

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

UYGULAMA KATMANI İÇİN GÜVENLİK DUVARI
GELİŞTİRİLMESİ

METİN ŞAHİN
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

GEBZE
2018

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

UYGULAMA KATMANI İÇİN GÜVENLİK
DUVARI GELİŞTİRİLMESİ

METİN ŞAHİN
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMANI
PROF. DR. İBRAHİM SOĞUKPINAR

GEBZE
2018

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF COMPUTER ENGINEERING

**AN EFFICIENT FIREWALL FOR WEB
APPLICATIONS (EFWA)**

METİN SAHİN
A THESIS SUBMITTED FOR THE DEGREE OF
MASTER OF SCIENCE
DEPARTMENT OF COMPUTER ENGINEERING

THESIS SUPERVISOR
PROF. DR. İBRAHİM SOGUKPINAR

GEBZE
2018

GTÜ Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 03/01/2018 tarih ve 2018/02 sayılı kararıyla oluşturulan jüri tarafından 22/01/2018 tarihinde tez savunma sınavı yapılan METİN ŞAHİN'in tez çalışması BİLGİSAYAR MÜHENDİSLİĞİ Anabilim Dalında YÜKSEK LİSANS tezi olarak kabul edilmiştir.

JÜRİ

ÜYE

(TEZ DANIŞMANI) : PROF. DR. İBRAHİM SOĞUKPINAR



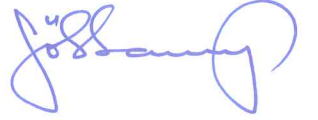
ÜYE

: DOÇ. DR. MEHMET GÖKTÜRK



ÜYE

: DOÇ. DR. ALİ GÖKHAN YAVUZ



ONAY

Gebze Teknik Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun

... / ... /2018 tarih ve 2018/ ... sayılı kararı.

ÖZET

Günümüz teknolojisinin gelişmesi bilgisayar, telefon, tablet vb. iletişim araçlarını http protokolünü kullanmaya yönlendirmiştir. Büyük veri akışı işlemlerin http protokolü üzerinden olması, saldırganlar için bu protokolün hedef haline gelmesine neden olmuştur. Saldırı önleme sistemleri için http trafiğini analiz etmek ve saldırı girişimini tespit etmek önemli bir hal almıştır. Http çalışma prensibine bağlı olarak yapılabilecek saldırılar nelerdir? Web uygulamalarında kullanılan içerik ile ilgili olarak yapılabilecek saldırılara karşı ne gibi önlemler alınabilir? Web uygulamaları güvenliği hakkında ne gibi çalışmalar yapılmıştır? Bu gibi sorunlar incelenmiştir. Daha sonra günlük hayatta kullanılan yöntemleri referans alarak bu sorunlara farklı bir bakış açısı kazandırılmaya çalışılmıştır.

Saldırı tespit sistemi için imza tabanlı, anormallik tabanlı ve karma yöntemler kullanılmaktadır. Saldırganların http trafiğini kullanarak en sık başvurdukları saldırı yöntemleri OWASP'ın (The Open Web Application Security Project) açıklamalarına göre SQL enjeksiyonu, siteler arası betik yazma (XSS), siteler ötesi istek sahteciliği (CRLF enjeksiyonu), zararlı dosya çalıştırma, güvensiz doğrudan nesne başvurusu vb. yöntemlerden oluşmaktadır. Bu saldırılar için anormallik tabanlı saldırı önleme yönteminden bahsedilmektedir. Anormallik saldırı yöntemlerinde bilinmeyen saldırılar oluşturulabilmektedir. Saldırı için oluşturulacak algoritma bu bilinmeyen yöntemlerin tanımlanmasına çalışılmıştır.

Bu çalışmada popüler saldırı türlerini barındıran ve açık kaynaklı olarak erişilebilen CSIC 2010 HTTP veri kümesi kullanılmıştır. Veri setinde 36000 normal ve 25000 den fazla anormallik isteklerinden oluşmaktadır. Veri seti üzerinde yapılan çalışma ile öznel tablosu ortaya çıkarılmıştır. Bu öznel tablosu kullanılarak saldırılara karşı sınıflandırmalar oluşturulmuştur.

Ortaya çıkan sınıflar için Karar Ağacı (C4.5), K En Yakın Komşuluğu (KNN) algoritmaları kullanılmıştır. Sonuç olarak %96.26 gibi önemli sonuca ulaşılmıştır.

Anahtar Kelimeler: Web Saldırıları Algılama, Web Uygulamaları Güvenlik Duvarı, Makine Öğrenme, Veri Madenciliği Algoritmaları

SUMMARY

Today, with the development of technology, communication tools like computers, phones, tablets and etc. use the http as default protocol. The fact that large data stream transactions are over the http protocol has caused this protocol to become a target for attackers. It is important to analyze HTTP traffic for attack prevention systems and to detect attack attempts. What are the attacks that can be done depending on the HTTP working principle? What precautions can be taken against the attack that can be done with respect to the content used in web applications? What kind of work has been done about security of web applications? Such questions have been examined. Later on, we tried to give a different perspective to these problems by taking the methods we use in daily life as reference. Signature based, anomaly based and mixed methods are used for the intrusion detection system. The most common attack methods applied by attackers using http traffic are SQL injection, cross-site scripting (XSS), cross-site request forgery (CRLF injection), malicious file execution, unsafe direct object reference (OSS) etc. For these attacks, we will talk about anomalous attack prevention. Unknown attacks can be created in abnormal attack methods.

The algorithm to be created for the attackers is trying to identify these unknown methods. The study used the CSIC 2010 HTTP dataset, which contains popular types of attacks and is openly accessible. There are 36000 normal and 25000 abnormal requests in the data set. The feature table is revealed by the study on the data set.

This feature table is used to classify against attack. Decision Tree (C4.5), K Nearest Neighborhood (KNN) algorithms were used for the emerging classes. As a result, a significant result of 96.26% has been achieved.

Keywords: Web Attacks Detection, Web Applications Security Walls, Machine Learning, Data Mining Algorithms

TEŞEKKÜR

Başta, yüksek lisans eğitimimde ve akademik hayatımda desteğini ve yardımlarını hiçbir zaman esirgemeyip bilgisi ile bu çalışmamın oluşmasını sağlayan danışmanım Prof. Dr. İbrahim SOĞUKPINAR'a,

ve göstermiş olduğu desteklerinden dolayı sevgili eşim Esengül UZUN ŞAHİN'e en içten teşekkürlerimi sunarım.

İÇİNDEKİLER

ÖZET	v
SUMMARY	vi
TEŞEKKÜR	vii
SİMGELER VE KISALTMALAR DİZİNİ	ix
ŞEKİLLER DİZİNİ	x
TABLolar DİZİNİ	xi
1. GİRİŞ	1
2. TEMEL BİLGİLER VE İLGİLİ ÇALIŞMALAR	3
2.1.Çok Bilinen Saldırı Türleri	3
3. ÖNERİLEN YÖNTEM	7
3.1. Matematiksel Yöntemler	9
3.1.1.Kelime Frekans Analizi	9
3.1.2.Uzunluk Analizi	10
3.1.3.Özel İfade Kontrol Analizi	11
3.2. Kullanılan Algoritmalar	12
3.2.1.Karar Ağacı	13
3.2.2.K En Yakın Komşuluk Algoritması (KNN)	15
3.2.2.1.Uzaklık Fonksiyonları	15
4. DENEYSEL SONUÇ VE DEĞERLENDİRME	17
4.1. Veri Kümesi	17
4.2. Performans Hesaplama	19
4.3. Değerlendirme	20
5. SONUÇ VE ÖNERİLER	25
KAYNAKLAR	26
ÖZGEÇMİŞ	28
EKLER	30

SİMGELER ve KISALTMALAR DİZİNİ

Simgeler ve Açıklamalar

Kisaltmalar

ATD	: Anormallik Tabanlı Denetim
C4.5	: Karar Ağacı Algoritması
CRLF	: Cross-Site Request Forgery
DARPA	: The Defense Advanced Research Projects Agency
GeFS	: General Feture Selection
HTTP	: HyperText Transfer Protocol
İTD	: İmza Tabanlı Denetim
OWASP	: The Open Web Aplication
SQL	: Structured Query Language
WUGD	: Web Uygulaması Güvenlik Duvarı
XSS	: Cross Site Scripting

ŞEKİLLER DİZİNİ

<u>Sekil No:</u>	<u>Sayfa</u>
3.1: Yapılan İşlemlerin Basit Olarak Gösterimi.	7
3.2: Öznitelik Tanımlama.	8
3.3: Öznitelik Belirleme Adımları.	9
3.4: Sınıflandırma Adımları.	13
3.5: Doğruluk Değeri Sözde Kod.	13
3.6: Bilgi Kazanma Sözde Kod.	14
3.7: K NN Sözde Kod.	16
4.1: Özniteliklerin Weka Uygulamasındaki Görünümü.	17
4.2: POST İsteği Veri Örneği.	18
4.3: GET İsteği Veri Örneği.	19
4.4: GET İsteği Parçalaması.	19
4.5: CSIC 2010 Veri Kümesi Kullanan Çalışmaların Karşılaştırma.	22

