Web Saldırı Saptama ve Engelleme Sistemi Temelleri

E. Karaarslan[1], T. Tuğlular[2], H. Şengonca[3]
[1]Ege Üniversitesi Uluslararası Bilgisayar Enstitüsü, 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

Özetçe

Günümüzde web sunucularını hedefleyen, özellikle web uygulaması seviyesinde artan sayıda saldırılar yaşanmaktadır. Kullanıcılar ve web uygulamaları arasında konuşlanacak yeni güvenlik önlemlerine gereksinim duyulmaktadır. Bu çalışmada bu tür bir sistemin temelleri verilmiş ve modellenmiştir. Web trafiği için özerkleştirilen saldırı saptama ve engelleme sistemi olan Web IDS/IPS tanıtılmıştır. Ayrıca Web altyapısı farkındalığı ve zayıflık inceleme sistemleriyle tümleştirilen sistem ve Ege Üniversitesi kampüs ağından alınan istatistiksel sonuçlar da sunulmuştur.

Abstract

There is an increasing number of attacks aiming web servers; mostly at the application level. There is a need for new security technologies which are deployed between the users and web applications. In this work, such a system's fundamentals will be given and modeled. Web IDS/IPS is introduced which is a specialized intrusion detection and prevention system. Also web infrastructure awareness, vulnerability analysis integrated system and statistical results from Ege University campus network are presented.

1 Giriş

Günümüzde şirketler, müşterilerine ve iş ortaklarına doğru bilgiyi zamanında sağlamak zorundadır. Aynı zamanda para işlemlerini içeren e-ticaret ve internet bankacılığının çevrimiçi olarak kişisel makinelerden gerçekleştirilmesi gerekmektedir. Bu işlemler, en kolay ve en etkin olarak WWW aracılığı ile gerçekleştirilebilmektedir. Her istemcinin makinesinde web tarayıcısı bulunmaktadır ve sunucu tarafında da WWW servislerinin kurulması kolaydır. Bunun yanı sıra, yerel ağ anahtarı, yazıcı, kablosuz erişim noktası, kamera sistemleri, kesintisiz güç kaynağı...vb aygıtları yönetmek için de web sunucu yazılımları en kolay araçlar haline gelmiştir.

Bir http sunucu çalıştığı zaman bütün http isteklerine açıktır. Sunucuya erişimin sağlanması için ağ güvenlik duvarlarında da http kapısı (port) açık bırakılmaktadır. Http istekleri saldırı kodu içerebilir ama geçerli http istekleri olarak gözüktükleri için geleneksel güvenlik duvarları tarafından kabul edilmekte ve ayrıntılı olarak incelenmemektedirler [1].

Web sistemlerini hedefleyen saldırılar artmaktadır. Zone-H 'in incelemesine göre, 2004 yılında web sunucu saldırıları ve web sitesi tahrifatları 2003 yılına göre 400,000 (%36) artmıştır [2]. CSI/FBI'in "Bilgisayar Suç ve Güvenlik Anketi"ne göre, ankete katılanların %95'i 2005 yılı içinde güvenlikle ilgili 10'dan fazla web sitesi olayı yaşamıştır [3].

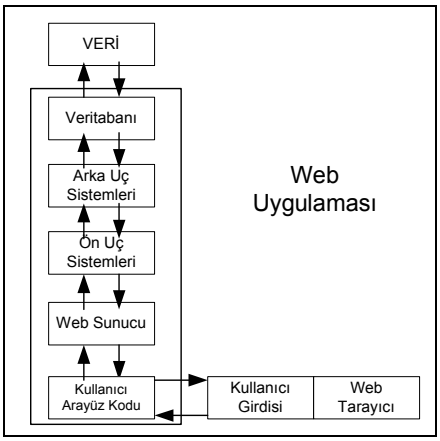
Web tabanlı saldırılar çoğunlukla meta kod veya geçersiz veri girişleri kullanılarak gerçekleştirilmektedir. Bu saldırılar sayesinde, saldırgan sistem hakkında bilgi edinebilmekte ve uygulamanın kapsamı dışındaki veri ve kaynaklara erişim kazanabilmektedir [1]. Açık Web Uygulama Güvenlik Projesi (OWASP), en kritik 10 web uygulama ve veritabanı güvenlik zayıflıklarını listelemekte ve bu zayıflıklardan kaçınmak için en etkili yolları sunmaktadır. Şu anki liste için bkz. [4]. Web tabanlı saldırılar, hedef gözeten saldırılar olarak bilinmektedir ama bu da Web Uygulaması Solucanları tehdidi ile değişmektedir. Solucanlar arama motorlarının yardımıyla yayılabilmekte ve zayıflıklara sahip web sitelerini bularak onları etkileyebilmektedir. Ayrıntılı bilgi için bkz [5].

Web sunucuların ve üzerlerinde çalışan uygulamaların güvenliğinin sağlanması gerekmektedir. Ne yazık ki alınması gereken önlemler gerektiği kadar uygulanamamaktadır. Ağ yöneticileri bütün sunuculara yamaları zamanında uygulayamamakta veya bütün güvenlik tehditlerini bilemeyebilmektedirler. Ağ genişledikçe problem de büyümektedir. Büyük kurumsal ağlarda çok sayıda değişik web sunucuları ve çeşitli web uygulamaları çalışmaktadır. Güvenlik uzmanları çoğunlukla kendi kurumsal web uygulamalarının nasıl çalışması gerektiğini bilmemektedir. Uygulamalar geliştirilirken, var olan ivedi gereksinimleri karşılamak için hızlı bir şekilde ve çoğunlukla güvenlik gereksinimleri düşünülmeden geliştirilmektedir. Aynı zamanda programcıların birçoğu da güvenli kodlamayı bilmemektedir. Bu nedenlerden dolayı, saldırıları saptayacak ve olabilirse saldırı yaşanmadan engelleyecek güvenlik önlemlerine gereksinim duyulmaktadır [1]. Bu çalışmada temel olarak bu tür bir sistemin nasıl çalışması gerektiği tanımlanacak, bu konuda Ege Üniversitesi ağında yapılmakta olan çalışmalar ve alınan ilk sonuçlar sunulacaktır.

Bir sonraki bölüm, web uygulamalarının nasıl çalıştığını tanımlamaktadır. Bölüm 3 'de web tabanlı saldırıların saptanması ve engellenmesi tanımlanmıştır. Bölüm 4 'de Web IDS/IPS modeli tanıtılmıştır. Bölüm 5 'de uygulama verilmiştir. Bölüm 6'da konuşlandırma verilmiştir. Bölüm 7'de sonuçlar verilmektedir.

