Kurumsal Web Güvenliği Yapısı

Enis KARAARSLAN[1], Tuğkan TUĞLULAR[2], Halil ŞENGONCA[3]

[1] Ege Üniversitesi, Bilgi İletişim ve Teknolojileri Araştırma Merkezi, İzmir

[2] İYTE, Bilgisayar Mühendisliği Bölümü, İzmir

[3] Ege Üniversitesi, Bilgisayar Mühendisliği Bölümü, İzmir

enis.karaarslan@ege.edu.tr, tugkantuglular@iyte.edu.tr, halil.sengonca@ege.edu.tr

Özet: Web altyapısına artan sayıda saldırı girişimi yaşanmaktadır, bu nedenle web ve web uygulaması güvenliği her geçen gün daha hayati hale gelmektedir. Nüfuz veya saldırı yaşanmadan saldırıları saptayacak ve saldırıya açıklıkları engelleyecek güvenlik düzeneklerine ihtiyaç duyulmaktadır. Bu çalışmada, güvenliği daha iyi sağlamak için değişik tekniklerin birlikte çalıştığı bir Kurumsal Web Güvenlik Altyapısı modeli tanımlanmıştır. Bu modelde, ağ farkındalığı ve eğitim konularına yoğunlaşılmıştır.

Anahtar Sözcükler: Web Güvenliği, Web Uygulama Güvenliği, Ağ Farkındalığı, Web Sistem Farkındalığı, Çok Katmanlı Güvenlik.

Enterprise Wide Web Security Infrastructure

Abstract: There is increasing number of intrusion attempts to the web infrastructure, so web and web application security are becoming more vital everyday. There is need for security measures, which will detect and prevent vulnerabilities before intrusion and attack occur. In this work, an Enterprise-wide Web Security Infrastructure model where different techniques work cooperatively to achieve better security is defined. Network awareness and training are focused on.

Keywords: Web Security, Web Application Security, Network Awareness, Web System Awareness, Multi Layer Security.

1. Giriş

Üniversite ağları gibi büyük kurumsal ağlarda, farklı web sistemleri, bunları yöneten ve üzerindeki yazılımları hazırlayan farklı ekipler bulunmaktadır. Bu sistemlerin güvenliğini sağlamak için daha kapsamlı sistemlere gereksinim duyulmaktadır.

Güvenlik önlemleri, hiçbir zaman mükemmel değildir ve her güvenlik önleminin bazı zayıflıkları bulunabilmektedir. "Bir zincir ancak en zayıf halkası kadar güçlüdür." sözünün de belirttiği üzere, güvenliğin sağlanması için zayıf noktalardan doğacak sorunların mümkün olduğu kadar çözülmesi gerekmektedir. Bu da birbirini tamamlayan ve birlikte etkileşimli çalışan güvenlik sistemleri ile mümkündür. Bu tür bir yapıya çok katmanlı güvenlik ve kademeli savunma (defense in depth) denilmektedir. Sel suyunu engelleyen ardışık bentler gibi; her katman, bir sonraki katmana geçilmeden önce sorunun bir kısmını çözmüş olacaktır [8].

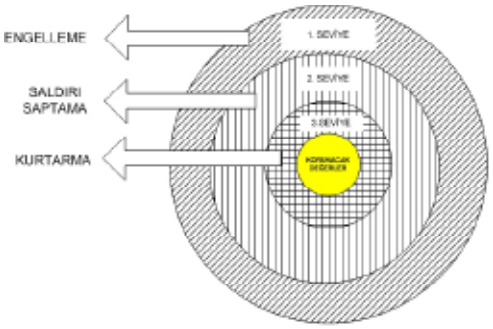
Güvenlik için gereken süreçleri tanımlarken, farklı ve birbirini tamamlayan işlevlere ait çok katmanlı güvenlik sistemlerinden söz edilmektedir. Katman yapısını, kurulacak güvenlik sistemlerinin özelliğine göre farklılaştırmak ve her katmanda alt katmanlar kullanmak mümkündür. Genel olarak üç katmandan oluşan bir yapıdan söz etmek mümkündür. Bu

genel model Şekil 1'de gösterilmiştir. Katmanlar aşağıdaki gibidir [9], [8]:

1. İlk katman- Engelleme: Saldırı olasılıklarını azaltmakla sorumlu olan ilk katmandır.

2. İkinci katman - Saptama: Saldırıları saptama ve uygun alarmları oluşturma bu katmanda gerçekleşecektir.

3. Üçüncü katman – Kurtarma: Saldırının etkilerini temizleme ve sistemi yeniden çalışır hale getirme bu katmanda gerçekleşecektir.



