ewsEndpoint);
    s1= "ipfw delete " + ruleNumber.getMSXPPort(port);
    }
} else if (serverType.equals("Linux")) {
    if (ar.equals("A")) {
        s1= "iptables -A INPUT -p TCP --dport " + port + " -j REJECT";
    } else {
        s1= "iptables -D INPUT -p TCP --dport " + port + " -j REJECT";
    }
}
}
return s1;
}

private String send2Node(String IP, String rule) {
    String ewsEndpoint = "http://" + IP.trim()
        + ":8080/m2node/ExecuteWebServiceService";
    String result1= "";
    try {
        ExecuteWebServiceService ewsService =
            new ExecuteWebServiceService();
        ExecuteWebService ewsPort = ewsService.getExecuteWebServicePort();
        ((BindingProvider)ewsPort).getRequestContext().put(
            BindingProvider.ENDPOINT_ADDRESS_PROPERTY, ewsEndpoint);
        // System.out.println("ewsEndpoint: " + ewsEndpoint);
        result1 = ewsPort.execute(rule, "FW");
        // System.out.println("rule: " + rule);
    } catch (Exception e) {
        e.printStackTrace();
        result1= "Başarızlı!";
    }
    // System.out.println("sure: " + durationTime + " result1: " + result1);
    return result1;
}

/**
 * Web service operation
 */
@WebMethod(operationName = "add")
public String[] add(@WebParam(name = "servers")
String servers, @WebParam(name = "port")
String port, @WebParam(name = "addremove") java.lang.String addremove) {
    //TODO write your implementation code here:
    String s1= "";
    ArrayList<String> rs1 = new ArrayList<String>();

    while (servers.indexOf("::") > -1) {

```

```

        if (servers.indexOf("|") > -1) {
            s1 = servers.substring(0, servers.indexOf("|"));
            servers = servers.substring(servers.indexOf("|")+1,
                servers.length());
        } else {
            s1 = servers;
            servers = "";
        }
        String nodeIP = s1.substring(s1.indexOf(":")+3, s1.length());
        String nodeCommand = get4Node(s1.substring(0, s1.indexOf(":")),
            port, addremove, nodeIP);
        // System.out.println("nodeCommand: " + nodeCommand + " nodeIP: "
        // + nodeIP);
        rs1.add(nodeIP + ": " + send2Node(nodeIP, nodeCommand));
    }
    String[] rsa = new String[rs1.size()];
    return rs1.toArray(rsa);
}

/**
 * Web service operation
 */
@WebMethod(operationName = "setservers")
public String setservers(@WebParam(name = "server")
String server) {
    //TODO write your implementation code here:
    String s1 = "";
    s1 = server + "\r\n";
    try {
        // Open the file that is the first
        // command line parameter

        BufferedWriter writer = new BufferedWriter(new FileWriter(filePath + "servers.txt",
true));
        writer.write(s1);
        writer.close();
    } catch (Exception e) { //Catch exception if any
        System.err.println("Error: " + e.getMessage());
    }
    return "";
}

/**
 * Web service operation
 */
@WebMethod(operationName = "getServerList")
public String[] getServerList(@WebParam(name = "server")
String server) {
    //TODO write your implementation code here:
    ArrayList<String> rs1 = new ArrayList<String>();
    try {
        BufferedReader br = new BufferedReader(new FileReader(new File(
filePath + "servers.txt")));

```

```

String strLine;
while ((strLine = br.readLine()) != null) {
    if ((strLine = strLine.trim()).equals("")) {
        continue;
    }
    rs1.add(strLine);
}
//Close the input stream
br.close();
} catch (Exception e){//Catch exception if any
    System.err.println("Error: " + e.getMessage());
}

String[] rsa= new String[rs1.size()];
return rs1.toArray(rsa);
}

/**
 * Web service operation
 */
@WebMethod(operationName = "serverDeleteChange")
public String serverDeleteChange(@WebParam(name = "oldServer")
String oldServer, @WebParam(name = "newServer") String newServer) {
    //TODO write your implementation code here:
    String s1= null;
    try {
        File inFile = new File(filePath + "servers.txt");
        File newFile = new File(inFile.getAbsolutePath() + ".tmp");
        BufferedReader readerServer = new BufferedReader(new FileReader(
            filePath + "servers.txt"));
        PrintWriter writerServer = new PrintWriter(new FileWriter(newFile));
        while ((s1 = readerServer.readLine()) != null) {
            if (!s1.trim().equals(oldServer)) {
                writerServer.println(s1);
                writerServer.flush();
            } else if (!newServer.equals("")) {
                writerServer.println(newServer);
                writerServer.flush();
            }
        }
        writerServer.close();
        readerServer.close();
        if (!inFile.delete()) {
            // System.out.println("Could not delete file");
        }
        if (!newFile.renameTo(inFile)) {
            // System.out.println("Could not rename file");
        }
    } catch (Exception e){//Catch exception if any
        System.err.println("Error: " + e.getMessage());
    }
    return "";
}

```

```

}

/**
 * Web service operation
 */
@WebMethod(operationName = "getActiveServerPorts")
public String[] getActiveServerPorts(@WebParam(name = "IP")
String IP) {
    //TODO write your implementation code here:
    ArrayList<String> rs1 = new ArrayList<String>();
    Runtime ProcessCommand = Runtime.getRuntime();
    try {
        Process ProcessProcess = ProcessCommand.exec("nmap -P0 " + IP);
        InputStream inStd = ProcessProcess.getInputStream();
        InputStreamReader inStdR = new InputStreamReader(inStd);
        BufferedReader bStd = new BufferedReader(inStdR);
        String line = null;
        boolean bStart= false;
        while(((line = bStd.readLine()) != null)) {
            if (line.indexOf(IP) > -1) {
                bStart= true;
            }
            if (bStart) {
                if (line.indexOf("Nmap done:") > -1) {
                    bStart= false;
                } else {
                    rs1.add(line);
                }
            }
            // System.out.println(line);
        }
    } catch (IOException ioe) {
        ioe.printStackTrace();
    }
    if (rs1.size()== 0) {
        rs1.add("Başarısız!");
    }
    String[] rsa= new String[rs1.size()];
    return rs1.toArray(rsa);
}

/**
 * Web service operation
 */
@WebMethod(operationName = "getStatistics")
public String[] getStatistics() {
    //TODO write your implementation code here:
    String s1= null;
    ArrayList<String> rs1 = new ArrayList<String>();
    try {
        BufferedReader readerServer = new BufferedReader(new FileReader(
            filePath + "statistics.txt"));
        while ((s1 = readerServer.readLine()) != null) {

```

```

        rs1.add(s1);
    }
    readerServer.close();
} catch (Exception e){//Catch exception if any
    System.err.println("Error: " + e.getMessage());
}
String[] rsa= new String[rs1.size()];
return rs1.toArray(rsa);
}