2. Web Uygulaması

Web uygulamaları, kullanıcının web sitesi ile etkileşimini sağlar, böylece bütün arka uç veri sistemleri ile işlemler gerçekleşebilmektedir. Kullanıcı, web uygulamasını kullanarak sisteme veri girebilir veya verilere erişebilir. Kod, kullanıcının sistem verilerine erişmesi ve işlem yapmasını sağlamaktadır. Web uygulaması değişik programlama dilleri ile hazırlanmış olabilir. Web uygulaması bileşenleri Şekil-1'de gösterilmiştir [6]. Olağan web uygulaması URL yapısında, dosya adını parametrelerden ayırmak için "?" karakteri, birden fazla koşul kullanıldığında koşulları ayırmak için "&" karakteri kullanılmaktadır.



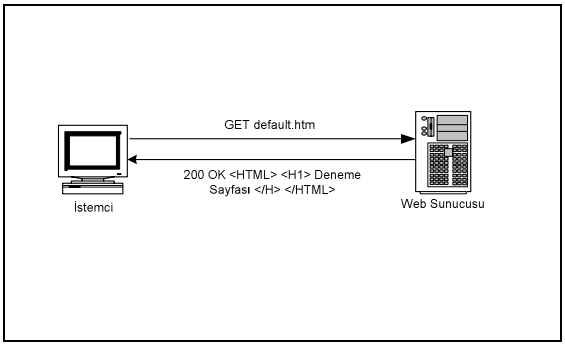
Şekil-1: Web Uygulaması Bileşenleri

Web uygulamalarının nasıl çalıştığını ve nasıl güvenli hale getirilebileceklerini anlayabilmek için aşağıdakilerin olağan davranışları modellenmelidir:

- HTTP İletişim Kuralları
- Web Uygulama Evreleri
- Web Servisleri

2.1. HTTP İletişim Kuralları

HTTP, WWW ortamına veri geçirmek için kullanılan birincil yöntemdir. Şu anda çoğunlukla kullanılan sürüm HTTP 1.1'dir ve RFC 2616'da ayrıntılı olarak tanıtılmıştır [7]. HTTP, istemci isteği ve sunucu yanıtından oluşan iletişim kurallarıdır. İstemci, sunucunun http kapısına (varsayılan 80 ama değişik de olabilir) TCP bağlantısı kurar. İstemci, "GET / HTTP/1.1" ve onu izleyen ve isteği tanımlayan bilgileri içeren MIME iletisi içeren bir TCP paketi ile web sayfasını ister. Sunucu duruma göre bir cevap dizgisi dönecektir. Bu yanıt olumlu olduğunda "200 OK" ve istenilen sayfayı içeren ileti veya olumsuz olduğunda bir hata iletisi veya başka bir bilgi olabilir [7]. Tipik bir HTTP iletişimi Şekil-2'de gösterilmiştir.



Şekil-2: Tipik HTTP İletişimi

HTTP standardı 8 istek yöntemi tanımlar; Get, Head, Post, Put, Delete, Trace, Options ve Connect [7]. Bir web sunucunun en azından Get, Put ve Options yöntemlerini uyguladığı varsayılmaktadır. En yaygın istek yöntemleri aşağıdaki gibidir:

- Get: Web sayfası isteği,
- Head: Meta bilgilerini edinmek için web sayfası isteği,
- Post: Kullanıcı verisinin bir html sayfasına veya örneğin bir form'a gönderilmesi,
- Options: Sunucunun desteklediği http yöntemlerinin döndürülmesidir.

HTTP yanıtının ilk satırına durum satırı denir ve bir sayısal durum kodu (örneğin 200) ve neden tümceciği metini (örneğin "OK") içerir. Neden tümcecikleri isteğe göre uyarlanabilir ve özel durum kodları kullanılabilir. Durum kodlarının listesi için bkz [7].

2.2. Web Uygulama Evreleri

Web uygulamasının çalışması dört evrede incelenebilir:

- Oturum Açma Evresi
- Oturum Evresi
- Hata Evresi
- Oturum Kapama Evresi

2.2.1. Oturum Açma Evresi

Web yazılımını kullanmak için geçilmesi gereken kilit bir evredir. Bu evrede, kullanıcı sisteme kendi kullanıcı kimliği ve şifresini girerek bağlanır. Şifre abecesayısal, sayısal veya özel karakterlerle birleşik bir biçimde olabilir. Saldırganlar/şifre kırıcılar bu evreyi en kolay yoldan "or" anahtar sözcüğünü kullanarak geçmeyi denerler. Bu yöntemde şifre alanında kullanılan söz dizim, kullanılan programlama diline göre değişmekle beraber aşağıdaki gibidir:

'[Herhangi bir şifre dizgisi]' or '1' = '1'

Şifre yanlış olsa da ikinci denklem doğru olacağı için koşulu sağlayacak ve saldırgan sisteme girecektir. Bu tür verilerin girileceği alanlarda 'or' ve benzeri anahtar kelimelerin denetlenmesi ve engellenmesi ile bu tür saldırılardan kaçınmak mümkündür.

Saldırgan, kullanıcı kimliği ve şifrelerini deneme yoluyla da sisteme girmeye çalışacaktır. Bu tür saldırıları engellemek veya zaman kazanmak için, sistem yanlış kullanıcı ve/veya yanlış şifre için aynı hata iletisini vermelidir [9]. Koşut zamanlı başarısız oturum açma girişimlerinde, sisteme giriş sayfası her seferinde artan gecikmelerle geri gelmelidir [10]. Belirli bir sayıdan fazla başarısız oturum açma girişiminden sonra sistem yöneticisine bir uyarı verilebilir veya o kullanıcı ve IP adresi belirli bir süre için engellenebilir [11].

Oturum açma evresi ayrıntılı olarak incelenmeli ve güvenliği sağlanmalıdır. Standart dışı uygulamalar bu evrenin ayrıntılı çözümlemesini zorlaştırmaktadır. Oturum açma sürecinde kullanılan ince ayrıntılar, html ve yanıt kodlarının istatistiksel değerlendirilmesi ile öğrenilebilir [10].

2.2.2. Oturum Evresi

Veri girişi ve verilere erişim bu evrede gerçekleştirilir. Kullanıcının sisteme girdikten sonra davranışları izlenmeli ve sistemi kötüye kullanma girişimleri engellenmeye çalışılmalıdır. Belirli bir kullanıcının o anda sistemde ne yaptığı (kullanıcı durumu) bilinmelidir. HTTP durum bilgisi tutmayan bir iletişim standardıdır. Her HTTP bağlantısı "GET" ile ayrı bir TCP bağlantısı kurar ve hiç bir durum bilgisi tutulmaz. Durum bilgisi tutmak için çerez (cookies), URL değiştirgeleri (URL yeniden yazma), gizli değerler veya meydan okuma/yanıt (challenge/response) yöntemleri kullanılabilir. Ayrıntılı bilgi için bkz. [12] [13]. Gizli değerler kullanılması tehlikeli olduğu için önerilmemektedir [14].

