WEB UYGULAMA GÜVENLİĞİ AÇIKLIKLARI VE GÜVENLİK ÇÖZÜMLERİ ÜZERİNE BİR ARAŞTIRMA

Durmuş AYDOĞDU, M. Sedef GÜNDÜZ
Gazi Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Maltepe, Ankara
aydogdu.durmus@gazi.edu.tr, sedefgunduz@gazi.edu.tr

ÖZET

Bu çalışmada, OWASP tarafından yayınlanan ilk 10 web uygulama güvenliği açıklıkları, açıklıkların kaynakları ve bu açıklıkları istismar eden saldırıları önlemek için kullanılan güvenlik çözümleri araştırılmış, bu açıklıkları kullanarak gerçekleştirilebilecek saldırılara karşı alınabilecek önlemler, kullanım alanları, platform-bağımsızlığı, çalışma mantığı ve verimlilikleri açısından değerlendirilerek karşılaştırılmıştır. Elde edilen bilgiler ve bulgular doğrultusunda, hangi tür saldırılara karşı nasıl bir güvenlik çözümünün alınması, tercih edilmesi konusunda öneriler, farkındalığın ve web uygulama güvenliğinin arttırılmasına yönelik çözümler sunulmuştur.

Anahtar Kelimeler: OWASP Top 10, Web uygulama güvenliği, Açıklık, Bilgi güvenliği, Güvenlik çözümleri

A STUDY ON WEB APPLICATION SECURITY VULNERABILITIES AND SOLUTIONS

ABSTRACT

In this paper, OWASP Top 10 vulnerabilities and sources of them have been presented. Security solutions used in preventing possible attacks caused by these vulnerabilities have been also compared with usage fields, intrusion detection mechanisms, performances and platform-independencies. Finally, based on the outcomes and findings of this research, this article presents some solutions, contributions and suggessions on security awareness, web application security level, and security solutions for web-based attacks. This article also summarises and suggests the prevention methods and methodologies.

Keywords: OWASP Top 10, Web application security, Vulnerability, Information security, Security solutions

I. GİRİŞ (INTRODUCTION)

İnternet kullanımının yaygınlaşmasıyla beraber web servisleri ve uygulamaları bilgi paylaşımını şekillendirmiş, web uygulamalarının kullanımını arttırmıştır [1]. Web uygulamaları finansal işlemler, bilgi paylaşımı, sosyalleşme ve iletişim kurma gibi ihtiyaçlardan dolayı kullanılabildiği gibi, devletler tarafından kamu hizmetlerinin internet üzerinden sağlanabilmesi için de kullanılabilmektedir [2-4]. Birçok işlemin yapılmasını sağlayan web uygulamaları, kişisel bilgiler, banka hesap bilgileri ve kurumsal bilgiler gibi çeşitli bilgileri barındırdığından saldırıların hedefi haline gelmektedir [1]. Bu ortamların güvenliğinin sağlanabilmesi web uygulamalarının güvenliğiyle doğrudan ilişkilidir. Web uygulama güvenliği, üzerinde bulundurduğu verilerin gizlilik, bütünlük ve erişilebilirliğin sağlanabilmesi amacıyla alınan tedbirlerin tümünü ifade etmektedir. Web uygulama açıklıklarının istismar edilmesi, kişi ve kurumlar için maddi ve manevi zararlara neden olabilmektedir. 10 Ekim 2014 tarihinde, çevirim içi ödeme yöntemi olan PayPal'in uygulama kodundaki bir açıklık istismar edilerek kullanıcı hesabı ele geçirilebildiği ortaya çıkmıştır [5]. Web uygulamaları üzerinden yapılan siber saldırıların bir örneği de Rusya tarafından Estonya'ya yapılan siber saldırılardır. Verdiği hizmetlerin büyük bir bölümünü web üzerinden veren Estonya'da saldırılar sonucu hayat durma noktasına gelmiştir [6]. Bu saldırıları engellemek için saldırı tespit/önleme sistemi, güvenlik duvarı, uygulama katmanı güvenlik duvarı, veritabanı güvenlik duvarı ve uygulama kodunun güvenliği sağlamaya yönelik güvenlik tedbirleri uygulanmaktadır [1-4, 7-11]. Bu şekilde, pek çok teknik ve teknolojiyle bu ortamların

güvenliğini sağlamak amacıyla çalışılmaya devam edilmektedir.