/**
 * Web service operation
 */
@WebMethod(operationName = "setStatistics")
public String setStatistics(@WebParam(name = "item")
String item, @WebParam(name = "duration") String duration) {
    //TODO write your implementation code here:
    String s1= null;
    try {
        File inFile = new File(filePath + "statistics.txt");
        File newFile = new File(inFile.getAbsolutePath() + ".tmp");
        BufferedReader readerServer = new BufferedReader(new FileReader(
            filePath + "statistics.txt"));
        PrintWriter writerServer = new PrintWriter(new FileWriter(newFile));
        while ((s1 = readerServer.readLine()) != null) {
            if (s1.substring(0, s1.indexOf(":")).equals(item)) {
                int itemCount= Integer.parseInt(s1.substring(s1.indexOf(":")
                    + 1, s1.indexOf("::"))) + 1;
                int newDuration= Integer.parseInt(s1.substring(
                    s1.indexOf("::") + 2, s1.length())
                    + Integer.parseInt(duration));
                writerServer.println(s1.substring(0, s1.indexOf(":")) + ":"
                    + Integer.toString(itemCount) + "::"
                    + Integer.toString(newDuration));
            } else {
                writerServer.println(s1);
            }
        }
        writerServer.flush();
    }
    writerServer.close();
    readerServer.close();
    if (!inFile.delete()) {
        // System.out.println("Could not delete file");
    }
    if (!newFile.renameTo(inFile)) {
        // System.out.println("Could not rename file");
    }
} catch (Exception e){//Catch exception if any
    System.err.println("Error: " + e.getMessage());
}
return "";
}
}
}

```

EK 3

Mobil Kullanıcı için geliştirilen kod:

```
/**
 *
 * @author Yafes Duygulutuna
 * Bahcesehir University, Computer Science, Master Education, 2007-2010
 */

package main;

import addrulewebservice.AddRuleWebServiceService_Stub;
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import javax.xml.rpc.Stub;
import org.netbeans.microedition.lcdui.SplashScreen;

    //</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: mainForm ">

public class MainMIDlet extends MIDlet implements ItemCommandListener, CommandListener {

    private boolean midletPaused = false;
    String ewsEndpoint =
        //"http://127.0.0.1:8080/m2gateway/AddRuleWebServiceService";
        "http://192.168.56.111:8080/m2gateway/AddRuleWebServiceService";

    //<editor-fold defaultstate="collapsed" desc=" Generated Fields ">
    private Form mainForm;
    private ChoiceGroup mainFormCG;
    private Form portAddDeleteForm;
    private StringItem portAddDeleteFormResultSI;
    private TextField portAddDeleteFormPortTF;
    private ChoiceGroup portAddDeleteFormServerListCG;
    private ChoiceGroup portAddDeleteFormOptionCG;
    private SplashScreen splashScreen;
    private Form serversForm;
    private ChoiceGroup serverFormOSTypeCG;
    private ChoiceGroup serverFormServerListCG;
    private ChoiceGroup serverFormOptionCG;
    private TextField serverFormIPTF;
    private Form aboutForm;
    private StringItem aboutsi;
    private ImageItem aboutii;
    private Form activePortsForm;
    private ChoiceGroup activePortsFormCG;
    private StringItem activePortsFormsi;
    private Form statisticsForm;
    private StringItem serverPortListingCountSI;
    private StringItem serverPortListingAverageDurationSI;
```



```

private StringItem deleteRuleCountSI;
private StringItem deleteRuleAverageDurationSI;
private StringItem addRuleCountSI;
private StringItem addRuleAverageDurationSI;
private StringItem stringItem;
private Form helpForm;
private StringItem helpFormStatisticsSI;
private StringItem helpFormServerAddvsSI;
private StringItem helpFormAboutSI;
private StringItem helpFormHelpSI;
private StringItem helpFormServerPortvsSI;
private StringItem helpFormPortAddvsSI;
private Command portAddDeleteFormbackC;
private Command exitCommand;
private Command mainFormGoC;
private Command portAddDeleteFormokC;
private Command aboutFormbackC;
private Command serverFormbackC;
private Command activePortsFormbackC;
private Command activePortsFormokC;
private Command serverFormokC;
private Command statisticsFormbackC;
private Command statisticsFormUpdateC;
private Command helpFormbackC;
private Image aboutI;
private Image splashI;
//</editor-fold>

/**
 * The MainMIDlet constructor.
 */
public MainMIDlet() {
}

//<editor-fold defaultstate="collapsed" desc=" Generated Methods ">
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Method: initialize ">
/**
 * Initalizes the application.
 * It is called only once when the MIDlet is started. The method is called before the
<code>startMIDlet</code> method.
 */
private void initialize() {
    // write pre-initialize user code here

    // write post-initialize user code here
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Method: startMIDlet ">
/**
 * Performs an action assigned to the Mobile Device - MIDlet Started point.

```

```

*/
public void startMIDlet() {
    // write pre-action user code here
    switchDisplayable(null, getSplashScreen());
    // write post-action user code here
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Method: resumeMIDlet ">
/**
 * Performs an action assigned to the Mobile Device - MIDlet Resumed point.
 */
public void resumeMIDlet() {
    // write pre-action user code here

    // write post-action user code here
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Method: switchDisplayable ">
/**
 * Switches a current displayable in a display. The <code>display</code> instance is taken from
<code>getDisplay</code> method. This method is used by all actions in the design for switching
displayable.
 * @param alert the Alert which is temporarily set to the display; if <code>null</code>, then
<code>nextDisplayable</code> is set immediately
 * @param nextDisplayable the Displayable to be set
 */
public void switchDisplayable(Alert alert, Displayable nextDisplayable) {
    // write pre-switch user code here
    Display display = getDisplay();
    if (alert == null) {
        display.setCurrent(nextDisplayable);
    } else {
        display.setCurrent(alert, nextDisplayable);
    }
    // write post-switch user code here
}
//</editor-fold>

public void commandsListAction() {
    if (mainFormCG.getString(mainFormCG.getSelectedIndex()).equals(
        "Kapı //ekle, sil")) {
        switchDisplayable(null, getPortAddDeleteForm());
    } else if (mainFormCG.getString(mainFormCG.getSelectedIndex()).equals(
        "Hakkında")) {
        switchDisplayable(null, getAboutForm());
    } else if (mainFormCG.getString(mainFormCG.getSelectedIndex()).equals(
        "Sunucu //ekle, sil, düzelt")) {
        switchDisplayable(null, getServersForm());
    } else if (mainFormCG.getString(mainFormCG.getSelectedIndex()).equals(
        "Sunucu //aktif kapı listesi")) {
        switchDisplayable(null, getActivePortsForm());
    }
}

```

