Bilgisayar Bilimleri ve Mühendisliği Dergisi

Cilt-18
Sayı-1
2025

Sızma Denemeleri: Ölçünler, Süreçler ve Kandırma Teknikleri Üzerine Bir İnceleme

Penetration Testings: An Examination of Standards, Processes, and Exploitation Techniques

Mustafa Furkan CEYLAN
Balıkesir Üniversitesi
Bilgisayar Mühendisliği Bölümü
Balıkesir, Türkiye
furkan.ceylan@balikesir.edu.tr
0009-0005-5609-395X

Selçuk KAVUT
Balıkesir Üniversitesi
Bilgisayar Mühendisliği Bölümü
Balıkesir, Türkiye
skavut@balikesir.edu.tr
0000-0002-9460-1418

Öz

Bu çalışma, siber güvenlik alanında yaygın olarak kullanılan sızma denemesi yöntemlerini, ölçünlerini ve güncel istismar tekniklerini ele almaktadır. Sızma denemesi, sistemlerdeki zayıflıkları önceden tespit ederek potansiyel saldırılara karşı etkili savunma stratejileri geliştirmek amacıyla uygulanan temel bir güvenlik sürecidir. Bu çalışmada OWASP, OSSTMM, PTES, ISSAF ve NIST gibi ölçünler incelenmiş; sızma denemesi süreçleri detaylandırılmış ve güvenlik açıklarının belirlenmesine yönelik örnekler sunulmuştur. Ayrıca, SQL enjeksiyonu, MITM (ortadaki adam) saldırıları, DoS (hizmet dışı bırakma) saldırıları, XSS (çapraz site betik çalıştırma) ve parola saldırıları gibi güncel istismar tekniklerine dair uygulamalı örnekler verilmiş ve bu tehditlere karşı alınabilecek önlemler tartışılmıştır. Buna ek olarak, söz konusu tekniklerin uygulanabilirliğini göstermek amacıyla senaryo tabanlı bir sızma denemesi vakası benzetimi düzenlenmiş; elde edilen bulgular CVSS (ortak zafiyet puanlama sistemi) kullanılarak yapılandırılmış bir risk analiziyle değerlendirilmiştir. Teknik ve yönetsel seviyede hazırlanmış örnek bir raporlama çıktısı da çalışmaya dahil edilmiştir. Literatürde sızma denemesi süreçleri ve ölçünleri üzerine ulusal bazda sınırlı sayıda çalışma bulunması, bu makalenin hem sektöre hem de Türkçe literatüre katkı sağlaması açısından önem arz etmektedir.

Anahtar sözcükler: Sızma Denemesi Ölçünleri, Siber Güvenlik, OWASP, İstismar Teknikleri, Vaka Analizi

Abstract

This study addresses commonly used penetration testing methodologies, standards, and recent exploitation techniques in cybersecurity. Penetration testing is a fundamental security procedure applied to identify vulnerabilities in systems in advance and develop effective defense strategies against potential attacks. This work examines standards such as OWASP, OSSTMM, PTES, ISSAF, and NIST, detailing the penetration testing processes and providing examples of identifying security weaknesses. Additionally, practical examples of current exploitation techniques, including SQL injection, MITM (Man-in-the-Middle) attacks, DoS (Denial of Service) attacks, XSS (Cross-Site Scripting), and password attacks, are presented, along with a discussion of countermeasures against these threats. Furthermore, a scenario-based penetration testing case is simulated to demonstrate the application of these techniques, and the findings are evaluated through a structured risk analysis using the CVSS (Common Vulnerability Scoring System). A reporting sample addressing both technical and managerial levels is also included. The limited number of national studies on penetration testing processes and standards in the literature highlights this paper's importance in contributing to the industry and Turkish literature.

Keywords: Penetration Testing Standards, Cybersecurity, OWASP, Exploitation Techniques, Case Study

1 Giriş

Teknolojinin hızla ilerlemesiyle birlikte, web ve mobil uygulamalar modern yaşamın ayrılmaz bir parçası haline gelmiştir. Ancak, yapay zekâ destekli saldırılar, sıfırıncı gün istismarları ve bulut güvenliği sorunları, IoT sistemlerinin

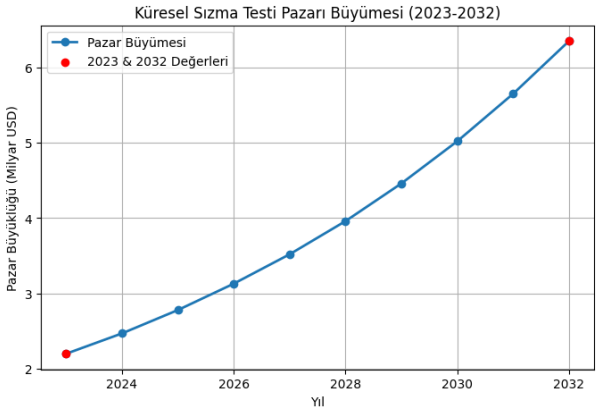
Gönderme, düzeltme ve kabul tarihi: 05.03.2025 - 22.05.2025 – 25.03.2025
Makale türü: Derleme

zayıflığı, siber güvenlik tehditlerini daha karmaşık hale getirmektedir[1]. Dijitalleşmenin yaygınlaşması, kişisel verilerin, kurumsal varlıkların ve kritik altyapının korunmasını zorunlu hale getirmektedir.

Günümüzde, internet tabanlı hizmetlerin artışı ve COVID-19 pandemisiyle birlikte uzaktan çalışmaya ani geçiş, siber güvenlik risklerini önemli ölçüde artırmıştır. Kuruluşlar, yeterli hazırlık yapamadan yeni teknolojileri hızla benimsemek zorunda kalmış ve bu durum, dolandırıcılık amaçlı oltalama saldırılarında artışa, güvenlik açıkları bulunan cihazların yaygınlaşmasına ve genel güvenlik zayıflıklarının ortaya çıkmasına neden olmuştur [2].

Bu güvenlik açıklarının temel nedenlerinden biri, geliştirilen yazılım ve donanım ürünlerinin OWASP [3] güvenlik ilkeleri, NIST [4] ve PTES [5] gibi uluslararası güvenlik yöntemlerine tam uyum sağlamadan üretilmesidir. Eksik veya hatalı uygulanan güvenlik protokolleri, sistemleri savunmasız hale getirerek saldırganlar için kolay hedefler oluşturmaktadır. Nitekim, yakın zamanda yayımlanan bir rapor, web uygulamalarındaki güvenlik açıklarının %82'sinin doğrudan kaynak kodundaki hatalardan kaynaklandığını ortaya koymaktadır [6]. Bu bulgu, güvenli bir yazılım geliştirme sürecinde uluslararası ölçünler ve yöntemlerin hayati önem taşıdığını açıkça göstermektedir. Yakın zamanda yayımlanan bir siber güvenlik raporuna göre, 2021 yılında siber saldırı sayısı 125 milyonken 2022 yılında iki katına çıkarak 250 milyona ulaşmıştır [7]. Bu saldırıların büyük bir kısmı insan kaynaklı değil, otomatikleştirilmiş botlar tarafından gerçekleştirilmiştir.