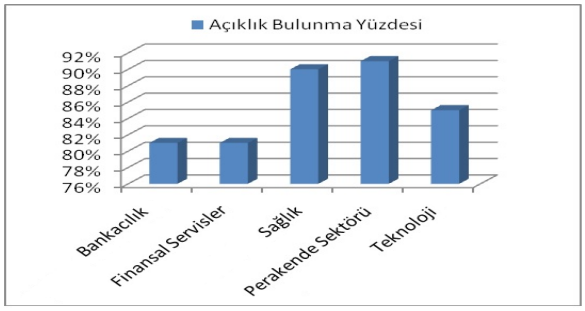
Bu çalışmada web uygulama güvenliğinin arttırılması amacıyla, web uygulamaları açıklıkları ve bu açıklıkların bulunduğu ortamlar araştırılmıştır. Bu ortamlar üzerindeki açıklıklar istismar edilerek yapılan saldırılara karşı kullanılan güvenlik çözümleri konusundaki çalışmalar incelenmiştir. Güvenlik çözümleri çalışma mantığı, performansı ve saldırılara karşı başarıları ile hangi tür saldırılarda hangi çözümlerin uygulanması gerektiği kriterleri bakımından karşılaştırılmıştır. Bu sayede güvenlik uzmanlarının hangi tür saldırılara karşı nasıl bir güvenlik çözümünü tercih etmeleri konusunda kapsamlı bir araştırma sunulmuştur.

Makale aşağıdaki şekilde düzenlenmiştir. II. Bölümde web uygulamalarının kullanım alanları ve güvenlik durumları hakkında bilgi verilmiştir. III. Bölümde OWASP (Open Web Application Security Project) tarafından 2013 yılında yayınlanan listedeki ilk on web açıklığı; uygulama, sunucu ve iletişim altyapısı güvenliğini tehdit eden unsurlar şeklinde üç başlık altında ele alınmıştır. IV. Bölümde ise bu açıklıklara yönelik yapılan saldırı yöntemlerine karşı kullanılan güvenlik çözümleri ve hangi tür saldırılara ne tür önlemler alınabileceği incelenerek güvenlik çözümlerinin karşılaştırılması yapılmıştır. V. Bölümde ise sonuç ve öneriler üzerinde durulmuştur.

II. WEB UYGULAMALARINA GENEL BAKIŞ VE GÜVENLİK DURUMLARI (WEB APPLICATION OVERVIEW AND SECURITY SITUATIONS)

İnternet en büyük ağdır ve internetin kullanımının artmasıyla beraber bilgi sistemlerinde kullanılan uygulamalar da internet ağı odaklı bir şekilde gelişmektedir. İnternetin yaygınlaşmasıyla beraber masaüstü uygulamaları yerine web uygulamaları kullanılmaya başlanmıştır [1,2]. Web uygulamaları günlük hayatımızda önemli bir yer tutmakta, etkileşimli bir yapı ile resimler, videolar, sesli materyaller gibi zengin özellikler sunmaktadır [1,2].

Web uygulamalarını saldırıların hedefi haline gelmesinde merak, zarar verici niyet, para kazanma umudu gibi birçok etken rol oynamaktadır [1,2,8]. Bu etkenleri ise, geliştirilen uygulamanın güvenliğinden ziyade fonksiyonelliğinin göz önünde bulundurulması, güvenlik bilgisinden yoksun kişilerin web uygulaması geliştirmesi ve bunun sonucunda oluşan açıklıklar tetiklemektedir [8]. Web uygulamalarının güvenlik durumunu ortaya koyan çeşitli raporlar bulunmaktadır. Web güvenliği için çözüm sağlayan bir şirket olan "Beyaz Şapka Güvenlik (White Hat Security)" şirketinin yayınladığı, "2013 Web Sitesi Güvenlik İstatistikleri" raporuna göre analiz edilen farklı alanlardaki web sitelerine ait açıklık tespit değerleri tarafımızdan grafik haline getirilerek yüzdelik dilimler şeklinde Şekil 1'de ifade edilmiştir [17].



Şekil 1. Farklı endüstrilerde analiz edilen sitelerin açıklık yüzdeleri

Bulut, mobil ve web açıklıklarının sürekli değerlendirilmesi için uygulama güvenliği sağlayan bir kurum olan CENZIC'in "Uygulama Güvenlik Açıklığı Eğilimleri Raporu 2014'e (Application Vulnerability Trends Report)" göre ise 2013 yılında uygulamaların %96'sında bir ya da birden fazla ciddi açıklık bulunduğu bilinmektedir [18]. Bunun yanında, uygulama başına 2012 yılında düşen açıklık sayısı 13 iken, bu rakam 2013 yılında 14'e yükselmiştir [18].

İstatistiki veriler bu konuda bir şeylerin yanlış olduğunu ve artık farklı güvenlik çözümlerinin geliştirilip, uygulanması gerektiğini göstermektedir [19]. Bu çalışmada da web uygulamalarındaki açıklık kaynakları ve güvenlik çözümleri incelenmiş ve güvenlik yöneticilerine doğru güvenlik çözümünü tercih etmelerinde dikkat etmeleri gereken hususlar konusunda öneriler sunulmuştur.