```

    } else if (mainFormCG.getString(mainFormCG.getSelectedIndex()).equals(
        "İstatistik")) {
        switchDisplayable(null, getStatisticsForm());
    } else if (mainFormCG.getString(mainFormCG.getSelectedIndex()).equals(
        "Yardım")) {
        switchDisplayable(null, getHelpForm());
    }
}
//</editor-fold>

public void getServers(ChoiceGroup cg, String wcg) {
    try {
        AddRuleWebServiceService_Stub service1 =
            new AddRuleWebServiceService_Stub();
        service1._setProperty(Stub.ENDPOINT_ADDRESS_PROPERTY, ewsEndpoint);
        String[] result1 = service1.getServerList("get");

        // System.out.println("cg: " + cg);
        cg.deleteAll();
        for (int i = 0; i < result1.length; i++) {
            cg.append(result1[i], null);
        }
        if (!wcg.equals("L")) {
            cg.append("ALL", null);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

//<editor-fold defaultstate="collapsed" desc=" Generated Method: commandAction for Items ">
/**
 * Called by a system to indicated that a command has been invoked on a particular item.
 * @param command the Command that was invoked
 * @param displayable the Item where the command was invoked
 */
public void commandAction(Command command, Item item) {
    // write pre-action user code here
    if (item == mainFormCG) {
        if (command == mainFormGoC) {
            // write pre-action user code here

            // write post-action user code here
            commandsListAction();
        }
    }
    } else if (item == portAddDeleteFormPortTF) {
        if (command == portAddDeleteFormokC) {
            // write pre-action user code here

            // write post-action user code here
            String s1= "";

```

```

String ar="A";

short i1= (short) (portAddDeleteFormServerListCG.size() - 1);
boolean bALL= portAddDeleteFormServerListCG.isSelected(
    portAddDeleteFormServerListCG.size()-1);

while (i1 > -1) {
    if ((portAddDeleteFormServerListCG.isSelected(i1)== true)
        || (bALL== true)) {
        if (s1.equals("")) {
            if (!portAddDeleteFormServerListCG.getString(i1).equals(
                "ALL")) {
                s1= portAddDeleteFormServerListCG.getString(i1);
            }
        } else {
            s1 = s1 + "|"
                + portAddDeleteFormServerListCG.getString(i1);
        }
    }
    i1--;
}
// portAddDeleteFormResultSI.setText(s1 + "|"
//     + portAddDeleteFormPortTF.getString());
if (portAddDeleteFormOptionCG.isSelected(1)== true) {
    ar= "D";
}
if (s1.equals(""))
    || portAddDeleteFormPortTF.getString().equals("")) {
    portAddDeleteFormResultSI.setText(
        "Lütfen sunucu seçin ve/ya kapı numarasını girin!");
} else {
    try {
        long startTime = System.currentTimeMillis();

        AddRuleWebServiceService_Stub service1 =
            new AddRuleWebServiceService_Stub();
        service1._setProperty(Stub.ENDPOINT_ADDRESS_PROPERTY,
            ewsEndpoint);

        String s = portAddDeleteFormPortTF.getString();
        // System.out.println("s: " + s);
        String[] result1 = service1.add(s1, s, ar);
        // System.out.println("Sonuç: " + result1);
        long durationTime = System.currentTimeMillis() - startTime;
        String result2= service1.setStatistics(
            portAddDeleteFormOptionCG.getString(
                portAddDeleteFormOptionCG.getSelectedIndex()),
            Long.toString(durationTime));

        // System.out.println("sure: " + durationTime + " result1: "
        //     + result1);
        portAddDeleteFormResultSI.setText(
            Long.toString(durationTime) + " milisaniye."

```

```

        + "\r\n");
        for (int i = 0; i < result1.length; i++) {
            portAddDeleteFormResultSI.setText(
                portAddDeleteFormResultSI.getText() + result1[i]
                + "\r\n");
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
} else if (item == serverFormIPTF) {
    if (command == serverFormokC) {
        // write pre-action user code here

        // write post-action user code here
        if ((!serverFormIPTF.getString().equals("")
            && (serverFormOptionCG.isSelected(0)== true))) {
            try {
                AddRuleWebServiceService_Stub service1 =
                    new AddRuleWebServiceService_Stub();
                service1._setProperty(Stub.ENDPOINT_ADDRESS_PROPERTY,
                    ewsEndpoint);
                String result1 = service1.setservers(
                    serverFormOSTypeCG.getString(
                    serverFormOSTypeCG.getSelectedIndex())
                    + ":: " + serverFormIPTF.getString());
                // System.out.println(result1);
            } catch (Exception e) {
                System.out.println(e.toString());
            }
        } else if (serverFormOptionCG.isSelected(1)== true) {
            try {
                AddRuleWebServiceService_Stub service1 =
                    new AddRuleWebServiceService_Stub();
                service1._setProperty(Stub.ENDPOINT_ADDRESS_PROPERTY,
                    ewsEndpoint);

                String result1 = service1.serverDeleteChange(
                    serverFormServerListCG.getString(
                    serverFormServerListCG.getSelectedIndex()), "");
                // System.out.println(result1);
            } catch (Exception e) {
                System.out.println(e.toString());
            }
        }

    } else if (serverFormOptionCG.isSelected(2)== true) {
        try {
            AddRuleWebServiceService_Stub service1 =
                new AddRuleWebServiceService_Stub();
            service1._setProperty(Stub.ENDPOINT_ADDRESS_PROPERTY,
                ewsEndpoint);

```

```

        String result1 = service1.serverDeleteChange(
            serverFormServerListCG.getString(
                serverFormServerListCG.getSelectedIndex()),
            serverFormOSTypeCG.getString(
                serverFormOSTypeCG.getSelectedIndex())
            + ":: " + serverFormIPTF.getString());
        // System.out.println(result1);
    } catch (Exception e) {
        System.out.println(e.toString());
    }
}
}
getServers(serverFormServerListCG, "L"); // for List
getServers(activePortsFormcG, "L"); // for List
getServers(portAddDeleteFormServerListCG, "W"); // for Work
}
}
// write post-action user code here
}
//</editor-fold>
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: mainForm ">
/**
 * Returns an initialized instance of mainForm component.
 * @return the initialized component instance
 */
public Form getMainForm() {
    if (mainForm == null) {
        // write pre-init user code here
        mainForm = new Form("m2 G\u00FCvenlik Duvar\u0131 Y\u00F6netimi", new Item[] {
getMainFormCG() });
        mainForm.addCommand(getExitCommand());
        mainForm.setCommandListener(this);
        // write post-init user code here
    }
    return mainForm;
}
//</editor-fold>