Bu bağlamda, sızma denemeleri (penetration testings), güvenlik açıklarını saldırganlardan önce tespit etmeyi ve zayıflıkların kullanımı sonucunda potansiyel zararları göstermeyi amaçlayan kritik bir siber güvenlik stratejisidir [8]. Önalıcı bir yaklaşımla riskleri belirleyerek, sistemlerin güvenliğini artırır ve saldırılara karşı savunma mekanizmalarını güçlendirir.

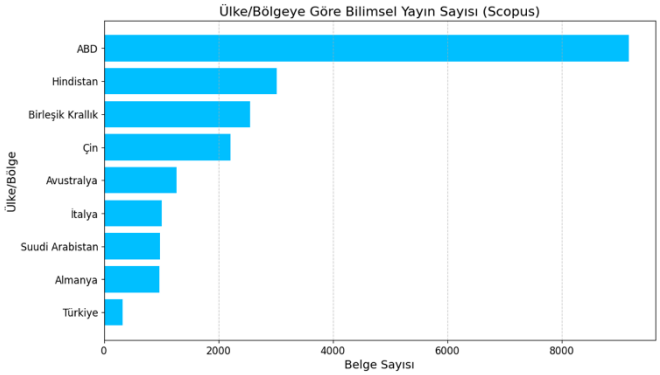


Şekil-1: Küresel sızma denemesi pazar büyümesi (2023-2032 Milyar Dolar) [9]

Artan siber tehditlere karşı kuruluşlar güvenlik stratejilerini güçlendirmekte ve sızma denemelerine olan talep hızla yükselmektedir. Şekil-1'de sunulan küresel sızma denemesi pazarı 2023'te 2,20 milyar USD olarak değerlenmiş olup, 2024-2032 yılları arasında %12,6'lık bir büyüme oranı ile 2032'de 6,35 milyar USD'ye ulaşması beklenmektedir [9]. Bu büyüme, sızma denemelerinin yalnızca teknik bir gereklilik değil, aynı zamanda ticari bir zorunluluk haline geldiğini göstermektedir.

Toplumun yaygın olarak kullandığı sistemlere yönelik saldırıların artmasıyla birlikte, özellikle finans, sağlık ve devlet kurumları gibi kritik sektörlerde güvenlik açıklarını en aza indirmek amacıyla sızma denemelerinin kullanımı giderek yaygınlaşmaktadır. Son yıllarda Türkiye'de siber saldırılar önemli ölçüde artış göstermiş ve bu durum özellikle kritik altyapılar için ciddi tehditler oluşturmuştur. Örneğin, Türkiye'nin DNS sunucularına, bankacılık sistemlerine ve e-devlet hizmetlerine yönelik gerçekleştirilen geniş çaplı bir DDoS saldırısı, saniyede 220 Gbps'ye ulaşarak yaklaşık 40.000 '.tr' uzantılı alan adının devre dışı kalmasına yol açmıştır [10]. Bu tür saldırılar, toplumun günlük yaşamda yaygın olarak kullandığı hizmetleri hedef alarak etki alanını genişletmeyi ve kamuoyu tepkisini artırmayı amaçlamaktadır. Bunun yanı sıra, STM'nin Ocak 2024 Siber Tehdit Durum Raporu'nda [11] yer alan 'The Mother of All Breaches' veri sızıntısında, Twitter, LinkedIn ve Dropbox gibi platformlardan 26 milyar kullanıcının verilerinin sızdırıldığı ve Türkiye'nin de etkilenen ülkeler arasında bulunduğu belirtilmektedir.

Dünyada siber saldırıların artışıyla birlikte bu alandaki bilimsel çalışmalar da hız kazanmış ve akademik araştırmaların sayısı önemli ölçüde artmıştır. Ancak Türkiye, siber güvenlik alanında yapılan akademik yayınlar açısından küresel ölçekte geride kalmaktadır. Şekil- 2'de görüldüğü üzere ABD'de 9174, Hindistan'da 3015, İngiltere'de 2551 makale yayımlanmışken, Türkiye'de yalnızca 329 makale yayımlanmıştır [12]. Bu durum, Türkiye'de siber güvenlik alanındaki akademik üretkenliğin yeterli olmadığını ve daha fazla araştırmaya ihtiyaç duyulduğunu göstermektedir.



Şekil- 2: Ülkelere göre siber güvenlik alanında yapılan bilimsel yayın sayıları

Bu çalışmanın amacı, Türkiye'de artan siber saldırı tehditlerine karşın yeterli olmadığı görülen Türkçe literatüre katkı sağlamaktır. Özellikle, uluslararası sızma denemesi ölçünleri ve yöntemleri hakkında farkındalığın artırılması, geliştiriciler ve şirketler için daha güvenli sistemler tasarlanmasına yardımcı olacaktır. Siber güvenlik alanında, özel olarak sızma denemesi süreçlerinde yaygın olarak kullanılan PTES, OWASP, OSSTMM, ISSAF ve NIST gibi yöntemlerin Türkçe literatürde yeterince ele alınmadığı görülmektedir. Örneğin, [20-27] çalışmalarında yalnızca temel zafiyet tarama süreçleri ele alınmış, ancak uluslararası

yöntemler arasında kapsamlı bir karşılaştırma yapılmamıştır. Ayrıca, bahsedilen çalışmalarda mevcut ölçünlerin uygulanabilirliği ve organizasyonlara uygun yöntemlerin seçimi konusunda yeterli akademik çalışma bulunmamaktadır.

Bu doğrultuda, ilgili çalışmaların ele alındığı bir sonraki bölümde görülebileceği gibi, çalışmamızın katkıları aşağıda özetlenmiştir:

- *Senaryo temelli ve zincirleme bir sızma denemesi vaka analizi sunumu.* Literatürde genellikle bağımsız zayıflıklar veya araç odaklı incelemeler öne çıkarken, bu çalışmada gerçekçi bir saldırı zinciri modeli kurgulanmıştır. Özel olarak, bilgi toplama, XSS, oturum ele geçirme, SQL enjeksiyonu, kaba kuvvet ve log temizleme adımlarının her biri analiz edilmiş ve bu süreç teknik bir rapor formatıyla sonuçlandırılmıştır.

- *Uluslararası sızma denemesi ölçünlerinin bütüncül bir yaklaşımla sistematik biçimde karşılaştırılması.* Önceki çalışmalarda çoğunlukla OWASP ve NIST gibi belirli ölçünlere odaklanıldığı gözlemlenebilir. Çalışmamızda, OWASP, OSSTMM, PTES, ISSAF ve NIST çerçeveleri teknik süreçlerle ilişkilendirilmiş ve kapsamlı biçimde değerlendirilerek Türkçe literatüre kazandırılmıştır.

