Derleme Makalesi

E-TİCARET SİTELERİ İÇİN KATMANLI AÇIK TESPİT MODELİ: TASARIM VE DEĞERLENDİRME

Kenan İslamoğlu[†], Fatma Nur Akı[††]
[†] Yüksek Lisans Öğrencisi, İstanbul Ticaret Üniversitesi, Lisansüstü Eğitim Enstitüsü, Siber Güvenlik Tezli Yüksek Lisans Programı
[††] Dr. Öğr. Üyesi, İstanbul Ticaret Üniversitesi, Mühendislik Fakültesi, Yazılım Mühendisliği (İngilizce) Bölümü
[†] islamoglukenan@gmail.com, [††] fnaki@ticaret.edu.tr
0009-0007-1381-9364, 0000-0003-4103-8876

Atıf/Citation: İslamoğlu, K., Akı, F.N., E-Ticaret Siteleri İçin Katmanlı Açık Tespit Modeli: Tasarım ve Değerlendirme, İstanbul Ticaret Üniversitesi Teknoloji ve Uygulamalı Bilimler Dergisi, 8(2), s.105-118, doi: 10.56809/icujtas.1755050

ÖZET

E-ticaret platformları, kullanıcı verileri ve finansal işlemleri bünyesinde barındırmaları nedeniyle siber saldırıların öncelikli hedefleri arasında yer almaktadır. Literatürdeki mevcut çalışmaların önemli bir kısmı, yalnızca otomatik araçlarla yapılan taramalara veya sınırlı senaryolarla yürütülen manuel testlere odaklanmakta; risk önceliklendirme ve yönetici seviyesine uygun raporlama boyutunu çoğunlukla göz ardı etmektedir. Bu çalışma, literatürde manuel ve otomatik analizleri entegre eden, senaryoya dayalı ve katmanlı bir tespit-raporlama modeli önermektedir. Model; hedef belirleme, bilgi toplama, zafiyet tarama, açık doğrulama ve raporlama olmak üzere beş aşamalı bir yapıya sahiptir. OWASP ZAP(Open Web Application Security Project Zed Attack Proxy), Burp Suite(Burp Suite Web Vulnerability Scanner and Testing Platform) gibi otomatik araçların yanı sıra, kullanıcı davranışlarına dayalı manuel testlerle desteklenen yapı sayesinde tespitlerin doğruluk ve güvenilirlik seviyesi artırılmıştır. Farklılaştırıcı olarak, model yalnızca teknik analizle sınırlı kalmayıp CVSS(Common Vulnerability Scoring System) tabanlı risk önceliklendirme ve yönetici seviyesine hitap eden, senaryo temelli karar destek raporlaması sunmaktadır. Değerlendirme sonuçları, SQL Injection(Structured Query Language Injection) ve XSS(Cross-Site Scripting) gibi kritik açıkların yüksek başarı oranıyla tespit edildiğini göstermektedir. Bu yönüyle çalışma, e-ticaret sistemlerinde sürdürülebilir güvenlik için literatüre katkı sunan özgün ve genişletilebilir bir yaklaşım ortaya koymaktadır.

Anahtar Kelimeler: E-ticaret Güvenliği, Güvenlik Açığı Tespiti, Zafiyet Tarama Araçları, CVSS, OWASP, Siber Güvenlik Modeli.

A LAYERED VULNERABILITY DETECTION MODEL FOR E-COMMERCE PLATFORMS: DESIGN AND EVALUATION

ABSTRACT

E-commerce platforms, which host user data and financial transactions, are among the primary targets of cyberattacks. Most existing studies in the literature focus solely on automated scanning tools or limited-scope manual testing scenarios, often overlooking risk prioritization and management-level reporting. This study proposes, for the literature, a scenario-based and layered detection-reporting model that integrates both manual and automated analyses. The model consists of five stages: target identification, information gathering, vulnerability scanning, vulnerability verification, and reporting. In addition to automated tools such as OWASP ZAP and Burp Suite, the structure is supported by manual tests based on user behaviors, which enhances the accuracy and reliability of detections. As a distinguishing feature, the model not only focuses on technical analysis but also offers CVSS-based risk prioritization and scenario-oriented decision-

Geliş/Received : 31.07.2025
Gözden Geçirme/Revised : 10.08.2025
Kabul/Accepted : 14.08.2025

support reporting tailored for management-level stakeholders. Evaluation results demonstrate that the model successfully detects critical vulnerabilities such as SQL Injection and XSS with a high success rate. In this respect, the study presents an original and expandable approach that contributes to sustainable security in e-commerce systems.

Keywords: E-commerce Security, Vulnerability Assessment, Vulnerability Scanning Tools, CVSS, OWASP, Cybersecurity Architecture.

1. GİRİŞ

Dijital dönüşümün ivme kazandığı günümüzde, e-ticaret platformları hem bireysel kullanıcılar hem de kurumsal aktörler için temel ticaret altyapılarından biri hâline gelmiştir. Ancak bu hızlı büyüme, beraberinde ciddi siber güvenlik tehditlerini de getirmektedir. Özellikle kullanıcı bilgileri, ödeme sistemleri ve kişisel veriler gibi hassas içeriklerin barındırılması; e-ticaret altyapılarını saldırganların öncelikli hedefi hâline getirmiştir (Anderson, 2020).

Bu sistemler, barındırdıkları açık yüzeylerin çokluğu nedeniyle yalnızca geleneksel güvenlik önlemleriyle korunamayacak kadar karmaşık hâle gelmiştir. Güvenlik açıklarının zamanında ve doğru biçimde tespit edilememesi, veri ihlalleri, kimlik hırsızlığı ve doğrudan finansal kayıplar gibi çok boyutlu sonuçlar doğurmakta; ayrıca kurumların itibarı üzerinde uzun vadeli ve telafisi zor etkiler bırakmaktadır (ENISA, 2023). Bu bağlamda, e-ticaret sistemlerinde güvenliğin reaktif değil, proaktif şekilde yönetilmesi; açıklara karşı önleyici ve bütüncül mekanizmaların entegre edilmesi bir zorunluluk hâlini almıştır.