TABLÖLAR DİZİNİ

<u>Sekil No:</u>	<u>Sayfa</u>
3.1: Popüler Saldırı Anahtar İfadeleri	8
3.2: Popüler Saldırı Anahtar İfadeleri	8
3.3: Frekans Analizine Göre Öznitelik İsimleri	10
3.4: Uzunluk Analizine Göre Öznitelik İsimleri	11
3.5: Özel İfade Kontrol Analizine Göre Öznitelik İsimleri	12
3.6: Karar Ağacı Algoritması Başarı Oranları	14
3.7: 3 NN Algoritması Başarı Oranları	16
4.1: Normal ve Anormallik İstek Sayısı	19
4.2: İstek Türleri Sayısı	20
4.3: Karışıklık Matrisi	20
4.4: Farklı İstek Sayılarına Göre Doğruluk Değeri	21
4.5: Öznitelik Vektörleri için Sınıflandırma Algoritmalarının Doğruluk Değerleri	22
4.6: CSIC 2010 veri kümesi kullanan çalışmaların Karşılaştırma	24

1.GİRİŞ

Teknolojinin hızla gelişmesi ile birlikte günlük hayata dair birçok faaliyetin web ortamına taşınmasına sebep olmuş bu da web uygulamaları(e-ticaret, e-devlet, e-sağlık vb.) güvenliğinin önemini arttırmıştır. Web uygulama sitelerinden %92'si çeşitli saldırılara maruz kalmakta, bu saldırılarının ise %70 gibi büyük bir oranı başarıya ulaşmış, istediği sonuçları almışlardır. [1] Wannacry virüsü son günlerde bütün dünyada etkisini göstermektedir. İnternet sayesinde dünyanın herhangi bir yerinde herhangi bir yerine saldırı yapıp çok büyük maddi ve manevi zararlar verilebilmektedir. Web ortamında saldırılar ağ katmanı ve uygulama katmanı kısımlarından yoğun olarak yapılmaktadır. Saldırı önleyici sistemlerin ağ katmanında aldıkları önlemler nedeniyle saldırganlar uygulama katmanına yönelmektedir.[2] Bu çalışmada yapılan saldırı önleyici sistemin uygulama katmanı seviyesinde saldırı analizi yaptığı görülmektedir. Uygulama katmanı web uygulamaları iletişim için http yapısını kullanmaktadır. Genel olarak saldırılar bu protokol üzerinden gelmektedir. Saldırı önleyici sistemler http trafik ve anormalliği hakkında çeşitli çalışmalar yapmışlardır. Bu konu ile alakalı gönüllü olarak en popüler çalışmalar yapan topluluk OWASP topluluğudur. Bu topluluğun en son yayınladığı popüler saldırı istatistiğini göre sıralama şu şekildedir. SQL enjeksiyon, kırık kimlik doğrulama yöntemi, siteler arası komut çalıştırma(XSS) şeklinde devam etmektedir.[3]Saldırı önleyici sistemler genel olarak saldırı tespit sistemleri için imza tabanlı ve anormallik tabanlı yöntemler kullanmaktadır. Açık kaynaklı ve imza tabanlı bir modül olarak örneklenecek olan bu çalışma Modsecurity imza tabanlı olarak çalışır ve bilinen saldırılar için etkilidir. İmza tabanlı saldırı yöntemleri için büyük bir zafiyet olan sıfırıncı gün saldırısına çok sıkı bir politika kullanmaktadır.[2]Bir diğer yöntem olan anormallik tabanlı saldırı önleyici sistem ise gelen trafiği bir takım öznelikler kullanarak sınıflar oluşturmaktadır. Bu sınıflar için normal davranış ve anormallik davranış olarak istekleri etiketlemektedir. Yapılan sistemin ise sürekli olarak yeni saldırılar öğrenmeye yönelik olması gerekmektedir.

Hwang ve diğerleri [17] yaptıkları çalışmada yeni bir hibrit saldırı tespit sistemin tasarımı gerçekleştirmişlerdir. Hibrit model düşük, yanlış, pozitif oranına sahip imza tabanlı denetim modelinin ve bilinmeyen yeni saldırıları denetleyen anormallik tabanlı denetim modelinin avantajlarını birleştirmektedir. SNORT'da bulunan

anormallik verilerin belirlenmesini saęlayan anormallik denetim modeliyle imza üretimi gerçekleştirilmiştir. Hibrit saldırı tespit sisteminin anormallik denetim süreci ile tespit edilen imzalar, daha hızlı denetim gerçekleştirmek için SNORT imza veri tabanına eklenmektedir. Deneysel sonuçlara göre %60 denetim oranı elde edilmiştir. anormallik denetimle üretilen imzalar SNORT performansını %33 oranında artırmıştır. Yapılan bu çalışma mimari olarak yazarların yaptıkları çalışmaya benzer olsa da bu çalışma web tabanlı saldırıların denetimini gerçekleştirmek için geliştirilmiş hibrit bir modeldir. Fakat yazarların yaptığı çalışma aę tabanlı saldırıların denetimini yapmak için geliştirilmiştir.

Hendry ve dięerleri [18] yaptıkları çalışmada, aę tabanlı saldırılara karşı anormallik tabanlı denetim algoritması gerçekleştirecek, imza üretimi yapmışlardır. Böylece imza tabanlı denetimin sıfır gün saldırılarına karşı etkili olmama özellięi bertaraf edilmiştir. İmza üretimi için hibrit, danışmanlı ve danışmansız sınıflandırma algoritması önerilmiştir. Gerçek zamanlı olarak oluşturulan imzalar yeni saldırıları denetlemek için gerçek zamanlı olarak devreye girmektedir. Yazarlar farklı yöntemler kullanarak önerilen çalışmaya benzer hibrit bir çalışma gerçekleştirmişlerdir, ama yazarların yaptığı çalışmada aę tabanlı yapılan saldırılara karşı denetim gerçekleştirilirken, yapılan bu çalışmada web tabanlı saldırılara karşı denetim gerçekleştirilmesi amaçlanmıştır.

Bu önerilen modelin dięer çalışmadan farkı hem HTTP isteklerine karşı hibrit bir denetim modelinin önerilmesi, hem de web tabanlı saldırıların en çok yapıldığı HTTP istek satırının denetimini gerçekleştirmesidir. Bu çalışmada web uygulamalarına yapılan saldırıları önlemek için anormallik tabanlı bir yöntem geliştirilmiştir. Yöntem, bilinen saldırı yöntemlerinde kullanılan anahtar ifadeler ve bu ifadeler ile saldırılar arasındaki ilişkisinin ortaya çıkarılmasına dayanır. Buna baęlı olarak da seçilmiş olunan özniteliklerin hesaplanan kazanç deęerlerinin büyüklüęü normal ile anormallik isteklerinin birbirinden ayrılmasına yardımcı olmaktadır.

Makalenin kalan bölümleri aşığıdaki şekilde düzenlenmiştir: Bölüm 2’de temel bilgiler ve ilgili çalışmalar, bölüm 3’de önerilen yöntem ve kullanılan algoritmalar, bölüm 4’de deneysel sonuçlar ve deęerlendirme, bölüm 5’de Sonuç ve gelecek çalışmalar verilmiştir

2.TEMEL BİLGİLER VE İLGİLİ ÇALIŞMALAR

2.1.Çok Bilinen Saldırı Türleri

Güvenlik araştırma topluluklarının keşiflerine göre ilk on saldırı, yöntemlerin tanımlanmasına yön vermiştir. Bu yöntemlerin sıralaması zaman zaman değişmektedir fakat popülerliği devam etmektedir. Popüler saldırı yöntemlerinden bir kaç tanesini aşağıda açıklanmıştır.

- SQL Enjeksiyonu

SQL kodlarıyla veri tabanından yapılan sorgular sonucu ortaya çıkan zayıflıkları saldırganların kullanması ile yapılan saldırı yöntemidir. Bu tarz saldırılar için veri alışverişi sırasında sql sorgu parametrelerinin çok iyi filtrelenmesi gereklidir.

- Siteler Arası Betik Yazma

Bir web uygulamasında kullanıcının veri girişi yapabileceği dinamik alanlarda script kodları kullanarak zararlı kod gönderilme yöntemidir. Bu tarz durumdan korunmak için script kullanabilecek alanlara engel koyma mesela `<script></script>` bu gibi işlemler için özel çözümler üretilmelidir.

- Siteler Ötesi İstek Sahteciliği (CSRF)

Bu saldırıda kurbanların farkında olmadan web uygulamasına komut göndermesi temeline dayanır. CSRF açıkları, gelen taleplerde oturum kontrolü yapmayı unutması nedeniyle gerçekleşir.

- Güvensiz Doğrudan Nesne Başvurusu

Doğrudan Nesne Referanslama; geliştirici dosya, izin, veri kümesi kaydı gibi bir bilgiyi URL veya form parametresi olarak alıp bunu uygulamaya referans olarak tanımladığı zaman oluşur. Böylece saldırgan referansı manipüle ederek yetkisi olmayan nesnelere erişebilir.