Çerez kullanımında çeşitli güvenlik ve mahremiyet sorunları bulunduğu için durum tutmak için kullanılması tercih edilmemelidir. Çerezler sadece kullanıcı hakkında tanımlayıcı bilgi içermeli, hassas bilgiler tutulmamalıdır [15]. Günümüzde durum yönetimi için çoğunlukla URL yeniden yazma ve ortama özgü oturum nesneleri kullanılmaktadır. Bu teknikler SOAP gibi web servislerinin oturum bilgilerini tutmak için uygun değildir ve yeni tekniklerin kullanılması gerekecektir. Örneğin yeni bir çalışmada SOAP iletişim kurallarının oturum yönetimi için IETF'in oturum başlatma iletişim kurallarının (SIP) kullanımı önerilmektedir. Ayrıntılı bilgi için bkz [13].

2.2.3. Hata Evresi

Hata durumunda sistem bu evreye girmektedir. Kural dışı durumlar ve hata koşulları dikkatlice incelenmelidir. Kural dışı durumlar yaşandığında sistem hassas bilgileri açığa çıkarmamalıdır. Oluşacak hata sayfası saldırgana bilgi vermeyecek şekilde düzenlenmeli, mümkünse daha önceden hazırlanan bir şablon sayfa gösterilmelidir.

2.2.4. Oturum Kapama Evresi

Kullanıcı bu evrede sistemden ayrılmaktadır. Oturum kapama aşağıdaki nedenlerden dolayı olabilmektedir [10]:

- Oturum kapatma bağlantılarına tıklatmak,
- Zaman aşımı,
- Uygulama hatası,
- Oturum sonu.

Oturum kapatma sürecinde kullanılan ince ayrıntılar, html ve yanıt kodlarının istatistiksel değerlendirilmesi ile öğrenilebilmektedir [10].

2.3. Web Servisleri

Günümüzün koşulları çok farklı yerlerden bilgi paylaşımını gerektirmektedir. Ortak çalışmayı, güncel ve geçerli bilgiye ulaşmayı sağlayacak dağıtık bilgi işlem düzeneklerine gereksinim duyulmaktadır. Farklı yerlerde veya farklı firmalarda çalışan çeşitli yazılım ve servislerin birlikte işlerliğini sağlayacak açık standartlar ve iletişim kuralları gerekmektedir. Bu gereksinimler web servislerini doğurmuştur. Web servisi, birlikte işler makineden makineye ağ üzerinden etkileşimi desteklemek için tasarlanmış bir yazılım sistemidir. Genellikle HTTP ve XML tabanlı çözümler kullanılmaktadır [16]. Web servisleri, yazılım sistemlerinin bir servis grubu olarak dağıtık çalıştığı servise dayalı mimariye (SOA) dayanmaktadır [13]. Ayrıntılı bilgi için bkz [16] [13].

3. Web Tabanlı Saldırıları Saptamak ve Engellemek

Özellikle kurumsal ağlarda web tabanlı saldırıları saptayacak ve engelleyecek bir web güvenlik sistemine gereksinim duyulmaktadır. Büyük kurumsal ağlarda ağ yönetim ve güvenlik gruplarının kurumsal web uygulamalarının özellikleri hakkında yeterli bilgiye sahip olmamaları en büyük eksiklik olarak karşımıza çıkmaktadır [1]. Öncelikle kurulacak güvenlik sisteminin web sunucu altyapısı hakkında bilgi sahibi olması hedeflenmelidir [20].

Sunucu makinelerinin ve üzerlerinde çalışan web uygulamalarının bilinen zayıflıklara sahip olup olmadıklarının denetlenmesi için zayıflık (saldırıya açıklık) inceleme süreçleri kullanılabilir. Böylece sistemdeki zayıf noktalar bulunabilecek ve olası saldırı öncesinde önlemler alınarak tehdit ortadan kaldırılabilecektir. Web uygulamalarının zayıflıklarının taranma süreci, ağ/sunucu zayıflık taramalarından daha karmaşıktır. Ayrıntılı bilgi için bkz [10].

Kurumsal ağlarda saptama ve engelleme düzenekleri, kullanıcılar ve web sunucuları arasında konuşlandırılabilir. Bu düzenekler http belirtimlerini ve web uygulaması evrelerini denetlemelidir. Bu sistemler, kullanıcı girdilerinin geçerliliğini sunuculara iletmeden önce denetleyecek ve hatta süzecek şekilde kurulabilirler.

Bilgisayar sistemlerini kötüye kullanmaya yönelik başarılı veya başarısız girişimleri tanımak için Saldırı Saptama Sistemleri (IDS) kullanılmaktadır [17]. Standart IDS'lerdeki temel sorunlar aşağıdaki gibidir:

- IDS işlem kapasitesi: IDS üzerinden geçen her paketi denetlemek zorundadır. Aşırı miktarda ağ trafiği veya bir saldırgan tarafından sistem aşırı yüklenebilir. Eşik değeri aşıldığında, sistem üzerinden geçen paketleri denetleyemez duruma gelecek ve bazı paket denetlemelerini iptal etmek zorunda kalacaktır.
- Günlük büyüklüğü: IDS tarafından yüksek sayıda günlük kayıdı üretilmektedir. Bu günlük kayıdı fazlalığı, ağ yöneticilerinin etkin bir şekilde bu günlükleri incelemelerini ve olası saldırı hakkında bilgi edinmelerini engelleyebilecektir. Bilgilerin kullanışlı olmasının sağlanması için sistemin mümkün olduğunca az ve öz günlük üretmesi hedeflenmelidir.
- Yanlış uyarılar: Saldırı olduğuna dair uyarılar doğru olmayabilir. Bu uyarıların sayısının fazla olması, sistem yöneticisinin incelemeyle zamanını harcamasına yol açtığı gibi, bu uyarılar arasındaki gerçek saldırıları görmesini de engelleyebilecektir.

IDS'in kurulduğu ağı tanıması ve ağdaki birimlerin özelliklerine göre sürekli olarak yapılandırmasını ayarlanması gerekmektedir, bu da ağ farkındalığı özelliği olması demektir. Ağ farkındalığı yeteneğine sahip özerkleşmiş IDS'ler, Hedef Tabanlı IDS (TIDS) olarak adlandırılabilir. Sistem hakkında bilgi sahibi bir IDS, yukarıda anlatılan sorunları azaltacaktır. Sunucu ve ağ hakkında yeterli bilgiye sahip olmama durumunda yaşanabilecek sorunlar, Ptacek ve Newsham'in çalışmasında [23] anlatılmıştır. Eğer saldırgan, sistemler hakkında IDS'den daha fazla bilgiye sahipse, bu bilgileri kullanarak IDS'e yakalanmadan saldırılarını gerçekleştirebilecektir [24].