III. WEB UYGULAMA GÜVENLİĞİ AÇIKLIKLARI (WEB APPLICATION SECURITY VULNERABILITIES)

Literatür incelendiğinde güvenlik riskleri hakkında en kapsamlı çalışan kuruluş olarak OWASP göze çarpmaktadır. OWASP dünya çapında yazılım güvenliğini geliştirmeye odaklanmış bir kuruluştur. Kuruluşun misyonu, dünya çapında birey ve kuruluşların gerçek yazılım güvenlik riskleri hakkında bilinçli karar verebilmesi için yazılım güvenliğini öne çıkarmaktır. Kuruluş web uygulama güvenliğinde farkındalığı arttırmak için en bilindik web uygulama açıklıklarını yayınlamaktadır [22]. Açıklığın yayınlanan listedeki sırası Tablo I'de sunulmuştur.

Web uygulamalarının güvensiz olması; uygulamanın kaynak kodu, uygulamanın üzerinde çalıştığı sunucu veya sunucu istemci arasındaki iletişim altyapısının üzerindeki açıklardan kaynaklanmaktadır [20,21].

Bu çalışma kapsamında OWASP tarafından 2013 yılında yayınlanmış web uygulamalarının güvensizliğine sebep olan açıklıklar; Dalai (2011) ve Microsoft tarafından yapılan güvenlik riski

değerlendirmesi ölçütlerine göre (uygulama kodu, sunucu ve iletişim altyapısı) kaynağı ve açıklıktan etkilenen taraflar başlıklarında sınıflandırılarak Tablo I'de sunulmuştur.

TABLO I. OWASP İLK 10 AÇIKLIK, AÇIKLIK KAYNAKLARI VE ZARAR GÖREN TARAFLAR [22]

Sıra Nu.	Açıklıklar	Kaynak	Etkilenen
1	Enjeksiyon (Injection)	Uygulama	Sunucu
2	Kırık Kimlik Doğrulama ve Oturum Yönetimi (Broken Authentication and Session Management)	Uygulama	İstemci
3	Siteler Arası Betik Çalıştırma (Cross-Site Scripting -XSS)	Uygulama	Sunucu/ İstemci
4	Güvensiz Doğrudan Nesne Başvurusu (Insecure Direct Object References)	Uygulama	Sunucu
5	Güvenlik Yanlış Yapılandırma (Security Misconfiguration)	Sunucu	Sunucu
6	Hassas Veriyi Açıkta Bırakma (Sensitive Data Exposure)	Uygulama	Sunucu
7	İşlev Seviyesi Erişim Kontrolü Eksikliği (Missing Function Level Access Control)	Uygulama	Sunucu
8	Siteler Arası İstek Sahteciliği (Cross-Site Request Forgery (CSRF))	İletişim Altyapısı	İstemci
9	Bilinen Açıklık Bileşenlerini Kullanma (Using Known Vulnerable Components)	Uygulama	Sunucu
10	Doğrulanmayan Yönlendirme ve İletme (Unvalidated Redirects and Forwards)	Uygulama	Sunucu/ İstemci

Etkilenen taraf değerlendirilirken, sunucu etkilendiğinde, doğrudan ya da dolaylı olarak istemci de etkileneceğinden dolayı etkilenen taraf sunucu olarak değerlendirilmiştir. Sadece kullanıcının etkilendiği durumlarda istemci, hem sunucunun hem de istemcinin ayrı ayrı etkilendiği durumlarda etkilenen taraf sunucu/istemci olarak değerlendirilmiştir.

OWASP kuruluşu web uygulama güvenliğinde farkındalığı arttırmak için en bilindik web uygulama açıklıklarını yayınlamaktadır. OWASP tarafından yayınlanan listeye göre 2013'ün en bilindik 10 açıklıkları, uygulama kodu açıklıkları, sunucu kaynaklı açıklıkları ve iletişim altyapısı kaynaklı açıklıklar olarak aşağıda başlıklar altında ele alınmıştır. Açıklıkların 8 tanesi uygulama kodu iken diğerleri sunucu ve iletişim altyapısı başlıkları altında ele alınmıştır.

A. Uygulama Kodu Kaynaklı Açıklıklar (Source Code Based Vulnerabilities)

Kod içerisindeki kusurlar bu kategori altında bulunmaktadır [20,21]. Bu açıklıklar üzerinde sisteme ciddi zararlar verebilecek saldırılar düzenlenebilmektedir. Web uygulamalarına yapılan saldırıların büyük çoğunluğu kullanıcıdan alınan giriş değerlerinin kontrol edilmemesinden kaynaklanmaktadır [4]. SQL enjeksiyon ve çapraz-site betikleme saldırıları da giriş değerlerinin uygunluğunun kontrol edilmemesini istismar eden saldırılarıdır [4,23]. OWASP 2013 en bilindik 10 açıklıklar listesinde yer alan ve bunlardan uygulama kodu kaynaklı olan açıklıklar ile istismar yöntemleri aşağıda ele açıklanmıştır.