- Dağıtık Hizmet Engelleme

Amaç hedefteki sunucunun bütün kaynakların (Memory, CPU, Network, vb.) çalışamaz hale gelmesini sağlamaktır. Saldırlara çözüm olarak işletim sistemini kuvvetlendirilerek log kayıtları periyodik olarak denetlenmeli ve takibi yapılmalıdır. Kaynağı yoğun kullanan sistemler için özel bir protokol belirleyerek kullanılan bant genişliğini sınırlandırmak, sisteme erişimi istenmeyen ip ve portlar için ve sadece gerekli iletişimi sağlamak için güvenlik duvarını yapılandırılmalıdır. Ayrıca hedef ip trafikleri üzerinden anormallik tabanlı yöntem belirlenmeli, ağ üzerinde şüpheli bir durum olduğunda ikinci aşamaya geçilmeli bu aşamada kullanıcıyı davranışına göre saldırıda olup olmadığı tespit edilmelidir.

- Kırık Kimlik Doğrulama ve Oturum Yönetimi

Oturum anahtarlarının korunmaması ya da düzgün bir şekilde doğrulanmaması nedeniyle ortaya çıkmaktadır. Saldırganlar şifrelerden ve kimlik denetimi ögelerinden kullanıcıya ait diğer bilgilerin elde edilmesi amaçlanmaktadır.

- Kaba Kuvvet Yöntemi

Çeşitli şifre kombinasyonlarının denenerek hedefe sızmaya çalışma yöntemidir. Kaba Kuvvet Yönteminden korunmak için şifreleme işlemleri kuvvetlendirilmeli örnek olarak password seçiminde tahmini zor, büyük küçük harf, nokta virgül ve ascii karakterleri kullanılmalıdır. İlave olarak log girişi için belirli bir zaman koyma, belirli birkaç girişten sonra sistemin durdurulması bir çözümdür.

- Dizin Aşım Açıklığı (PathTraversal)

Uygulamalar sunucuda bulunan dosyaları direkt isimleriyle çağırabilir. Path değiştirilerek erişilmesi istenmeyen dosyalara uygulama aracılığıyla erişim sağlanabilir. Yama uygulamalara karşı web sunucu düzenli şekilde güncelleme yapmalı ayrıca dinamik sayfalarından korumak ve sakıncalı kullanıcıların dosyalara direkt erişimini engelleme yöntemlerden bir tanesidir.

- Zararlı Dosya Çalıştırma

Uzaktan Dosya Ekleme (RFI) ataklarına karşı zayıflıkları olan kod, saldırganın sunucunun güvenliğini tehlikeye düşürebilecek kadar tehlikeli düşmanca kodlar ve veriler yüklemesine yol açar. Kötü amaçlı dosya çalıştırılması kullanıcıdan dosya adı

veya dosya kabul eden PHP, XML veya herhangi bir çerçeveyi (framework) etkileyebilir

- Güvensiz Kriptografik Depolama

Web uygulamaları verilerin ve bilgilerin güvenliğini sağlamak için nadiren kriptografik fonksiyonları kullanırlar. Saldırganlar zayıfça korunan veriyi, kimlik hırsızlığı ve kredi kartı dolandırıcılığı gibi diğer suçları işlemek için kullanırlar.

- Güvensiz İletişimler

Uygulamalar önemli iletişim bilgilerini ve veri alışverişlerini korumaları gerekirken, çoğu zaman ağ trafiğini şifrelemekte yetersiz kalmaktadırlar.

- URL Erişimini Kısıtlamada

Bozukluk Çoğu zaman uygulamalar, bağlantı ve URL satırlarını yetkili olmayan kullanıcılara sadece göstermeyerek hassas işlevselliği korumaktadırlar.Saldırganlar bu URL'lere doğrudan erişerek, yetkisi olmayan işlemlerigerçekleştirmek için bu zayıflığı kullanabilirler.

Web Uygulama Güvenlik Duvarı ile ilgili literatürde birçok çalışma mevcuttur. WUGD teknik olarak web trafiğini analiz eder ve sqlekleme siteler arası komut çalıştırma, parametre veya urltahrifi, oturum ele geçirme gibi saldırıları arabellek aşımına engelleme gibi yukarıda açıkladığım saldırı yöntemlerini kullanarak sisteme zarar vermeye çalışan saldırganları önlemeye çalışmaktadır.

Bunlardan bu çalışmaya ilham kaynağı olan birkaç çalışmadan bahsedilmektedir. Kruegel ve arkadaşları yaptıkları çalışmada yöntem olarak karakter istatistiğini parametre uzunluğu ve karakter dağılımı ve karakter yapısı ile sırasına bakarak inceler. Bu metot sorgu parametresine odaklanır fakat http cevabını ihmal etmektedir.[5][6]

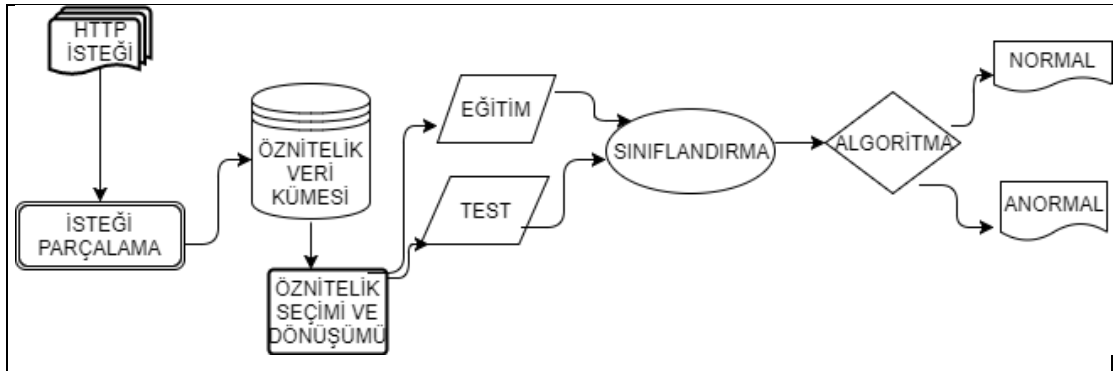
SQLrand isimli çalışmada ise sqlanahtar kelimeleri yerine rasgele talimatlar kullanarak sorgu yapar[7] . Rastgele olmayan anahtarları ve veri tabanı kullanarak vekil filtrelemede başarılı olurlar. Bu sayede rastgele küme ile zararlı kod eklemenin önüne geçilmiş olur.

Bazı web güvenlik duvarı siteler arası betik yazma saldırılarından korunmak için statik ve dinamik analizler yapmaktadır.[8] Amnesia isimli çalışmada ise sql enjeksiyondan korunma için statik analiz ve çalışma zamanı denetimi yöntemlerini

kullanılmaktadırlar. Bu yöntem illegal işlemleri tespit eder. Bununla birlikte kaynak kodunda incelenmesi gerekmektedir[9]. Nguyen ve arkadaşları [10] yaptıkları çalışmada web saldırı denetimi için genel öznitelik seçimi (GeFS) kullanarak 30 farklı öznitelik ortaya çıkarak denetim yapmışlardır. Çalışmalarında siteler arası kod yazma (xss), sqlenjeksiyonu, basit indeks erişim protokolü enjeksiyonu (LDAP),xmldosyasının parçasının adreslemesi enjeksiyonu (XPATH)saldırı türleri için sınıflandırma yapılmıştır. Basile ve arkadaşları [11] yaptıkları çalışmada web uygulama güvenlik duvarında paket filtrelemek için anormallik denetim gerçekleştirmişlerdir. Önerdikleri modelde düzenli ifadeler ile belirtilen metin tabanlı içerik filtreleme yapmışlardır. Modelin etkinliği http proxy olan SQUID'in erişim kontrol özelliğine karşı başarılı bir şekilde test edilmiştir. Torrano-Gimenez ve arkadaşları [12] yaptıkları çalışmada, anormallik tabanlı denetim sonucunda bilinmeyen web tabanlı saldırıları tespit edebilen bir web uygulama güvenlik duvarı geliştirmişlerdir. Geliştirilen model bir http isteğinin saldırı olup olmadığına bir xmldosyasının yardımı ile karar vermektedir. XML dosyası web uygulamasını hedef alan istatistiksel olarak normal davranışlara sahip yapay zekâ tabanlı olarak üretilmiş normal http isteklerini arındırmaktadır. Normal davranışın dışında sapma gösteren istekler, sistem tarafından anormallik olarak nitelendirilmektedir. Tekerek ve arkadaşları http isteklerinin anormallik tespitleri, alfa nümerik karakter, harf frekans ve cümle uzunluğu olmak üzere üç öznitelik kullanılarak gerçekleştirilmiştir. ATD (Anormallik Tabanlı Denetim) yöntemiyle elde edilen sonuçlar Bayes sınıflandırma kullanılarak itd(imza tabanlı denetim) veri tabanı güncellenmektedir.[13] itd hızlı çalışırken sıfır gün saldırılarına karşı etkili değildir. atdyöntemi ise sıfır gün saldırılarında etkilidir, fakat itd, atdmethoduna göre daha hızlı çalışmaktadır[14]Cho ve arkadaşları [15] ise yaptıkları çalışmada kullanıcılar tarafından yapılan istekler ile web sayfalarının benzer özniteliklere sahip olduğunu tespit etmiştir. Web erişim verilerinden web oturumlarını açığa çıkarmak için Bayes parametre tahmin tekniği kullanılarak, oturum bilgilerinden anormallik tahmini gerçekleştirilen oturum anormallik tespiti (Session Anomaly Detection - SAD) ismi verilen bir yöntem geliştirilmiştir.Bu çalışmada web uygulamasına yapılan saldırılar, oturum bilgileri kullanılarak Bayes sınıflandırma yöntemi ile anormallik denetim gerçekleştirilerek tespit edilmiştir. Sadece oturum bilgileri kullanılarak ve anormallik denetim yapılarak web uygulamalarına yapılan saldırıların denetimi yetersiz kalmaktadır.