//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: portAddDeleteFormResultSI ">
/**
 * Returns an initialized instance of portAddDeleteFormResultSI component.
 * @return the initialized component instance
 */
public StringItem getPortAddDeleteFormResultSI() {
    if (portAddDeleteFormResultSI == null) {
        // write pre-init user code here
        portAddDeleteFormResultSI = new StringItem("Sonu\u00E7:", null);
    }
}
//</editor-fold>

```

```

        // write post-init user code here
    }
    return portAddDeleteFormResultSI;
}
//</editor-fold>
//</editor-fold>
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: portAddDeleteFormPortTF ">
/**
 * Returns an initalized instance of portAddDeleteFormPortTF component.
 * @return the initialized component instance
 */
public TextField getPortAddDeleteFormPortTF() {
    if (portAddDeleteFormPortTF == null) {
        // write pre-init user code here
        portAddDeleteFormPortTF = new TextField("G\u00FCvenlik Duvar\u0131nda i\u015Flem
yap\u0131lacak Kap\u0131", null, 200, TextField.ANY);
        portAddDeleteFormPortTF.addCommand(getPortAddDeleteFormokC());
        portAddDeleteFormPortTF.setItemCommandListener(this);
        // write post-init user code here
    }
    return portAddDeleteFormPortTF;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Method: commandAction for
Displayables ">
/**
 * Called by a system to indicated that a command has been invoked on a particular displayable.
 * @param command the Command that was invoked
 * @param displayable the Displayable where the command was invoked
 */
public void commandAction(Command command, Displayable displayable) {
    // write pre-action user code here
    if (displayable == aboutForm) {
        if (command == aboutFormback) {
            // write pre-action user code here
            switchDisplayable(null, getMainForm());
            // write post-action user code here
        }
    } else if (displayable == activePortsForm) {
        if (command == activePortsFormbackC) {
            // write pre-action user code here
            switchDisplayable(null, getMainForm());
            // write post-action user code here
        } else if (command == activePortsFormokC) {
            // write pre-action user code here

            // write post-action user code here
            activePortsFormsi.setText("L\u00fctfen bekleyin...");
            try {
                long startTime = System.currentTimeMillis();

```

```

String s1= activePortsFormcG.getString(
    activePortsFormcG.getSelectedIndex());
s1= s1.substring(s1.indexOf(":".)+3, s1.length());

    AddRuleWebServiceService_Stub service1 =
        new AddRuleWebServiceService_Stub();
    service1._setProperty(Stub.ENDPOINT_ADDRESS_PROPERTY,
        ewsEndpoint);

String[] result1= service1.getActiveServerPorts(s1);

s1= "";
long durationTime = System.currentTimeMillis() - startTime;
    s1= Long.toString(durationTime) + " milisaniye.";
for (int i = 0; i < result1.length; i++) {
    s1= s1 + "\r\n" + result1[i];
}
activePortsFormsi.setText(s1);

service1.setStatistics("Nmap",
    Long.toString(durationTime));

    // System.out.println("Sonuç : " + activePortsFormsi.getText());
} catch (Exception e) {
    e.printStackTrace();
    portAddDeleteFormResultSI.setText(e.toString());
}
}
} else if (displayable == helpForm) {
    if (command == helpFormbackC) {
        // write pre-action user code here
        switchDisplayable(null, getMainForm());
        // write post-action user code here
    }
} else if (displayable == mainForm) {
    if (command == exitCommand) {
        // write pre-action user code here
        exitMIDlet();
        // write post-action user code here
    }
} else if (displayable == portAddDeleteForm) {
    if (command == portAddDeleteFormbackC) {
        // write pre-action user code here
        switchDisplayable(null, getMainForm());
        // write post-action user code here
    }
} else if (displayable == serversForm) {
    if (command == serverFormbackC) {
        // write pre-action user code here
        switchDisplayable(null, getMainForm());
        // write post-action user code here
    }
} else if (displayable == splashScreen) {

```



```

if (command == SplashScreen.DISMISS_COMMAND) {
    // write pre-action user code here
    switchDisplayable(null, getMainForm());
    // write post-action user code here
}
} else if (displayable == statisticsForm) {
    if (command == statisticsFormUpdateC) {
        // write pre-action user code here

        // write post-action user code here
        try {
            AddRuleWebServiceService_Stub service1 =
                new AddRuleWebServiceService_Stub();
            service1._setProperty(Stub.ENDPOINT_ADDRESS_PROPERTY,
                ewsEndpoint);

            String[] result1 = service1.getStatistics();
            // System.out.println("Sonuç: " + result1);
            long count= 0;
            long duration= 0;
            for (int i = 0; i < result1.length; i++) {
                if (result1[i].substring(0, result1[i].indexOf(
                    ":")).equals("Ekle")) {
                    count= Long.parseLong(result1[i].substring(
                        result1[i].indexOf(":") + 1,
                        result1[i].indexOf("::")));
                    duration= Long.parseLong(result1[i].substring(
                        result1[i].indexOf("::") + 2,
                        result1[i].length())) / count;
                    addRuleCountSl.setText(Long.toString(count));
                    addRuleAverageDurationSl.setText(
                        Long.toString(duration));
                } else if (result1[i].substring(
                    0, result1[i].indexOf(":")).equals("Sil")) {
                    count= Long.parseLong(result1[i].substring(
                        result1[i].indexOf(":") + 1,
                        result1[i].indexOf("::")));
                    duration= Long.parseLong(result1[i].substring(
                        result1[i].indexOf("::") + 2,
                        result1[i].length())) / count;
                    deleteRuleCountSl.setText(Long.toString(count));
                    deleteRuleAverageDurationSl.setText(
                        Long.toString(duration));
                }
                if (result1[i].substring(
                    0, result1[i].indexOf(":")).equals("Nmap")) {
                    count= Long.parseLong(result1[i].substring(
                        result1[i].indexOf(":") + 1,
                        result1[i].indexOf("::")));
                    duration= Long.parseLong(result1[i].substring(
                        result1[i].indexOf("::") + 2,
                        result1[i].length())) / count;
                    serverPortListingCountSl.setText(

```

```

                Long.toString(count));
            serverPortListingAverageDurationSl.setText(
                Long.toString(duration));
        }
    }
} catch (Exception e) {
    e.printStackTrace();
    portAddDeleteFormResultSl.setText(e.toString());
}
} else if (command == statisticsFormbackC) {
    // write pre-action user code here
    switchDisplayable(null, getMainForm());
    // write post-action user code here
}
}
// write post-action user code here
}
//</editor-fold>
//</editor-fold>
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: portAddDeleteFormbackC ">
/**
 * Returns an initialized instance of portAddDeleteFormbackC component.
 * @return the initialized component instance
 */
public Command getPortAddDeleteFormbackC() {
    if (portAddDeleteFormbackC == null) {
        // write pre-init user code here
        portAddDeleteFormbackC = new Command("Geri", Command.BACK, 0);
        // write post-init user code here
    }
    return portAddDeleteFormbackC;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: splashScreen ">
/**
 * Returns an initialized instance of splashScreen component.
 * @return the initialized component instance
 */
public SplashScreen getSplashScreen() {
    if (splashScreen == null) {
        // write pre-init user code here
        splashScreen = new SplashScreen(getDisplay());
        splashScreen.setTitle("m2 G\u00FCvenlik Duvar\u0131 Y\u00F6netimi");
        splashScreen.setCommandListener(this);
        splashScreen.setImage(getSplashI());
        splashScreen.setText("Ho\u015Fgeldiniz...");
        // write post-init user code here
    }
    return splashScreen;
}
}

```