Saldırı önleme sistemleri (IPS), ağ trafiğini sunuculara gelmeden önce üzerlerinden geçirip (inline) denetleyen ve bazı kesin saldırıları engellemeye yarayan sistemlerdir. IPS'in saptama düzenekleri IDS'in saptama motoruna bağlıdır. Özetle IPS, IDS'in gereksinimlere göre etkinleştirilmiş halidir.

Bu çalışmada web tabanlı saldırılar için uzmanlaşmış bir IDS/IPS ön ürünü sunulmuştur. Sunulan sistem, web altyapısı hakkında bilgi toplayıp daha etkin çalışmak için kurallarını düzenlemektedir. Sistem IDS olarak çalışabildiği gibi, vekil sunucu olarak çalışması sağlandığında IPS olarak da çalışabilecektir.

4. Web IDS/IPS Modeli

Web IDS/IPS, sadece web güvenliği ile ilgilenen uzmanlaşmış bir IDS/IPS sistemidir. IDS olarak çalıştığında, Berkley paket süzgeçleri (BPF) ve bit maske süzgeçleri kullanılarak [19], http dışındaki ağ trafiğinin süzgeçten geçirilmesi söz konusu olabilecektir. Web trafiği sadece tanınmış kapılara göre yapılmayacak, sisteme özgü ayarlar saptanıp tanımlanacaktır. Web altyapısında çalışan sunucular ve kullandıkları web kapıları sistem tarafından saptanacak ve süzgeçler bu bilgilere göre düzenlenecektir. Ayrıntılı bilgi için bkz [20].

Bu sistemin amacı kurum sunucularının ve sunuculardaki tutulan bilgilerin güvenliğidir. Sadece web sunucularına gelen ve giden http trafiği IDS tarafından işlenecektir. Kurumsal web sunucularının HTTP trafiği incelendiğinde, trafiğin çoğunun (%95) sunucu yanıtı ve %5'inin istemci isteği olduğu görülmektedir. Sunucu yanıtının büyük bir kısmı (%95+) paket yüküdür ve küçük bir kısmı (%3-%5) paket başlığıdır. Paketin yük kısmı kurumsal ağlardan gelen içerikle dolu olduğu için ayrıntılı incelenmesine gerek yoktur. Bu nedenle bütün http trafiğinin sadece %10'unun ayrıntılı incelenmesi gerekmektedir [8]. Böylece incelenecek veri azaltılmakta ve IDS'in işlem kapasitesi aşılmamaktadır.

Sistemin etkin çalışabilmesi için, web altyapısına ait bileşenleri ve özelliklerini tanıyor olması gerekmektedir. Bileşenlere ait bilgiler aşağıdaki yöntemlerle temin edilebilir:

- İstatistiksel Veri: Web sunucularının ve uygulamalarının geçerli çalışmasının izlenerek istatistiksel çözümlemenin gerçekleştirilmesidir. Bunun için sistemin eğitilmesi ve yapay anlayış sistemlerinin kullanılması gerekecektir.
- Web Altyapı Bilgi Toplayıcısı : Web sunucu ve web yazılımları hakkında mümkün oldukça çok bilgi toplamayı hedefleyen sistemlerdir. Etken (tarayıcı) ve edilgen (ağ trafiği çözümleme) yöntemleri kullanılabilir.
- Elle girilen veri: Bazı bilgilerin sistem hakkında bilgisi olan ağ yöneticisi tarafından elle girilmesi gerekebilecektir. Bu bilgiler ilk yapılanış için olabileceği gibi başka yöntemlerle toplanan bilgileri düzeltmek için de kullanılabilecektir.

Bu çalışmada sunulan "Web IDS/IPS" sistemi, birlikte işbirliği içerisinde çalışan aşağıdaki alt sistemlerden oluşmaktadır:

- Web Altyapısı Bilgi Toplayıcısı
- Zayıflık Tarama Sistemi
- Gelen Trafik Çözümleyicisi
- Giden Trafik Çözümleyicisi

4.1. Web Altyapısı Bilgi Toplayıcısı

Bu alt sistem, kurumsal ağdaki web sunucular ve web yazılımları hakkında ayrıntılı veri toplamakla görevlidir. Toplanması hedeflenen veriler aşağıdaki gibidir [20]:

- Web sunucusu L3 adresleri
- Web sunucusu kullanılan kapılar (80, 8080 …vb)
- İşletim sistemi türleri (Linux, Windows …vb)
- Web sunucu yazılım türleri ve sürümleri (Apache 2.0, IIS 6.0…vb)
- Web uygulamalarının geliştirildiği programlama dilleri türleri (cgi, php, asp, .net, jsp …vb)
- Uygulama dosya adı
- Uygulama çalışma yolu (Dizin yapısı)
- Kullanılan değiştirgeler ve tipleri

Hedeflenen sistemde, standart dışı URL kullanımındaki öğeler de saptanmaya çalışılmalıdır. Örneğin:

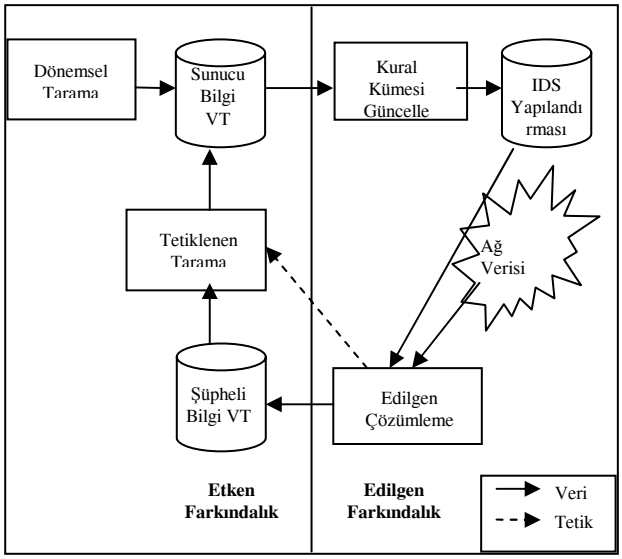
- Değiştirgelerden önce "?" dışında karakterlerin kullanılması,
- Koşulları ayırmak için "&" karakteri dışında karakterlerin kullanılması,
- Uygulamalarda uygulama türünü belirtmeyen dosya uzantılarının kullanılması (Örneğin .html),
- Değişken ve standart dışı yanıt kodlarının kullanılmasıdır.