3.ÖNERİLEN YÖNTEM

Gerçek hayatta suçlar (hırsızlık, cinayet, yaralama vb.) kendi içlerinde de oluş şekillerine göre farklılık göstermektedir. Suçlarda kullanılan suç aletleri ve suçların oluş biçimleri bu çalışmaya ilham kaynağı olmuştur. Bu çalışmanın diğerlerinden farkı, saldırı yöntemleri farklılığı ile saldırılar arasındaki bağı kullanarak sınıflandırma için öznitelik vektörlerinin ortaya çıkarılmasıdır. Referanslarda verilen çalışmalarda detaylı incelendiğinde öznitelik seçimi sınıflandırması için Tablo 3.1 de ifade edilen anahtar özniteliklerin kullanılabileceği görülmüştür. Buna ek olarak ise metin sınıflandırma algoritmaları ile http istekleri incelendiğinde bazı ifadeler diğerlerinden farklı olarak değerler atandığı bazı ifadelerin ise değer beklediği bazı ifadelerin de normal davranış sergileyen isteklerde yoğunca kullanıldığı görülmüştür. [19] Bu yaklaşım ile Tablo 3.2 de ifade edilen anahtar öznitelikler ortaya çıkartılmıştır. Karar ağacı algoritması ile bu özniteliklerin kazanç değerleri hesaplanmış sınıflandırma için bir ölçü olabileceği görülmüştür.



Şekil 3.1: Yapılan işlemlerin basit olarak gösterimi.

Bu çalışmada http isteği iki kısma isteği parçalama kısmında ayrılmaktadır. Tablo 3.1 ve Tablo 3.2 de istek üzerinde yapılan çalışma neticesinde owasp'ın da ifade ettiği (SQL enjeksiyonu, Arabellek Taşması, Bilgi Toplama, Dosyaların İfşa Edilmesi, Siteler Ötesi İstek Sahteciliği Enjeksiyonu, Siteler Arası Betik Yazama, Parametre Müdahalesi) popüler saldırı türleri için saldırı önleme sistemlerinde kullanılabilecek anahtar kümeler göz önünde bulundurularak oluşturulmuştur. Bu özniteliklerin önemi ise karar ağacı algoritması yardımı ile kazanç değeri hesaplanır

ve belirli bir eşik değerin üzerindeki öznitelikler önemli öznitelikler olarak değerlendirilir. Elde edilen bu önemli öznitelikler eğitim aşamasında eğitilir ve teste sokulur sınıflandırma aşamasında önemli öznitelikler ile karar ağacı veya KNN algoritmaları ile isteğin normal veya anormallik davranışına karar verilmektedir.

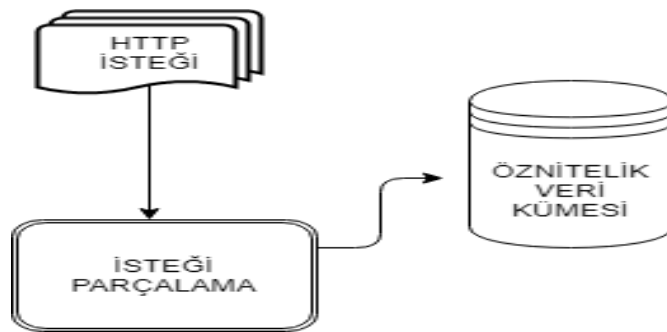
Tablo 3.1:Popüler Saldırı Anahtar İfadeleri

Parametreler	Parametre Değerleri
SQL ifadeleri	Alter, Create, Delete, Select vb.
JSS ifadeleri	Document, Script, Body,!>, Input, vb.
Özel karakterler	!, %23, %26,',%29,%2A,%2B, %2D vb.
Dosya uzantıları	.db, .dbf, .log, .mdb, .sav, .sql, .tar, .exe vb.
İstek tipleri	Get,Post,Put vb.

Tablo 3.2: Özel saldırı anahtar ifadeleri

Parametreler	Parametre Değerleri
Anlamlı ifadeler	<i>remember, B1, modo, login, passwordvb</i>
Nümerik ifadeler	<i>ntc, cp, precio, id vb.</i>
Zararsız ifadeler	<i>.css,.jsp,.gif,.js,.jpg vb.</i>

Tablo 3.1 ve Tablo 3.2 de ifade edilen parametre değerleri için kelime frekans analizi, Özel ifade kontrol analizi, Uzunluk Analizi matematiksel yöntemleri kullanarak 25 adet alt öznitelik çıkartılmıştır.

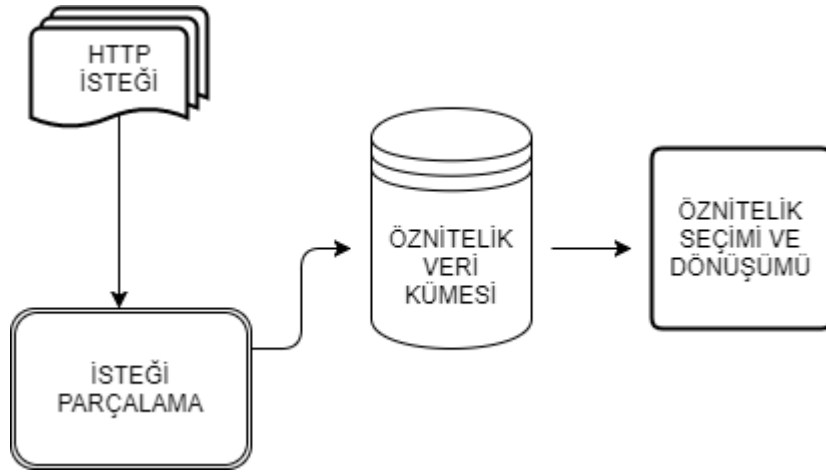


Şekil 3.2: Öznitelik Tanımlama.

3.1. Matematiksel Yöntemler

3.1.1. Kelime Frekans Analizi

Saldırı yöntemleri için Tablo 3.1 ve Tablo 3.2 de belirtilen anahtar kelimeler Şekil 3.3 de ifade edilen den suç türlerine göre öznitelik seçimi ve dönüşümü aşamasında alt sınıf oluşturulmuş, her bir alt sınıf için ayrı ayrı Şekil 3.3 de aşağıdaki matematiksel dönüşüm yöntemi kullanarak Tablo 3.3 de gösterilen öznitelik tablosu oluşturulmuştur.



Şekil 3.3: Öznitelik belirleme adımları(Tablo 3.3,3.4,3.5).

$$T = \sum_{i=0}^N F_i \in H \quad (3.1)$$

Burada;

- T= Toplam değeri
- H=Http isteği
- N=İsteği oluşturan karakter sayısı
- F=K_i

Tablo 3.3: Frekans Analizine Göre Öznitelik İsimleri

S.N	Öznitelikİsimleri
1	sqlscount
2	jsscount
3	pathfilecount
4	special_charcount
5	urlsqlscount
6	urljsscount
7	urlpathfilecount
8	urlspecial_charcount
9	url_passive_slash_count
10	file_path_percent_count

Burada sqlcount da SQL parametrelerinin veri kümesinde her bir istek için ne kadar geçtiğinin hesaplanmaktadır. Diğer öznitelik isimleri de tablo 3.1ve tablo 3.2 deki öznitelikler kullanılarak her istek için değerleri hesaplanmaktadır.

3.1.2.Uzunluk Analizi

Anormallik tabanlı saldırı önleme sistemlerinin kullandığı uzunluk analizi Şekil 3.3 de ifade edilen öznitelik seçimi ve dönüşümü işlemi sırasında aşağıda (3.2) yöntemde ifade edilen matematiksel dönüşümden faydalanılarak sorgu parametresinin tamamı ve url kısmı için Tablo 3.4 de ifade edilen öznitelik isimleri ortaya çıkartılmıştır.

$$T = Length (H) \quad (3.2)$$

- T= Toplam değeri
- H=Http isteği

Tablo 3.4: Uzunluk Analizine Göre Öznitelik İsimleri

S.N	Öznitelikİsimleri
1	<i>totallength</i>
2	<i>urllength</i>

Burada her bir istek iki parçaya ayrılıyor ve kümeleme için uzunluk değeri hesaplanmaktadır.