Enjeksiyon (Injection), Enjeksiyon açıklıkları komutun ya da sorgunun bir kısmında güvenilmeyen verinin yorumlayıcıya gönderilmesiyle meydana gelmektedir. Saldırganın kötücül verisi istenmeyen komutların çalıştırılması ya da uygun yetkilendirme olmadan verilere erişimi sağlayacak şekilde yorumlayıcı yanıltabilmektedir. Enjeksiyon saldırıları SQL, OS ve LDAP komutları üzerinden yapılabilmektedir [22].

Kırık Kimlik Doğrulama ve Oturum Yönetimi (Broken Authentication and Session Management), Uygulama fonksiyonlarının kimliklendirme ve oturum yönetimiyle ilgili fonksiyonlarının uygulanmaması sonucu meydana gelmektedir. Bu açıklık üzerinden saldırganlar şifreler, anahtarlar, oturum jetonu (token) ya da diğer kullanıcıların bilgilerini tahmin etme şeklinde açıklığı kullanabilmektedir [22].

Siteler Arası Betik Çalıştırma (Cross-Site Scripting (XSS)), Siteler Arası Betik Çalıştırma açıklığı uygulamaların web tarayıcı üzerinden, güvenilmeyen verinin alınması ya da gönderilmesi sırasında, verilerin düzgün doğrulanmaması sonucu meydana gelmektedir. Bu açıklık saldırganlara, kurbanının tarayıcısı üzerinde betik çalıştırmasını sağlamaktadır [22]. Saldırı sonucu, kullanıcının oturum bilgisi çalınabilmekte, web sitelerine zarar verilebilmekte ya da kullanıcılar kötücül sitelere yönlendirilebilmektedir [22,24].

Güvensiz Doğrudan Nesne Başvurusu (Insecure Direct Object References), Doğrudan nesne başvurusu, uygulamayı geliştiren kişinin uygulama içerisinde kullanılan dâhili uygulama nesnelerinin referanslarına uygun erişim kontrolü yapılmaması sonucu meydana gelmektedir. Referansların erişim kontrolü ya da diğer korumalar yapılmaması sonucu, saldırganlar verilere yetkisiz erişerek bu referansları değiştirebilmektedir [22].

Hassas Veriyi Açıkta Bırakma (Sensitive Data Exposure), Web uygulamaların birçoğunda kredi kartları, vergi numaraları ve kimliklendirme bilgilerini düzgün olarak korumamaktadır. Saldırganlar da zayıf korunan verileri kredi kartı dolandırıcılığı, kimlik hırsızlığı ya da diğer suçlar için çalabilmekte ya da değiştirebilmektedir. Bu açıklık uygulama üzerinde bulunan hassas bilgilerin yetkisiz kişilere erişimi imkân sağlamaktadır. Hassas veriler

açığa çıkmaması için saklanırken ya da gönderilirken şifreleme gibi ekstra önlemler alınmalıdır [22].

İşlev Seviyesi Erişim Kontrolü Eksikliği (Missing Function Level Access Control), Web uygulamalarının birçoğunda kullanıcı ara yüzü kullanılabilir olmadan önce erişim hakları seviyesinde fonksiyonlar doğrulanmaktadır. Ancak uygulamaların her bir fonksiyona erişim sağlandığında bu erişim kontrollerini sunucu üzerinde yapması gerekmektedir. Eğer istek doğrulanmazsa, saldırganlar düzgün kimliklendirme yapmadan sahte isteklerle fonksiyonlara erişimleri mümkün olabilmektedir [22].

Bilinen Açıklık Bileşenlerini Kullanma (Using Known Vulnerable Components), Yazılım modülleri, çerçeveler (frameworks) ve kütüphaneler gibi bileşenler, genellikle tam yetkiyle çalışmaktadırlar. Eğer bu bileşenlerin açıklıkları istismar edilebilir ise, saldırganlar açıklıklar üzerinden sistem üzerinde, veri kaybına yol açmasına ya da sunucuyu ele geçirmesini kolaylaştırabilir. Uygulamalarda bilindik açıklıkları olan bileşenlerin kullanılması uygulamanın güvenlik seviyesini düşürebilir ve mümkün saldırı alanları ve etkilerine olanak sağlamaktadır [22].

Doğrulanmayan Yönlendirme ve İletme (Unvalidated Redirects and Forwards), Web uygulamaları sık sık kullanıcıları diğer sayfa ve web sitelerine yönlendirmekte ve hedef sayfalara karar vermek için güvenilmeyen veri kullanmaktadırlar. Bu saldırı türünde, saldırganlar kullanıcıları sahte ya da kötücül sitelere yönlendirebilir ya da yetkisiz sayfalara erişim için iletebilirler [22].