Belirtilen standart dışı uygulamalar yüzünden yazılımlar hakkında her zaman yeterli bilgi toplanması mümkün olamayabilecektir. Yazılımların mantıklarının anlaşılması zordur ve daha iyi sonuçlar için sistemi bilen bir uzmanın sürecin eniyilemesi için katılması gerekebilecektir. Toplanan bilgiler bir veritabanı dosyasında tutulmalı ve IDS'in yapılanışı toplanan bilgiye göre etkinleştirilmelidir. Bilgi toplama teknikleri aşağıdaki gibi sınıflandırılabilir:

- Edilgen teknik: IDS, dinlediği ağ trafiğini çözümleyerek ağdaki web sunucularını ve yazılımlarını saptamaktadır.
- Etkin teknik: Bu teknikte ayrıntılı veri toplamak için ağ ve uygulama tarayıcıları kullanılmaktadır.
- Karma Teknik: Karma teknikte, edilgen ve etkin teknikler birlikte en iyi başarım için kullanılmaktadır.

Önerilen sistemde karma teknik kullanılmaktadır. Sistem modeli Şekil 3 'de gösterilmiştir. Fazlar aşağıdaki gibidir:

- Etken Çözümleme: Dönemsel taramalar yapılarak web sunucuları hakkında ayrıntılı bilgi toplanmakta ve bilgiler Sunucu Bilgi veritabanında tutulmaktadır.
- IDS Yapılandırmasının Güncellenmesi: Bulunan bilgiler veritabanından alınmakta ve bu bilgilerle IDS yapılandırılması düzenlenmekte ve güncellenmektedir.
- Edilgen Çözümleme: Ağ sürekli olarak dinlenmekte ve daha önceden bulunamamış sunuculara ait bilgiler toplanmakta ve Şüpheli Bilgi veritabanına yazılmaktadır.
- Tetiklenen Tarama: Şüpheli Bilgi veritabanından alınan bilgilerle web sunucu taraması başlatılmakta ve eğer yeni bilgiler bulunursa Sunucu Bilgi veritabanı güncellenmektedir.

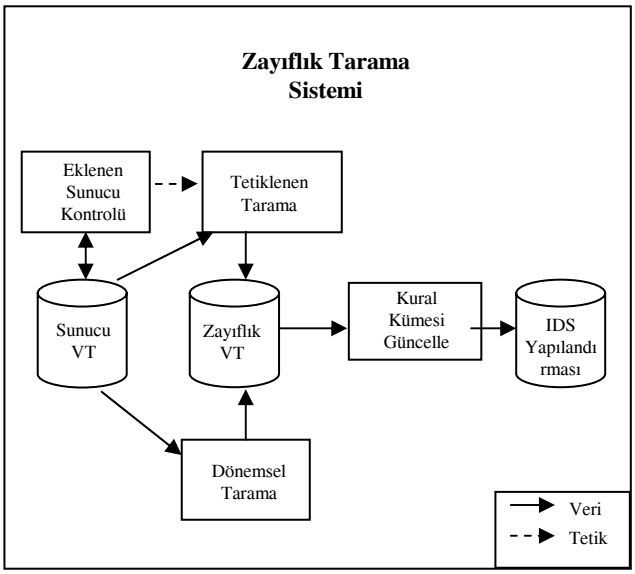


Şekil-3: Etken ve Edilgen Ağ Farkındalığı Sistemi

4.2. Zayıflık Tarama Sistemi

Bulunan web sunucuların, zayıflık (saldırıya açıklık) inceleme sistemleri tarafından denetlenmesi bu aşamada gerçekleşmektedir. Böylece sistemdeki zayıflıklar, bir saldırı yaşanmadan öğrenilecek ve sorumlu kişiler uyarılacaktır. Önerilen sistem modeli Şekil 4 'de gösterilmiştir. Web yazılımlarının kaynak koduna erişim sağlanmadan, dışarıdan özdevinimli (kara kutu) zayıflık taramaları yapılacaktır. Sistem dönemsel olarak sunucu veritabanından sunucuların L3 adreslerini alarak zayıflık taraması yapacaktır. Aynı zamanda veritabanı devamlı denetlenerek yeni eklenen sunucu olduğunda o sunucu için tarama işlemi tetiklenecektir.

Zayıflık tarama süreci sonucunda çıkan raporlar Zayıflık veritabanına yazılacaktır. Bu bilgiler ayrıştırılarak zayıflıklara sahip sistemler için öncelik artırılacak ve ayrıntılı günlük tutma özelliği etkinleştirilecektir. Sistemin IPS olarak çalışması durumunda; belirli bir kritik güvenlik açığının bulunduğunda, açığın bulunduğu sayfaya erişimin kısıtlanması ve engellenmesi söz konusu olabilecektir.



Şekil-4: Zayıflık Tarama Sistemi

4.3. Gelen Trafik Çözümleyicisi

Bu aşama, kurumsal web sunucularına gelen trafiği çözümleyecektir. Aşağıdaki yedi evreden oluşmaktadır:

1. SSL Şifre Çözme
2. Kod Çözme
3. İstem Çözümleme
4. HTTP Paket Durum Korumalı Çözümleme
5. URL Süzgeci
6. Davranış Denetlemesi
7. Web Uygulaması Durum Korumalı Çözümleme

4.3.1. SSL Şifre Çözme

Eğer paketin içerisindeki veri SSL ile şifrelenmişse, gönderilen verinin denetlenebilmesi için bu şifre çözülmelidir. Bu işlem için IDS'in web sunucularına ait özel anahtarları tutması gerekecektir. Bu denetleme kurumun gereksinimlerine ve yapısına göre tümden iptal edilebileceği gibi, anahtarları bilinmeyen sunucuların SSL şifrelenmiş trafiğinin denetimden geçmemesi de sağlanabilir.

4.3.2. Kod Çözme

Saldırganlar saldırı saptama sistemine fark edilmemek için paketi değişik kodlamalarla (unicode, iis_alt_unicode, double encode, iis_flip_slash, full_whitespace) gönderebilmektedirler. Bu evrede; gelen paketlerde eğer farklı kodlamalar kullanıldıysa, diğer evrelere geçmeden önce içerik ASCII karakterlere dönüştürülmektedir.

4.3.3. İstem Çözümleme

Paketler HTTP standartlarına göre denetlenmektedir. Eğer bir paket şüpheli ise günlüğe kaydedilmektedir.

4.3.4. Http Paket Durum korumalı Çözümleme

Eğer web sunucusuna gönderilen veri, çoklu paketten oluşmaktaysa bitiştirilmektedir.

4.3.5. URL Süzgeci

Saldırıları tanımlayan imzalar veya örüntüler kullanılarak URL'ler denetlenmektedir. Eğer URL yapısı şüpheliyse günlüğe kaydedilecektir. Günümüzde bu işlem için, açık kaynak olan ve özel imzaların da eklenebildiği Snort imzalarının kullanılması iyi bir fikir olarak karşımıza çıkmaktadır. Sistemin daha etkin çalışması için imzalar sınıflandırılmalı ve sunucular kendi özelliklerine özgü kurallarla denetlenmelidir. Bu sınıflandırma aşağıdaki bilgilere göre yapılabilir:

- İşletim Sistemi
- Web Sunucu Yazılımı
- Programlama dilleri (uygulama tipi)

4.2.6. Davranış Denetlemesi:

Bir uygulama oturumu saptandığında bu evreye girilecektir. Bu evrede, kullanıcının davranışlarının izin verilen sınırlar içerisinde olup olmadığı denetlenmektedir. Kullanıcıyı oturum boyunca takip etmek için çerezler veya URL yeniden yazma yöntemleri kullanılmalıdır. Eğer şüpheli bir durum saptanırsa o kullanıcının oturum boyunca bütün davranışları kaydedilmelidir.