Mevcut güvenlik açığı tespit uygulamaları genellikle ya tamamen manuel sızma testlerine ya da otomatik zafiyet tarama araçlarına dayanmaktadır. Ancak bu yöntemlerin her biri önemli sınırlılıklar içermektedir. Manuel testler, uzmanlık gerektirmesi nedeniyle yüksek zaman ve maliyet unsurları barındırırken; otomatik araçlar, dinamik ve kullanıcı etkileşimli yapıları yeterince yorumlayamadığından kritik bazı açıkları gözden kaçırabilmektedir (Chaffey & Ellis-Chadwick, 2019). Bu durum Tablo 1'de karşılaştırmalı olarak özetlenmiştir:

Tablo 1. Manuel test ve otomatik tarama avantajları ve sınırlılıkları

Yöntem	**Avantajlar**	**Sınırlılıklar**
Manuel Test	Derinlemesine analiz, bağlama duyarlı yorumlama	Zaman alıcı, yüksek uzmanlık ve maliyet gereksinimi
Otomatik Tarama	Hızlı ve geniş kapsamlı tarama	Statik yapılarla sınırlı analiz, yüksek false positive oranı

Bu analiz çerçevesinde, her iki yöntemin güçlü yönlerini bir araya getiren, bağlam farkındalığına sahip bütüncül ve katmanlı bir güvenlik açığı tespit modeline ihtiyaç duyulduğu açıktır.

Bu çalışma kapsamında önerilen model, manuel ve otomatik analiz tekniklerini entegre ederek güvenlik açığı tespit sürecini yapılandırılmış ve çok katmanlı bir mimari çerçevede ele almaktadır. Modelin bileşenleri — hedef tanımlama, bilgi toplama, zafiyet tarama, açık doğrulama ve raporlama — birbiriyle bütünleşik biçimde kurgulanmış; yalnızca teknik analiz değil, aynı zamanda CVSS(*Common Vulnerability Scoring System*)tabanlı risk skorlaması, yönetici düzeyine uygun raporlama çıktıları ve operasyonel müdahaleye olanak tanıyan senaryo temelli analiz ile desteklenmiştir.

Literatürde OWASP(Open Web Application Security Project) tarafından yayımlanan Top 10 listesi (OWASP, 2023), bu alandaki en kritik güvenlik açıklarını tanımlamak açısından temel bir başvuru kaynağıdır. Geliştirilen model, bu listeye dayalı olarak SQL Injection(Structured Query Language Injection), Cross-Site Scripting (XSS) ve kimlik doğrulama zaafiyetleri gibi yaygın tehdit türleri üzerinde uygulanmış ve senaryoya dayalı test ortamında geçerliliği ölçülmüştür.

Bu makale, söz konusu güvenlik açığı tespit modelinin kuramsal temelini, uygulama sürecini ve değerlendirme çıktılarının bütününü sunmayı hedeflemektedir. Aynı zamanda küçük ve orta ölçekli e-ticaret işletmeleri açısından düşük maliyetli, uygulanabilir ve sürdürülebilir bir siber güvenlik yaklaşımı ortaya koyarak literatüre hem akademik hem de pratik katkı sağlamaktadır.

Bu bağlamda çalışma, öncelikle e-ticaret sistemlerinde yaygın olarak karşılaşılan güvenlik açıklarını ve bunların işletmeler üzerindeki etkilerini literatür ışığında ortaya koymakta; ardından geliştirilen katmanlı modelin tasarım ilkeleri, mimari yapısı ve uygulama sürecini ayrıntılı biçimde sunmaktadır. Modelin geçerliliği, gerçekçi senaryolar üzerinden gerçekleştirilen testlerle değerlendirilmiş; elde edilen bulgular teknik analiz çıktıları ve risk önceliklendirme perspektifinden yorumlanmıştır. Son olarak, çalışma kapsamında ulaşılan sonuçlar ile hem akademik hem de uygulamaya dönük öneriler paylaşılmıştır.

2. LİTERATÜR TARAMASI

E-ticaret sistemleri, sundukları çevrim içi alışveriş imkânı, yüksek işlem hacmi ve geniş kullanıcı tabanları nedeniyle siber saldırganların öncelikli hedefleri arasında yer almaktadır (Anderson, 2020). Bu platformlarda meydana gelen güvenlik ihlalleri; finansal kayıplar, kullanıcı verilerinin sızdırılması ve kurumsal itibarın zedelenmesi gibi çok boyutlu sonuçlar doğurmaktadır (ENISA, 2023). Literatürde e-ticaret