3.1.3.Özel İfade Kontrol Analizi

Bu kısımda saldırılar için belirlenen Tablo 3.1 ve Tablo 3.2 deki anahtar kelimeler Şekil 3.1 de ifade edilen öznitelik seçimi ve dönüşümü aşamasında her birHTTP istek için (3.2) numaralı denklem kullanılarak kontroller yapılmış Tablo 3.5 de ifade edilen öznitelik isimleri ortaya çıkmıştır.

$$P = \sum_{i=0}^N (F_i \in H) / (T + 1) \quad (3.3)$$

- T= Toplam değeri
- H=Http isteği
- N=İsteği oluşturan karakter sayısı
- F=Ki
- P= Her bir elamanın http isteğinde olup olmadığına bakılır.

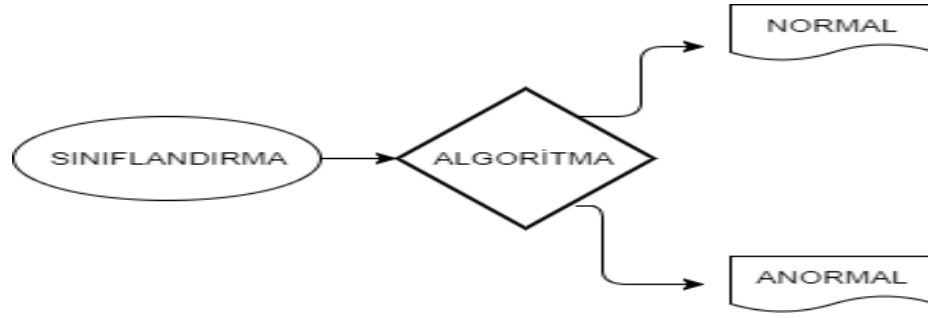
Tablo 3.5: Özel İfade Kontrol Analizine Göre Öznitelik İsimleri

S.N	Öznitelikİsimleri
1	sqlsexists
2	jssexists
3	pathfileexists
4	urlsqlsexists
5	urljssexists
6	urlpathfileexists
7	urlspecial_charexists
8	urldoubleextension
9	unknown_vars
10	non_numeric
11	urlslash_after_extension
12	good_extension
13	method_type

Burada sqlsexists da SQL parametrelerinin veri kümesinde her bir istek için ve bölümlenen url kısmı içinde geçip geçmediğine bakılmaktadır. Diğer öznitelik isimleri de Tablo 3.1ve Tablo 3.2 deki öznitelikler kullanılarak her istek için değerleri hesaplanmaktadır.

3.2.Kullanılan Algoritmalar

Saldırı türlerinde kullanılan yöntemlerden de faydalanılarak ortaya çıkarılan öznitelik tablosu Şekil3.1 de ifade edilen sınıflandırma kısmında kullanılmaktadır. Bu kısımda WEKA uygulamasının paket algoritma programlarından karar ağacı algoritması ve k en yakın komşuluğu algoritmaları kullanılmıştır. İstekler normal ve anormallik davranışlı olarak sınıflara ayrılmıştır.



Şekil 3.4: Sınıflandırma adımları.

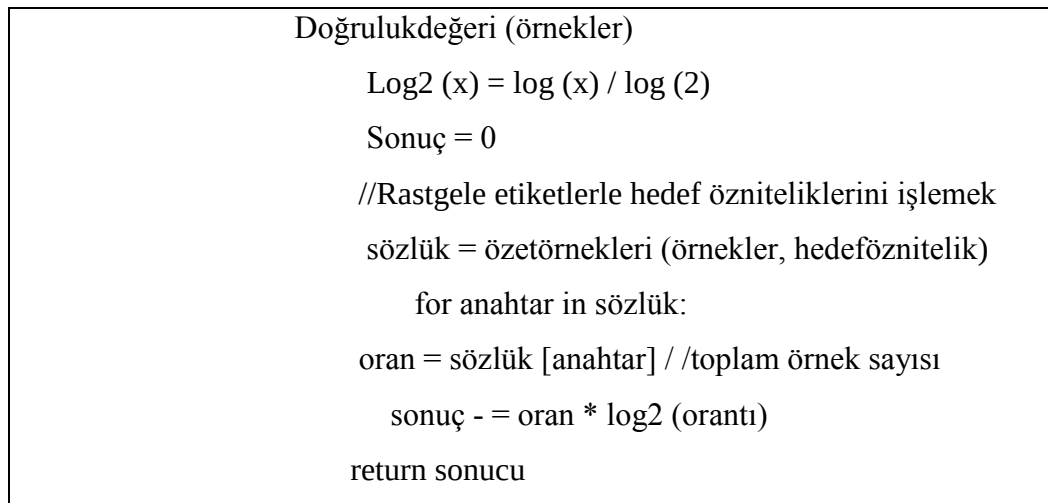
3.2.1.Karar Ağacı

Karar ağacı algoritmaları sınıflandırma için iyi bir çözüm yöntemidir. Optimum bir karar ağacı için ise kural gereklidir. Entropi (doğruluk değeri) formülünü kullanarak (3.4) ile hesaplanmıştır. Kazanç için kullanılan denklemlerin sözde kodu Şekil 3.4 ve Şekil 3.3 de ifade edilmiştir.[18]

$$\text{Entropi} = \text{Info}(D) = \sum_{p=1}^M S_p (\log_2 S_p) \quad (3.4)$$

$M=S_k$ aldığı farklı değerler

S_k =Veri kümesindeki sonuç bilgisidir. Yani normal ya da anormallik olması durumudur.



Şekil 3.5: Doğruluk Değeri Sözde Kod.

```

Bilgikazancı (örnekler, nitelik, doğrulukdeğeri)
kazanç = doğrulukdeğeri//entropi
for değer in öznitelikdeğeri(örnekler, öznitelik):
    sub = altküme (örnekler, nitelik, değer)
    kazanç - = (sub değeri) / (toplam örnek sayısı) * doğruluk
değeri (sub)
return kazancı

```

Şekil 3.6: Bilgi Kazanma Sözde Kod.

Doğruluk değeri hesabını yukarıda ki K kümesinin her alt özelliğini kullanarak oluşturulan öznitelik tablosu için (3.5) kullanarak bilgi değerini hesaplanmaktadır.

$$\text{Sonuç} = \text{kazanç} (A) = \text{Info}(D) - \text{InfoA}(D) \quad (3.5)$$

Sonuç olarak (3.4) de hesaplanan doğruluk değeri ve bilgi değerlerine göre her bir özelliğin kazancını hesaplanması yapılmaktadır. En yüksek kazançlı özelliği kök kabul edip dallar için her bir basamakta tekrar (3.5) kullanarak kazanç hesabı yapılarak karar ağacı oluşturulmaktadır.

Bilgi kazancı (InformationGain) seçilen özniteliklerin sınıflandırma için ne kadar önemli olduğunu da göstermektedir. Bilgi kazancı doğruluk değeri hesaplamasına göre yapılmaktadır. Bilgi kazancı en büyük olan öznitelik bize sınıflandırma için en önemli özelliğidir ayrıca bu öznitelik ağacın kökünü de ifade etmektedir. Sonuç olarak doğruluk değeri rastgele ligin belirsizliğin ve beklenen durumun ortaya çıkma olasılığını gösterir.

Tablo 3.6: Karar Ağacı Algoritması Başarı Oranları

	Öngörülen Sınıf	
	Sınıf=1	Sınıf=0
Sınıf=1	10488	260
Sınıf=0	504	7067

Burada doğru olması gereken 260 tane küme elemanın yanlış olarak tahmin edildiğini yanlış olması gereken 504 değer ise doğru olarak tahmin edildiğini görülmektedir. Başarı oranı ise % 96.26 olarak hesaplanmıştır.

3.2.2.K En Yakın Komşuluk Algoritması:

Çalışmada kullanılan diğer algoritma ise KNN algoritmasıdır ve sözde kodu Şekil 3.4 de gösterilmiştir. [20] KNN algoritmasında amaç birbirine en çok benzeyen veri noktalarını aynı veri kümesi içinde tutmaktır. Benzerlik kavramı çoğu kez uzaklığın tersiyle ifade edilir. Veriler arasındaki benzerlik veriler arasındaki uzunluk hesaplaması (3.6),(3.7) denklemleri ile bulunmuştur. Uzunluk hesaplama için Öklid ve Minkowski uzaklık fonksiyonlarını kullanılmaktadır. Sınıflandırmak istenilen yeni bir bireyin daha önceki bireylerden k tanesine yakınlığına bakılması ile bulunur.