- *Saldırı türleri ile sızma denemesi yöntemlerinin tematik olarak eşleştirilmesi.* Literatürde çoğunlukla saldırı teknikleri ile yöntemler ayrı ayrı incelenmektedir. Bu çalışmada SQL enjeksiyonu, XSS, MITM, DoS ve parola saldırılarının hangi deneme ölçününde nasıl ele alındığı gösterilmiş; yöntemlerin kapsam ve yeterliliği teknik olarak karşılaştırılmıştır.

Bu çalışmada betimsel bir araştırma yöntemi benimsenmiş; sızma denemesi alanındaki kuramsal yaklaşımlar ve uygulamalı yöntemler, literatür taraması yoluyla çok boyutlu olarak ele alınmıştır. Kaynak taraması, tematik bir yapı çerçevesinde gerçekleştirilmiş; yöntemler, saldırı türleri, risk değerlendirme yaklaşımları ve vaka analizleri gibi başlıca temalar altında sınıflandırılmıştır. Uluslararası literatürde OWASP, NIST, OSSTMM gibi otoritelerin yayımladığı güvenlik ölçünlerinin yanı sıra hakemli akademik makaleler, sektörel teknik raporlar ve açık kaynak güvenlik veri tabanları (Common Vulnerabilities and Exposures - CVE, Common Weakness Enumeration - CWE, Common Vulnerability Scoring System - CVSS) temel kaynak olarak değerlendirilmiştir. Kaynak seçiminde özellikle çalışmanın konusuyla doğrudan ilişkili olması, teknik içerik sunması ve güncel (ağırlıklı 2015 sonrası) olması gibi ölçütler dikkate alınmıştır. Vaka analizinde ise senaryo temelli yapı izlenmiş; tespit edilen zayıflıklar OWASP ve CVE/CWE eşleştirmeleriyle analiz edilmiş, risk seviyeleri CVSS puanlamasına göre değerlendirilmiş ve bulgular hem teknik hem yönetsel düzeyde sistematik biçimde raporlanmıştır. Bu kapsamda çalışmanın bütünsel yapısını destekleyen içerik, aşağıda sıralanan bölümler doğrultusunda yapılandırılmıştır.

Makalenin bir sonraki bölümü, sızma denemeleriyle ilgili ulusal ve uluslararası literatürü inceleyerek bu çalışmanın özellikle Türkçe literatüre olan katkılarını ortaya koymaktadır. Üçüncü bölüm, sızma denemelerinin temel kavramlarını ve tarihsel gelişimini ele almaktadır. Dördüncü bölümde, kara kutu, beyaz kutu ve gri kutu gibi farklı sızma denemesi yöntemlerine yer verilmiştir. Beşinci bölüm, OWASP, OSSTMM, PTES, ISSAF ve NIST gibi yaygın kullanılan sızma denemesi ölçünlerini kapsamlı bir şekilde incelemektedir. Altıncı bölüm, sızma denemesi sürecinin aşamalarını detaylandırmakta; bilgi toplama, tehdit modelleme, zafiyet analizi, istismar ve raporlama gibi temel adımlara odaklanmaktadır. Yedinci bölüm, SQL enjeksiyonu, MITM, DoS, XSS ve parola saldırıları gibi yaygın istismar tekniklerini teknik detaylarıyla ele almaktadır. Sekizinci bölüm, bu saldırı türlerinin sızma denemesi ölçünleriyle olan ilişkisini değerlendirerek yöntemsel bütünlük sağlamaktadır. Dokuzuncu ve son bölümde ise, örnek bir vaka analizi üzerinden uygulamalı bir sızma denemesi senaryosu sunulmakta, deneme zinciri içerisindeki adımlar (bilgi toplama, XSS, oturum ele geçirme, SQL enjeksiyonu, kaba kuvvet ve log silme) OWASP Top 10 ve CVE/CWE eşleştirmeleriyle ilişkilendirilmektedir. Bu bölüm ayrıca, tespit edilen zayıflıklara yönelik risk analizini CVSS değerlendirmesiyle gerçekleştirmekte ve hem teknik hem de yönetsel düzeyde örnek bir raporlama çıktısı sunarak çalışmanın uygulamaya dönük katkılarını ortaya koymaktadır.

2 İlgili Çalışmalar

Sızma denemesi, kuruluşların güvenlik açıklarını önalıcı olarak tespit edip gidermelerini sağlayan önemli bir siber güvenlik uygulamasıdır. Sızma denemesi süreçlerini yapılandırmak ve güvenlik değerlendirmelerini ölçünleştirmek etmek amacıyla OWASP, OSSTMM, PTES, ISSAF ve NIST gibi çeşitli yöntemler geliştirilmiş olup, güvenlik değerlendirme süreçlerini ölçünleştirmeyi amaçlamaktadır. Bu çerçeveler, deneme aşamaları, kullanılan araçlar ve yöntemler açısından rehberlik sunarak sistematik bir güvenlik değerlendirmesi sağlar. Söz konusu ölçünler, uluslararası literatürde de geniş bir şekilde ele alınmış olup, yapılan akademik çalışmalar bu ölçünlerin etkinliğini, karşılaştırmalı analizlerini ve uygulama alanlarını detaylı biçimde incelemektedir.

Shah ve Mehtre [13], güvenlik açığı değerlendirme ve sızma denemesi konusuna odaklanarak, özellikle finans sektöründe yaygın kullanımını ve OWASP ile OSSTMM gibi ölçünlerin önemini vurgulamaktadır. Buna ek olarak, Bertoglio ve Zorzo [14] de benzer bir yaklaşımla sızma denemesi yöntemlerini inceleyerek OWASP, OSSTMM ve PTES'in en yaygın kullanılan çerçeveler olduğunu tespit etmiştir. Çalışma, tek bir ölçünün tüm ihtiyaçları karşılamadığını vurgulayarak, farklı yöntemlerin bir arada kullanılmasının daha kapsamlı güvenlik değerlendirmeleri sağlayacağını öne sürmektedir.

Adam ve ark.[15], sızma denemesi ölçünlerini ve araçlarını değerlendirerek farklı siber güvenlik alanlarındaki etkinliklerini incelemektedir. Araştırmaları, OWASP'ın en yaygın kullanılan çerçeve olduğunu ve PTES'in onu takip ettiğini göstermektedir. Özellikle web güvenlik denemelerinde her iki çerçevenin önemli bir rol oynadığı belirtilmektedir. Araştırmacılar, OWASP ve PTES'in birleştirilmesinin kapsamlı bir sızma denemesi yaklaşımı sağlayabileceğini öne sürmektedir. Bunun yanı sıra, ISSAF, OSSTMM ve NIST'in de yaygın kullanılan ölçünler arasında yer aldığı vurgulanmaktadır.