güvenliği, çoğunlukla yaygın güvenlik açıklarının analizi ve bu açıkların tespit yöntemleri üzerinden ele alınmaktadır. En güncel referanslardan biri olan OWASP Top 10 listesi (OWASP, 2023), web tabanlı sistemlerde en kritik güvenlik açıklarını tanımlamakta ve SQL Injection (SQLi), Cross-Site Scripting (XSS), Broken Authentication, Security Misconfiguration ve Sensitive Data Exposure gibi zafiyetlerin e-ticaret platformlarında en sık rastlanan tehditler arasında olduğunu vurgulamaktadır (Choudhary & Sharma, 2022). Çeşitli çalışmalar, SQL Injection'ın veri tabanı manipülasyonu yoluyla yetkisiz erişim sağladığını, XSS'in ise kullanıcı oturumlarının ele geçirilmesine neden olabileceğini ortaya koymuştur (Halfond et al., 2006; Gupta & Gupta, 2015). Yaygın olarak kullanılan otomatik güvenlik tarama araçları arasında OWASP ZAP, Burp Suite, Nikto ve Nmap öne çıkmakta olup, bu araçlar önceden tanımlanmış imza veritabanları ve davranışsal analiz yöntemleri aracılığıyla bilinen zafiyetleri hızlı bir şekilde tespit edebilmektedir (Suto, 2010). Ancak yapılan çalışmalar, otomatik araçların yanlış pozitif üretme olasılığının yüksek olduğunu ve karmaşık kullanıcı etkileşimlerini simüle etmede yetersiz kaldığını ortaya koymuştur (Doupe et al., 2010; Bau et al., 2010). Manuel sızma testleri ise güvenlik uzmanlarının deneyim ve sezgilerini kullanarak sistemdeki açıkları hedefe yönelik test senaryolarıyla keşfetmesini sağlamakta ve özellikle kimlik doğrulama zafiyetleri, yetkilendirme hataları ve iş mantığı açıkları gibi otomatik araçların tespit edemediği durumlarda etkili olmaktadır (Jamil & Zaki, 2011). Bununla birlikte, manuel testlerin yüksek maliyetli, zaman alıcı ve tekrarlanabilirlik açısından sınırlı olduğu da vurgulanmaktadır (Geer, 2015). Son yıllarda yapılan bazı çalışmalar, manuel ve otomatik test yöntemlerinin bir arada kullanılmasının tespit doğruluğunu artırabileceğini göstermektedir (Alshamrani et al., 2019). Ancak mevcut araştırmaların çoğu, bu iki yöntemin entegrasyonunu sistematik ve katmanlı bir mimari çerçevesinde ele almamış, ayrıca risk önceliklendirme ve yönetici odaklı raporlama gibi çıktılara yeterince yer vermemiştir. Dolayısıyla, hem teknik doğruluk hem de karar destek süreçleri açısından bütüncül bir güvenlik açığı tespit modeli ihtiyacı devam etmektedir. Bu çalışmada geliştirilen model, literatürde tespit edilen bu boşluğu doldurmayı hedeflemekte; manuel ve otomatik analiz tekniklerini entegre eden, senaryoya dayalı ve katmanlı bir yaklaşım sunarak hem teknik ekipler hem de karar vericiler için uygulanabilir bir çözüm önermektedir.

3. YÖNTEM

Bu çalışmada geliştirilen katmanlı güvenlik açığı tespit modeli, e-ticaret sistemlerinde sık karşılaşılan zafiyetlerin sistematik ve tekrarlanabilir bir yaklaşımla belirlenmesini hedeflemektedir. Model, hedef belirleme, bilgi toplama, zafiyet tarama, açık doğrulama ve raporlama olmak üzere beş temel bileşenden oluşmaktadır. Bu yapının temel dayanağı, hem manuel hem de otomatik analiz tekniklerinin bir arada kullanılarak açıkların daha yüksek doğrulukla ve düşük yanlış pozitif oranıyla tespit edilmesidir.

Hedef Belirleme: İlk aşamada testin kapsamı net olarak tanımlanır. Bu kapsam; alan adı, IP(Internet Protocol) aralıkları, port bilgileri, test edilecek modüller, API(Application Programming Interface) uç noktaları ve sistem altyapısına dair diğer parametreleri içerir. Bu aşamada yasal ve etik çerçeve de belirlenerek, yalnızca yetkilendirilmiş alanlarda test yapılması garanti altına alınır.

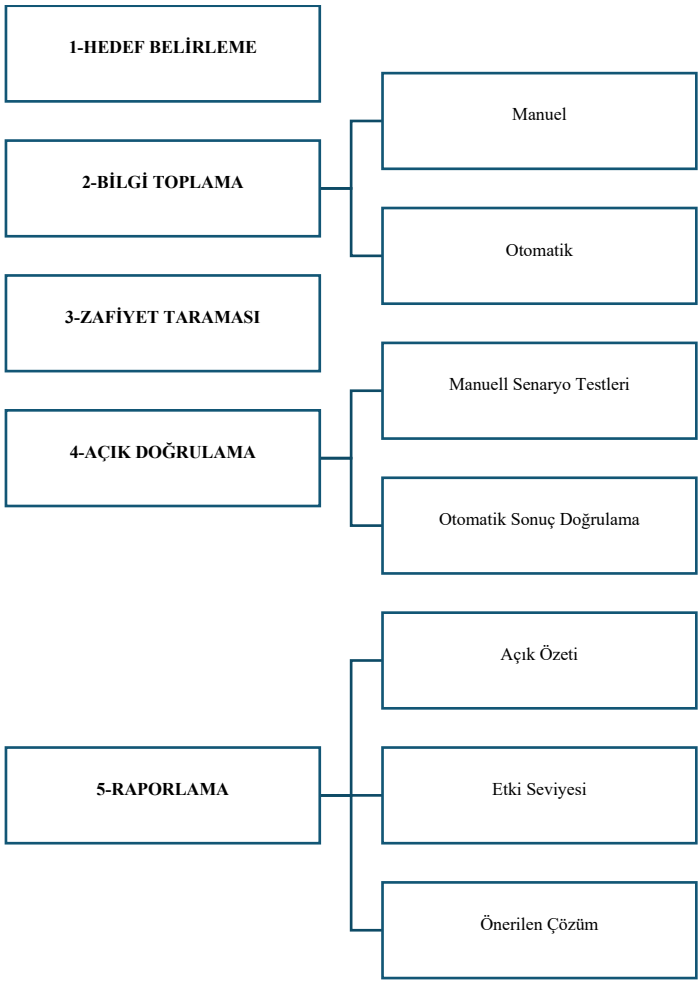
Bilgi Toplama: Sistem hakkında teknik ve yapısal veriler toplanır. Pasif bilgi toplama yöntemleri (Whois sorguları, BuiltWith teknolojisi analizi, Shodan veritabanı taramaları) ile sistemin yazılım yığını, kullanılan framework'ler, web sunucusu tipi, açık portlar ve çalışan servisler belirlenir. Ardından aktif tarama yöntemleri (Nmap servis ve versiyon tespiti, port durumu analizi) ile ağ topolojisi ve servis haritası çıkarılır. Bu veriler, sonraki aşamada hedefe yönelik zafiyet taramalarının temelini oluşturur.

