

Güvenlik Duvarı Kurallarına Birim Test Yöntemi Uygulanması

Remzi Emre Başar, Can Burak Çilingir

İstanbul Bilgi Üniversitesi (27 Ocak 2007)

Özet: Bir yazılım geliştirme metodolojisi olarak ortaya çıkmış olmasına karşın birim testleri bilişim dünyasının farklı alanlarında da uygulanabilirlik sunarlar. Tasarımın genişlediği, karmaşıklaştığı ve hata takibinin zorlaştığı birçok durumda birim testler kullanılarak tasarım örneği oluşturacak bir belgelendirme üretilirken ayrıca tasarımda yapılacak olan değişikliklerin sistemin geri kalanının çalışmasını etkilemesine engel olunabilir. Birim testlerin getirdiği bu avantaj özellikle ağ güvenliği gibi hassas bir konuda kullanılarak güvenlik duvarı tasarımında ortaya çıkabilecek hatalar ve bu hatalardan kaynaklanabilecek güvenlik açıkları büyük ölçüde bertaraf edilebilir. Doğru tasarlanmış birim testler kullanılarak karmaşık ve heterojen ağlarda dahi güvenlik duvarı mimarileri sağlıklı bir şekilde oluşturulabilir.

İçindekiler

- 1 Giriş
- 2 Akademik Ortamlarda Güvenlik Pol.
 - 2.1 Akademik Kadro
 - 2.2 öğrenciler
 - 2.3 Diğer Kullanıcılar
 - 2.4 Sunucuların "Özel İhtiyaçları
- 3 Birim Testleri
 - 3.1 Yazılım testleri
 - 3.1.1 Yazılım testleri geçmişi
 - 3.1.2 Yazılım testi aşamaları
 - 3.2 Uyarılma
- 5 Test yöntemi
 - 5.1 Kullanılan yazılımlar
- 6 Gelecek planları
 - 6.1 Bant genişliği kontrolü
 - 6.2 Paket üretici kullanımı
 - 6.3 Nagios entegrasyonu
- 7 Sonuç

1 Giriş

İnternet, ilk ortaya çıktığı yıllardan başlayarak güven temeline oturan protokoller üzerine inşa edilmiştir. İnternet'in ortaya çıktığı askeri ve akademik ortamlarda bu güven durumu akademik dünyanın ihtiyaçları ile aynı doğrultuda bir hareket olduğundan, bu protokollerdeki güvenlik eksikliği İnternet'in askeri ve akademik ortamlar dışında yaygınlaşmasına kadar farke-

dilememiştir. İnternet'in güvene dayalı doğasının değişmesi ve bunun yerini dışarıdan gelecek saldırılara karşı gerekli önlemleri almayı mecburi kılan güvensizlik temelli bir anlayışın ortaya çıkması ile güvenlik duvarı kavramı hayatımıza girmiştir. önce büyük servis sağlayıcıların veri merkezlerinde, ardından teker teker sunucuların üzerinde görülmeye başlanan güvenlik duvarı uygulamaları, günümüzde masaüstü bilgisayarlarda dahi kullanılmaktadır. Güvenlik duvarının işlevi, bir sistem ile bağlı olduğu ağ arasında gidip gelen veri paketlerini kaynak adresi, hedef adresi, hedef port numarası ve benzeri kriterlere dayanarak değerlendirmek ve bu değerlendirme sonucunda söz konusu veri paketine önceden belirlenmiş bir kurallar dizisini uygulamaktır. Bu kurallar veri paketini kabul veya reddetmek gibi basit kurallar olabileceği gibi, paketi diğer bir sisteme iletmek veya özniteliklerini değiştirmek gibi daha karmaşık işlemler de içerebilir. Güvenlik duvarının iç işleyişine bağlı olarak bu kuralların uygulanış biçimi farklılık gösterebilir.