B. Sunucu Kaynaklı Açıklıklar (Server Based Vulnerabilities)

Uygulamanın çalıştığı sunucu üzerinde güvenlik yapılandırmasının düzgün şekilde yapılmamasından dolayı ortaya çıkan açıklıklar bu kategori altında değerlendirilmektedir [20,21]. OWASP 2013 en bilindik 10 açıklıklar listesinde yer alan ve bunlardan sunucu kaynaklı olan açıklık ile istismar yöntemleri aşağıda ele alınmıştır.

Güvenlik Yanlış Yapılandırma (Security Misconfiguration), Uygulamanın güvenliğinin sağlanabilmesi için uygulamalar, çerçeveler (frameworks), uygulama sunucu, web sunucu, veritabanı sunucu ve ortamlar için güvenlik ihtiyaçları bulunmalı, güvenli yapılandırma tanımlanmalı ve uygulanmış olması gerekmektedir. Varsayılan güvenlik ayarlarının güvensiz olmasından dolayı, güvenlik yapılandırması tanımlanmış, uygulanmış ve sürdürülmüş olması ve bunlara ek olarak yazılımlar güncel olmalıdır [22]. Güvenlik kontrollerinin düzgün uygulanmaması sonucu, sunucu üzerindeki açıklık istismar edilerek uygulama üzerindeki verilere erişim sağlanabilmektedir. Veritabanı bağlantısında şifre belirlenmemesi ve uzaktan bağlantı özelliğinin açık olması durumu bu açıklığa örnek olarak verilebilir.

C. İletişim Altyapısı Kaynaklı Açıklıklar (Communication Infrastructure Based Vulnerabilities)

Sunucu ile istemci arasındaki iletişim altyapısı üzerindeki açıklıklar bu kategori altında ele alınmıştır. Bunlar iletim katmanı, ağ katmanı ya da OSI modelinin diğer katmanlarındaki zafiyetlerden ortaya çıkan açıklıklardır [20,21]. OWASP 2013 en bilindik 10 açıklıklar listesinde yer alan ve bunlardan iletişim altyapısı kaynaklı olan açıklık ile istismar yöntemi aşağıda verilmiştir.

Siteler Arası İstek Sahteciliği (Cross-Site Request Forgery (CSRF)), Bu saldırılar sisteme kendini tanıtmış kullanıcının tarayıcısını web uygulaması üzerinde açıklık oluşturmak amacıyla, kullanıcının oturum çerezi ya da diğer otomatik olarak eklenen kimliklendirme bilgisini içeren, sahte HTTP istekleri göndermesine zorlamaktadır. Bu da saldırganlara, açıklık bulunan web uygulamasının gelen istekleri kullanıcıdan gelen meşru istek olarak görmesini sağlamaktadır. Bu şekilde kullanıcıların sistem üzerindeki eposta adresi gibi bilgiler değiştirilebilmektedir [22].

IV. WEB UYGULAMALARI SALDIRILARINDAN KORUNMA YÖNTEMLERİ (PREVENTION METHODS OF THE WEB APPLICATION ATTACKS)

Web uygulamalarına yönelik yapılan saldırıları önleme amacıyla geliştirilen akademik ve ticari olarak birçok güvenlik çözümleri bulunmaktadır. Güvenlik çözümleri incelendiğinde uygulama kodunun güvenliğinin sağlanması için PHAN, SWAP, Diglossia, PSIAQOP güvenlik çözümleri önerilmiştir. Ayrıca, imza tabanlı güvenlik sistemleri ve anormallik tabanlı güvenlik sistemleri olarak Saldırı Tespit Sistemleri yaklaşımları kullanılmaktadır [1-4,7-11]. Bu bölümde Saldırı Tespit Sistemleri ve uygulama kodunun güvenliğinin sağlanmasında kullanılan yöntemler anlatılmıştır.

Yapılan araştırma sonucunda güvenlik çözümlerinin özellikleri arasında özgün ve etkili çözüm sağlayan kriterler ayıklanmıştır. Bu kriterler; imza tabanlı, performans etkin, programlama diline özel, sıfır gün saldırısında başarılı olma olarak belirlenmiştir. Bu kriterler ve güvenlik çözümleri Tablo II'de sınıflandırılmıştır.

A. Saldırı Tespit Sistemleri (Intrusion Detection Systems)

Saldırı tespit sistemleri bir sisteme saldırı olup olmadığının tespitinde kullanılan, hem dâhili hem de harici saldırılara karşı koruma sağlayan bir güvenlik sistemidir [13,14]. Bu sistemin; uç birim ve ağ tabanlı

saldırı tespit sistemi olmak üzere iki türü olup, uç birim tabanlı saldırı tespit sistemleri bir ya da birkaç uç birim sistem üzerindeki verileri analiz etmektedir [14,25]. Bu bilgiler genellikle; sistem çağrıları, uygulama kayıtları, dosya sistemi değişiklikleridir [25]. Ağ tabanlı saldırı tespit sistemleri ise, ağ topolojisinde tüm trafiğin üzerinden geçecek bir konumda üzerinden geçen trafiği analiz etmektedirler [14,25]. Saldırı tespit sistemleri anormallik tabanlı ve imza tabanlı olmak üzere iki başlık altında ele alınmaktadır. Bu yöntemler aşağıda açıklanmıştır.