Zafiyet Taraması: Otomatik araçlar OWASP ZAP(Open Web Application Security Project Zed Attack Proxy), Nikto(Nikto Web Server Scanner), Burp Suite(Burp Suite Web Vulnerability Scanner and Testing Platform) kullanılarak geniş kapsamlı taramalar yapılır. Araç seçiminde, OWASP Top 10 kapsamındaki en kritik açıkları tespit etme yeteneği, güncel imza veritabanı ve aktif topluluk desteği kriterleri dikkate alınmıştır. Taramalar sonucunda; SQL Injection(Structured Query Language Injection), XSS (Cross-Site Scripting), kimlik doğrulama zafiyetleri, güvenlik yapılandırma hataları ve potansiyel bilgi sızıntıları gibi riskler öncelikli olarak incelenir.

Açık Doğrulama (Manuel Testler): Otomatik araçların ürettiği çıktılar doğrudan raporlanmaz. Yanlış pozitifleri elemek ve yalnızca gerçekten istismar edilebilir açıkları belirlemek amacıyla manuel doğrulama yapılır. Bu süreçte, kullanıcı davranışlarını taklit eden senaryolar, oturum yönetimi testleri, giriş kontrol bypass denemeleri ve özel hazırlanmış payload'lar kullanılır. Ayrıca, istismar edilebilirliğin kanıtlanması için kontrollü ortamda proof-of-concept (PoC) kodları geliştirilmiş ve çalıştırılmıştır. PoC geliştirmede Python betikleri, Burp Suite Repeater/Intruder modülleri ve tarayıcı geliştirici araçları gibi teknikler kullanılmıştır.Raporlama ve Risk Önceliklendirme: Son aşamada, doğrulanmış açıklar ayrıntılı teknik bilgiler ve istismar senaryoları ile birlikte raporlanır. Rapor, iki ayrı formatta sunulur:

- Teknik rapor: Geliştiriciler ve güvenlik ekipleri için, açıklığın konumu, istismar yöntemi, PoC kodları ve giderme adımlarını içerir.
- Yönetici raporu: Karar vericiler için, CVSS tabanlı risk skorlaması, açıklığın potansiyel etkisi, iş süreçlerine yansıması ve uygulanabilir çözüm önerileri özetlenir.

Bu şekilde geliştirilen modelin tüm bileşenleri, Şekil 1'de gösterilen mimari yapı içerisinde bütünleşik biçimde çalışmakta, her katman bir sonrakini besleyerek bütüncül bir güvenlik değerlendirme süreci oluşturmaktadır.



Şekil 1. Güvenlik açığı tespit modeli

Literatürdeki mevcut güvenlik açığı tespit uygulamaları genellikle yalnızca otomatik tarama araçlarına ya da manuel sızma testlerine dayalı olarak yürütülmektedir. Otomatik araçlar hızlı tarama ve geniş kapsam sağlasa da yanlış pozitif oranı yüksek olup karmaşık kullanıcı etkileşimlerini simüle etmede yetersiz kalmaktadır. Manuel testler ise dinamik senaryoları ve iş mantığı açıklarını tespit etmede etkili olsa da yüksek zaman ve maliyet gerektirir. Önerilen model, bu iki yaklaşımın güçlü yönlerini bir araya getirerek hem kapsam hem doğruluk açısından üstünlük sağlamaktadır.

Katmanlı yapı sayesinde, otomatik tarama ile geniş ölçekli ön değerlendirme yapılmakta; ardından manuel doğrulama ile yalnızca istismar edilebilir açıklar raporlanarak yanlış pozitifler elenmektedir. Ayrıca model, literatürde çoğunlukla göz ardı edilen CVSS tabanlı risk önceliklendirme ve yönetici odaklı senaryo raporlama bileşenlerini entegre ederek, tespit sürecini teknik analizden karar destek düzeyine taşımaktadır. Bu yönüyle önerilen yaklaşım, hem teknik ekipler hem de yönetim için uygulanabilir, tekrarlanabilir ve genişletilebilir bir güvenlik değerlendirme çerçevesi sunmaktadır.

4. BULGULAR VE DEĞERLENDİRME

4.1. Test Ortamı ve Uygulama Süreci

Modelin geçerliliğini değerlendirmek amacıyla oluşturulan test ortamı, orta ölçekli bir e-ticaret sitesini temsil edecek şekilde yapılandırılmıştır. Test ortamı, ASP.NET MVC kullanılarak geliştirilen basit bir e-ticaret simülasyonu üzerine kurulmuştur. Uygulamada temel olarak kullanıcı kaydı/girişi, ürün listeleme ve basit arama işlevleri yer almakta; yönetici paneli ile kontrol mekanizmaları da test kapsamına dahil edilmiştir. Bu yapılandırma sayesinde hem istemci hem de sunucu taraflı güvenlik açıkları test edilebilecek gerçekçi bir ortam sağlanmıştır.

Test süreci, modelin sunduğu beş temel katmana uygun olarak adım adım yürütülmüştür. İlk olarak, "hedef belirleme" adımında test kapsamına alınacak URL(Uniform Resource Locator)'ler, IP(Internet Protocol) blokları ve alt alan adları belirlenmiş; sistemin erişim noktaları, teknolojik altyapısı ve dışa açık servisleri saptanmıştır. Bu aşamada hedef tanımı yapılırken etik kurallara dikkat edilmiş ve sistem içindeki yalnızca simülasyon amaçlı kullanıcı verileri kullanılmıştır. Bu yaklaşım, Solove'un (2007) belirttiği "veri mahremiyetine saygılı test ortamı oluşturma" ilkesine paralel biçimde yürütülmüştür.