Her ne kadar İnternet'in ortaya çıktığı dönemde akademik ortamlarda hakim olan güven hissiyatı şu anda kullandığımız ağ protokollerinin güvene dayalı olarak tasarlanmasına sebep olacak kadar yoğunsa da günümüzde akademik kurumlar da en az diğer kurumlar kadar İnternet üzerinden gelecek saldırılar için hedef teş-

kil etmektedir. Bunun yanında, akademik ortamlarda kişilerin rollerinden (öğrenci, öğretim görevlisi, v.b.) kaynaklanan güvenlik politikası farklılıkları ve yük dengeleme gibi ihtiyaçlar da akademik ortamlarda güvenlik duvarlarının kullanımını mecburi kılmaktadır.

2 Akademik Ortamlarda Güvenlik Politikaları

Güvenlik duvarı politikaları, uygulanacağı yerin niteliğine göre farklılık gösterir. Bir ev kullanıcısının güvenlik politikası ile ticari bir kuruluşun, ya da akademik bir kurumun güvenlik ihtiyaçları birbirinden çok farklı olacaktır. Güvenlik politikası bu ihtiyaçlar göz önünde bulundurularak biçimlendirilir ve güvenlik duvarı kuralları bu politikaları yansıtacak biçimde ayarlanır.

Akademik bir kurum içerisinde, kişilerin İnternet'ten ve yerel ağ üzerinde sunulan hizmetlerden faydalanma biçimleri kurum içerisindeki rollerine bağlı olarak belirgin farklılıklar gösterir. Bu nedenle her bir grup için uygulanacak güvenlik kuralları belirlenirken o grubun kullanım biçimi göz önünde bulundurulup, bu kullanım biçimine uygun bir politika tasarlanmalıdır. Bu grupları ve uygulanacak örnek politikaları bir gözden geçirmek faydalı olacaktır.

2.1 Akademik Kadro

Akademik alanda yapılan çalışmaların doğası gereği, akademik çalışmalar ile uğraşan kullanıcıların normalden daha geniş bir alanda erişim ihtiyaçları söz konusudur. Bunun yanında akademik kadronun kendi içinde iletişimi ve birlikte çalışma ihtiyacı sebebiyle yerel ağ üzerinde de bazı servislerin sunulması da mümkündür. Bu tip servisler üzerinde dışarıdan erişilmemesi gereken bilgilerin bulunması durumunda bu servislerin de sadece akademik kadro tarafından erişilebilir kılınması son derece önemlidir.

2.2 Öğrenciler

Akademik ağlar üzerinde öğrencilerin ağ hizmetleri ve İnternet'ten faydalanmaları okul/bölüm bünyesinde kurulan bilgisayar laboratuvarları aracılığıyla mümkün olmaktadır. öğrencilerin ağ üzerindeki etkinlikleri genellikle okul tarafından öğrencilere yönelik olarak sunulan dosya depolama, e-posta ve benzeri servisler ile İnternet'te, web üzerindeki kaynaklara erişim şeklinde ortaya çıkmaktadır. Bazı özel durumlar haricinde öğrencilerin web dışındaki bilgi edinme kaynaklarından çok fazla faydalanmadıkları bilinmektedir.

Bu gibi olağan kullanımların yanında öğrenciler tarafından kullanılan sistemlerin eğitim dışındaki amaçlarla, dosya paylaşım hizmetlerine ve anında mesajlaşma sistemlerine bağlanmak veya dışarıya yönelik saldırılar gerçekleştirmek amacıyla da kullanılabildiği bilinen bir gerçektir. Bu nedenle öğrenciler tarafından kullanılan sistemlerin özellikle gerek yerel ağ içerisinde, gerekse İnternet üzerindeki kaynaklara erişimi konusunda koyulacak kuralların bu tip kötüye kullanımlara engel olabilecek biçimde kısıtlayıcı olması gerekmektedir.

2.3 Diğer Kullanıcılar

Üniversitelerin akademik çalışmaları içerisinde yaptıkları bir diğer uygulama da çeşitli toplantı ve konferansların düzenlenmesidir. Özellikle günümüzde yaygınlaşan taşınabilir ve kablo-suz teknolojiler neticesinde bu tip etkinliklere katılan konukların İnternet bağlantısı ihtiyacı gibi normal şartlarda o ağa bağlı olmayan kişileri de geçici olarak ağa dahil etme ihtiyacı ortaya çıkmıştır. Özellikle genel katılıma tamamen açık olan ortamlarda bu durum herkesin erişimine tamamen açık bir ağ işletmek ile eşit derecede risk içerir. Bu nedenle özellikle yetkisiz birinin kurumun ağ bağlantısı üzerinden dışarıya gerçekleştirebileceği saldırılara karşı önlem alınması şarttır.