```

//</editor-fold>

//</editor-fold>
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: aboutI ">
/**
 * Returns an initialized instance of aboutI component.
 * @return the initialized component instance
 */
public Image getAboutI() {
    if (aboutI == null) {
        // write pre-init user code here
        try {
            aboutI = Image.createImage("/splash2.jpg");
        } catch (java.io.IOException e) {
            e.printStackTrace();
        }
        // write post-init user code here
    }
    return aboutI;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: mainFormGoC ">
/**
 * Returns an initialized instance of mainFormGoC component.
 * @return the initialized component instance
 */
public Command getMainFormGoC() {
    if (mainFormGoC == null) {
        // write pre-init user code here
        mainFormGoC = new Command("Tamam", "Git", Command.OK, 0);
        // write post-init user code here
    }
    return mainFormGoC;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: exitCommand ">
/**
 * Returns an initialized instance of exitCommand component.
 * @return the initialized component instance
 */
public Command getExitCommand() {
    if (exitCommand == null) {
        // write pre-init user code here
        exitCommand = new Command("\u00C7\u0131k\u0131\u015F", Command.EXIT, 0);
        // write post-init user code here
    }
    return exitCommand;
}
//</editor-fold>

```

```
//</editor-fold>
//</editor-fold>
```

```
//<editor-fold defaultstate="collapsed" desc=" Generated Getter:
portAddDeleteFormServerListCG ">
/**
 * Returns an initialized instance of portAddDeleteFormServerListCG component.
 * @return the initialized component instance
 */
public ChoiceGroup getPortAddDeleteFormServerListCG() {
    if (portAddDeleteFormServerListCG == null) {
        // write pre-init user code here
        portAddDeleteFormServerListCG = new ChoiceGroup("Sunucular", Choice.MULTIPLE);
        portAddDeleteFormServerListCG.setSelectedFlags(new boolean[] { });
        // write post-init user code here
    }
    return portAddDeleteFormServerListCG;
}
//</editor-fold>
```

```
//<editor-fold defaultstate="collapsed" desc=" Generated Getter: portAddDeleteFormokC ">
/**
 * Returns an initialized instance of portAddDeleteFormokC component.
 * @return the initialized component instance
 */
public Command getPortAddDeleteFormokC() {
    if (portAddDeleteFormokC == null) {
        // write pre-init user code here
        portAddDeleteFormokC = new Command("G\u00F6nder", Command.OK, 0);
        // write post-init user code here
    }
    return portAddDeleteFormokC;
}
//</editor-fold>
//</editor-fold>
```

```
//</editor-fold>
//</editor-fold>
```

```
//<editor-fold defaultstate="collapsed" desc=" Generated Getter: portAddDeleteFormOptionCG ">
/**
 * Returns an initialized instance of portAddDeleteFormOptionCG component.
 * @return the initialized component instance
 */
public ChoiceGroup getPortAddDeleteFormOptionCG() {
    if (portAddDeleteFormOptionCG == null) {
        // write pre-init user code here
        portAddDeleteFormOptionCG = new ChoiceGroup("Se\u00E7enek (kap\u0131y\u0131
eri\u015Fime kapat ya da a\u00E7)", Choice.EXCLUSIVE);
        portAddDeleteFormOptionCG.append("Ekle", null);
        portAddDeleteFormOptionCG.append("Sil", null);
        portAddDeleteFormOptionCG.setSelectedFlags(new boolean[] { true, false });
        // write post-init user code here
    }
}
//</editor-fold>
```

```

    }
    return portAddDeleteFormOptionCG;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: aboutForm ">
/**
 * Returns an initialized instance of aboutForm component.
 * @return the initialized component instance
 */
public Form getAboutForm() {
    if (aboutForm == null) {
        // write pre-init user code here
        aboutForm = new Form("Hakk\u0131nda", new Item[] { getAboutii(), getAboutsi() });
        aboutForm.addCommand(getAboutFormback());
        aboutForm.setCommandListener(this);
        // write post-init user code here
    }
    return aboutForm;
}
//</editor-fold>
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: aboutii ">
/**
 * Returns an initialized instance of aboutii component.
 * @return the initialized component instance
 */
public ImageItem getAboutii() {
    if (aboutii == null) {
        // write pre-init user code here
        aboutii = new ImageItem("", getAboutI(), ImageItem.LAYOUT_DEFAULT, "<Missing
Image>", Item.PLAIN);
        // write post-init user code here
    }
    return aboutii;
}
//</editor-fold>
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: aboutsi ">
/**
 * Returns an initialized instance of aboutsi component.
 * @return the initialized component instance
 */
public StringItem getAboutsi() {
    if (aboutsisi == null) {
        // write pre-init user code here
        aboutsi = new StringItem("m2 G\u00FCvenlik Duvar\u0131 Y\u00F6netimi
Uygulamas\u0131", "Yafes Duygulutuna, Bahce\u015Fehir \u00DCniversitesi, Bilgi Teknolojileri,
Y\u00FCksek Lisans", Item.PLAIN);
        aboutsi.setLayout(ImageItem.LAYOUT_DEFAULT | Item.LAYOUT_TOP |
Item.LAYOUT_BOTTOM | Item.LAYOUT_VCENTER);
    }
}
//</editor-fold>
//</editor-fold>

```

```

        // write post-init user code here
    }
    return aboutsi;
}
//</editor-fold>
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: aboutFormback ">
/**
 * Returns an initialized instance of aboutFormback component.
 * @return the initialized component instance
 */
public Command getAboutFormback() {
    if (aboutFormback == null) {
        // write pre-init user code here
        aboutFormback = new Command("Geri", Command.BACK, 0);
        // write post-init user code here
    }
    return aboutFormback;
}
//</editor-fold>
//</editor-fold>
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: mainFormCG ">
/**
 * Returns an initialized instance of mainFormCG component.
 * @return the initialized component instance
 */
public ChoiceGroup getMainFormCG() {
    if (mainFormCG == null) {
        // write pre-init user code here
        mainFormCG = new ChoiceGroup("Se\u00E7enekler", Choice.EXCLUSIVE);
        mainFormCG.append("Kap\u0131 //ekle, sil", null);
        mainFormCG.append("Sunucu //aktif kap\u0131 listesi", null);
        mainFormCG.append("Sunucu //ekle, sil, d\u00FCzelt", null);
        mainFormCG.append("\u0130statistik", null);
        mainFormCG.append("Yard\u0131m", null);
        mainFormCG.append("Hakk\u0131nda", null);
        mainFormCG.addCommand(getMainFormGoC());
        mainFormCG.setItemCommandListener(this);
        mainFormCG.setDefaultCommand(getMainFormGoC());
        mainFormCG.setFitPolicy(Choice.TEXT_WRAP_DEFAULT);
        mainFormCG.setSelectedFlags(new boolean[] { true, false, false, false, false, false });
        // write post-init user code here
    }
    return mainFormCG;
}
//</editor-fold>
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: serversForm ">
/**

```