TABLO II. GÜVENLİK ÇÖZÜMLERİNİN KARŞILAŞTIRILMASI

Güven Yöntemleri	İmza Tabanlı	Performans Etkin	Programlama Diline Özel	Sıfır Gün Saldırısında Başarılı
İmza Tabanlı Sistemler	✓	✓	×	×
Anormallik Tabanlı Sistemler	×	×	×	✓
PHAN	✓	×	✓	✓
SWAP	✓	✓	×	✓
Diglossia	✓	✓	×	×
PSIAQOP	✓	×	×	×

Anormallik tabanlı saldırı tespit sistemi, anormallik tabanlı saldırı tespit sistemi daha önceden belirlenen bir davranış referansı üzerinden çalışmaktadır [8,13-16]. Bu yaklaşımda, belirlenen davranış ve sistemin anlık davranışı olmak üzere iki adet profil kullanılmaktadır [13]. Davranış belirleme işleminde çeşitli kaynaklardan alınan; işlemci kullanımı, TCP bağlantı sayısı, izleme olayları, basılan tuş kayıtları, sistem çağrıları, ağ paketleri, kullanıcıların oturum süreleri, eposta sayısı, dosya sistemlerine erişim ve güvenli çalışma şartları gibi çeşitli bilgiler kullanılmaktadır [8,13]. Davranış belirleme işlemi statik ya da dinamik şekilde yapılabilmektedir [8]. Davranış belirleme işleminden sonra bilinen davranış ile sistemin anlık davranışı karşılaştırılarak saldırı olup olmadığının tespiti yapılmaktadır [14,16]. Anormallik tabanlı sistemlerdeki başlıca sorunlar; sistemin normal çalışmasının daha önceden belirlenmesi gerekliliği, sistemin normal çalışması olarak belirlenen şartların zaman içerisinde değişmesi, normal davranışın öğrenilmesi esnasında saldırılarının normal davranış olarak algılanması ve makine öğrenme tekniklerinin zorluğudur [7,13,14]. Ayrıca sistemin normal davranışı zamanla çok sık değişiyorsa, kötü niyetli kişiler yapacağı saldırıyı geniş bir zaman aralığına yayabilir ve sistem üzerinde fark edilmemesini sağlayabilirler [14]. Anormallik tabanlı yaklaşımlar sistem üzerindeki olayları sürekli olarak takip ettiği ve belirli bir şarta bağlı olarak çalışmadığı için sıfır gün saldırılarına karşı daha başarılı koruma sağlamaktadır [7,13]. Fakat bu yaklaşımda yanlış uyarı üretilme sık görülen bir durumdur [7,13].

İmza tabanlı saldırı tespit sistemi, imza tabanlı sistemler daha önceden elde edilen bilgi birikimini kullanarak, bilinen saldırılardan oluşturulmuş imzalar üzerinden çalışan bir yaklaşımdır [8,13,14,16]. İmza tabanlı sistemlerde saldırı olduğu halde saldırı olduğu önceden tespit edilememiş, imza veritabanında bulunmayan özgün saldırılara karşı koruma sağlanamamaktadır [13]. Fakat önceden bilinen ve imza veritabanında bulunan saldırılara karşı efektif koruma sağlanmaktadır [13]. İmza tabanlı yaklaşım diğer yaklaşımlara göre daha az yanlış uyarı üretmesinden dolayı gerçek dünyada daha kabul edilebilir bir yaklaşımdır [7,16].

Saldırı tespit sistemlerinin amacı yetkisiz erişimler, bilgi sistemlerine sızma, istenmeyen ve kötücül ağ trafiğinin tespitidir ve virüsler, truva atları ve solucanlar gibi kötücül yazılımların zararlarını en aza indirmektedir. [14,15,17] Saldırı tespit sistemleri, bilgi sistemlerine ağ üzerinden gelebilecek saldırılara karşı koruma sağlamak amacıyla tasarlandıklarından dolayı, web uygulamalarına karşı yapılan saldırılarda da sistem üzerinden yapılabilecek saldırılara karşı koruma sağlayabilmektedir. Saldırı tespit sistemleri, bilinen açıklık bileşenlerini kullanma ve güvenlik yanlış yapılandırma açıklıklarını istismar eden saldırıları tespit edebilmektedir.