Shanley [16], çalışmasında dokuz farklı sızma denemesi yöntemini karşılaştırarak, bu ölçünlerin güçlü ve zayıf yönlerini analiz etmektedir. Araştırma, OWASP, OSSTMM, NIST ve PTES'in en yapılandırılmış ve yaygın olarak kabul edilen ölçünler olduğunu ortaya koymaktadır. Sarker ve ark.[17], sızma denemesi ölçünleri, araçları ve puanlama yöntemlerini inceleyerek kapsamlı ve esnek bir yaklaşımın gerekliliğini vurgulamaktadır. Çalışmaları, OWASP, OSSTMM, PTES, ISSAF ve NIST'in en yaygın kullanılan ölçünler olduğunu belirterek, her birinin farklı avantajlar sunduğunu ortaya koymaktadır.

Haq ve ark.[18], mobil uygulama güvenliği için sızma denemesi ölçünlerini inceleyerek, özellikle Android güvenlik açıklarına ve mobil uygulamaların korunmasındaki zorluklara odaklanmaktadır. Çalışmaları, OWASP Mobil Güvenlik Denemesi Kılavuzu'nun (MSTG) en etkili ölçün olduğunu vurgularken, OSSTMM, PTES ve NIST'in mobil odaklı deneme tekniklerinden yoksun olduğunu belirtmektedir. Wen ve ark.[19], yapay zekâ destekli otomasyonun sızma denemelerine entegrasyonunu inceleyerek, güvenlik açıklarının tespiti, tehdit analizi ve izleme süreçlerinde önemli iyileştirmeler sağladığını belirtmektedir. Ancak, mevcut çerçevelerin (OSSTMM, NIST, OWASP) AI yeteneklerine sahip olmaması, tehditlere uyum sağlama konusunda eksiklikler yaratmaktadır. Bu çalışma, güvenlik değerlendirmelerinin AI destekli yöntemlerle genişletilmesi gerektiğini öne sürmektedir.

Sızma denemesi yöntemleri, güvenlik değerlendirmelerini ölçünleştirmek amacıyla geliştirilmiş olup, OWASP ve OSSTMM web ve sistem güvenliğinde öne çıkarken, PTES ve NIST kapsamlı deneme süreçleri sunmaktadır. ISSAF ise detaylı güvenlik analizleri için önemli bir ölçün olarak değerlendirilmektedir. Son yıllarda, yapay zekâ destekli güvenlik denemeleri ve mobil platformlara yönelik çözümler sızma denemesi alanında yeni eğilimler yaratmakta ve mevcut çerçevelerin bu gelişmelere uyum sağlamasını gerekli kılmaktadır.

Bu alanda uluslararası düzeyde kapsamlı bir literatür bulunmasına karşın, Türkiye'deki akademik araştırmaların sınırlı olduğu görülmektedir [20-27]. Bu bölümde, Türkçe literatürde yer alan önemli çalışmalar incelenerek mevcut araştırmalarla karşılaştırılmıştır.

Özellikle OWASP ve NIST çerçevelerine odaklanan çalışmalar, güvenlik deneme süreçlerinin etkinliğini artırmaya yönelik yöntemler sunmaktadır. Vural ve ark. [20] çalışmasında OWASP, NIST ve OSSTMM çerçevesinde güvenlik deneme yöntemlerini ele alarak kurumsal sistemlerde güvenlik denetimlerinin nasıl yürütülebileceğini detaylandırmıştır. Benzer şekilde, Yılmaz ve ark. [21] çalışması, endüstriyel sistemlerde siber güvenlik risklerini azaltmak için NIST ve OWASP ölçünlerinin uygulanmasının önemini vurgulamaktadır. Tulgar ve ark. [22] çalışmasında, "Ulusal Bilgi ve İletişim Güvenliği Rehberi" hazırlanmasına odaklanılmış ve özellikle IoT güvenliği alanındaki uygulama süreci ele alınmıştır. Çeşitli uluslararası bilgi güvenliği ölçünlerinden yararlanılan bu çalışmada, çalışmamızda ele alınan sızma denemesi ölçünlerinden sadece OWASP ve NIST bulunmaktadır. Benzetim ortamında gerçekleştirilen uygulama ile bahsedilen rehberin IoT güvenliği tedbirlerine uyumun nasıl sağlanabileceği gösterilmiş ve kurumlara yönelik uyum yol haritası önerilmiştir.

Web uygulamalarına yönelik güvenlik denemeleri üzerine yapılan çalışmalar, OWASP yöntemine dayalı olarak güvenli yazılım geliştirme süreçlerini iyileştirmeye odaklanmaktadır. Of [23] çalışmasında OWASP çerçevesinde yazılım güvenliği açıklarını ele almış ve güvenli yazılım geliştirme süreçlerine yönelik önerilerde bulunmuştur. Aydoğdu ve ark. [24] çalışmasında, OWASP tarafından tanımlanan web güvenlik açıklarını ele almakta ve güvenlik önlemleri sunmaktadır. Yalçınkaya ve ark. [25] çalışmasında, web uygulamalarında güvenlik denemelerinin kapsamını genişletmeye yönelik yöntemsel bir yaklaşım önermektedir.

Kurumsal sistemlerde güvenlik riskleri ve tehdit modellemeleri üzerine yapılan çalışmalar, ağırlıklı olarak NIST çerçevesinde gerçekleştirilmiştir. Şentürk ve ark. [26] çalışmasında, NIST çerçevesinde aktif dizin ortamlarında karşılaşılabilecek güvenlik risklerini analiz etmiştir. Bu çalışma, genel güvenlik denemeleri yerine, belirli tehdit senaryolarına dayalı daha spesifik bir güvenlik tehdit modeli üzerinden değerlendirme yaparak, olası saldırı ve savunma stratejilerini ele almaktadır. Kestane ve ark. [27] makalesinde, NIST çerçevesinde bulut bilişim sistemlerinde iç güvenlik denetim süreçlerini incelemiştir. Bu çalışmada, güvenlik denemeleri ile ilgili detaylı bir yöntemsel yaklaşım yerine denetim süreçleri üzerinde durulmuştur.

Bu çalışmalar incelendiğinde, Türkçe literatürde OWASP ve NIST'in öne çıktığı, ancak OSSTMM, PTES ve ISSAF gibi diğer önemli yöntemlerin yeterince ele alınmadığı görülmektedir. Ayrıca, Çizelge 1'de görüleceği üzere mevcut çalışmaların çoğu belirli bir yönteme odaklanmakta ve ölçünler arasında sistematik bir karşılaştırma sunmamaktadır. Bu doğrultuda, çalışmamızın PTES, OWASP, OSSTMM, ISSAF ve NIST gibi uluslararası sızma denemesi ölçünlerini karşılaştırarak, uluslararası literatürü referans alıp değerlendirerek Türkçe literatüre katkı sağladığı düşünülmektedir.