"Bilgi toplama" aşamasında sistemin mimari detayları analiz edilmiş; Nmap(Network Mapper) ile port taraması yapılmış, manuel analizle birlikte kullanılan teknolojiler örneğin .NET sürümü, veri tabanı tipi, kullanılan JS(*JavaScript*) kütüphaneleri belirlenmiştir. Bu sayede saldırı yüzeyi detaylı şekilde haritalanmış ve sonraki aşamalarda uygulanacak testler için hedefleme isabeti artırılmıştır. Ayrıca sistemin HTTPS(Hypertext Transfer Protocol Secure)protokolü, TLS(*Transport Layer Security*) sürümü ve SSL sertifika geçerliliği gibi güvenlik protokolleri de incelenmiş ve yapılandırma açıkları ilk etapta tespit edilmiştir.

"Zafiyet taraması" adımında hem OWASP ZAP hem de Burp Suite araçları birlikte kullanılarak çift yönlü analiz gerçekleştirilmiştir. OWASP ZAP daha çok pasif taramalarda ve otomatik saldırı kalıplarında kullanılırken, Burp Suite'in proxy,

intruder ve repeater modülleri ile özellikle girdi alanları üzerindeki açıklar hedeflenmiştir. Böylelikle hem davranışsal hem de imza tabanlı açıklar taranmıştır. Sunucu yapılandırması manuel olarak kontrol edilerek dizin listelemeleri ve varsayılan dosyalar gibi zayıf noktalar incelenmiştir.

Tarama süreci sonunda elde edilen bulgular doğrudan raporlanmamış; "açık doğrulama" aşamasına geçilmiştir. Bu aşamada, örnek saldırı senaryoları (proof-of-concept) yazılarak, açıkların kontrollü saldırı senaryoları altında sömürülebilir olup olmadıkları test edilmiştir. Örneğin, ürün arama kutusunda SQL Injection testi için ' OR '1'='1 ifadesi kullanılmış ve yönetici paneli altında kullanıcı verilerinin sızdırılabilir olduğu doğrulanmıştır. XSS açıkları ise kullanıcı yorumlarına yerleştirilen <script>alert('xss')</script> gibi zararlı kodlarla test edilmiş; çıktılar doğrudan tarayıcı etkileşiminde analiz edilmiştir.

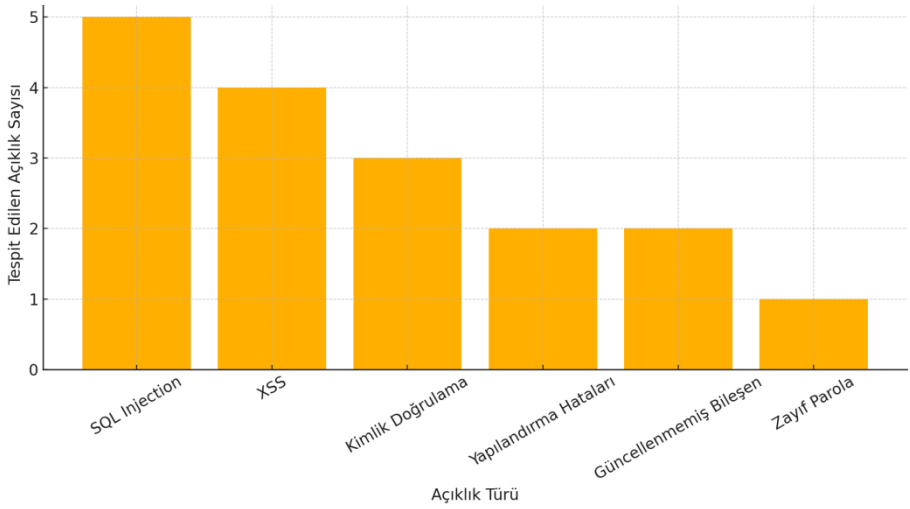
Tüm bu adımlar sonucunda geliştirilen modelin farklı sistem bileşenlerinde başarılı şekilde uygulanabilir olduğu görülmüştür. Özellikle hem istemci taraflı (UI) hem de sunucu taraflı açıkların birlikte analiz edilmesi sayesinde, sadece yüzeysel taramalarla yetinilmeyen derinlemesine bir güvenlik denetimi elde edilmiştir. Bu da modelin hem akademik düzeyde teorik bir katkı sunduğunu hem de sektörel olarak uygulanabilirlik taşıdığını göstermektedir.

4.2. Tespit Edilen Açıkların Türleri ve Dağılımı

Yapılan testler sonucunda hedef sistemde birçok farklı güvenlik zafiyeti tespit edilmiştir. Bu açıkların büyük çoğunluğu, OWASP (2023) tarafından yayımlanan "Top 10 Web Application Security Risks" listesinde doğrudan karşılık bulmaktadır. Bu zafiyetlerin dağılımı; 5 SQL Injection, 4 Cross-Site Scripting (XSS), 3 kimlik doğrulama zafiyeti, 2 güvenlik yapılandırma hatası, 2 güncellenmemiş bileşen ve 1 zayıf parola politikası şeklindedir. Elde edilen veriler, OWASP'in en yaygın ve tehlikeli açık türlerinin e-ticaret sistemlerinde de yoğun biçimde karşılaşıldığını açıkça ortaya koymaktadır.