Şekil 1: Çok Katmanlı Güvenlik Modeli

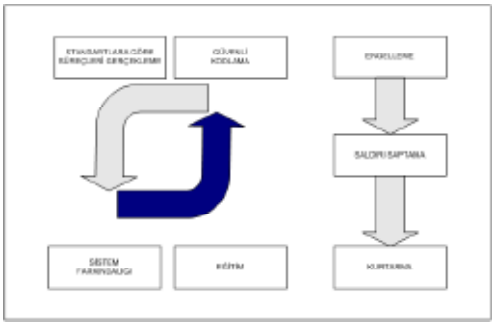
Web güvenliğini sağlamak için çok katmanlı güvenlik modeli kullanılmalıdır. Alt sistemlerin birbirleriyle etkileşimli çalışması sağlanmalıdır.

Bu bildiride, oluşturulan kurumsal web güvenliği modeli tanıtılacaktır. Bu modelde anlatılan birimler, üniversite ağlarında web güvenliği konusunda gözlemlenen gereksinimleri karşılamayı hedeflemektedir. Her ne kadar web güvenliğine özelleşmiş bir model olsa da, farklı ağ iletişim kuralları için de bu tür bir bütünleşik sisteme olan gereksinim günümüz kurumsal ağlarında her geçen gün artmaktadır.

2. Kurumsal Web Güvenliği Modeli

Kurumsal web güvenliği modeli, birbirleriyle etkileşimli çalışan çeşitli değişik güvenlik yöntemlerinden oluşmaktadır. Bu model Şekil 2'de gösterilmiştir. Model aşağıdaki birimlerden oluşmaktadır [7]:

1. Standartlaştırma
2. Güvenli Kodlama
3. Sistem Farkındalığı
4. Eğitim / Sınama
5. Saldırı Saptama
6. Saldırı Engelleme
7. Kurtarma
8. Eşgüdüm Merkezi



Şekil 2 Kurumsal Web Güvenliği Modeli

2.1. Standartlaştırma

Daha istikrarlı ve daha güvenli bir web bilgi sistemi için; standartların ve kuralların, kullanım ve güvenlik politikalarında tanımlanması ve uygulanması gerekmektedir. SANS Güvenlik Politikaları Projesi sayfasından (http://www.sans.org/resources/policies/) ayrıntılı bilgi edinmek mümkündür. Kampüs ağlarında güvenlik politikalarının uygulanması [6]'da incelenmiştir. Bu konuda örnekler [5], [18] incelenmeli ve kurumun ihtiyaç ve yapısına göre güvenlik politikaları düzenlenmelidir. Kurumsal olarak tanımlanabilecek web güvenliği standartlarına örnek olarak aşağıdakiler verilebilir:

1. Web Sunucu düzlem (işletim sistemi, web sunucu yazılımı, yazılım geliştirme ortamı...vb.) kısıtlamaları, saldırıya açıklık

çözümleme yöntemleri, yama yönetimi ve yedekleme yordamlarının tanımlanması,

2. Daha iyi yönetim ve güvenliğin sağlanması için web sunucuların sayısı mümkün olduğunca azaltılmalıdır. Merkezi bir web sunucusu kullanmak ve kuruma ait birimlerin sadece bu sunucuyu kullanmasını sağlamak iyi bir uygulama olarak karşımıza çıkmaktadır.

3. Web sitelerinde, güvenlik ve kişisel gizlilik politikaları verilmelidir. Dünya çapında kamu web sitelerinin sadece %29'unda kişisel gizlilik politikasının bulunduğu ve sadece %21'inde güvenlik politikası açıklamalarının bulunduğu saptanmıştır [19].

4. Web uygulamaları geliştirilirken güvenli kodlama esaslarına uyulması,

5. Saldırıya açıklık sınamasının, yazılım geliştirme yaşam döngüsü (YGYD) içinde de gerçekleştirilmesinin sağlanması [2],

6. Web uygulamalarına ait belgelemelerin javadoc, phpdoc gibi uygun araçlarla yapılması,

7. Kurumsal web sayfalarının belirli bir yapıya ve şablona uymasının sağlanmasıdır. E-dönüşüm Türkiye Projesi kapsamında hazırlanan kılavuzla [17], kamu kurumlarının İnternet sitelerinde asgari düzeyde içerik ve tasarım uyumunun sağlaması için gerekenler anlatılmıştır. Bu belge, web sitelerinin yapısal gereksinimleri anlatmakla beraber, güvenliğin sağlanması konusunda yapılması gerekenlere ilişkin bir bilgi içermemektedir. Güvenlik gereksinimlerini ve yöntemlerini anlatan bir belgeye ihtiyaç duyulmaktadır.