B. Uygulama Kodunun Güvenliğinin Sağlanmasında Kullanılan Yöntemler (Methods used for Ensuring Application Code Safety)

Web uygulama güvenliğinin arttırılması amacıyla uygulama kodunun güvenliğinin sağlanması yöntemleri de bulunmaktadır. Bu yöntemlerin belirli saldırı tipine ve programlama diline yönelik olması gibi kısıtları bulunmaktadır. Literatürde uygulama kodunun güvenliğinin sağlanması için birçok yöntem mevcuttur.

Diglossia, web uygulamalarında SQL ve NoSQL sorgularında enjeksiyon saldırı olup olmadığını PHP programlama dilinde kesin ve etkili bir biçimde tespit edebilen bir araçtır [10]. Diglossia, enjeksiyon açıklığını kullanan saldırılarından olan SQL ve NoSQL enjeksiyon saldırılarına karşı koruma sağlayabilmektedir [10].

SWAP (Secure Web Application Proxy), sunucu tarafında çalışan bir güvenlik çözümüdür. SWAP, sistem üzerinde ters vekil sunucu olarak çalışan ve bu şekilde bütün HTML cevaplarının içeriğini kontrol ederek, çapraz-site betikleme açıklığını kullanan saldırılara karşı koruma sağlayabilmektedir [12].

PHAN (PHP Hybrid Analyzer), PHP programlama dilinde çalışan, web uygulamalarının güvenliğini statik ve dinamik yöntemlerin güçlü yönlerini birleştirerek hibrid bir yaklaşım kullanan bir araçtır [4]. Phan, enjeksiyon açıklıklarını kullanan saldırılara karşı koruma sağlayabilmektedir [4].

PSIAQOP (Preventing SQL Injection Attack based on Query Optimization Process), sorgu optimizasyonu yaparak bilinen bütün SQL enjeksiyon saldırılarına karşı koruma sağlayan özgün bir yaklaşımdır [26]. Sorgu optimizasyonu sezgisel kurallara bağlıdır [26]. PSIAQOP, enjeksiyon açıklığını kullanan SQL enjeksiyon saldırılarına karşı koruma sağlayabilmektedir [26].

V. SONUÇLAR VE ÖNERİLER (CONCLUSIONS AND RECOMMENDATIONS)

Bu makalede, OWASP tarafından yayınlanan 2013 yılına ait ilk 10 web uygulama açıklıkları, açıklık kaynakları ve bu açıklık kaynaklarını istismar eden saldırılara karşı kullanılan güvenlik çözümleri ele alınmıştır. Yapılan araştırma sonucunda elde edilen sonuçlara göre:

1. Web uygulama güvenliğinde açıklıkların uygulama kodu, sunucu ve iletişim altyapısından oluştuğu ve bu çalışmada özetlenen açıklıkların kontrol edilmesi ve açıklıkların kapatılması,
2. Güvenlik çözümlerinin çeşitlilik gösterdiği ve bu güvenlik çözümlerinin seçilmesinde;
    a. Uygulamanın geliştirildiği yazılım dili,
    b. Uygulamanın üzerinde çalıştığı sunucu sayısı,
    c. Uygulamanın üzerinde çalıştığı sunucunun yazılım ve donanım özellikleri,
    d. Uygulamanın üzerinde çalıştığı web sunucu programı.

    gibi kriterlere dikkat edilmesi,
3. Açıklıkların büyük bir bölümünün uygulama kodu kusurlarından dolayı oluştuğu,
4. Web uygulama geliştiricileri yansıra web uygulamalarını kullanan kişilerin bilgi güvenliği farkındalık seviyesinin de büyük önem taşıdığı,
5. Web uygulama güvenliğini arttırmaya yönelik olarak bu çalışmada sunulan uygulama kod güvenliği yöntemlerinden mutlaka faydalanılması,
6. Uygulama güvenliğinin sağlanabilmesi için saldırıların zamanında tespit edilmesinin çok önemli olduğu ve Bölüm IV'te sunulan tespit yöntemlerinin mutlaka kullanılması,
7. Güncel açıklıkların takip edilmesi ve giderilmesi gerektiği görülmüştür.

Bu çalışma ile güvenlik uzmanlarının web uygulama güvenliğinde hangi tür açıklıklara karşı nasıl bir güvenlik çözümünü tercih etmelerine yardımcı olarak, web uygulama güvenliğinin arttırılmasına katkı sağlamak amaçlanmıştır.

IX. KAYNAKLAR (REFERENCES)

[1] Manikanta, Y. V. N., & Sardana, A. (2012). Protecting web applications from SQL injection attacks by using framework and database firewall. In Proceedings of the International Conference on Advances in Computing, Communications and Informatics (609-613). ACM.

[2] Scholte, T., Balzarotti, D., & Kirda, E. (2012). Have things changed now? An empirical study on input validation vulnerabilities in web applications. Computers & Security, 31(3), 344-356.