3.2.2.1.Uzaklık Fonksiyonları

X verisi öğrenme, Y verisi test verisi kullanılarak hesaplama yapılır.

$$X=\{x_1,x_2,x_3,\dots\} \quad Y=(y_1,y_2,y_3,\dots) \in \mathbb{R}^n$$

- Öklid Uzaklık Fonksiyonu:

$$\text{Öklid} = \sum_{i=1}^k (X_i - Y_i)^2 \quad (3.6)$$

- Minkowski Uzaklık Fonksiyonu:

$$\text{Minkowski} = \left(\sum_{i=1}^k (|X_i - Y_i|)^p \right)^{1/q} \quad (3.7)$$

Minkowski hesaplamasında her bir özelliğin farkının p'inci kuvvetinin toplamının 1/p kuvveti hesaplanır. P = 1 olduğu durumda işlemin sonucu Manhattan yöntemiyle aynı olur.

P = 2 olduğu durumda işlemin sonucu Öklid yöntemiyle aynı olur

Bir a noktasının sınıfını belirlemek için veride bulunan tüm X kümesinin elemanları kullanılır. K değeri, bilinmeyen noktaya en yakın kaç adet sınıflı noktanın kullanılacağını belirler. Örneğin 1-NN için hedef sınıf, bilinmeyen noktaya en yakın

sınıflı veri noktası tarafından belirlenir, 3-NN için hedef sınıf ise bilinmeyen noktaya en yakın 3 adet sınıflı noktadan çoğunluğu sağlayan sınıf olmalıdır.

Sınıflandırma(X,Y,x)

// X: öğrenme verisi, Y:X sınıf. etiketi, x: bilinmeyen yeni örnek

for i=1 to m:

d(xi, x) mesafesini hesapla

endfor

k en yakın x örneğine I sınıfı denilerek

return x için en sık rastlananları Y kümesine dâhil edilmelidir.

Şekil 3.7 : K NN Sözde Kod.

Tablo 3.7: 3 NN Algoritması Başarı Oranları

		Öngörülen Sınıf	
		Sınıf=1	Sınıf=0
	Sınıf=1	10508	240
	Sınıf=0	497	7064

Burada doğru olması gereken 240 tane küme elemanın yanlış olarak tahmin edildiğini yanlış olması gereken 497 değer ise doğru olarak tahmin edildiğini görülmektedir. Başarı oranı ise %95.97 olarak hesaplanmıştır.

4.DENEYSEL SONUÇLAR VE DEĞERLENDİRME

Tablo 3.3,3.4,3.5 de elde edilen öznitelikler isimlerine göre elde edilen değerlerden oluşan veri kümesi WEKA uygulaması kullanılarak sınıflandırılmıştır.

```
@ATTRIBUTE method REAL
@ATTRIBUTE sqlscount REAL
@ATTRIBUTE jsscount REAL
@ATTRIBUTE pathfilecount REAL
@ATTRIBUTE special_charcount REAL
@ATTRIBUTE urlsqlscount REAL
@ATTRIBUTE urljsscount REAL
@ATTRIBUTE urlpath_filecount REAL
@ATTRIBUTE urlspecial_charcount REAL
@ATTRIBUTE sqlexists REAL
@ATTRIBUTE jssexists REAL
@ATTRIBUTE path_lileexists REAL
@ATTRIBUTE urlsqlexists REAL
@ATTRIBUTE urljssexists REAL
@ATTRIBUTE urlpath_lileexists REAL
@ATTRIBUTE urlspecial_charexists REAL
@ATTRIBUTE check_double_extension REAL
@ATTRIBUTE len REAL
@ATTRIBUTE urlen REAL
@ATTRIBUTE urlunknown_vars REAL
@ATTRIBUTE urlnon_numeric REAL
@ATTRIBUTE urlslicecount REAL
@ATTRIBUTE urlpercent REAL
@ATTRIBUTE urlslash_after_extens REAL
@ATTRIBUTE urlgoodexten REAL
@ATTRIBUTE class {0,1}
@DATA
0,0,6,1,1,0,6,1,1,0,1,1,0,1,1,1,0,902,398,0,1,6,20,0,1,1
2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,818,247,0,0,6,5,0,1,1
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,498,0,0,0,6,0,0,1,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,537,34,1,0,6,0,0,1,1
0,2,0,0,1,2,0,0,1,1,0,0,1,0,0,1,0,559,47,0,1,6,8,0,1,1
1,1,0,0,0,1,0,0,0,1,0,0,1,0,0,0,0,616,46,0,0,6,1,0,1,1
0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,516,0,0,0,7,0,0,0,1
1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,604,33,0,0,6,0,0,1,0
1,2,0,0,0,2,0,0,0,1,0,0,1,0,0,0,0,645,74,0,0,6,7,0,1,1
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,502,0,0,0,6,0,0,1,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,485,0,0,0,5,0,0,0,1
```

Şekil 4.1: Özniteliklerin Weka Uygulamasındaki Görünümü.

4.1. Veri Kümesi

CSIC 2010 HTTP veri seti, sanal bir e-ticaret uygulaması kullanılarak ve kullanıcılara ait özel bilgileri tutarak üretilen binlerce web isteğinden oluşmaktadır ve Web saldırılarından korunma sistemini test etmek için üretilmiştir. [17] DARPA veri seti güncel saldırıları test etmede yersiz kaldığı için uygun değildir. Veri setinde 36000 normal ve 25000 den fazla anormallik istekten oluşmaktadır.

Http istekleri normal ve anormallik olarak etiketlenmiş SQL enjeksiyonu, arabellek taşması, bilgi toplama, dosyaların ifşa edilmesi, CRLF enjeksiyonu, XSS, sunucu tarafı dâhil, parametre müdahalesi gibi saldırıları içerir.

Üç tip normal istekten oluşmuştur.

- Statik saldırılar gizli kaynak talep etmeye çalışmaktadır. Bu istekler olmayan bir dosyayı yapılandırarak ya da varsayılan olarak tekrar yazma şeklinde olan saldırılardır.
- Dinamik saldırılar geçerli bir istek argümanlar değiştirerek şu saldırı yöntemlerini denemektedir.(ör: SQL enjeksiyonu, CRLF enjeksiyonu, siteler arası komut dosyası, önbellek taşmaları, vb.)
- Kasıtsız yasadışı istekleri bu isteklerin kötü niyeti yoktur, ama normal istekte değildirler. (ör: harflerden oluşan bir telefon numarası)

```
Http://localhost:8080/tienda1/publico/anadir.jsp HTTP/1.1
User-Agent: Mozilla/5.0 (compatible; Konqueror/3.5; Linux) KHTML/3.5.8
Pragma: no-cache
Cache-control: no-cache
Accept:
text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q
=0.8,image/png,*/*;q=0.5
Accept-Encoding: x-gzip, x-deflate, gzip, deflate
Accept-Charset: utf-8, utf-8;q=0.5, *;q=0.5
Accept-Language: en
Host: localhost:8080
Cookie: JSESSIONID=933185092E0B668B90676E0A2B0767AF
Content-Type: application/x-www-form-urlencoded
Connection: close
Content-Length: 68
id=3&nombre=Vino+Rioja&precio=100&cantidad=55&B1=A%F1adir+al+
```

Şekil 4.2:POST İsteği Veri Örneği.

```

Http://localhost:8080/tienda1/publico/anadir.jsp?id=3&nombre=Vino+Rioja&precio=100&cantidad=55&B1=A%F1adir+al+carrito HTTP/1.1
User-Agent: Mozilla/5.0 (compatible; Konqueror/3.5; Linux) KHTML/3.5.8
Pragma: no-cache
Cache-control: no-cache
Accept:
text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Encoding: x-gzip, x-deflate, gzip, deflate
Accept-Charset: utf-8, utf-8;q=0.5, /*;q=0.5
Accept-Language: en
Host: localhost:8080
Cookie: JSESSIONID=81761ACA043B0E6014CA42A4BCD06AB5
Connection: close

```

Şekil 4.3: GET İsteği Veri Örneği.

```

id=3&nombre=Vino+Rioja&precio=100&cantidad=55&B1=A%F1adir+al+carrito
HTTP/1.1

```

Şekil 4.4: GET İsteği Parçalaması.

4.2. Performans Hesaplama

Çalışmada performans hesaplaması için WEKA uygulaması kullanılmıştır. Seçilen özniteliklerin karar ağacında kazanç hesaplaması yapılarak sınıflandırmaya katkısı görülmüştür. Karar ağacı algoritmasında kazanç hesaplaması doğruluk değeri ve bilgi kazancı (3.5),(3.6),(3.7) şimdi bunların nasıl hesaplandığına bir bakılmalıdır. Tablo 4.1 de normal anormallik istek sayılarını, Tablo 4.2 de bunların istek tiplerine göre dağılımı görülmektedir.