Çizelge 1: Türkçe Literatürde ve Çalışmamızda Ele Alınan Ölçünleri Karşılaştırması

Çalışmalar	OWASP	OSSTMM	PTES	ISSAF	NIST
[20]	✓	✓	✗	✗	✓
[21]	✓	✗	✗	✗	✓
[22]	✓	✗	✗	✗	✓
[23]	✓	✗	✗	✗	✗
[24]	✓	✗	✗	✗	✗
[25]	✗	✗	✗	✗	✗
[26]	✗	✗	✗	✗	✓
[27]	✗	✗	✗	✗	✓
Çalışmamız	✓	✓	✓	✓	✓

3 Sızma Denemeleri

Sızma denemeleri, güvenlik değerlendirme sürecinde, deneme uzmanlarının gerçek dünya saldırılarını taklit ederek bir uygulama, sistem veya ağın güvenlik mekanizmalarını aşabilecek yöntemleri belirlemeye çalıştığı bir güvenlik denemesidir [28]. Sızma denemesi "bir açık kapı bulma

sanatı" olarak da tanımlanmaktadır [29]. Sızma denemesinde sürekli değişen tehdit ortamına göre saldırganların kullandığı yöntemleri canlandırılır. Bu sırada sistemlerin güvenlik seviyesi ölçülür ve olası saldırı senaryoları analiz edilir. Günümüzde sızma denemesi, organizasyonların siber güvenlik stratejilerinin temel bir bileşeni haline gelmiş olup, birçok sektörde güvenlik gereksinimleri doğrultusunda uygulanmaktadır.

Sızma denemelerinin kökeni, 1970'lerin başlarına kadar uzanır. ABD Savunma Bakanlığı, askeri sistemlerdeki güvenlik açıklarını tespit etmek amacıyla ilk girişimlerde bulunmuş ve bu çalışmalar modern sızma denemelerinin temelini atmıştır [30]. Bu denemeler, bilgisayar sistemlerinin güvenliğini değerlendirmek için kullanılan ilk sistematik yaklaşımlar olmuştur.

1990'lı yıllarda, ağ güvenliği alanında ilk otomatik deneme araçları geliştirilmeye başlanmıştır. 1995 yılında piyasaya sürülen Security Administrator Tool for Analyzing Networks (SATAN), ilk geniş çaplı otomatik zafiyet tarama araçlarından biri olarak kabul edilir [31]. Bu araç, 100'den fazla bilinen güvenlik açığını tarayarak sistemlerin savunmasız noktalarını belirleyebilme yeteneği sunmuştur. Ancak, kötü niyetli saldırganların da bu aracı kullanmasıyla, siber güvenlik topluluğunda tartışmalara yol açmıştır.

Bu dönemin ardından, Nmap, Nessus ve Metasploit gibi araçlar geliştirilmiş ve sızma denemelerinin daha geniş kitleler tarafından uygulanabilir hale gelmesini sağlamıştır [32]. Bu araçlar, sızma denemelerini manuel süreçlerden otomatik sistemlere doğru taşıyarak, daha hızlı ve kapsamlı güvenlik değerlendirmeleri yapılmasını mümkün kılmıştır.

Günümüzde sızma denemeleri, gelişen tehdit ortamına uyum sağlayarak daha kapsamlı ve sistematik hale gelmiştir. Yapay zekâ destekli otomatik sistemler [33] ve Snort [34] gibi otonom güvenlik araçları, gerçek zamanlı verilere dayalı olarak stratejilerini dinamik bir şekilde ayarlayarak geleneksel yöntemlere kıyasla daha karmaşık siber saldırı senaryolarını daha gerçekçi bir şekilde canlandırabilmektedir [19].

Bununla birlikte, sızma denemelerinin etkinliği, belirli bir yönteme dayanılarak yürütülmesiyle doğrudan ilişkilidir. PTES, OWASP, OSSTMM, ISSAF ve NIST gibi uluslararası ölçünler, sızma denemelerinin sistematik olarak planlanmasını ve güvenlik açıklarının tutarlı bir şekilde tespit edilmesini sağlamaktadır. Yapay zekâ destekli araçlar, bu yöntemlere entegre edildiğinde, zafiyet analizlerinin hızlanmasını ve kapsamlı güvenlik değerlendirmelerinin daha verimli hale getirilmesini mümkün kılmaktadır [35]. Bu entegrasyon, manuel deneme süreçlerinin zaman alıcı ve maliyetli doğasını azaltarak, sızma denemelerinin ölçeklenebilirliğini artırmakta ve kurumsal güvenlik stratejilerinin daha önalıcı hale gelmesine katkı sunmaktadır.

4 Sızma Denemelerinin Yöntemleri

Sızma denemeleri, yazılım deneme yöntemleri arasında önemli bir yer işgal eder ve sistematik deneme süreçlerinin kritik bir parçasıdır. Bu denemeler, çeşitli saldırı senaryoları tasarlanarak gerçekleştirilir ve yazılım veya sistemdeki güvenlik zayıflıklarını belirlemeyi amaçlar. Şekil-3'de görüldüğü üzere üç farklı türü bulunur: Kara kutu, gri kutu ve beyaz kutu denemeleri [36]. Kara kutu denemesi saldırganın hiçbir bilgiye sahip olmadığı bir senaryoyu varsayarken, beyaz kutu denemesi sistemin iç işleyişinin tam olarak bilindiği bir senaryoyu varsayarak gerçekleştirilir. Gri kutu denemesi ise sınırlı bilgiye sahip bir saldırganın bakış açısıyla gerçekleştirilir. Bu farklı yaklaşımlar, farklı türde zayıf noktaların tespit edilmesine ve güvenlik açıklarının giderilmesine yardımcı olur [36], [37]. Bu sayede yazılımlar daha güvenli hale getirilir ve potansiyel saldırılara karşı daha dirençli hale gelir.

4.1 Kara Kutu Denemesi

Bu deneme türü, bilgi gereksiniminin düşük olduğu bir yaklaşımla icra edilir ve güvenlik uzmanları sanki gerçek bir saldırganmış (black hat) gibi hareket eder [38]. Ürün ya da şirket ortamının dışından yapılan bir deneme yöntemidir.

Şekil- 3: Sızma denemesi yöntemleri

Deneme yapanlar ile yazılım geliştiricileri genellikle bağımsız çalıştıklarından, bu denemeler geliştiricilerin gözünden kaçan hataların tespit edilmesine yardımcı olur. Aynı zamanda alınan güvenlik tedbirlerinin ne kadar etkili olduğunu da ölçer. Dolayısıyla, kara kutu yöntemi ile gerçekleştirilen denemeler, yazılımın güvenilirliğini artırmak için önemli bir araç olarak kabul edilir.