[3] Qian, L., Wan, J., Chen, L., & Chen, X. (2013). Complete Web Security Testing Methods and Recommendations. In Computer Sciences and Applications (CSA), 2013 International Conference on (86-89). IEEE.

[4] Monga, M., Paleari, R., & Passerini, E. (2009). A hybrid analysis framework for detecting web application vulnerabilities. In Proceedings of the 2009 ICSE Workshop on Software Engineering for Secure Systems (25-32). IEEE Computer Society.

[5] İnternet : Hacking PayPal Accounts with one click, URL: http://yasserali.com/hacking-paypal-accounts-with-one-click/.

[6] İnternet: 2007 cyberattacs on Estonian, URL: http://en.wikipedia.org/wiki/2007_cyberattacks_on_Estonia, Son Erişim Tarihi: 02.08.2014

[7] Jain, P., & Goyal, S. (2009). An Adaptive Intrusion Prevention System Based on Immunity. In Advances in Computing, Control, & Telecommunication Technologies, 2009. ACT'09. International Conference on (759-763). IEEE.

[8] H. J., Richard Lin, C. H., Lin, Y. C., & Tung, K. Y. (2013). Intrusion detection system: A comprehensive review. Journal of Network and Computer Applications, 36(1), 16-24.

[9] Yip, A., Wang, X., Zeldovich, N., & Kaashoek, M. F. (2009). Improving application security with data flow assertions. In Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles (291-304). ACM.

[10] Son, S., McKinley, K. S., & Shmatikov, V. (2013). Diglossia: detecting code injection attacks with precision and efficiency. In Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security (1181-1192). ACM.

[11] Krishnamurthy, A., Mettler, A., & Wagner, D. (2010). Fine-grained privilege separation for web applications. In Proceedings of the 19th international conference on World wide web (551-560). ACM.

[12] Wurzinger, P., Platzer, C., Ludl, C., Kirda, E., & Kruegel, C. (2009). SWAP: Mitigating XSS attacks using a reverse proxy. In Proceedings of the 2009 ICSE Workshop on Software Engineering for Secure Systems (33-39). IEEE Computer Society.

[13] Guan, X., Wang, W., & Zhang, X. (2009). Fast intrusion detection based on a non-negative matrix factorization model. Journal of Network and Computer Applications, 32(1), 31-44.

[14] Chakrabarti, S., Chakraborty, M., & Mukhopadhyay, I. (2010). Study of snort-based IDS. In Proceedings of the International Conference and Workshop on Emerging Trends in Technology (43-47). ACM.

[15] Hoang, X. D., Hu, J., & Bertok, P. (2009). A program-based anomaly intrusion detection scheme using multiple detection engines and fuzzy inference. Journal of Network and Computer Applications, 32(6), 1219-1228.

[16] Meng, Y., & Kwok, L. F. (2014). Adaptive blacklist-based packet filter with a statistic-based approach in network intrusion detection. Journal of Network and Computer Applications, 39, 83-92.

[17] WhiteHat Security, Website Security Statistics Report. 2013.

[18] CENZIC Application Vulnerability Trends Report, 2014.

[19] Du, W., Jayaraman, K., Tan, X., Luo, T., & Chapin, S. (2011). Position paper: why are there so many vulnerabilities in web applications?. In Proceedings of the 2011 workshop on New security paradigms workshop (83-94). ACM.

[20] Dalai, A. K., & Jena, S. K. (2011, Şubat). Evaluation of web application security risks and secure design patterns. In Proceedings of the 2011 International Conference on Communication, Computing & Security (565-568). ACM.

[21] İnternet: Web Application Security Fundamentals, URL:http://msdn.microsoft.com/en-us/library/ff648636.aspx, Yayınlanma Tarihi: Haziran 2003.

[22] İnternet: "Top 10 2013 - Top 10 ", URL: https://www.owasp.org/index.php/Top_10_2013-Top_10, Değiştirilme Tarihi: 26 Ağustos 2014.

[23] Liu, A., Yuan, Y., Wijesekera, D., & Stavrou, A. (2009). SQLProb: a proxy-based architecture towards preventing SQL injection attacks. In Proceedings of the 2009 ACM symposium on Applied Computing (2054-2061). ACM.

[24] Silva Pinto, B., & Barnett, R. (2011). A novel algorithm for obfuscated code analysis. In Information Forensics and Security (WIFS), 2011 IEEE International Workshop on (1-5). IEEE.

[25] Khairkar, A. D., Kshirsagar, D. D., & Kumar, S. (2013). Ontology for Detection of Web Attacks. In Communication Systems and Network Technologies (CSNT), 2013 International Conference on (612-615). IEEE.

[26] Al-Khashab, E., Al-Anzi, F. S., & Salman, A. A. (2011). PSIAQOP: preventing SQL injection attacks based on query optimization process. In Proceedings of the Second Kuwait Conference on e-Services and e-Systems(10).ACM.