Tablo 4.1: Normal Ve Anormallik İstek Sayısı

Normal	Anormallik	Toplam
36000	25064	61064

- $Info(D)=I(36000,25064)=-36000/61064*\log_2(36000/61064) - 25064/61064*\log_2(25064/61064)$

Tablo 4.2: İstek Türleri Sayısı

İstek Tipi	Normal	Anormallik	Toplam
GET	28000	15087	43.087
POST	8000	9580	17580
PUT	0	397	397

- $Info_{istek}(D) = 43087/61064 * I(28000,15087) + 17580/61064 * I(8000,9580) + 397/61064 * I(0,397)$
- $Kazanç = Info_{istek}(D) - Info(D) = 0.00387$

Doğruluk değeri kullanarak bilgi kazançları hesaplanmıştır. Doğruluk değerlerine göre kullanılan 8 öznitelikten türetilen 25 yeni alt öznitelik ortaya çıkarılmıştır. Bu öznitelik veri kümesi için bir eşik değeri belirlenirse bu değeri 0.00387 den büyük olduğunu düşünüldüğünde öznitelik sayısı 23'e düşmektedir. Bu ise önerilmemekte çünkü sıfırdan farklı her bir öznitelik değerinin farklı veri kümelerinde da saldırı tespiti için sınıflandırmaya katkısı olacaktır.

4.3.Değerlendirme

Değerlendirme işleminin nasıl yapıldığına bir göz atıldığında sınıflandırma aşamasına getirilmiş işlenmiş veri kümesinin %70 öğrenme verisi %30 unu ise test verisi olarak kullanılmış olduğu görülmektedir.

Tablo 4.3: Karışıklık Matrisi

		Öngörülen Sınıf	
		Sınıf=1	Sınıf=0
Sınıf=1		TP	FN
Sınıf=0		FP	TN

- Doğruluk

Sistem üzerinde doğru olarak yapılan tahminlerin bütün tahminlere oranına denir.

$$\text{Doğruluk} = (TP+TN) / (TP+TN+FN+FP) \quad (4.1)$$

- Anma

Sistem üzerinde pozitif durumların kendi ne kadar başarılı tahmin edildiğini gösterir ve sınıflandırma için önemli bir argümandır.

$$\text{Anma} = TP / (TP + FN) \quad (4.2)$$

- Hassas

Pozitif olarak tahmin edilen bir durumdaki başarı oranını gösteren durumdur.

$$\text{Hassas} = TP / (TP + FP) \quad (4.3)$$

- F-ölçütü

Anma ve Hassasın olarak hesaplanan değerlerin harmonik ortalaması hesaplanması sonucu ortaya çıkan değere *denir*.

$$\text{F-Ölçütü} = 2 * \text{Hassas} * \text{Anma} / (\text{Hassas} + \text{Anma}) \quad (4.4)$$

Sınıflandırma aşamasında kullanılan algoritmaların veri kümesini sınıflandırmada başarısı Doğruluk, Anma, Hassas ve F-ölçütü Tablo 11 da CSIC 2010 veri kümesini kullanarak yapılan çalışmaların başarısı ise Şekil 4.4 de ifade edilmiştir.

Tablo 4.4: Farklı İstek Sayılarına Göre Doğruluk Değeri

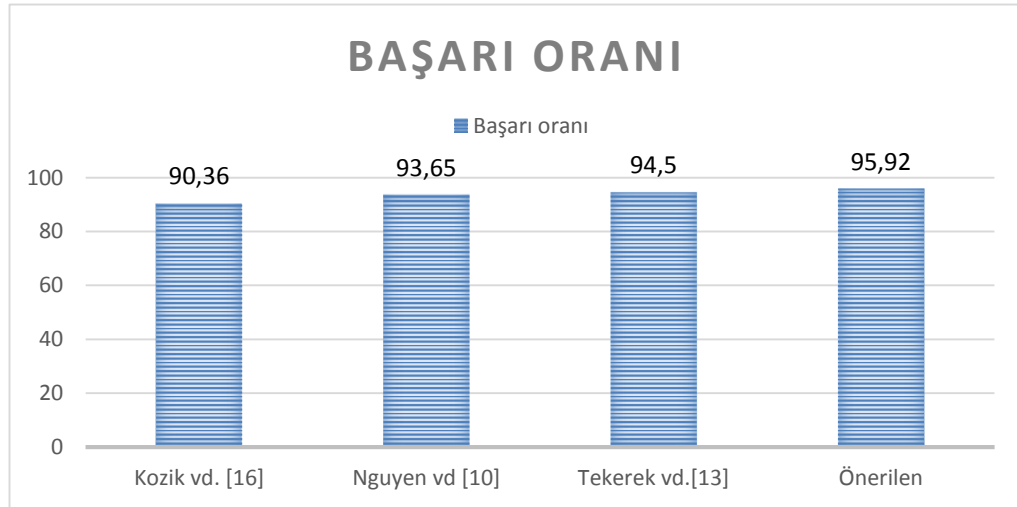
Test No	Doğru sınıflanmış eleman sayısı	Doğruluk değeri (%)
Test1	12028	95.9783
Test2	12018	95.8985

Yukarıdaki tabloda veri kümesindeki istek tipleri üç parçaya bölünmüş ve 30.000 istek eğitim için kullanılmış 15.000 adet istek Test1 de test için kullanılmış kalan 15.000 ise Test2 de doğrulama amacıyla kullanılmıştır. Sonuç olarak

sınıflandırma değerlerinde çok büyük oranda değişiklik göstermemiş ve birbirine yakın sonuçlar elde edilmiştir. Bu da çalışmada seçilen özniteliklerin başarısını ortaya çıkarmaktadır.

Tablo 4.5: Öznitelik Vektörleri için Sınıflandırma Algoritmalarının Doğruluk Değerleri

Algoritma	Öznitelik Sayısı	Doğruluk	Anma	Hassas	F- ölçütü
C4.5	25	0,963	0,963	0,963	0,963
1NN	25	0,957	0,957	0,957	0,957
1NN	25	0,958	0,958	0,958	0,958
3NN	25	0,960	0,960	0,960	0,960
3NN	25	0,960	0,960	0,960	0,960
5NN	25	0,960	0,960	0,960	0,960
5NN	25	0,960	0,960	0,960	0,960



Şekil 4.5: CSIC 2010 veri kümesi kullanan çalışmaların Karşılaştırma.

4.4.Yapılan Ana Katkılar

Bu çalışmada yapılan kullanılan başlıca yöntemler aşağıdaki gibidir.

- i) Http isteği iki parçaya bölünmüştür.
- ii) OWASP’ın belirlediği popüler saldırılar için Tablo 3.1 de belirlenen anahtar kelimeler ortaya çıkarılmıştır.
- iii) Genel olarak yapılan anormal istekler incelenmiş ve bu isteklerden Tablo 3.2 de bulunan anahtar kelimeler ortaya çıkartılmıştır.

Tablo 3.3,3.4,3.5 de bulunan özgün öznitelikler oluşturularak elde edilen veri kümesi sınıflandırma algoritmaları uygulanmıştır.

Aşağıda tabloda ifade edilen öznitelik isimleri, owaspın belirlediği popüler saldırı yöntemleri için belirlenen anahtar kelimelerden oluşmaktadır. Belirtilen öznitelikler tespit edilirken karar ağacı algoritması kullanılmış, her bir özniteliğin bilgi kazanç değerleri (3.4) ve (3.5) olarak ifade edilen matematiksel formüller ile hesaplanmıştır. Bilgi kazancı öznitelik çıkarımı için ise 0.00387 eşik değeri taban alınmış bu doğrultuda öz nitelik çıkarımı yapılmıştır. Öznitelik çıkarımı konusunda anormal isteklerin karakteristik yapıları bu alan ile ilgili yapılan güncel çalışmalardan yararlanarak ortaya çıkartılmıştır. Makiou ve arkadaşları istek içerisinde gereksiz tekrar, yanlış sorgu kontrolü, birleşme sorgu kontrolü ve header seviye sql kontrolü yaparak saldırı tespitinde bulunmuştur. [25] Tekerek ve arkadaşları ise http isteklerini alfa nümerik karakter, harf frekans ile cümle uzunluğu öznitelikler oluşturmuştur. [11] M. Kirchner isimli şahsın çalışmasında ise düzenli ifadeler kullanarak 32 adet öznitelik sınıfı oluşturmuştur. [24] Aşağıda maddeler halinde açıklanan yöntemler ve bu alanda bahsettiğimiz güncel çalışmalar kullanılarak 25 adet öznitelik oluşturulmuştur.