4.2 Beyaz Kutu Denemesi

Beyaz kutu denemesi, kara kutu denemesiyle birlikte yazılım deneme süreçlerinde önemli bir rol oynar ve birbirlerini tamamlayıcı olarak kullanırlar. Beyaz kutu denemesinde saldırgan, ürünün ağ topolojisi, portları, sistem bilgileri, servisler dahil olmak üzere hedefler hakkında detaylı bilgiye hatta kaynak koda sahiptir. Dolayısıyla, deneme edilmek istenen unsurlar daha özel olarak belirlenebilir ve denemeler daha derinlemesine tasarlanabilir. Beyaz kutu denemesinde, tasarım ve planlamanın detaylarına hâkim olunduğu için, deneme senaryoları ve stratejileri daha doğrudan ve özelleştirilebilir bir şekilde oluşturulabilmektedir [38]. Kara kutu denemesinde gereken bilgi toplama adımına ihtiyaç duyulmadığından dolayı bu deneme yöntemi zaman açısından daha az maliyetlidir. Bu deneme türü genellikle şirket içinde gerçekleştirilir ve kara kutu denemesi ile kullanıldığında, yazılımın farklı yönlerini kapsamlı bir şekilde deneme etmeye yardımcı olur. Bu nedenle, beyaz kutu denemesi, yazılımın güvenilirliğini ve kalitesini artırmak için önemli bir araç olarak kabul edilir.

4.3 Gri Kutu Denemesi

Gri kutu denemesi, kara kutu ve beyaz kutu deneme tekniklerinin birleşiminden oluşan melez bir yaklaşımdır. Bu yöntemde güvenlik uzmanı, sistemin tüm kaynak koduna sahip değildir; ancak tamamen dış kaynaklı bir saldırgan gibi de değildir. Genellikle kurum içinden bir çalışanın bakış açısını taklit ederek, fiziksel ortama, sistem mimarisine ve belirli web uygulamalarına sınırlı düzeyde erişim sağlar [38]. Gri kutu denemesinde, senaryoların tasarımında beyaz kutu yaklaşımı; uygulama aşamasında ise kara kutu teknikleri kullanılır. Diğer bir ifadeyle, yazılımın iç yapısını kısmen anlamak ve kodu incelemek için içsel bilgilerden yararlanılırken, dışarıdan gelen bir saldırganın bakış açısıyla davranışlar ve güvenlik açıkları deneme edilir. Bu sayede hem içsel zayıflıklar hem de dışa açık tehdit yüzeyleri tespit edilebilir. Gri kutu deneme yöntemi, sistemin hem içeriden hem de dışarıdan değerlendirilebilmesini sağladığı için dengeli bir yaklaşım sunmakta; ayrıca yazılım geliştirme sürecinde güvenlik açıklarının erken tespitiyle ürünün toplam maliyetini azaltma potansiyeli taşımaktadır [39].

Ele alınan üç farklı sızma denemesi yönteminin bilgi seviyesi, gerçekçilik, zaman ve maliyet, avantajlar ve dezavantajlar açısından karşılaştırmalı analizi Çizelge 2'de sunulmaktadır. Yöntemlerin güçlü ve zayıf yönlerini sistematik bir çerçevede göstererek, farklı senaryolar için en uygun yöntemin belirlenmesine katkı sağlamayı amaçlamaktadır.

Çizelge 2: Kara Kutu, Beyaz Kutu ve Gri Kutu Denemesi Karşılaştırması

Özellikler	Kara Kutu Denemesi	Beyaz Kutu Denemesi	Gri Kutu Denemesi
Bilgi Seviyesi	Sisteme dair hiçbir bilgiye sahip olunmaz.	Tüm sistem bilgileri, kaynak kodu ve yapı detayları bilinmektedir.	Sınırlı sistem bilgisine sahiptir.
Gerçekçilik	Gerçek saldırgan senaryolarına en yakın denemedir.	İç güvenlik açıklarını belirlemeye odaklanır, ancak dış tehditleri yansıtmaz.	Hem iç hem dış tehdit perspektifinden analiz sağlar.
Zaman ve Maliyet	Uzun sürebilir, maliyetlidir.	Daha az zaman ve kaynak gerektirir.	Orta düzeyde zaman ve maliyet gerektirir.
Avantajları	Dış tehditlerin benzetimini sağlar.	Derinlemesine analiz ile spesifik güvenlik açıklarını ortaya çıkarır.	İç ve dış tehditleri birlikte değerlendirme imkânı sunar.
Dezavantajları	İç tehditleri ve sistem içi güvenlik açıklarını tespit etmede yetersizdir.	Gerçek saldırgan perspektifini tam olarak yansıtmaz.	Her iki yöntemin bazı sınırlamalarını içerebilir.

5 Sızma Denemesi Ölçünleri

Sızma denemesi, sistemlerin güvenlik açıklarını belirlemek ve riskleri minimize etmek amacıyla uzman ekipler tarafından yürütülen sistematik bir süreçtir. Yeni güvenlik açıklarının sürekli keşfedilmesi ve saldırı tekniklerinin gelişmesi, bu alanda ölçünleştirilmiş yöntemlere duyulan ihtiyacı artırmaktadır. Bu ihtiyacı karşılamak üzere, OSSTMM, ISSAF, PTES, NIST ve OWASP gibi uluslararası kabul görmüş sızma denemesi ölçünleri geliştirilmiştir [14]. Bu ölçünler, sızma denemelerinin belirli yöntemlere uygun şekilde yürütülmesini sağlayarak, deneme süreçlerini daha sistematik ve tekrarlanabilir hale getirmektedir. Ayrıca, ölçünler hem kurumlar hem de bağımsız güvenlik araştırmacıları için ortak bir çerçeve sunarak güvenlik değerlendirmelerinin tutarlılığını artırmaktadır.

5.1 Açık Web Uygulama Güvenlik Projesi

Açık Web Uygulama Güvenlik Projesi (OWASP – Open Web Application Security Project) kâr amacı gütmeyen ve web uygulamalarının güvenliğini artırmaya odaklanan küresel bir organizasyondur [40]. 2001 yılında kurulan bu platform, güvenlik araştırmacıları, geliştiriciler ve sektördeki uzmanlar tarafından desteklenmekte olup yaygın güvenlik hatalarını belirlemek, güvenlik açıklarını azaltmak ve güvenli yazılım geliştirme süreçlerini teşvik etmek amacıyla çeşitli projeler yürütmektedir [13].