```

* Returns an initialized instance of serversForm component.
* @return the initialized component instance
*/
public Form getServersForm() {
    if (serversForm == null) {
        // write pre-init user code here
        serversForm = new Form("Sunucu Listesi", new Item[] { getServerFormServerListCG(),
getServerFormOptionCG(), getServerFormOSTypeCG(), getServerFormIPTF() });
        serversForm.addCommand(getServerFormbackC());
        serversForm.setCommandListener(this);
        // write post-init user code here
        getServers(serverFormServerListCG, "L"); // for List
    }
    return serversForm;
}
//</editor-fold>
//</editor-fold>
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: serverFormOSTypeCG ">
/**
* Returns an initialized instance of serverFormOSTypeCG component.
* @return the initialized component instance
*/
public ChoiceGroup getServerFormOSTypeCG() {
    if (serverFormOSTypeCG == null) {
        // write pre-init user code here
        serverFormOSTypeCG = new ChoiceGroup("\u0130\u015Fletim Sistemi T\u00FCr\u00FC",
Choice.POPUP);
        serverFormOSTypeCG.append("Linux", null);
        serverFormOSTypeCG.append("MS W7", null);
        serverFormOSTypeCG.append("MS XP", null);
        serverFormOSTypeCG.setFitPolicy(Choice.TEXT_WRAP_DEFAULT);
        serverFormOSTypeCG.setSelectedFlags(new boolean[] { false, false, false });
        // write post-init user code here
    }
    return serverFormOSTypeCG;
}
//</editor-fold>
//</editor-fold>
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: serverFormbackC ">
/**
* Returns an initialized instance of serverFormbackC component.
* @return the initialized component instance
*/
public Command getServerFormbackC() {
    if (serverFormbackC == null) {
        // write pre-init user code here
        serverFormbackC = new Command("Geri", Command.BACK, 0);
        // write post-init user code here
    }
}

```

```

    return serverFormbackC;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: portAddDeleteForm ">
/**
 * Returns an initialized instance of portAddDeleteForm component.
 * @return the initialized component instance
 */
public Form getPortAddDeleteForm() {
    if (portAddDeleteForm == null) {
        // write pre-init user code here
        portAddDeleteForm = new Form("Kap\u0131: EKLE, S\u0130L", new Item[] {
getPortAddDeleteFormServerListCG(), getPortAddDeleteFormOptionCG(),
getPortAddDeleteFormPortTF(), getPortAddDeleteFormResultSI() });
        portAddDeleteForm.addCommand(getPortAddDeleteFormbackC());
        portAddDeleteForm.setCommandListener(this);
        // write post-init user code here
        getServers(portAddDeleteFormServerListCG, "W"); // for Work
    }
    return portAddDeleteForm;
}
//</editor-fold>
//</editor-fold>
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: serverFormServerListCG ">
/**
 * Returns an initialized instance of serverFormServerListCG component.
 * @return the initialized component instance
 */
public ChoiceGroup getServerFormServerListCG() {
    if (serverFormServerListCG == null) {
        // write pre-init user code here
        serverFormServerListCG = new ChoiceGroup("Sunucu", Choice.EXCLUSIVE);
        serverFormServerListCG.setSelectedFlags(new boolean[] { });
        // write post-init user code here
    }
    return serverFormServerListCG;
}
//</editor-fold>
//</editor-fold>
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: serverFormOptionCG ">
/**
 * Returns an initialized instance of serverFormOptionCG component.
 * @return the initialized component instance
 */
public ChoiceGroup getServerFormOptionCG() {
    if (serverFormOptionCG == null) {
        // write pre-init user code here
        serverFormOptionCG = new ChoiceGroup("Se\u007enek", Choice.EXCLUSIVE);

```



```

        serverFormOptionCG.append("Ekle", null);
        serverFormOptionCG.append("Sil", null);
        serverFormOptionCG.append("D\u00FCzelt", null);
        serverFormOptionCG.setSelectedFlags(new boolean[] { true, false, false });
        // write post-init user code here
    }
    return serverFormOptionCG;
}
//</editor-fold>
//</editor-fold>
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: serverFormIPTF ">
/**
 * Returns an initialized instance of serverFormIPTF component.
 * @return the initialized component instance
 */
public TextField getServerFormIPTF() {
    if (serverFormIPTF == null) {
        // write pre-init user code here
        serverFormIPTF = new TextField("Sunucu IP Adresi", null, 32, TextField.ANY);
        serverFormIPTF.addCommand(getServerFormokC());
        serverFormIPTF.setItemCommandListener(this);
        // write post-init user code here
    }
    return serverFormIPTF;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: serverFormokC ">
/**
 * Returns an initialized instance of serverFormokC component.
 * @return the initialized component instance
 */
public Command getServerFormokC() {
    if (serverFormokC == null) {
        // write pre-init user code here
        serverFormokC = new Command("Tamam", Command.OK, 0);
        // write post-init user code here
    }
    return serverFormokC;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: activePortsForm ">
/**
 * Returns an initialized instance of activePortsForm component.
 * @return the initialized component instance
 */
public Form getActivePortsForm() {
    if (activePortsForm == null) {
        // write pre-init user code here

```

```

        activePortsForm = new Form("Sunucu Aktif Kap\u0131lar", new Item[] {
getActivePortsFormcG(), getActivePortsFormsi() });
        activePortsForm.addCommand(getActivePortsFormbackC());
        activePortsForm.addCommand(getActivePortsFormokC());
        activePortsForm.setCommandListener(this);
        // write post-init user code here
        getServers(activePortsFormcG, "L"); // for List
    }
    return activePortsForm;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: activePortsFormcG ">
/**
 * Returns an initialized instance of activePortsFormcG component.
 * @return the initialized component instance
 */
public ChoiceGroup getActivePortsFormcG() {
    if (activePortsFormcG == null) {
        // write pre-init user code here
        activePortsFormcG = new ChoiceGroup("Sunucu", Choice.EXCLUSIVE);
        activePortsFormcG.append("Test", null);
        activePortsFormcG.setSelectedFlags(new boolean[] { false });
        // write post-init user code here
    }
    return activePortsFormcG;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: activePortsFormsi ">
/**
 * Returns an initialized instance of activePortsFormsi component.
 * @return the initialized component instance
 */
public StringItem getActivePortsFormsi() {
    if (activePortsFormsi == null) {
        // write pre-init user code here
        activePortsFormsi = new StringItem("Sonu\u00E7 :", null);
        // write post-init user code here
    }
    return activePortsFormsi;
}
//</editor-fold>
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: activePortsFormbackC ">
/**
 * Returns an initialized instance of activePortsFormbackC component.
 * @return the initialized component instance
 */
public Command getActivePortsFormbackC() {
    if (activePortsFormbackC == null) {
        // write pre-init user code here

```