Tablo 4.6:Öznitelik İsimleri

S.N	Öznitelikİsimleri	Açıklama
1	Sqlscount	Şekil4.3 da bulunansqlsayısı
2	Jsscount	Şekil4.3 da bulunanjsssayısı
3	pathfilecount	Şekil4.4 da pathfilenamesayısı
4	special_charcount	Şekil4.3 da bulunanözelcharsayısı
5	urlsqlscount	Şekil4.4 da bulunansqlsayısı
6	urljsscount	Şekil4.4 da bulunanjsssayısı
7	urlpathfilecount	Şekil4.4 da pathfilenamesayısı
8	urlspecial_charcount	Şekil4.4 da bulunanözelcharsayısı
9	url_passive_slash_count	Şekil 4.4 da bulunan slash sayısı
10	file_path_percent_count	Şekil 4.3 da bulunan percent sayısı
11	totallength	Şekil 4.3 un uzunluğu
12	urllength	Şekil 4.4 un uzunluğu
13	sqlsexists	Şekil 4.3da sqlvarmı ?
14	jssexists	Şekil 4.3da jssvarmı ?
15	pathfileexists	Şekil 4.3 da pathfilenamevarmı ?
16	urlsqlsexists	Şekil 4.4da sqlvarmı ?
17	urljssexists	Şekil 4.4da jssvarmı ?
18	urlpathfileexists	Şekil 4.4 da pathfilenamevarmı ?
19	urlspecial_charexists	Şekil 4.4özelcharvarmı?
20	urldoubleextension	Şekil 4.4 da double extvarmı?
21	unknown_vars	Şekil 4.3da unknownvarmı ?
22	non_numeric	Şekil 4.3da non numeric varmı ?
23	urlslash_after_extension	Şekil 4.4 da after extension varmı ?
24	good_extension	Şekil 4.3da goodextensionvarmı ?
25	method_type	Şekil 4.3 da type

5.SONUÇ VE ÖNERİLER

Bu çalışmada; web saldırılarını önlemek için http protokol yapısının GET, POST, PUT kısımları detaylı bir şekilde ele alınmış olup bu kısma ek olarak veri kümesi istek kısımları çeşitlendirilebilir ise bu durum öznetelik çıkarımına daha fazla katkı sağlayacaktır. Ayrıca web uygulamaları güvenliği için kullanılan imza tabanlı, anormallik tabanlı yöntemler ile hibrit yöntemi incelenmiştir. Yeni saldırıları tanımlayacak ve bu saldırılara karşı önlem alabilecek yöntemler konusunda kendini yenileyebilen algoritmalar üzerinde çalışmalar yapılmıştır. Günlük hayatta karşılaşılan problemlerden faydalanılarak atak tespiti için anormallik tabanlı bir yaklaşım ile çalışmalar yapılmıştır. Bilinen saldırı yöntemlerinde kullanılan anahtar ifadeler ve bu ifadelerin saldırılar ile nasıl bir ilişkisinin olduğu yukarıda gösterilen Şekil 4.2 ve 4.3’ de bulunan veri küme elamanları kullanılarak gösterilmiştir. Buna bağlı olarak da seçilen özneteliklerin kazanç değeri hesaplanmıştır. Kazanç değerlerinin büyüklüğü normal ile normal olmayan isteklerin birbirinden ayrılmasına yardımcı olmaktadır. Bu çalışmaya ilave olarak eldeki mevcut veri kümesi ve diğer veri kümeleri için saldırıların karakteristik özelliği çıkarılabilir. Her bir saldırıda saldırgan kişinin çıkarılan karakteristik özelliğine göre anormallik yöntemler için önlemler alınabilir. Veri kümesi daha detaylı incelendiğinde örnek olarak tekrar eden değişkenler gibi saldırılar için daha fazla başarı elde edilebilir. Bu ise saldırı için belirlenen profilden normal davranıştan sapma miktarına göre, yapılan isteğin saldırı olup olmadığı kararının verilmesine yardımcı olacaktır. Bu kararın saldırı tespiti için belirli bir eşik değeri kullanılarak sonuca ulaşılabilecektir. Web uygulamalarının kullandıkları parametreler ağ ve uygulama katmanı arasında kalan kısımlarda değerlendirilebilir. Bu kullanılan ara işlemlerden de faydalanılarak çıkarılacak öznetelikler yine belirli bir eşik değerin üzerinde olma koşulu ile kazanç değeri hesaplanarak sınıflandırma algoritması ile daha iyi sonuçlar elde edilebilir.

KAYNAKLAR

- [1] Namit G., Abakash S., Dheeraj S.,(2008), “Web Application Firewall”, Tech Project Final Report,499-511
- [2] Tajpour A., Massrumand M. , Heydari M.Z.,(2010),”Comparison of SQL Injection Detection and Prevention Techniques”, 2nd International Conference on EducationTechnologyandComputer (ICETC).
- [3] Kruegel C., Vigna. G.,(2003),”Anomalydetection of web-basedattacks”, 10th ACM Conference on ComputerandCommunication Security (CCS 03), 251-261
- [4] Mahmoud M. Bill C. Heather R.L, (2016),“Automatic web security unit testing: XSS vulnerability detection “,Proceedings of the 11th International Workshop on Automation of Software Test
- [5] Halfond W.G., Orso,Nov A., (2005),” AMNESIA: Analysis and Monitoring for Neutralizing SQL-InjectionAttacks”, Proc. 20th IEEE and ACM IntlConf. Automated Software Eng, 65-81.
- [6] Nguyen H.T.,Torrano-Gimenez C., Alvarez C., Franke K., Petrovic S.,(2011), “Enhancing the effectiveness of Web Application Firewalls by generic feature selection, Logic Journal of IGPL, 21 (4), 560-57
- [7] Torrano-Gimenez C.,Perez-Villegas A., Alvarez G.,(2009),” A Self-learning Anomaly-Based Web Application Firewall”, Information Systems. Springer-Verlag Berlin Heidelberg, 85–92
- [8] Tekerek A.,Gemci C., Bay O.,(2014.)” Development of a Hybrid Web Application Firewall to Prevent Web BasedAttacks”, 2014 IEEE 8th International Conference on Application of Information and Communication Technologies IEEE, Astana Kazakhstan, 15-17 Ekim 2014.
- [9] Cho S.,Cha S.,(2004),”SAD: Web Session Anomaly Detection Based on Parameter Estimation”, Computers & Security, 23 (4), 312-31
- [10] Kozik R. Choras M. Renk R. Holubowicz W.,(2014), “A Proposal of Algorithm for Web Applications Cyber Attack Detection” 13th IFIP International Conference on Computer Information Systems and Industrial Management (CISIM).
- [11] Tutkan M.,Ganiz M., Akyokuş S.,(2014),” An Unsupervised Semantic Attribute Selection Method for Text Classification”, Eleco 2014.

- [12] Basile C., Lioy A., (2015), "Analysis of Application-Layer Filtering Policies With Application to http", IEEE/ACM Transactions On Networking, IEEE/ACM Transactions On Networking, 23 (1), 28-41
- [13] Hwang K., Cai M., Chen Y. Qin M., (2007), "Hybrid intrusion detection with weighted signature generation over anomalous Internet episodes", IEEE Transactions On Dependable And Secure Computing. And Secure Computing, 4 (1), 41-55
- [14] Hendry G., Yang S., (2008), "Intrusion signature creation via clustering anomalies, Data Mining, Intrusion Detection, Information Assurance", And Data Networks Security, Proceedings of SPIE.
- [15] Kirchner M. , (2010), "A Framework for Detecting Anomalies in HTTP Traffic Using Instance-Based Learning and K-Nearest Neighbor Classification", 2nd International Workshop on, 1–8
- [16] Makiou A., Begriche Y. Serhrouchni A., (2014), "Hybrid approach to detect SQLi attacks and evasion techniques "
- [17] Roesch M., (1999) Snort-lightweight intrusion detection for networks, 13th Systems Administration Conference LISA'99, Seattle Washington-USA, 229–238.
- [18] Ingham K., Anomaly Detection for HTTP Intrusion Detection: Algorithm Comparisons and the Effect of Generalization on Accuracy, Doktora Tezi, University of New Mexico, Albuquerque-USA
- [19] S. W. Boyd and A. D. Keromytis. (2004) SQLrand: Preventing SQL Injection, Attacks. In Proceedings of the 2nd Applied Cryptography and Network Security (ACNS) Conference, 292-302.
- [20] Web 1, (2017), "<http://www.isi.csic.es/dataset/>", (Eriřim tarihi: 10.03.2017)
- [21] Web 2, (2017), "<https://www.cs.swarthmore.edu/~meeden/cs63/f05/id3.html>", (Eriřim tarihi: 18.02.2017)
- [22] Web 3, (2017), "<https://www.researchgate.net/figure/260397165>" (Eriřim tarihi: 25.01.2017)
- [23] Web 4, (2017), "<https://www.owasp.org/index.php>", (Eriřim tarihi: 25.05.2017)

ÖZGEÇMİŞ

Metin ŞAHİN 1985 yılında Kozan’da doğdu. 2004 yılında başladığı Gebze Yüksek Teknoloji Enstitüsü Fen Fakültesi Matematik Bölümünü 2010 yılında başarıyla tamamladı. 2012 yılında Emniyet Genel Müdürlüğünde Polis Memuru olarak çalışmaya başladı. Yine aynı yıl Gebze Teknik Üniversitesinde Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans eğitime başladı. Halen Kocaeli Emniyet Müdürlüğü bünyesinde çalışmaya devam etmektedir. İşten fırsat bulduğu aralıklarda kendini geliştirmeye ve yetiştirmeye çalışmaktadır.

EKLER

Ek A: Tez Çalışması Kapsamında Yapılan Yayınlar

Sahin M.,Sogukpınar I., (2017), “An Efficient Firewall for Web Applications (EFWA)”,Computer Science and Engineering (UBMK), 2017 International Conference on, 707-712.