OWASP tarafından yayımlanan en önemli kaynaklardan biri OWASP Top Ten listesi [3] olup, belirli dönemlerde güncellenen ve sektörde en yaygın görülen güvenlik açıklarını kritiklik derecesine göre sıralayan bir çalışmadır. Bu liste, yazılım geliştiriciler, güvenlik uzmanları ve deneme mühendisleri için rehber niteliğinde olup, en kritik güvenlik zayıflıklarını tanımlayarak bu zayıflıkların nasıl tespit edileceği ve nasıl önlenebileceği konusunda yol gösterici bilgiler sunmaktadır. Örneğin, SQL enjeksiyonu, kimlik doğrulama açıkları ve siteler arası betik çalıştırma (XSS) gibi saldırı türleri OWASP Top Ten listesinde yıllar boyunca en kritik zayıflıklar arasında yer almıştır [41]. Çalışmamız yapılırken en son çıkan ilk 10 zafiyet listesi aşağıdaki gibidir [3]:

OWASP İlk On Zafiyet – 2021

I. Bozuk Erişim Kontrolü (A01:2021 - Broken Access Control)
II. Şifreleme Hataları (A02:2021 - Cryptographic Failures)
III. Enjeksiyon (A03:2021 - Injection)
IV. Güvensiz Tasarım (A04:2021 - Insecure Design)
V. Güvenlik Yanlış Yapılandırma (A05:2021 - Security Misconfiguration)
VI. Savunmasız ve Eski Bileşenler (A06:2021 - Vulnerable and Outdated Components)
VII. Tanımlama ve Kimlik Doğrulama (A07:2021 - Hataları Identification and Authentication Failures)
VIII. Yazılım ve Veri Bütünlüğü Hataları (A08:2021 - Software and Data Integrity Failures)
IX. Güvenlik Loglama ve İzleme (A09:2021 - Security Logging and Monitoring)
X. Sunucu Taraflı İstek Sahteciliği (A10:2021 - Server Side Request Forgery)

5.2 Açık Kaynak Güvenlik Deneme Yöntemi Kılavuzu

ISECOM (The Institute for Security and Open Methodologies) tarafından ilk olarak 2000 yılında ortaya çıkarılan bu proje

[42], çeşitli disiplinlerden ve dış meslek gruplarından uzmanların oluşturduğu açık bir topluluk tarafından sürdürülmektedir. Ticari ve siyasi etkilerden bağımsız olarak finanse edilen ISECOM projeleri, ortaklıklar, abonelikler, sertifikalar ve lisanslara dayalı araştırmalarla desteklenmektedir. Teknolojik ortamların giderek karmaşıklaşması, uzaktan operasyonlar, sanallaştırma ve bulut bilişim gibi gelişmelere paralel olarak, OSSTMM buna uyum sağlamıştır. Yönteminin günümüzdeki üçüncü versiyonu, insan, fiziksel, kablosuz, telekomünikasyon ve veri ağları da dahil olmak üzere geniş bir kanal yelpazesini kapsamaktadır. Bu esneklik, OSSTMM'nin bulut bilişiminden mobil iletişim ağlarına kadar çeşitli altyapıları değerlendirmek için ideal olmasını sağlamakta ve birden fazla kanalda karşılaşılan karmaşık güvenlik sorunlarını ele alınmasında yardımcı olmaktadır. İnsan faktörünün sistemde ciddi bir güvenlik zafiyeti yaratabileceğine değinen ve sosyal mühendisliğe karşı yöntemler öneren ilk ölçün olarak bilinmektedir [43].

5.3 Sızma Denemesi Yürütme Ölçünü

Penetration Testing Execution Standard (PTES), sızma denemelerinin sistematik olarak gerçekleştirilmesini sağlayan kapsamlı bir yöntemdir [5]. Yedi aşamadan oluşan bir deneme çerçevesi sunarak, saldırı vektörlerinin belirlenmesinden güvenlik açıklarının analizine kadar tüm süreci detaylandırmaktadır. Şekil-4'te gösterildiği gibi, PTES yalnızca teknik değerlendirmeleri değil, aynı zamanda deneme stratejileri ve uzman deneyimlerini de içeren bir rehber olarak geliştirilmiştir. Günümüzde, Bilgi teknolojileri güvenliği alanında yaygın olarak kabul gören ölçünlerden biri olarak kullanılmaktadır [44].

Şekil- 4: Sızma denemesi süreç akış şeması [45]

5.4 Bilgi Sistemleri Güvenlik Değerlendirme Çerçevesi

Bilgi Sistemleri Güvenlik Değerlendirme Çerçevesi [46] (ISSAF – Information Systems Security Assessment Framework), Açık Sistem Bilgi Güvenliği Grubu (OISSG) tarafından geliştirilen ve sızma denemesi süreçlerini ölçünleştirmeyi amaçlayan bir güvenlik deneme yöntemidir. Kurumlara güvenlik değerlendirmelerinde kapsamlı bir çerçeve sunarak, güvenlik denemelerinin planlanması, yürütülmesi ve raporlanması aşamalarını detaylandırmaktadır. ISSAF'ın temel hedefi, sızma denemesi sürecinin neden ve nasıl uygulanması gerektiğine dair eksiksiz bir rehber oluşturmaktır [46]. Diğer yöntemlerin genellikle belirli teknik süreçlere odaklandığı eleştirisiyle geliştirilen ISSAF, ağ güvenliği, sistem güvenliği, fiziksel güvenlik ve uygulama güvenliği gibi geniş bir yelpazeyi kapsayan bir deneme süreci sunmaktadır [18].

Ancak, güncellenmemiş olması nedeniyle modern tehdit ortamında etkinliğini yitirdiği belirtilmektedir [17].

5.5 Ağ Güvenliği Deneme Kılavuzu

Ulusal Ölçünler ve Teknoloji Enstitüsü (NIST) tarafından geliştirilen güvenlik deneme yöntemi, başlangıçta Ağ Güvenliği Deneme Kılavuzu (GNST) olarak tanıtılmış ve SP 800-42 [47] çerçevesinde şekillendirilmiştir. Daha sonra bu rehberin genişletilmiş versiyonu SP 800-115 [28] – Bilgi Güvenliği Deneme ve Değerlendirme Teknik Kılavuzu olarak yayımlanmış ve deneme süreçleri daha kapsamlı hale getirilmiştir.

Her iki kılavuz da deneme süreçlerini dört temel aşamada ele almaktadır: planlama, keşif, saldırı ve raporlama [28] [47]. SP 800-42, özellikle ağ güvenliği denemelerine odaklanırken, SP 800-115 daha geniş kapsamlı bir çerçeve sunarak genel bilgi güvenliği değerlendirmelerini içermektedir. Bu yöntemler, kurumların güvenlik açıklarını tespit etmelerini ve sızma denemelerini daha verimli hale getirmelerini sağlamaktadır [14].

Sızma denemeleri için geliştirilen çeşitli uluslararası ölçünler, güvenlik değerlendirme süreçlerinin sistematik ve etkin bir şekilde yürütülmesini sağlamaktadır. Bu ölçünler, yöntemsel yaklaşımlar, deneme kapsamları ve değerlendirme kriterleri açısından farklılık göstermektedir. Çizelge 3'te, NIST, OWASP, PTES, OSSTMM ve ISSAF gibi yaygın kullanılan sızma denemesi ölçünleri kapsam, avantaj ve dezavantajları açısından karşılaştırılmıştır. Bu karşılaştırma, her bir ölçünün güçlü ve zayıf yönlerini ortaya koyarak, güvenlik uzmanlarının ihtiyaçlarına en uygun yöntemi seçmelerine yardımcı olmayı amaçlamaktadır.