8. Kurumlar, Web İçerik Yönetim Sitemlerini ve portal yazılımlarını mümkünse kendileri geliştirmelidir. Böylece daha esnek ve daha güvenli sistemlerin kurulması sağlanacaktır. Kurum bunun yerine, özel bir içerik yönetim sistemini seçip bu sistemin yamalarını düzenli olarak izlemeyi de tercih edebilir. Tercih edilen sistemlere özgü güvenlik ayarlarının yapıldığından ve gerekli güncellemelerin yerine getirildiğinden emin olunmalıdır.

9. Web sayfası yüklenirken kasti veya teknik nedenlerden dolayı web yazılımlarında oluşan hatalarda, kullanıcıya kurumun şablon hata sayfası gönderilmelidir. Böylece saldırganların sistemle ilgili hassas bilgiler edinmesi engellenmiş olacaktır.

10. Ağ yönetim grubu, standartlara uyulmaması durumunda web servisine erişimini kapatma hakkına sahip olmalıdır.

2.2. Güvenli Kodlama

Web uygulamaları geliştirilirken tasarım sürecinden itibaren güvenli kodlama esaslarına uyulmalıdır. OWASP'ın (http://www.owasp.org) Güvenli Kodlama İlkeleri belgesi [11], Güvenli Web Uygulamaları Geliştirme Kılavuzu [12] belgesi kurum içi eğitimlerde kullanılmalı ve yazılım geliştiricilerin bu esaslara uyup uymadıkları denetlenmelidir. Yazılım geliştirme yaşam döngüsünün ilk aşamalarından itibaren güvenlik süreçlerini eklemek için yapısal bir yaklaşım sunan OWASP CLASP Projesi [13] incelenmelidir. Microsoft'un yazılım geliştirme süreci için oluşturulan, güvenilir bilişim güvenlik geliştirme yaşam döngüsü [10]'da verilmiştir.

Yazılım geliştirirken uyulması gereken temel ilkeleri aşağıdaki gibi özetlemek mümkündür [11]:

- Görev tanımlama: Kullanıcıların görevleri ve yetkileri tanımlanmalıdır. Her kullanıcıya veya kullanıcı grubuna farklı güvenlik düzeyleri tanımlanmalı,

• Farklı güvenlik düzeyleri: Yazılımın ve işlenen verinin önemine göre farklı güvenlik düzeyleri ve önlemleri kullanılmalı,

• En az yetki: Kullanıcıya ve alt sistemlere mümkün olan en az yetkinin verilmesi sağlamalı,

• Güvenli varsayılan ayarlar: Güvenlik mümkün olan en yüksek düzeyde tutulmalı, kullanıcının isterse güvenlik düzeyini ayarlamasına izin vermeli,

• Kademeli savunma: Bir denetimin yeterli olması durumunda bile, birden fazla denetimi uygulayarak daha güvenli ortamlar sağlanmalı,

• Önlemleri yalın tutmak: Alınan önlemlerin karmaşık olması, her zaman daha güvenli olacağı anlamına gelmez. Yazılımcının nasıl çalıştığını anladığı ve sorun olduğunda çözüm sürecinde kolaylıkla çözebileceği yalınlıkta sistemler daha başarılı olabilmektedir.

• Saldırı alanını azaltmak: Yapılan denetlemeler ve kısıtlamalarla, olabilecek saldırı etkisi ve alanı azaltılmalı,

• Güvenli düşmek: Yazılımın hata vermesi durumunda, güvenlik açığına yol açmadan sonlanması sağlanmalıdır.

• Kurumlar ortak projelerde kullanmak üzere, kendilerine ait yazılım kütüphanelerini ve sınıflarını güvenli kodlama esaslarına uyarak geliştirmelidir. Bu yazılım kodlarının bütün yazılım projelerinde kullanılması sağlanmalıdır. Bu kodları kullanarak geliştirilen projeler daha güvenli temeller üzerine inşa edilmiş olacaktır.

• Yazılımcıları güvenli kodlama esaslarına göre yazılım geliştirmeye zorlayan Struts yazılım geliştirme çerçeve sistemleri (http://struts.apache.org/) gibi ortamların kullanılması sağlanmalıdır.

2.3. Sistem Farkındalığı

Ağ ve güvenlik yöneticileri, ağ üzerinde nelerin yaşanmakta olduğunu bilmelidir. Bu, ağ üzerinde korunması gereken aygıtları, onların şu anki durumunu, tehditleri ve saldırıya açık noktaları bilmeyi içerir. Sistem farkındalığı olarak adlandırdığımız bu sistem, aşağıdaki düzeneklerden oluşmalıdır:

- Web Bilgi Sistemi Fakındalığı
- Saldırıya Açıklık Çözümlemesi
- Web Sunucu İzleme

Web Bilgi Sistemi Farkındalığı:

Web Bilgi Sistemi Farkındalığı ile hedeflenen, kurumsal web sunucuları ve üzerlerinde çalışan web uygulamalarına ilişkin ayrıntılı bilgilerin saptanması ve belirlenmesidir. Etkin (servis ve port tarayıcıları) ve edilgen (pasif ağ dinleyicileri) tekniklerin birbirleriyle etkileşimli kullanıldığı ve web sunucu altyapısına ilişkin güncel bilgilerin edinildiği bir yapı önerilmiştir. Bu sisteme, kurumun gereksinimlerine göre özelleştirilmiş bir arama motorunun eklenmesi ile kurumun bilgi sistemindeki web içeriği endekslenebilecektir [8].