```

        activePortsFormbackC = new Command("Geri", Command.BACK, 0);
        // write post-init user code here
    }
    return activePortsFormbackC;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: activePortsFormokC ">
/**
 * Returns an initialized instance of activePortsFormokC component.
 * @return the initialized component instance
 */
public Command getActivePortsFormokC() {
    if (activePortsFormokC == null) {
        // write pre-init user code here
        activePortsFormokC = new Command("Tamam", Command.OK, 0);
        // write post-init user code here
    }
    return activePortsFormokC;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: statisticsForm ">
/**
 * Returns an initialized instance of statisticsForm component.
 * @return the initialized component instance
 */
public Form getStatisticsForm() {
    if (statisticsForm == null) {
        // write pre-init user code here
        statisticsForm = new Form("\u0130statistik", new Item[] { getAddRuleCountSI(),
getAddRuleAverageDurationSI(), getDeleteRuleCountSI(), getDeleteRuleAverageDurationSI(),
getServerPortListingCountSI(), getServerPortListingAverageDurationSI(), getStringItem() });
        statisticsForm.addCommand(getStatisticsFormbackC());
        statisticsForm.addCommand(getStatisticsFormUpdateC());
        statisticsForm.setCommandListener(this);
        // write post-init user code here
    }
    return statisticsForm;
}
//</editor-fold>
//</editor-fold>
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: addRuleCountSI ">
/**
 * Returns an initialized instance of addRuleCountSI component.
 * @return the initialized component instance
 */
public StringItem getAddRuleCountSI() {
    if (addRuleCountSI == null) {
        // write pre-init user code here
        addRuleCountSI = new StringItem("Bloklanan Kural Say\u0131s\u0131 :", null);
    }
}
//</editor-fold>

```

```

        // write post-init user code here
    }
    return addRuleCountSI;
}
//</editor-fold>
//</editor-fold>
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: addRuleAverageDurationSI ">
/**
 * Returns an initialized instance of addRuleAverageDurationSI component.
 * @return the initialized component instance
 */
public StringItem getAddRuleAverageDurationSI() {
    if (addRuleAverageDurationSI == null) {
        // write pre-init user code here
        addRuleAverageDurationSI = new StringItem("Ortalama Bloklama S\u00FCresi :", null);
        // write post-init user code here
    }
    return addRuleAverageDurationSI;
}
//</editor-fold>
//</editor-fold>
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: deleteRuleCountSI ">
/**
 * Returns an initialized instance of deleteRuleCountSI component.
 * @return the initialized component instance
 */
public StringItem getDeleteRuleCountSI() {
    if (deleteRuleCountSI == null) {
        // write pre-init user code here
        deleteRuleCountSI = new StringItem("Silinen Kural Say\u0131s\u0131 :", null);
        // write post-init user code here
    }
    return deleteRuleCountSI;
}
//</editor-fold>
//</editor-fold>
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: deleteRuleAverageDurationSI ">
/**
 * Returns an initialized instance of deleteRuleAverageDurationSI component.
 * @return the initialized component instance
 */
public StringItem getDeleteRuleAverageDurationSI() {
    if (deleteRuleAverageDurationSI == null) {
        // write pre-init user code here
        deleteRuleAverageDurationSI = new StringItem("Ortalama Silinme S\u00FCresi :", null);
        // write post-init user code here
    }
}

```

```

    return deleteRuleAverageDurationSI;
}
//</editor-fold>
//</editor-fold>
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: serverPortListingCountSI ">
/**
 * Returns an initialized instance of serverPortListingCountSI component.
 * @return the initialized component instance
 */
public StringItem getServerPortListingCountSI() {
    if (serverPortListingCountSI == null) {
        // write pre-init user code here
        serverPortListingCountSI = new StringItem("Sunucu Kap\u0131 Listeleme Say\u0131s\u0131");
    }
    // write post-init user code here
}
return serverPortListingCountSI;
}
//</editor-fold>
//</editor-fold>
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter:
serverPortListingAverageDurationSI ">
/**
 * Returns an initialized instance of serverPortListingAverageDurationSI component.
 * @return the initialized component instance
 */
public StringItem getServerPortListingAverageDurationSI() {
    if (serverPortListingAverageDurationSI == null) {
        // write pre-init user code here
        serverPortListingAverageDurationSI = new StringItem("Ortalama Listeleme S\u00FCresi :");
    }
    // write post-init user code here
}
return serverPortListingAverageDurationSI;
}
//</editor-fold>
//</editor-fold>
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: statisticsFormbackC ">
/**
 * Returns an initialized instance of statisticsFormbackC component.
 * @return the initialized component instance
 */
public Command getStatisticsFormbackC() {
    if (statisticsFormbackC == null) {
        // write pre-init user code here
        statisticsFormbackC = new Command("Geri", Command.BACK, 0);
    }
    // write post-init user code here
}
}
//</editor-fold>
//</editor-fold>
//</editor-fold>

```

```

    }
    return statisticsFormbackC;
}
//</editor-fold>
//</editor-fold>
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: statisticsFormUpdateC ">
/**
 * Returns an initialized instance of statisticsFormUpdateC component.
 * @return the initialized component instance
 */
public Command getStatisticsFormUpdateC() {
    if (statisticsFormUpdateC == null) {
        // write pre-init user code here
        statisticsFormUpdateC = new Command("Tamam", "G\u00FCncelle", Command.OK, 0);
        // write post-init user code here
    }
    return statisticsFormUpdateC;
}
//</editor-fold>
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: splashl ">
/**
 * Returns an initialized instance of splashl component.
 * @return the initialized component instance
 */
public Image getSplashl() {
    if (splashl == null) {
        // write pre-init user code here
        try {
            splashl = Image.createImage("/splash1.jpg");
        } catch (java.io.IOException e) {
            e.printStackTrace();
        }
        // write post-init user code here
    }
    return splashl;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: helpForm ">
/**
 * Returns an initialized instance of helpForm component.
 * @return the initialized component instance
 */
public Form getHelpForm() {
    if (helpForm == null) {
        // write pre-init user code here
        helpForm = new Form("Yard\u0131m", new Item[] { getHelpFormPortAddvsSI(),
getHelpFormServerPortvsSI(), getHelpFormServerAddvsSI(), getHelpFormStatisticsSI(),
getHelpFormHelpSI(), getHelpFormAboutSI() });
    }
}

```