Tespit edilen SQL Injection açıklarının tamamı, veri girişi doğrulamasının yapılmadığı ürün arama ve kullanıcı girişi modüllerinde görülmüştür. Bu durum, OWASP (2023) tarafından "en tehlikeli 3 açıktan biri" olarak tanımlanan bu risk türünün, hâlâ temel form bileşenlerinde sıklıkla atlandığını göstermektedir. Kullanıcıdan alınan girdilerin doğrudan SQL sorgularına entegre edilmesi, veri sızıntısından tüm veri tabanının silinmesine kadar ciddi etkiler yaratabilecek bir zemin oluşturmaktadır. Bu çalışmada kullanılan testlerde ürün arama kutusu gibi kullanıcı girdisi alınan alanlarda klasik SQL Injection payload'ları (örn. ' OR '1'='1) kullanılmış, yönetici paneline yetkisiz erişim başarıyla gerçekleştirilmiş, sistemin sorgu güvenliğinde ciddi açıklar bulunduğu gözlemlenmiştir.

XSS (Cross-Site Scripting) açıklarının ise ağırlıklı olarak yorum panellerinde ve kullanıcı profil sayfalarında ortaya çıktığı görülmüştür. Bu tür açıklar, zararlı JavaScript kodlarının kullanıcı tarayıcılarında çalıştırılmasına ve oturum bilgisi gibi hassas verilerin dışarıya aktarılmasına neden olmaktadır. Bu çalışmada uygulanan senaryolarda <script>alert('xss')</script> gibi payload'lar, hem stored hem de reflected XSS açıklarının mevcudiyetini ortaya koymuştur. Bu durum, kullanıcı arayüzü bileşenlerinde istemci taraflı doğrulamanın tek başına yeterli olmadığını, sunucu tarafında da mutlaka filtreleme yapılması gerektiğini vurgulamaktadır. Elde edilen bulgular bu uyarının pratik karşılığını gözler önüne sermiştir.



Şekil 2. Tespit edilen güvenlik açıkları

Kimlik doğrulama zafiyetleri, özellikle parola sıfırlama ekranlarında ve oturum yönetiminde karşımıza çıkmıştır. Testlerde kullanılan analiz senaryolarında, parola sıfırlama bağlantılarının süresiz geçerli olması, oturum çerezlerinin http Only bayrağı olmadan iletilmesi ve çok faktörlü kimlik doğrulamanın (2FA) uygulanmaması gibi ciddi yapı eksiklikleri gözlenmiştir. Bu tür açıkların, ENISA (2023) tarafından yayınlanan "Threat Landscape Report"ta da vurgulandığı üzere, kimlik hırsızlığına ve hesap ele geçirme saldırılarına doğrudan zemin hazırladığı belirtilmektedir. Test sonuçları, bu tür zafiyetlerin hâlâ sistematik güvenlik mimarisi içerisinde yeterince önceliklendirilmediğini göstermektedir.

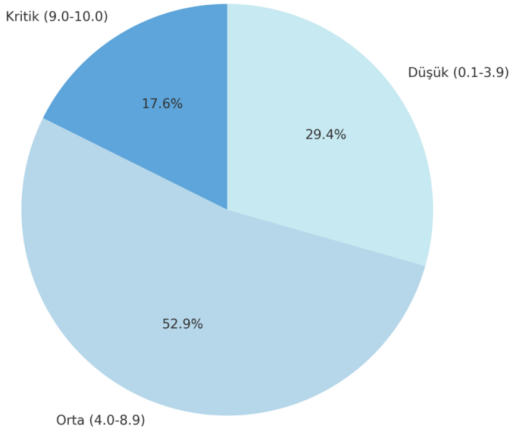
Güncellenmemiş bileşenler ve yapılandırma hataları, genellikle sistemin üçüncü taraf eklentilere bağımlı olduğu bölümlerde gözlemlenmiştir. Örneğin, kullanılan bir kargo takip modülünün iki yıldır güncellenmemiş olduğu ve bu bileşenin içerdiği bilinen bir

XSS açığının hâlâ sistemde aktif olduğu tespit edilmiştir. OWASP (2023) bu tür açıkları "Vulnerable and Outdated Components" başlığı altında değerlendirmekte ve özellikle açık kaynak platformlarda sık rastlandığını belirtmektedir. Bu da, sistem yöneticilerinin yalnızca ana yazılım değil, tüm modül ve eklenti altyapılarını düzenli olarak denetlemeleri gerektiğini ortaya koymaktadır.

4.3. Modelin Etkinliği ve Doğruluk Değerlendirmesi

Modelin uygulandığı test süreci sonucunda elde edilen en kritik kazanım, açıkların sadece bulunması değil; aynı zamanda geçerli ve sömürülebilir olup olmadıklarının yüksek doğrulukla doğrulanması olmuştur. Bu yönüyle model, sadece teknik çıktılar sunan klasik tarama sistemlerinden ayrılmakta ve tespit sonrası müdahale sürecine doğrudan katkı sağlayan bir yapı sunmaktadır. ENISA (2023), otomatik araçların her zaman doğru sonuçlar üretmediğini ve bu nedenle "manuel doğrulama ile desteklenmiş yarı otomatik yapıların" daha etkili olduğunu belirtmektedir. Bu çalışma da bu görüşü sahada test ederek desteklemiştir.

Özellikle OWASP ZAP ve Burp Suite gibi araçlarla yapılan ilk taramalarda, bazı zafiyetlerin "kritik risk" olarak işaretlendiği, ancak sistemdeki yapı nedeniyle gerçekte istismar edilemeyecek durumda olduğu gözlemlenmiştir. Örneğin, Burp Suite tarafından "yüksek riskli XSS" olarak işaretlenen bir açıklığın, uygulamanın Java Script filter mekanizması nedeniyle gerçekte kullanıcı tarayıcısında çalışmadığı; ancak tarayıcının farklı bir sürümünde çalışabileceği varsayımıyla işaretlendiği tespit edilmiştir. Bu gibi durumlarda modelin manuel açık doğrulama aşaması sayesinde bu açık, sistem yöneticilerine "izlenmesi gereken ama öncelikli olmayan" bir risk olarak sunulmuştur.