Saldırıya Açıklık Çözümlemesi:

Web sistemleri ve üzerlerinde çalışan web uygulamalarına ait güvenlik açıkları, saldırıya açıklık (zayıflık) çözümleme sistemleri kullanılarak saptanabilmektedir. Web uygulamalarının saldırıya açıklık çözümlemeleri için kullanılan iki tür yöntemden söz etmek mümkündür [8]:

• Açık Kutu Sınaması: Web yazılımının kaynak kodu üzerinde yapılan sınamadır.

Yazılım koduna erişimin mümkün olduğu durumlarda mutlaka kullanılması, sınamanın mümkünse yazılım geliştirme yaşam döngüsü (YGYD) içerisine eklenmesi önerilmektedir. Bu yöntemle birçok olası saldırıya açıklıkların bulunması sağlanabilecektir. OWASP'ın "WebScarab", Firefox'un "Web Developer Toolbar", "Greasemonkey" ve "XSS Assistant" bu sınama için kullanılabilecek programlara örnek olarak verilebilir [16]. Saldırıya açıklık çözümlemesinin YGYD'ye eklenmesine ilişkin [2]'de ayrıntılı bir çalışma yapılmıştır.

• Kapalı Kutu Sınaması: Kapalı kutu sınaması; web sistemine dışarıdan sınama sürecinin yapılmasının gerektiğinde ve/veya web yazılımı koduna erişim mümkün olmadığında tercih edilen bir saldırıya açıklık sınamasıdır. Kapalı kutu sınaması, sadece saldırıya açıklıkların bir kısmını saptayabilmektedir. Bu sınamanın daha etkin olması için çözülmesi gereken sorunlar bulunmaktadır [3], [4]. Nikto, Wapiti, Paros Proxy ve Burpsuite bu sınama için kullanılabilecek programlara örnek olarak verilebilir. Daha ayrıntılı güvenlik sınama araç listesi [14]'de verilmiştir.

Web sistemleri dönemsel olarak sınama araçları ile taranmalı ve sınamaya ilişkin rapor teknik sorumlulara ulaştırılmalıdır. Bu raporlar saldırı yaşanmadan önce teknik personelin saldırıya açık noktalarda gerekli önlemleri almalarını sağlayacaktır. Saldırıya açıklık sınama sistemleri, Web Bilgi Sistemi Farkındalığı ve Saldırı Saptama Sistemleri (SSS) ile işbirliği içerisinde çalışmalıdır.

Web Sunucu İzleme:
Kurumsal web sunucuları, olağan dışı etkinliklere karşı sürekli olarak izlenmelidir. Bunun için kullanılan en etkin yöntem, basit ağ yönetim protokolü (SNMP) aracılarının kullanılmasıdır. Web sunucularına ait ağ trafiği, işlemci gücü, bellek kullanımı ve süreç istatistikleri gibi bilgiler SNMP aracıları tarafından toplanmalı; çözümleme süreçleri ile daha ayrıntılı incelemeler yapılmalıdır.

2.4. Eğitim / Sınama
Kurumsal ağların, web güvenliği altyapıları için kılavuzları ve standartları bulunması gerekmektedir. Web uygulama geliştiricileri ve web sunucu yöneticilerinin var olan güvenlik tehditleri ve bu tehditlere karşı alınması gereken önlemler hakkında bilgilendirilmeleri gerekmektedir. Eğitim ve sınama için aşağıdaki düzenekler kullanılabilir:

- Çalıştay ve Çalışma Grupları
- Eğitim Portalı
- Sınama Sunucuları

Çalıştay ve Çalışma Grupları:
Kurumdaki uygulama geliştiricileri ve web sunucu yöneticilerini bilgilendirmek amacıyla toplantılar, çalıştaylar yapılmalıdır. Uygulama geliştiricilerle yapılan toplantılarda güvenli kodlamanın önemini göstermek hedeflenmelidir. Web uygulamalarına girdi denetimi ve çıktı süzmesinin önemi vurgulanmalıdır. Bu toplantılar sonucunda çalışma gruplarının oluşturulmalıdır. Bu çalışma gruplarına, kuruma özgü yazılımlar için girdi/çıktı denetimi yapan yazılım kütüphanelerini geliştirmek gibi görevler verilmesi sağlanmalıdır. Aynı zamanda içeri sızma sınamaları kullanılarak saldırganların web uygulaması açıklarını kullanarak neler yapabileceğini göstermek de durumun ciddiyetinin vurgulanmasında yardımcı olacaktır.