2.4 Sunucuların Özel İhtiyaçları

Bunların yanında akademik bir kurumda İnternet kullanımı tek yönlü değildir. Yerel ağdan

dışarıya yapılan bağlantılar söz konusu olduğu gibi hemen hemen bütün akademik kurumlar dışarıya yönelik FTP, web ve e-posta gibi servisler sunarlar. Bu servislere yapılacak bağlantıların da yerel ağ üzerinde bu servisleri sağlayan sistemlere yönlendirilmesi, iç ağa yapılacak doğrudan bağlantıların engellenmesi gibi görevlerin de yerine getirilmesi gerekmektedir.

Bütün bu ihtiyaçlar göz önüne alındığında özellikle ağ üzerinde sunulan servislerin ve bu servisleri kullanan kullanıcıların sayısı arttıkça güvenlik duvarı kurallarının yapısı da karmaşıklaşmaktadır. Bu karmaşıklığın en önemli etkisi, yeni eklenen kuralların hali hazırda çalışmakta olan sistemin tutarlılığını bozup bozmadığının takip edilmesini gerekli kılmıştır.

3 Birim Testleri

Birim testleri, yazılım doğruluk kontrolü yöntemlerinden birisidir. Birim testleri tanımını ve bu makalede ne bağlamda işimize yarayacağını tartışmadan önce, yazılım testleri tanımı ve tarihine göz atmak yararlı olacaktır.

3.1 Yazılım testleri

Yazılımların üretilme sebeplerinin başında müşteri istekleri geldiğinden testi bu bakış açısı ile değerlendirmeye çalışacağız. Testler ile 2 ana derdin önüne geçilmeye çalışılır.

1. Yazılımın kullanıcının umduğu işi yapmaması.
2. Yazılımın tasarlanmadığı şekilde çalışması.

İlk durum tasarım sorunu iken diğer durum yazılım sorunlarına, belki de daha doğru olarak teknik sorunlara denk gelmektedir. Yazılım testlerinin amaçlarından bir tanesi özellikle ikinci gruptaki sorunları ortadan kaldırmaktır.

3.1.1 Yazılım testleri geçmişi

... - 1956 Hata ayıklama ile yazılım testi arasında net bir ayırım yoktu.

1957 - 1978 Hata ayıklama ve yazılım testi kavramları ayrı ayrı işlenmeye başladı. Yazılımın gereksinimlere uyduğu yazılım testleri ile kontrol edilmeye başlanmıştır.

1979 - 1982 Yazılım geliştirmede temel amaç hata ayıklaması haline gelmiştir.

1983 - 1987 Yazılımın kullanım süresi boyunca kullanımı gözetlenmiş ve kalite testlerine tabi tutulmuştur.

1988 - günümüz Yazılım testleri, yazılımın belirtilmelere uyduğunu test etmenin yanı sıra, hataları bulma ve hataları engelleme gibi amaçlara hizmet etmeye başlamıştır. Günümüz yazılım test kültürünün temel dayanak noktaları olan IEEE'nin test dokümanları standardı² ve "The Complete Guide of Software Testing" kitabının üretimi bu döneme denk gelir.

3.1.2 Yazılım testi aşamaları

Yazılım testlerini kabaca 4 aşamaya ayırabiliriz.

1. Birim testler ile sistemi oluşturan ufak parçaların kendi içlerinde testi
2. Ayrı parçaların birbirleri ile etkileşimlerinin/birleşmelerinin testi
3. Parçalar birleştikten sonra ortaya çıkan bütün yazılımın gereksinimlere uygunluk testi
4. Üretilen yazılım müşteri tarafından testi. Bu aşamada firma içinde ilk testler için alpha sürümü üretilebilir, ardından müşteri testi için beta sürümleri üretilebilir.