Şekil 3. CVSS skorlarına göre güvenlik açığı dağılımı

CVSS (Common Vulnerability Scoring System) v3.1 kullanılarak yapılan risk skorlaması, açıkların etkisinin sistematik biçimde sınıflandırılmasına olanak sağlamıştır. Tespit edilen açıklardan üçü 9.0 ve üzeri skorlama ile "kritik" düzeyde sınıflandırılmış ve 24 saat içinde müdahale gerekliliği oluşturulmuştur. Bunlar arasında özellikle veri tabanına tam erişim sağlayan SQL Injection açıkları ve oturum çalma riski içeren XSS açıkları yer almıştır. Orta ve düşük düzeydeki açıklar için ise 7 gün ve üzeri düzeltme periyotları önerilmiştir.

Modelin yönetsel düzeye uyarlanabilirliği de test sürecinde değerlendirilmiştir. Geliştirilen raporlama şablonları sayesinde teknik ekipler detaylı hata kodlarına, açık açıklamalarına ve çözüm önerilerine erişebilirken; yönetici özetleri, açıkların işletme üzerindeki potansiyel etkisini sade bir formatta sunmuştur. Bu sayede yalnızca teknik değil; karar verici düzeyde de güvenlik farkındalığı oluşturulmuştur. Bu yapı, Chaffey ve Ellis-Chadwick (2019)'in dijital platformlarda "güvenliğin sadece teknik bir konu değil, aynı zamanda stratejik yönetim unsuru olduğu" görüşüyle doğrudan örtüşmektedir.

Ancak modelin bazı sınırlılıkları da değerlendirilmiştir. Öncelikle testler yalnızca web tabanlı e-ticaret sistemleri üzerinde yapılmış; mobil uygulamalar ve REST API'ler gibi modern bileşenler kapsam dışında bırakılmıştır. Ayrıca testlerin canlı müşteri trafiği üzerinde değil, izole bir simülasyon ortamında gerçekleştirilmiş olması, kullanıcı davranışına bağlı açıkların tespitini sınırlamıştır. Yine de modelin mimarisi genişletilmeye açık olduğundan, bu sınırlamaların gelecekte yapılacak çalışmalarda giderilebileceği değerlendirilmektedir.

5. SONUÇ VE ÖNERİLER

Bu çalışma kapsamında geliştirilen katmanlı güvenlik açığı tespit modeli, e-ticaret sistemlerinde sık karşılaşılan zafiyetleri hem teknik hem de yönetsel açıdan sistematik bir biçimde analiz etmeyi hedeflemiştir. Model; hedef belirleme, bilgi toplama, otomatik tarama, manuel doğrulama ve raporlama olmak üzere beş temel aşamadan oluşmakta; böylece yalnızca açıkların saptanmasına değil, bu açıkların gerçek risk potansiyelinin değerlendirilmesine de imkân tanımaktadır. Geliştirilen yapı sayesinde OWASP Top 10 listesinde yer alan temel zafiyet türleri başarıyla tespit edilmiş ve ayrıca CVSS tabanlı puanlama ile risk önceliklendirmesi gerçekleştirilmiştir.

Uygulama sonuçlarına göre en sık karşılaşılan açık türleri SQL Injection, XSS ve kimlik doğrulama zafiyetleri olmuştur. Bu bulgular, OWASP (2023) ve ENISA (2023) gibi literatürdeki çalışmalarda vurgulanan eğilimlerle uyumludur. Ayrıca modelin yöneticilere yönelik sadeleştirilmiş rapor sunabilmesi, yalnızca teknik personelin değil, karar verici üst kadroların da sürece dâhil olmasını sağlayarak güvenlik farkındalığını kurumsal düzeye taşımıştır.

Modelin en belirgin güçlü yönleri arasında, yanlış pozitiflerin manuel testlerle elenmesi, çok katmanlı analiz yapısı ve CVSS uyumlu raporlama yer almaktadır. Bununla birlikte modelin canlı trafik üzerinde test edilmemiş olması, mobil uygulamaların ve API yapıların kapsam dışı kalması gibi sınırlılıkları da göz ardı edilmemelidir. Ancak mimarisi gereği esnek ve modüler olan bu yapı, farklı sistem türlerine kolaylıkla uyarlanabilecek potansiyele sahiptir.

Bu doğrultuda gelecek çalışmalarda aşağıdaki öneriler dikkate alınabilir:

- Modelin mobil e-ticaret uygulamaları ve REST API servisleri gibi bileşenleri de kapsayacak şekilde genişletilmesi: Bu sayede, artan mobil kullanım oranları göz önüne alındığında daha kapsayıcı ve gerçekçi bir güvenlik değerlendirme elde edilmesi mümkün olacaktır.
- Yapay zekâ destekli zafiyet sınıflandırma sistemlerinin entegrasyonu ile sahte pozitiflerin otomatik ayıklanmasının sağlanması: Örneğin, Li et al. (2018) tarafından geliştirilen makine öğrenmesi tabanlı zafiyet tespit yaklaşımlarında olduğu gibi, sistem çıktılarının yapay zekâ algoritmaları ile yorumlanması analiz süresini kısaltırken doğruluk potansiyelini artırabilir.
- Gerçek zamanlı kullanıcı trafiği üzerinde çalışan davranışsal anomali tespit modülleri ile modelin proaktif tehdit algılama yeteneğinin güçlendirilmesi: Böylece bilinen açıklar dışında, sıfırıncı gün saldırılarına karşı da erken uyarı sistemleri oluşturulabilir.
- Açık kaynaklı bir sürümünün geliştirilerek küçük ve orta ölçekli işletmelerin ücretsiz olarak yararlanabileceği bir web ara yüzü hâline getirilmesi: Bu öneri, özellikle bütçesi sınırlı olan işletmelerin de güvenlik süreçlerini sürdürülebilir şekilde yönetmelerine olanak sağlayacaktır.
- CVSS skorlamasının yanı sıra işlevsel etkiler, hukuki sonuçlar ve operasyonel riskler gibi çok boyutlu değerlendirme yapılarının da modele entegre edilmesi: Bu tür çok boyutlu risk değerlendirme sistemleri, yalnızca teknik değil, stratejik güvenlik yönetimini de mümkün kılacaktır.