Eğitim Portalı:
Eğitim portalı, teknik sorumluların kurum iç ağından ulaşarak web güvenliği konusunda bilgilere ulaşabileceği bir bilgi ortamıdır. Bu bilgi ortamında sunulabilecek içeriğe örnek olarak aşağıdakiler verilebilir:

- Kurumda uyulması gereken web güvenliği standartları,
- İyi güvenli kodlama uygulama örnekleri,
- Yapılması ve yapılmaması gerekenlerin örneklerle anlatılması,
- Web sunucu yapılanış ayarları.

Sınama Sunucuları:
Kuruma özgü web uygulamaları, sunucu güvenliği sağlanmış bir web sunucu üzerine konulmalı ve güvenlik uzmanlarının bu uygulamalardaki güvenlik açıklıklarını bulmaları sağlanmalıdır. Kaynak kod çözümleme ve kara kutu sınama yöntemleri kullanılabilir. Web sunucular ve veritabanlarında saldırıya açıklık sınamaları yapmakla ilgili ayrıntılı bilgi [20]'de verilmiştir.

2.5. Saldırı Saptama

Bilgisayar ağındaki olağan dışı etkinliklerin saptanması için çeşitli düzenekler kullanılmalıdır. Saldırı saptama için aşağıdaki düzenekler kullanılabilir:

- Saldırı Saptama Sistemleri
- Günlük Denetimi
- Saldırgan Tuzağı

Saldırı Saptama Sistemleri:
Saldırı Saptama Sistemleri (SSS), saldırıların saptaması ve olağan dışı etkinliklerin saptanması için kurulan sistemlerdir. Bir veya daha fazla ağ tabanlı SSS, kritik ağ kesimlerinde konuşlandırılabilir. SSS'in etkin çalışması, güncel ağ ve sistem bilgisine sahip olması ile mümkündür. Bu da SSS'in ağ farkındalığı sistemi ile etkileşimli çalışmasını gerektirmektedir. SSS'in yapılanış ve kuralları web güvenliği için özerkleştirilmelidir. Bunun yanı sıra, sunucuların yerel güvenliği için sunucu tabanlı SSS'ler konuşlandırılabilir. Saldırganların sistemde yapabilecekleri değişikliklerin takibi açısından, Tripwire (http://sourceforge.net/projects/tripwire/) ve benzeri programlarla kritik sistem dosyalarımda yaşanan değişimler izlenmelidir [15].

Günlük Denetimi:
Sunucu günlükleri (log), sunucu makineleri üzerindeki web sunucu yazılımı (örneğin apache) ve web uygulama güvenlik duvarı (örneğin Mod Security) gibi çeşitli sistemlerin günlük içeriklerini içermektedir. Bu günlüklerin çözümlenmek üzere eşgüdüm merkezi olarak adlandırılan bir ortak merkezde toplanması önerilmektedir [8]. Web sunucu yazılımına ait erişim ve hata günlüklerinin ayrıntılı olarak çözümlenmesi, olağan erişimler dışındaki davranışların saptanmasını sağlayacağından, saldırı girişimlerine ilişkin bilgi sağlayacaktır.

Saldırgan Tuzağı:
Saldırgan tuzağı (honeypot), sisteme saldırı yapacak kişiler için kurulan tuzak sistemlerdir. Bu sistemler, saldırı saptama sistemleri tarafından yakalanamayan yeni zayıflıkların öğrenilmesi ve saldırganların saldırı anındaki davranışlarına ilişkin bilgi toplamak için kurulmaktadır. Kuruma özgü web uygulamaları sahte verilerle bu saldırgan tuzakları üzerinde konuşlandırılabilir. Saldırgan tuzakları, saldırgan etkinliklerine karşı sürekli olarak izlenmelidir. Böyle bir sistemin uygulama ayrıntıları [15] çalışmasında verilmiştir.

Birden fazla saldırgan tuzağının oluşturduğu ağa, saldırgan tuzağı ağı (honeynet) denir. Genellikle saldırgan tuzaklarını izleyerek ayrıntılı istatistik ve günlük toplamak amacıyla da bu tür ağlar kurulmaktadır. Bu konuda Honeynet (http://project.honeynet.org) projesinin ayrıntılı çalışmaları bulunmaktadır.