Davranış denetlemesi için sistem bir vekil sunucu veya bir IPS gibi çalışmalı, ya da web sunucunun kendisi durum yönetimi kullanıyor olmalıdır. Kullanıcı davranış denetimi oldukça karmaşıktır; ayrıntılı istatistikler ve yapay anlayış sistemlerinin kullanılması gerekebilecektir. Kullanıcı tarafından girilen verilerin en azından aşağıdakileri sağlaması gerekmektedir [15]:

- Girilen veri uzunluğu tanımlanan sınır değerden düşük olmalıdır. Bu sınır, arabellek taşma saldırılarını engellemek için kullanılmalıdır.
- Girilen veri sadece belirli karakterleri içermelidir. Sisteme saldırmak için kullanılabilecek karakterlerin ("../", "&", "?", "+", ";", ">", "<", "?" …vb) kullanılmasına mümkün olduğunca izin verilmemeli ve bu karakterlerin bulunduğu içerikler dikkatlice denetlenmelidir [21].

4.3.7. Web Uygulaması Durum korumalı Çözümleme:

Bu evrede uygulamanın durumu denetlenmektedir. İncelenen paket, uygulamaya bir yanıt (veri girişi) olabilir. Veri bir şifre veya uygulamanın alacağı herhangi bir veri olabilir. Eğer veri şüpheli ise günlüğe kaydedilecektir.

4.4. Giden Trafik Çözümleyicisi

Bu çözümlemede amaç, kurumsal web sunuculardaki web uygulamalarının düzgün çalıştığının ve açığa çıkartmamaları gereken hassas bilgileri verip vermediklerinin saptanmasıdır. Bu evrede, gelen trafiğe uygulanandan daha az sayıda URL süzme kuralı kullanılacaktır. Giden trafik 6 evrede incelenecektir:

1. SSL Şifre Çözme
2. Kod Çözme
3. HTTP Paket Durum Korumalı Çözümleme
4. URL Süzgeci
5. Davranış Denetlemesi
6. Web Uygulaması Durum Korumalı Çözümleme

Eğer sistem IPS olarak kullanılmaktaysa aşağıdaki durumlar engellenmeli ve önceden hazırlanan bir şablon sayfayı kullanıcıya döndürmelidir. IDS olarak kullanıldığında ise uyarı düzenekleri çalıştırılmalıdır. Denetlenmesi gereken durumlar:

- Başarısız bir oturum açma girişiminden sonra, sistem tarafından kullanıcı adı veya şifrenin hangisinin yanlış girildiğinin belirtilmesi,
- Bir hata veya saldırı girişimi sonucunda, sistem hakkında (veritabanı veya yazılımdaki değişkenler veya çalıştırılan dosyalar) bilgi veren sayfanın gösterilmesidir.

4.5. IPS Ek Evreleri

Eğer çözüm IPS olarak çalıştırılacaksa, bazı durumlarda paketin içeriğinin değiştirilmesi ve değiştirilmiş paketin iletilmesi

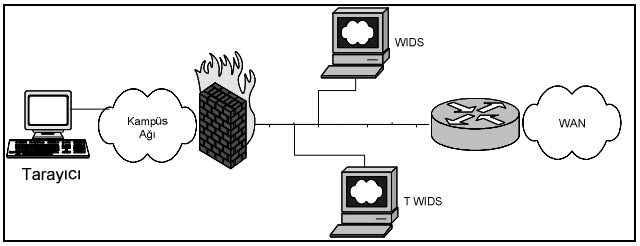
gerekecektir. Bu durumda en azından aşağıdaki işlevlere gereksinim duyulacaktır:

- Kodlama: Paketin içeriğinin gönderildiği ilk kodlamaya döndürülmesi gerekebilecektir.
- SSL Şifreleme: Web sunucuya iletmeden önce paket tekrar şifrelenebilir ya da şifresiz bir biçimde sunucuya gönderilebilir. Sunucunun başarımını artırmak için sunucunun bütün SSL şifreleme işlemlerinin web sunucu yerine IPS tarafından da gerçekleştirilmesi de mümkündür.
- Oturum Yönetimi Bilgisinin Eklenmesi: Sunucunun oturum bilgisi tutmadığı durumlarda, IPS kendisi oturum bilgisi ekleyebilecek ve çıkarabilecektir.
- Web Servislerinin Denetimi: İç veya dış web servislerine erişime izin verilebilir veya engellenebilir. Squid süzgeçleriyle bu tür bir uygulama için bkz [22].

5. Uygulama

Denemelerde Snort saldırı saptama sistemi temel olarak alınmış ve bu sistem, web saldırıları kuralları için etkinleştirilmiştir. Sistemde çoğunlukla yanlış alarmlara yol açan ayarlamalar çıkarılmış veya sorunsuz çalışması için ayarlanmıştır.

Yapılan denemelerde, donanımsal ve yazılımsal olarak aynı özelliklere sahip iki Web IDS, Ege Üniversitesi ağının girişine konuşlandırılmıştır. Konuşlandırma yapısı Şekil 5'de gösterilmiştir. Her iki sunucu da web saldırıları için özelleştirilmiştir. TWIDS, ağ farkındalığı özelliğine sahip ve kuralları etkinleştirilmiş sisteme verilen adlandırmadır. Tarayıcı makine, ağı taramakta ve bulunan bilgileri işleyerek TWIDS'in kurallarını etkinleştirmektedir.



Şekil-5: Deneme Uygulaması

Web sunucu altyapısı hakkında bilgi toplamak için açık kaynak kodlu NMAP tarayıcısı ve AMAP uygulama tespit yazılımı kullanılmaktadır. Bu programların istenilen etkinlikte çalışmasını sağlamak için Perl dilinde hazırlanan kod kullanılmaktadır.