Sonuç olarak, bu çalışma teknik sağlamlık, yönetsel farkındalık ve kurumsal uygulanabilirlik bakımından dengeli bir güvenlik açığı tespit modeli sunmakta; özellikle dinamik ve saldırıya açık yapıya sahip e-ticaret sistemlerinde sürdürülebilir güvenliğin sağlanmasına katkıda bulunmaktadır. Bu yönüyle, hem akademik literatüre hem de sektörel uygulamalara doğrudan katkı sağlayacak nitelikte bir öneri olarak değerlendirilmektedir.

KAYNAKÇA

Alshamrani, A., Myneni, S., Chowdhary, A., & Huang, D. (2019). A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities. *IEEE Communications Surveys & Tutorials, 21*(2), 1851–1877. https://doi.org/10.1109/COMST.2019.2891891

Anderson, R. (2020). *Security engineering: A guide to building dependable distributed systems* (3rd ed.). Wiley.

Bau, J., Bursztein, E., Gupta, D., & Mitchell, J. (2010). State of the art: Automated black-box web application vulnerability testing. *IEEE Symposium on Security and Privacy*, 332–345. https://doi.org/10.1109/SP.2010.27

Chaffey, D., & Ellis-Chadwick, F. (2019). *Digital marketing* (7th ed.). Pearson Education.

Chen, H., & Zhao, F. (2018). Vulnerability detection in web applications: A systematic literature review. *Journal of Systems and Software, 144*, 314–327. https://doi.org/10.1016/j.jss.2018.07.002

Choudhary, N., & Sharma, A. (2022). Web application vulnerabilities and security issues: A systematic mapping study. *Journal of King Saud University - Computer and Information Sciences, 34*(9), 6575–6590. https://doi.org/10.1016/j.jksuci.2021.04.008

Doupe, A., Cova, M., & Vigna, G. (2010). Why Johnny can't pentest: An analysis of black-box web vulnerability scanners. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment* (pp. 111–131). Springer. https://doi.org/10.1007/978-3-642-14215-4_6

ENISA. (2023). *Threat landscape 2023*. European Union Agency for Cybersecurity. https://www.enisa.europa.eu/publications

FIRST.org. (2019). *Common vulnerability scoring system v3.1: Specification document*. https://www.first.org/cvss/specification-document

Fitzgerald, B., & Stol, K. J. (2015). Continuous software engineering: A roadmap and agenda. *Journal of Systems and Software, 123*, 176–189. https://doi.org/10.1016/j.jss.2015.06.063

Geer, D. (2015). Penetration testing: A duel between attacker and defender. *IEEE Security & Privacy, 13*(3), 84–87. https://doi.org/10.1109/MSP.2015.65

Goseva-Popstojanova, K., & Perhinschi, A. (2015). On the capability of static code analysis to detect security vulnerabilities. *Information and Software Technology, 68*, 18–33. https://doi.org/10.1016/j.infsof.2015.08.002

Gupta, A., & Gupta, S. (2015). Cross-site scripting (XSS) attacks and defense mechanisms: Classification and state-of-the-art. *International Journal of System Assurance Engineering and Management, 6*(S1), 125–134. https://doi.org/10.1007/s13198-014-0270-z

Halfond, W. G. J., Viegas, J., & Orso, A. (2006). A classification of SQL-injection attacks and countermeasures. In *Proceedings of the IEEE International Symposium on Secure Software Engineering* (pp. 13–15).

Jamil, D., & Zaki, M. (2011). Security issues in cloud computing and countermeasures. *International Journal of Engineering Science and Technology, 3*(4), 2672–2676.

Jovanovic, N., Kruegel, C., & Kirda, E. (2006). Static analysis for detecting taint-style vulnerabilities in web applications. *Journal of Computer Security, 14*(1), 57–85. https://doi.org/10.3233/JCS-2006-14103

Li, Z., Xia, X., Lo, D., & Wang, X. (2018). Automatic detection of vulnerable source code in open source software: An empirical study. *IEEE Transactions on Reliability, 67*(3), 1231–1247. https://doi.org/10.1109/TR.2018.2834471

Mell, P., Scarfone, K., & Romanosky, S. (2007). A complete guide to the Common Vulnerability Scoring System version 2.0. *National Institute of Standards and Technology (NIST)*.

NIST. (2022). *Special publication 800-115: Technical guide to information security testing and assessment*. National Institute of Standards and Technology.

OWASP. (2023). *OWASP top ten web application security risks – 2023*. https://owasp.org/Top10

Solove, D. J. (2007). *The digital person: Technology and privacy in the information age*. NYU Press.

Suto, L. (2010). *Analyzing the accuracy and time costs of web application security scanners*. The Leviathan Security Group.

Zhou, Y., Sharma, A., & Sadeghi, A.-R. (2020). Automated risk assessment and mitigation of web vulnerabilities using hybrid analysis. *ACM Transactions on Privacy and Security, 23*(1), 1–30. https://doi.org/10.1145/3378384

Not: *Bu makale, İstanbul Ticaret Üniversitesi Lisansüstü Eğitim Enstitüsü, Siber Güvenlik Tezli Yüksek Lisans Programı'nda, "E -TİCARET SİTELERİ İÇİN GÜVENLİK AÇIĞI TESPİTİ VE RAPORLAMA MODELİ" başlıklı yüksek lisans tezinin ön çalışmalarından yararlanılarak hazırlanmıştır.*