2.6. Saldırı Engelleme

Engelleme ve risk azaltıcı sistemler, mümkün olduğunca kullanılmalıdır. Sistem aşağıdaki düzeneklerden oluşmaktadır:

- Erişim Denetimi
- Sunucu Yerel Güvenliği
- Web Uygulama Güvenlik Duvarı/Ters Vekil

Erişim Denetimi:
Kurum, web sunucularını kullanım amaçlarına ve içerdikleri bilgilerin/servislerin gizliliğine göre dış ağlara açık (public) veya dış ağlara kapalı olarak sınıflandırabilir. Kurum, bazı web sunucuların ve üzerlerinde çalışan servislerin sadece kurum ağından erişilebilmesini ve dış dünyaya kapalı olmasını hedefleyebilir. Bu kuruma özel sunuculara erişim, kurumun kendi iç ağı (intranet) veya kurumun birlikte çalıştığı ortak kurumları da içeren bir harici ağ (extranet) ile sınırlandırılabilir. Bunun yanı sıra, aygıtları yönetmek amaçlı olarak web sunuculara yapılan uzaktan erişimler denetlenmeli ve kısıtlanmalıdır. Bütün bu denetimler ağ üzerinde, ağ tabanlı güvenlik duvarındaki yapılandırmayla ve/veya yönlendirici aygıtlarında devreye alınacak erişim listeleri ile gerçekleştirilebilir. Dışarıdan erişime açık sunucular için yapılması önerilen önlemler aşağıdaki gibidir:

- Dış ağlara açık sunucuların sayısı mümkün olduğunca az tutulmalı,

- Dış ağlara açık sunucular tercihen ayrı bir sanal yerel ağda (VLAN) veya ağ güvenlik duvarına bağlı ayrı bir ağ kesimi olan "yarı güvenli ağ" (Demilitarized Zone) olarak da adlandırılan YGA'da bulundurulmalı,

- Veritabanı servisi, tercihen web hizmeti veren sunucudan ayrı bir sunucu üzerinden verilmelidir. Web hizmeti veren sunucu üzerinde veritabanı servisi çalıştırıldığı durumda, veritabanı servislerinin çalıştığı ağ kapısına erişim kısıtlanmalıdır [16].

Kuruma özel, dış ağlara kapalı sunucular için yapılması önerilen ek önlemler aşağıdaki gibidir:

- Bu sunuculara erişimde gelişmiş kimlik doğrulama düzenekleri kullanılmalıdır.

- Bu sunuculara erişim izni sadece belirli ağ kesimlerine verilmelidir. Bu önlemin güvenliği arttırdığı aşikardır, yalnız saldırganların bu izin verilen ağ kesimlerinden bir kullanıcı makinesini ele geçirebileceği ve ele geçirilen makine üzerinden bu özel sunuculara erişmeye çalışabileceği unutulmamalı ve diğer güvenlik önlemlerini almaya devam edilmelidir.

Sunucu Yerel Güvenliği:
Web sunucularının bulunduğu makinenin işletim sisteminin ve üzerinde çalışan sistemlerin güvenliği sağlanmalıdır. Web sunucusu üzerinde alınabilecek önlemler aşağıda özetlenmiştir:

- İşletim sistemi ve üzerinde çalışan yazılımların güvenlik güncellemelerinin düzenli olarak yapılması ve yazılımların gerekli yamalarının uygulanması [16],

- Sunucu üzerinde mümkün olduğunca az servisin çalıştırılması,

- Kurumda ağ tabanlı sistemler üzerinde erişim kısıtlamaları uygulanmakta olsa da, uzaktan erişim kısıtlamalarının sunucuda ayrı olarak tanımlanması,

- Web sunucu yazılımı ve veritabanları için gerekli güvenlik ayarlarının yapılması [16],

- Web uygulama güvenlik duvarı (örneğin, mod security) yazılımının sunucu üzerinde yerel olarak kurulması,

- Web sunucu üzerindeki günlüklerin düzenli olarak incelenmesidir.

Web Uygulama Güvenlik Duvarı / Ters Vekil:
Bazı sistemlerde çeşitli nedenlerden dolayı web güvenliğinin sağlanması mümkün olmayabilir. Bunun başlıca nedenleri:

• Sunucuların sistem yöneticilerine ulaşmada yaşanan problemler,

• Sistem üzerinde çalışan yazılımların yeterli belgelemelerinin bulunmaması ve o yazılımı hazırlayan programcılara ulaşılamaması yüzünden gerekli yazılım düzeltmelerinin gerçekleştirileme-mesi,

• Donanımsal kısıtlamalar yüzünden gerekli güncelleme veya sürüm terfilerinin gerçekleştirilememesidir.