Tarama işlemi için ciddi bir işlemci gücü ve zaman gerektiği için, tarama işlemin mümkün olduğunca basit tutulması gerkecektir. Örneğin bir sunucunun bir kapısının taranması Pentium-II işlemcili bir makinede yaklaşık 0,139 saniye sürmektedir. Herhangi bir sunucunun bütün 65535 kapısının taranması 28 dakika (1,668.916 saniye) sürmektedir. 2000 makinanın bu şekilde taranması ise 39 gün sürecektir ki mantıklı bir süreç değildir. Bu durumda bu sürecin http sunucuları için etkinleştirilmesi gerekmektedir. Yapılan çalışmada, öncelikle ilk taramada amaçlanan 80 kapısında çalışan makinelerin saptanmasıdır. NMAP ile 80 nolu kapılar atanmakta, XML biçimindeki sonuç dosya çözümlenerek 80 kapısında çalışan sunucular saptanmaktadır. Bulunan sunucular çeşitli uygulama tarayıcıları (örn. Httprint, AMAP) ile detaylı olarak incelenmekte ve geliştirilen Perl koduyla çözümlenerek sunucuların özellikleri tanımlanmaktadır. Ağ trafiğinin sürekli dinlenmesi sayesinde bulunan sunucular ve üzerinde çalışan uygulamalar hakkında daha detaylı bilgiler toplanmaktadır. Bunun yanı sıra aktif yöntemle tespit edilmeyen sunucular ve farklı kapılar saptanmaktadır. Bulunan bilgiler daha sonar aktif yöntemle ayrıntılı olarak taranmaktadır. Geliştirilen sistem hakkında ayrıntılı bilgi için bkz [20].

Mesai saatlerinde yapılan denemede, o anda aktif olan 2193 aktif makine taranmış, çözümlenmiş ve IDS 'in yapılandırma dosyaları oluşmuştur. Bu sürecin tümü yaklaşık 25 dakika sürmüştür. Çözümleme sonucunda kampus ağında 199 web sunucusu saptanmıştır. Bulunan web sunucuların %22,6 'sının (45 adet) Windows işletim sistemi, %15 'inin (30 adet) Unix/Linux işletim sistemi üzerinde çalıştığı belirlenmiştir. Bazı sunucuların işletim sistemi tam olarak belirlenememiştir, bu sistemlerin tanımlanması için daha ayrıntılı testler uygulanacaktır. Bulunan sunucuların %19,5 'i (39 adet) IIS, %24 'ü (48 adet) Apache web sunucusu kullanmaktadır. İnceleme sonucunda web sunucusu yazılımı kullanan çok sayıda yazıcı, kesintisiz güç kaynağı, kablosuz ağ cihazı, kamera sistemleri gibi cihazlar tanımlanmıştır. Özellikle ağ yönetiminden izinsiz takılan kablosuz iletişim cihazlarının tanımlanma süreci de bu projenin ek bir hedefi olarak ortaya çıkmıştır. Kablosuz ağ cihazları tanımlanmış ve bu cihazlarda varsayılan şifreler kullanılıp kullanılmadığı denetlenmiştir. Bu cihazların kampus politikasına uygun bir şekilde yapılandırılması sağlanmıştır.

Bunun yanı sıra dinlenen ağın çözümlenmesi ile 80 kapısı dışında çalışan vekil sunucu sistemleri ve özel amaçlı sistemler de ortaya çıkarılmıştır. Bunun yanı sıra uygulamaların bulunduğu sayfalar ve değiştirgeler gibi bilgiler de edilgen yöntemle toplanmıştır. Bulunan bilgilerle IDS'in yapılandırması düzenlenmiştir.

Sistemin izlendiği bir hafta boyunca, iki sistemde gözlenen ortalama günlük uyarı sayısı Tablo-1'de verilmiştir. Tablodan da görüleceği üzere, ağ farkındalığına sahip sistem ağ yöneticisinin kolayca inceleyebileceği miktarda daha anlamlı uyarılara sahip olmaktadır. Bu çalışmaya ait daha ayrıntılı bilgiler için bkz [20].

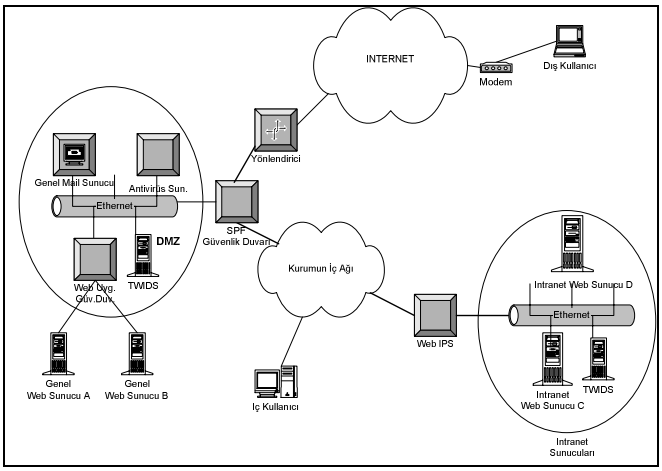
Tablo-1: Uyarıların Karşılaştırılması

	WIDS	TWIDS
Toplam Uyarı Sayısı	4678	185
Kaynak IP Adresleri	699	36
Varış IP Adresleri	30	5
Benzersiz IP Bağları	757	36
Benzersiz Uyarılar	4	3

Zayıflık tarama sistemleri ile üniversiteye ait sunucuların taranma işlemi özdevinimli olarak gerçekleştirilmektedir. Bu süreçte çeşitli açık kaynak kodlu yazılımlar kullanılmaktadır. Makinelerin zayıflık taramaları Nessus yazılımı ile gerçekleştirilmektedir. Web uygulamalarının zayıflık testi için imza tabanlı yazılımlar (örn. Nikto) kullanıldığı gibi, imza bilgisine ihtiyaç duymayan yazılımlar (örn. Wapiti) da kullanılmaktadır. Sistemlerde zayıflık saptandığı durumda özdevinimli olarak e-posta ve bir web sitesi aracılığı ile sistem yöneticilerinin uyarılması sürecinin gerçekleştirilmesi için çalışmalar sürdürülmektedir.

6. Topoloji

Kurumsal ağlarda, değişik gereksinimler için değişik konuşlanmalardan oluşan karma çözümlerin birlikte kullanılması gerekebilecektir. Bu tür bir topoloji (ilinge) Şekil 6'da verilmiştir.



Şekil-6: Kurumsal Web Güvenlik Topolojisi

Önerilen sistemde, kurumun ağı ile dış ağlar arasında bir durum korumalı paket süzücü ağ güvenlik duvarı konuşlanmaktadır. Bu ağ güvenlik duvarı kurumsal ağa giren bütün paketlerle ilgilenecektir. Etkin çalışması açısından OSI'nin ilk dört katmanı için çözümleme yapacaktır. Amaç geçerli olmayan trafiğin mümkün olduğunca azaltılmasıdır. Web IDS, web sunucularının bulunduğu ağ kesimlerinde veya ağ güvenlik duvarının iç ağa bakan bacağında konuşlandırılacaktır. IPS ise daha kritik sunucuların önünde konuşlanacaktır.