```

        helpForm.addCommand(getHelpFormbackC());
        helpForm.setCommandListener(this);
        // write post-init user code here
    }
    return helpForm;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: helpFormPortAddvsSI ">
/**
 * Returns an initialized instance of helpFormPortAddvsSI component.
 * @return the initialized component instance
 */
public StringItem getHelpFormPortAddvsSI() {
    if (helpFormPortAddvsSI == null) {
        // write pre-init user code here
        helpFormPortAddvsSI = new StringItem("Kap\u0131 //ekle, sil :", "Bir ya da birden fazla
sunucuda kap\u0131 bloklamas\u0131 yap\u0131lmas\u0131.");
        // write post-init user code here
    }
    return helpFormPortAddvsSI;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: helpFormServerPortvsSI ">
/**
 * Returns an initialized instance of helpFormServerPortvsSI component.
 * @return the initialized component instance
 */
public StringItem getHelpFormServerPortvsSI() {
    if (helpFormServerPortvsSI == null) {
        // write pre-init user code here
        helpFormServerPortvsSI = new StringItem("Sunucu //aktif kap\u0131 listesi : ", "Sunucuya
ait hangi kap\u0131lar\u0131n a\u00E7\u0131k oldu\u011Fu bilgisini getirir.");
        // write post-init user code here
    }
    return helpFormServerPortvsSI;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: helpFormServerAddvsSI ">
/**
 * Returns an initialized instance of helpFormServerAddvsSI component.
 * @return the initialized component instance
 */
public StringItem getHelpFormServerAddvsSI() {
    if (helpFormServerAddvsSI == null) {
        // write pre-init user code here
        helpFormServerAddvsSI = new StringItem("Sunucu //ekle, sil, d\u00FCzelt",
"D\u00FCzlemeler aras\u0131na yeni sunucu eklenmesi, sunucu silinmesi ya da
eklenmi\u015F sunucunun IP adresinin d\u00FCztilmesi i\u015Flemleri yap\u0131lmas\u0131.");
        // write post-init user code here
    }
}

```

```

    return helpFormServerAddvsSI;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: helpFormStatisticsSI ">
/**
 * Returns an initialized instance of helpFormStatisticsSI component.
 * @return the initialized component instance
 */
public StringItem getHelpFormStatisticsSI() {
    if (helpFormStatisticsSI == null) {
        // write pre-init user code here
        helpFormStatisticsSI = new StringItem("\u0130statistik : ", "Kap\u0131 ekleme, silme, aktif
kap\u0131 listesi s\u00FCresel istatistiki bilgileri bulunur.");
        // write post-init user code here
    }
    return helpFormStatisticsSI;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: helpFormHelpSI ">
/**
 * Returns an initialized instance of helpFormHelpSI component.
 * @return the initialized component instance
 */
public StringItem getHelpFormHelpSI() {
    if (helpFormHelpSI == null) {
        // write pre-init user code here
        helpFormHelpSI = new StringItem("Yard\u0131m : ", "Yard\u0131m penceresi.");
        // write post-init user code here
    }
    return helpFormHelpSI;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: helpFormAboutSI ">
/**
 * Returns an initialized instance of helpFormAboutSI component.
 * @return the initialized component instance
 */
public StringItem getHelpFormAboutSI() {
    if (helpFormAboutSI == null) {
        // write pre-init user code here
        helpFormAboutSI = new StringItem("Hakk\u0131nda : ", "Program ni\u00E7in ve kim
taraf\u0131ndan yaz\u0131ld\u0131\u011F\u0131 ile ilgili bilgiler bulunur.");
        // write post-init user code here
    }
    return helpFormAboutSI;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: helpFormbackC ">
/**

```



```

* Returns an initialized instance of helpFormbackC component.
* @return the initialized component instance
*/
public Command getHelpFormbackC() {
    if (helpFormbackC == null) {
        // write pre-init user code here
        helpFormbackC = new Command("Geri", Command.BACK, 0);
        // write post-init user code here
    }
    return helpFormbackC;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: stringItem ">
/**
* Returns an initialized instance of stringItem component.
* @return the initialized component instance
*/
public StringItem getStringItem() {
    if (stringItem == null) {
        // write pre-init user code here
        stringItem = new StringItem("-----",
"De\u011Ferler milisaniye cinsindedir.");
        // write post-init user code here
    }
    return stringItem;
}
//</editor-fold>

/**
* Returns a display instance.
* @return the display instance.
*/
public Display getDisplay () {
    return Display.getDisplay(this);
}

/**
* Exits MIDlet.
*/
public void exitMIDlet() {
    switchDisplayable (null, null);
    destroyApp(true);
    notifyDestroyed();
}

/**
* Called when MIDlet is started.
* Checks whether the MIDlet have been already started and initialize/starts or resumes the
MIDlet.
*/
public void startApp() {
    if (midletPaused) {

```

```

        resumeMIDlet ();
    } else {
        initialize ();
        startMIDlet ();
    }
    midletPaused = false;
}

/**
 * Called when MIDlet is paused.
 */
public void pauseApp() {
    midletPaused = true;
}

/**
 * Called to signal the MIDlet to terminate.
 * @param unconditional if true, then the MIDlet has to be unconditionally terminated and all
resources has to be released.
 */
public void destroyApp(boolean unconditional) {
}
}

```

ÖZGEÇMİŞ

Adı Soyadı : Yafes Murat Duygulutuna

Sürekli Adresi : Feyzullah Mh. Serap Cd. Nilüfer Sk. Özay Apt. No: 6 Daire 14 Maltepe
İstanbul, Türkiye

Doğum Yeri ve Yılı : Ankara, 1975

Yabancı Dili : İngilizce (iyi), Almanca (orta)

İlk Öğretim : Nene Hatun İlkokulu 1987, Balgat Ortaokulu 1990

Orta Öğretim : Balgat Teknik ve Endüstri Meslek Lisesi 1993

Lisans : Kırıkkale Üniversitesi 1995, Anadolu Üniversitesi 2006

Yüksek Lisans : Bahçeşehir Üniversitesi - (devam ediyor)

Enstitü Adı : Fen Bilimleri Enstitüsü

Program Adı : Bilgi Teknolojileri

Yayımları : Mobil Gelecek ve Güvenlik, Turkcell Akademi Dergisi, Mayıs 2008

Çalışma Hayatı : Turkcell 2001 – devam ediyor, Mynet 1999 – 2001, AdaNet 1997- 1999,
Kalem Mühendislik 1995 - 1997