Web sunucularının güvenliğini sağlamak için, sunuculara gelen ağ trafiğini süzen ve saldırı girişimlerini mümkün olduğunca engelleyen sistemler kurulmalıdır. Bu tür çözümlere Web Uygulama Güvenlik Duvarı (WUGD) denmektedir. Bu çözümler, daha çok ters vekil sunucu olarak uygulanmaktadır.

WUGD çözümlerinde, web sunucularına gidecek bütün ağ trafiği ters vekil sunucu üzerinden geçecek şekilde ayarlanmalıdır. WUGD; sadece web için değil, bütün ağ iletişim kuralları için de güvenlik duvarı olarak çalışacak şekilde kullanılabilir. WUGD'un, süzme işlemini gerçekleştirmek için süzme yeteneği olacak şekilde yapılandırılması gerekecektir.

2.7. Kurtarma

Saldırı sonrasında nelerin yapılması gerektiği tanımlanmalı ve mümkünse aralıklı olarak gerekli tatbikatlar gerçekleştirilmelidir. Saldırı sonrasında gerçekleştirilecek aşamaları aşağıdaki şekilde sınıflandırmak mümkündür :

• Sunucuya erişimin engellenmesi: Saldırganın sunucu üzerinden yayın yapması veya başka saldırılara kalkışmasını engellemek için öncelikle sunucuya erişim engellenmelidir.

• Ayrıntılı inceleme: Saldırının boyutu ve bilgi sisteminde yarattığı zarar saptanmalıdır. Sistemdeki hangi zayıflığın bu saldırıyı başarılı kıldığı belirlenmelidir. Saldırının kuruma ait başka sistemlerde de etkin olup olmadığı incelenmelidir. Saldırgan saldırıdan sonra sistemde bir arka kapı bırakmış olabilir. Sunucu, saldırıya ilişkin yeterli bilgi toplanması ve çekirdek düzeyinde sistemde bir değişiklik yapılıp yapılmadığının denetlenmesi için ayrıntılı incelemeye alınmalıdır.

• Saldırıların etkilerini temizleme: Sistemin tekrar etkinleştirilmeden önce eski haline geri getirilmesi gerekecektir. Yeni sistemin aynı saldırıya maruz kalmaması için, bu saldırıya neden olan zayıflık giderilmelidir. Mümkünse yeni bir sunucuda sistem ayağa kaldırılmalıdır.

• Sistemi yeniden çalıştırma: Verilerin yedeklerden alınarak yeni kurulan sisteme taşınması gerekecektir. Bunun için kurumun öncelikle düzenli bir sistem yedekleme politikasına sahip olması gerekmektedir.

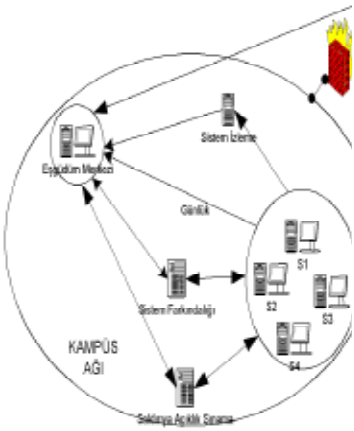
• Kullanıcıların durumdan haberdar edilmesi: Kullanıcıların kişisel bilgilerinin saldırganların eline geçmesi durumunda, kullanıcıların gerekli önlemleri alması için bir an önce gerekli bilgilendirme yapılmalıdır. Böylece kullanıcı, başka sistemlerde de aynı parolayı kullanıyorsa değiştirebilecek veya kredi kartı gibi mali değerleri için gerekli önlemleri alabilecektir.

• Saldırganın saptanması: Saldırganın saptanması için sunucu makinesi ve sistemdeki diğer aygıtlardaki kayıtlar ayrıntılı olarak incelenmelidir. Olayın teknik ve hukuki süreçleri bulunmaktadır. Türkiye'de hukuki süreçler yeterince gelişmiş değildir. İstanbul Üniversitesi'nde gerçekleşen web saldırısı sonrasında gerçekleşen teknik ve hukuki süreç [1]'de ayrıntılı olarak ele alınmıştır.

2.8. Eşgüdüm Merkezi

Kurumsal web güvenliği modelinde tanımlanan düzeneklerin her birinin belirli yarar kazanımları olduğu kadar, bazı yarar yitimleri de bulunmaktadır. Web güvenliği konusunda sağlam bir altyapı ancak çok katmanlı güvenlik modelinin uygulanması ile mümkündür Her eklenen ve etkinleştirilen güvenlik düzeneği ile riski en aza indirmek mümkün olacaktır.

Kurumsal web güvenliğinde tanımlanan düzenekler, ancak birlikte etkileşimli çalıştırıldığında en iyi sonuca ulaşmak mümkündür. Bu düzeneklerin eşgüdümü,merkezi bir sistemde toplanmalıdır. Şekil 3'de bu sistem yapısı gösterilmiştir.