Web altyapısı bilgi toplayıcısının etken biriminin, diğer ağ kesimlerinde gereksiz trafik yaratmaması için web sunucularının bulunduğu ağ kesimlerinde olması tercih edilmelidir. Edilgen olarak ağı dinleyecek birim ise web sunucularının bulunduğu ağ kesimlerinde veya güvenlik duvarının iç ağa bakan bacağında konuşlanmalıdır.

7. Sonuç

Web sunucularına yönelik tehditlerin arttığı günümüzde, olabilecek saldırı girişimlerini saptamak ve mümkünse engellemek gerekmektedir. Bu çalışmada bunun için kullanılabilecek bir yapının ana hatları verilmiştir. Web altyapısı hakkında yeterli ve güncel bilgi toplayan bir hedef tabanlı web saldırı saptama sistemi tanıtılmıştır. Aynı zamanda zayıflık tarama sistemlerinin kullanılması ile sistemde oluşabilecek sorunları engellemek için zayıf noktaların bulunması hedeflenmiştir.

Web altyapısı hakkında bilgi toplamak için karma bir teknik kullanılmıştır. Bu çalışma, http iletişim kurallarına özel etkin bir yöntem sunmasıyla benzer çalışmaları genişletmektedir. Amaç herhangi bir anda, sistem hakkında güncel ve doğru bilgiye sahip olmaktır. Bu bilgiler ışığında saldırı saptama sisteminin yapılandırılması düzenlenmiştir. Bu sayede sistem yöneticilerinin daha etkin kullanabileceği az ve öz günlük kayıtlarına ulaşılabilmiştir.

Gelecek bir çalışma olarak, zayıflık tarama sistemlerinin raporlarının ayrıntılı incelenmesi ve IDS kurallarının tekrar düzenlenerek önceliklendirme işleminin yapılması hedeflenmektedir. Zayıflık tarama sistemlerinin daha etkin çalışması sağlanmalıdır. Özellikle oturum açma evresindeki şifre deneme ve kırma tabanlı saldırıların daha etkin saptanmasının sağlanması planlanmaktadır. Saldırı girişimleri birden fazla adımdan oluştuğu için, ardışık girişimlerin takip edilmesi ve saldırı uyarılarının daha doğru ve ayrıntılı olması sağlanmalıdır. Denetleme ve adli inceleme ile ilgili çalışmalar da yapılacak gelecek çalışmalar olarak hedeflenmektedir.

Kaynakça

[1] Karaarslan Enis, Tuglular T, Sengonca, H, 2004. "Enterprise Wide Web Application Security: An Introduction", EICAR 2004.

[2] Zone-h, 2005. Independent observation of web server cybercrimes, 18 Aralık 2005 tarihinde erişilmiştir, http://www.zoneh.org

[3] CSI/FBI, 2005. Computer Crime and Security Survey, Computer Security Institute Publication, 21 Kasım 2005 tarihinde erişilmiştir, http://www.gocsi.com/

[4] OWASP, 2006. OWASP Top Ten Most Critical Web Application Security Vulnerabilities, 12 Ocak 2006 tarihinde erişilmiştir, http://www.owasp.org/documentation/topten.html

[5] Sima C., 2005. Web Application Worms - the next Internet infestation, (In)secure Magazine, Issue 2, pg 17-21, 20 Haziran 2005 tarihinde erişilmiştir, http://www.insecuremagazine.com/INSECURE-Mag-2.pdf

[6] Sanctum, 2003. Anatomy of a Web Application, 26 Aralık 2003 tarihinde erişilmiştir, http://www.sanctuminc.com/solutions/whitepapers/

[7] IETF, 1999. RFC2616, Hypertext Transfer Protocol - HTTP/1.1, 10 Ocak 2006 tarihinde erişilmiştir, http://www.ietf.org/rfc/rfc2616.txt

[8] Roelker D., Norton M., 2002. Snort 2.0: Protocol Flow Analyzer, 29 Mayıs 2004 tarihinde erişilmiştir, http://www.sourcefire.com/products/library.html#wp

[9] Durkee R., 2003. Java Web Application Security, 26 Aralık 2003 tarihinde erişilmiştir, http://www.rd1.net/present/Durkee_RJUG_WebAppSec.pdf

[10] Grossman J., 2004. Challenges of Automated Web Application Scanning "Why automated scanning only solves half the problem.", Blackhat Windows 2004, 12 Ocak 2006 tarihinde erişilmiştir, http://www.whitehatsec.com/presentations/challenges_of_scanning.pdf

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

[12] Newmarch J., 2000. HTTP Session Management, 10 Ocak 2006 tarihinde erişilmiştir, http://jan.netcomp.monash.edu.au/ecommerce/session.html

[13] Dong W., 2005. Adding Session and Transaction Management to XML Web Services by Using SIP, Minor Thesis, Monash University, 10 Ocak 2006 tarihinde erişilmiştir, http://jan.netcomp.monash.edu.au/publications/wendy_thesis.pdf

[14] Robin B., 2003. Web Application Security, Lesson Notes, 26 Eylül 2003 tarihinde erişilmiştir, http://josquin.cti.depaul.edu/~rburke/courses/f03/ect582/notes/w8/lec1106.ppt

[15] Dayioglu B., 2003. Php ve Web Güvenliği, 12 Temmuz 2006 tarihinde erişilmiştir, http://seminer.linux.org.tr/seminer-notlari/web-uygulama-guvenligi.sxi

[16] Wikipedia, 2006. Web Service, 8 Temmuz 2006 tarihinde erişilmiştir, http://en.wikipedia.org/wiki/Web_service

[17] McHugh J.: Intrusion and intrusion detection, International Journal of Information Security, Springer, ISSN: 1615-5262 (Paper), 1615-5270 (Online), Issue: Volume 1 - Number 1 (2001) 14 – 35

[18] Conry-Murray A., 2003. Emerging Technology: Detection vs. Prevention - Evolution or Revolution?, 26 Aralık 2003 tarihinde erişilmiştir, http://www.networkmagazine.com/shared/article/showArticle.jhtml?articleId=9400017

[19] Parker D., 2004. Filtering IDS Packets, 27 Kasım 2005 tarihinde erişilmiştir, http://www.onlamp.com/pub/a/security/2004/06/17/ids_filtering.html

[20] Karaarslan E, Tuglular T, Sengonca, H, 2006. Does Network Awareness Make Difference In Intrusion Detection of Web Attacks, ICHIT 2006.

[21] Chu B., 2002. Application Security, 26 Aralık 2003 tarihinde erişilmiştir, http://www.belkcollege.uncc.edu/nblong/ITIS2300/Application%20Security.ppt

[22] Newmarch J., Huang M., Chua K. G., 2003. Firewalling Web Services

[23] Ptacek T.H., Newsham T.N., 1998. Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection, http://www.snort.org/docs/idspaper/

[24] Snort Users Manual 2.4.0 , 2005.