Bu makale, bahsi geçen yöntemlerin sadece ilk aşaması ile, yani birim testler ile ilgileniyor.

3.2 Uyarılama

Birim testi aşaması için öncelikle test durumları ve bu durumdaki beklentilerin gerçekleşip gerçekleşmediği test edilir. örneğin, çok basit bir test durumu, belli bir butona basıldığında ekrana "Ad Soyad" içeren bir uyarı çıkıp çıkmadığını kontrol edebilir. Bu durum, mesela IEEE 829-1988 (Test Documentation Standard)

1 http://en.wikipedia.org/wiki/Software_testing

bir insan tarafından, butona tıklanarak ilgili uyarının çıkması gözlemlenerek çalıştırılabilir.

Test durumları kağıt üzerinde tanımlanıp elle kontrol edilebileceği gibi, uygun durumlarda bilgisayar ya da otomatik başka bir sistem tarafından da çalıştırılabilir. örneği basite indirgeyecek olursak, bir sayıyı ikiye bölen bir fonksiyonu test ederken kullanacağımız test durumları, “6 ile çağrıldığında 3”, “10 ile çağrıldığında 5” sonuçlarını kontrol edecektir ve bu tarz testleri de bilgisayar yardımı ile otomatik olarak yapmak mümkün olacaktır.

Birim testler için gerekli test durumları belirlendikten sonra, yazılım, bu makale tabanında düşünürsek güvenlik duvarı kurallarında rahatça yeniden düzenlemeye ya da kural değiştirmeye gidilebilir. Eğer yeni eklenen kod/kural eski kurallardan birisini bozacak olur ise, test durumlarından bir tanesi çalışmayacak ve hatanın farkına varma süresi en aza çekilecektir. Bunun sonucunda bu hatanın yol açacağı maddi kayıp ve iş yapamama durumu ortadan kalkacaktır.

4 Uygulama Detayları

Önceki kısımlarda da açıklandığı gibi güvenlik duvarlarının temel amaçları bir veri paketinin kaynağı ve hedefi arasında bir noktada durarak belirlenen kural dizisine uygun biçimde paketlerin geçişini kontrol altında tutmaktır. Bu davranışı bir birim test sistemi içerisinde test etmek için sistemimizin ki parçadan oluşması gerekir. İlk parçamız aradaki bağlantı testini yapacak olan uygulama motorudur. Uygulamanın bu parçası iki sistem arasında bağlantı kurulup kurulmadığını ve test sonucunun beklenen sonuç ile uyup uyumadığını kontrol eder.

Uygulamanın ikinci parçası ise test edilecek kuralların tanımlandığı ayar dosyasıdır. Bu dosyada sistemler arasında test edilmesi istenen bağlantılar ve beklenen sonuçlar listelenir. Bu kurallar dizisi birim test uygulaması için gerekli bilgiyi sağladığı gibi güvenlik duvarı politikasının belgelenmesi amacıyla da kullanılabilir.

5 Test Yöntemi

Bu tip bir uygulama için kullanılabilecek en basit test yöntemi ayar dosyasında belirtilen servislere bağlanmaya çalışmak ve sonucu kontrol etmektir. Bu yöntem her ne kadar tek noktadan diğer yerlere yapılacak bağlantıları test etmekte işe yararsa da özellikle birden fazla adres bloğu içerecek şekilde parçalanmış olan ağlarda bu yöntem yetersiz kalacaktır. Parçalı bir ağ yapısında ağın bazı bölümlerinin çeşitli servislere erişmesi bazı bölümlerinin ise erişmemesi istenen durumlarla sıkça karşılaşılır.

Birim testleri ağın farklı bölümleri için ayrı ayrı tanımlamak ve o yerlerden çalıştırmak özellikle karmaşık ağ yapılarında pratiklikten oldukça uzak bir durumdur. Bunun yerine birim test uygulamasının çalıştırıldığı sistemden diğer sistemlere de erişerek bu sistemler üzerinde çalışacak olan birim testleri de yönetebilmesi gerekir.