Şekil 3 Kurumsal Web Güvenliği Yapısı

Eşgüdüm merkezi ile hedeflenecekler aşağıdaki gibidir:

- Düzenekler arasındaki etkileşimi sağlamak,
- Sistemin başarımını eniyileme,
- Yanlış uyarıların azaltılması ve böylece gerçek uyarılar üzerinde yoğunlaşılmasının sağlanması,
- Sistemdeki saldırıya açık noktaların saptanması ve aciliyet/tehdit durumuna göre hareket planının yapılmasının saplanması,
- Saldırı saptama sistemi gibi etkin sistemlerin yapılanışlarını dinamik olarak değiştirmektir.

3. Kısaltmalar

SSS: Saldırı Saptama Sistemi (IDS)
SNMP: Basit Ağ Yönetim Protokolü
VLAN: Sanal Yerel Ağ
YGA: Yarı Güvenli Ağ (DMZ)
YGYD: Yazılım Geliştirme Yaşam Döngüsü (SDLC)
WUGD: Web Uygulama Güvenlik Duvarı

4. Sonuç

Kurumsal ağlarda, web güvenliği için yapılması gerekenler bir model şeklinde bu bildiride anlatılmıştır. Bu yöntemlerin mümkün olduğunca çoğunun uygulanması; daha sağlam bilgi altyapılarının oluşmasına ve toplumun bu tür bilgi sistemlerine güveninin artmasına yol açacaktır.

5. Kaynaklar

[1].Cimilli C., Doğan İ., İnternet Saldırıları Sonrasında Yapılması Gerekenler, Akademik Bilişim 2004, 2004

[2].Curphey M., Araujo R., Web Application Security Assessment Tools, IEEE Security & Privacy, 2006

[3] Grossman J., Challenges of Automated Web Application Scanning – "Why Automated scanning only solves half the problem", Blackhat Windows 2004, 2004, http://www.whitehatsec.com/presentations/challenges_of_scanning.pdf

[4] Grossman J., 5 challenges of web application scanning, http://jeremiahgrossman.blogspot.com/2006/07/5-challenges-of-web-application.html

[5] İTÜ BIDB, Dinamik Web Sayfaları ve Veritabanı Hizmetleri Kullanım Politikaları, http://www.bidb.itu.edu.tr/?i=54

[6] Karaarslan E., Teke A., Şengonca H., Bilgisayar Ağlarında Güvenlik Politikalarının Uygulanması, İletişim Günleri, 2003

[7] Karaarslan E., Tuglular T., Sengonca, H., Enterprise-wide Web Security Infrastructure, TERENA Network Conference, 2007

[8] Karaarslan E., Doktora Tezi, 2008

[9].Magiera J., Pawlak A., Security Frameworks for Virtual Organizations, In Virtual Organizations: Systems and Practices, Springer, 2005

[10].Lipner S., Howard M., The Trustworthy Computing Security Development Lifecycle, Microsoft Corporation, http://msdn2.microsoft.com/en-us/library/ms995349.aspx, 2005

[11].OWASP, Secure Coding Principles, http://www.owasp.org/index.php/Secure_Coding_Principles

[12].OWASP, OWASP Guide to Building Secure Web Applications, http://www.owasp.org/index.php/Category:OWASP_Guide_Project

[13].OWASP CLASP (Comprehensive, Lightweight Application Security Process) Projesi http://www.owasp.org/index.php/Category:OWASP_CLASP_Project

[14].Peine H., Security Test Tools for Web Applications, IESE Report-Nr. 048.06/D, A Fraunhofer IESE Publication, 2006

[15].Riden J., McGeehan R., Engert B., Mueter M., Know your Enemy: Web Application Threats, Using Honeypots to learn about HTTP-based attacks, http://honeynet.org/papers/webapp/

[16].SANS, Web Applications, SANS Top-20 Internet Security Attack Targets (2006 Annual Update), http://www.sans.org/top20/#c1

[17].TÜBİTAK-MAM, 2006, Kamu Kurumları İnternet Sitesi Kılavuzu (Sürüm 1.0), http://rega.basbakanlik.gov.tr/eskiler/2007/01/20070127-7-1.doc

[18] ULAKBİM, 2007, Güvenlik Politikaları, http://csirt.ulakbim.gov.tr/politika/

[19].West, D.M., Global E-Government, 2007 Yılı İnceleme Raporu, http://www.insidepolitics.org/egovt07int.pdf, 2007

[20].Whitaker A., Newman D., Penetration Testing and Network Defense, Cisco Press, ISBN:1-58705-208-3, 2005