Bu bilgiler ışığında bir güvenlik duvarı birim test uygulamasının test metodu şu şekilde özetlenebilir:

- Bulunduğu sistemden dışarıdaki servislere bağlanmayı dener
- Bağlantı denemesinin sonucunu beklenen sonuç ile karşılaştırır
- Diğer ağlardaki sistemlere bağlanarak bu sistemler üzerinden diğer ağlara yapılan bağlantıların kurallara uyup uymadığını test eder.

6 Gelecek Planları

Bu makalenin de konusunu oluşturan ve yukarıda anlattığımız araçlar kullanılarak geliştirdiğimiz birim test uygulamasımız şu anda Bilgi Üniversitesi Bilgisayar Bilimleri bölümünün güvenlik duvarı altyapısının test edilmesi amacıyla aktif olarak kullanılmaktadır. Uygulamanın tam teşekküllü bir birim test altyapısına dönüşmesi için ise daha bazı geliştirmelere ihtiyaç vardır. Bu geliştirmelerden bazıları aracın temel

test fonksiyonlarını geliştirmeye, diğerleri ise kullanım kolaylığı sağlamaya yöneliktir.

6.1 Bant genişliği kontrolü

Bugünkü haliyle, uygulama sistemler arasındaki bağlantı kurallarını test edebilmekle beraber bant genişliği kontrolüne yönelik kısıtlamalar üzerinde bir test yapma yeteneğinden uzaktır. Doğrudan adres bazında yapılacak bant genişliği kısıtlamaları üzerinde çalışan bir test geliştirmek görece kolaysa da servis bazında uygulanan bant genişliği kısıtlamalarını test edecek bir yapı test edilecek her servis için o servise özel ilgi gerektireceğinden gerçekleştirilmesi göreceli olarak daha zordur.

6.2 Paket üretici kullanımı

Şu anki birim test yöntemimizin yetersiz kaldığı diğer bir nokta ise özellikle bozulmuş paketler kullanılarak yapılan saldırıları karşılamak üzere geliştirilmiş olan güvenlik duvarı kurallarını test etmekte yetersiz kalmasıdır. Şu anki testler sadece bağlantıların gerçekleşip gerçekleşmemesi temeline dayandığından bozuk paketlere karşı güvenlik duvarının verdiği tepkiler test edilememektedir. Bu amaçla çeşitli tiplerde bozuk veri paketleri üretimini kolaylaştıran hping veya scapy gibi araçların kullanma dahil edilmesi planlanmaktadır.

6.3 Nagios entegrasyonu

Özellikle birden fazla kişi tarafından yönetilen ve kuralların dinamik olarak eklenip çıkartıldığı güvenlik duvarı ortamlarında her kural değişiminin ardından testleri çalıştırıp sonuçları incelemektense test işlemini otomatikleştirip Nagios gibi bir ağ izleme aracı yardımıyla kuralların işlenmesinde meydana gelen aksaklıkları kontrol etmek daha kolay olacaktır. Bu amaçla Nagios'un send nsca programının yardımıyla birim test aracının test sonuçlarını ekrana göndermek yerine Nagios'a bildirmesi sağlanabilir.

7 Sonuç

Bir yazılım geliştirme metodolojisi olarak ortaya çıkmış olmasına karşın birim testleri bilişim dünyasının farklı alanlarında da uygulanabilirlik sunarlar. Tasarımın genişlediği, karmaşıklığı ve hata takibinin zorlaştığı birçok durumda birim testler kullanılarak tasarım örneği oluşturacak bir belgelendirme üretilirken ayrıca tasarımda yapılacak olan değişikliklerin sistemin geri kalanının çalışmasını etkilemesine engel olunabilir.

Birim testlerin getirdiği bu avantaj özellikle ağ güvenliği gibi hassas bir konuda kullanılarak güvenlik duvarı tasarımında ortaya çıkabilecek hatalar ve bu hatalardan kaynaklanabilecek güvenlik açıkları büyük ölçüde bertaraf edilebilir. Doğru tasarlanmış birim testler kullanılarak karmaşık ve heterojen ağlarda dahi güvenlik duvarı mimarileri sağlıklı bir şekilde oluşturulabilir.