6 Sızma Denemesi Süreci

Bu bölümde, sızma denemelerinde genellikle kullanılan ve güvenlik endüstrisinde kabul gören Şekil- 4'te gösterildiği gibi deneme süreci ve alt başlıkları incelenmiştir [5], [13], [46], [48].

6.1 Deneme Öncesi Etkileşim

Sızma denemelerinin icrası öncesinde, müşteri ile yapılan toplantılar esas alınarak denemesin amaçları, kapsamı, süresi,

yöntemi, zaman planlaması, maliyeti ve ücretlendirme gibi hususlar ele alınır. Özellikle deneme kapsamının doğru şekilde belirlenmesi, kritik sistemlerin deneme sürecinde istem dışı hizmet kesintisine uğramasını önlemek açısından hayati öneme sahiptir [49].

Çizelge 3: Sızma Denemesi Ölçünlerin Karşılaştırılması

Ölçün	Kapsamı	Avantajları	Dezavantajları
NIST [4]	Ağ Güvenliği, Bilgi Güvenliği, Uygulama Güvenliği, Risk Yönetimi	Risk bazlı yaklaşımı destekler ve her organizasyon için uyarlanabilir. Güvenlik çerçevesi; risk yönetimi, tehdit farkındalığı ve adaptif süreçleri içerir.	Sızma denemesini uygulayacak kişinin çerçeveyi iyi bilmesi gerekir. Organizasyonun mevcut güvenlik durumu hakkında detaylı analiz gerektirir.
OWASP [40]	Web Uygulamaları, API Güvenliği, Mobil Güvenlik, Gömülü Sistemler (IoT, Firmware)	Web güvenliği denemeleri için detaylı yöntemler sağlar (OWASP Web Security Testing Guide vb.). Güncel tehditlere göre düzenli olarak güncellenir. Güvenlik bilinci oluşturma, yazılım geliştirme ve deneme süreçlerine katkı sağlar.	Yalnızca uygulama güvenliğine odaklanır, ağ veya sistem güvenliği konularını kapsamaz. Organizasyon güvenliği yerine bireysel deneme süreçlerine ağırlık verir.
PTES [5]	Ağ Güvenliği, Web Uygulamaları, Fiziksel Güvenlik	Sızma denemeleri için yapılandırılmış yedi aşamalı yöntem sunar. Araçlar, uygulama alanları ve teknikler içeren bir rehberdir. Teknik ve yönetsel raporlama süreçleri içerir.	Hala geliştirilmekte olup, belirli ölçünler açısından eksiklikler barındırmaktadır. Gerçek dünyadaki tüm saldırı senaryolarını kapsamaz.
OSSTMM [42]	Ağ Güvenliği, Fiziksel Güvenlik, Kablosuz Güvenlik, Bulut ve Telekomünikasyon Güvenliği	Ölçülebilir ve tekrarlanabilir deneme yöntemini sunar. İşlemsel güvenlik risk değerlendirmelerine odaklanır. Mobil, telekomünikasyon ve bulut güvenliğini kapsayan geniş bir perspektife sahiptir.	Oldukça karmaşıktır ve tam uygulanması zaman alır. Kuramsal kanıtlar yerine doğrudan işlemsel güvenliği ölçmeye odaklanır.
ISSAF [46]	Ağ Güvenliği, Web Uygulamaları, Veri Tabanı ve Sistem Güvenliği	Sızma denemesi aşamalarını belirli araçlarla ilişkilendirerek kapsamlı bir yöntem oluşturur. Deneme sürecinin kapsamını detaylıca tanımlar ve yürütme aşamalarını sistematikleştirir. Teknik ve yönetsel raporlamaları içeren yapılandırılmış bir süreç sunar.	Güncellenmemekte olup, modern güvenlik tehditlerine tam olarak uyum sağlamamaktadır. Eski araç ve yöntemleri içermesi nedeniyle günümüz tehditleri için sınırlı kalabilir.

Bu aşamada, deneme sürecinde karşılıklı iletişim ve koordinasyon gerektirecek kişilerin kimlik bilgilerini içeren bir liste oluşturulur. Denemesin yasal çerçevesi ve uygunluğu için gerekli olan tüm belgeler ve sözleşmeler hazırlanır ve imzalanır. Tüm bu hazırlıklar tamamlandıktan sonra, deneme uzmanları, belirtilen yetkilendirme ve izinler doğrultusunda sızma denemelerine başlamak üzere harekete geçerler. Bu şekilde, deneme süreci başarılı bir şekilde yürütülmüş ve aynı zamanda taraflar arasındaki güven ve iş birliği pekiştirilmiş olur.

6.2 Bilgi Toplama

Bilgi toplama süreci, sızma denemesinin temel bir unsurudur ve denemesin etkinliğini doğrudan etkiler. Bilginin çeşitliliği ve niteliği, yapılan denemelerde keşfedilebilecek zayıflıkların kapsamını belirleyen kritik faktörlerdir. Bu bilgiler arasında fiziksel bilgiler (sunucu odasının yeri, geçiş kontrol sistemleri, donanım özellikleri vb.), bireysel bilgiler (çalışan ismi, ünvanı, mail adresi ve sosyal mühendislikte kullanılmak üzere potansiyel hassas kişisel bilgiler vb.), mantıksal ilişkiler (ağ topolojisi, sunucu – istemci ilişkisi ve veri akış diyagramı vb.), organizasyon yapısı (organizasyonun yönetim yapısı, departmanlar ve çalışanları, anahtar kişiler vb.), kullanılan cihazlar (bilgisayar, mobil telefonlar, ağ ekipmanları ve diğer donanımlar vb.), gibi çeşitli türleri bulunmaktadır. Deneme uzmanı belirlenen hedefe üç farklı perspektiften yaklaşır.

Pasif Keşif: Pasif keşif, hedef sistemle doğrudan iletişime geçmeden gerçekleştirilen bilgi toplama faaliyetidir. Bu yöntemde, güvenlik uzmanı hedefin varlığından haberdar olmaksızın; WHOIS sorguları, Google Dorking gibi açık kaynak istihbarat teknikleriyle alan adı, barındırma hizmetleri, DNS kayıtları ve altyapı bileşenlerine dair bilgiler toplar [50]. Herhangi bir ağ trafiği oluşturulmadığından tespit edilme riski yoktur. Ancak, kullanılan veriler genellikle geçmişe dönük ve güncelliğini yitirmiş olabileceğinden analiz sırasında dikkatli olunmalıdır.

Yarı Pasif Keşif: Hedefin dikkatini üzerimize çekmeden, çeşitli kaynaklardan taramalar gerçekleştirilir. Bu kaynaklar, önceden yayınlanmış web sayfaları ve bunların barındığı sunucular, haberler, blog yazıları gibi çeşitli platformlar olabilir. Ağ düzeyinde aktif port taramaları ya da gizli içerik aramaları gibi müdahaleler yapılmaz. Bunun yerine, normal internet trafiği gibi görünen hareketlerle hedefin profil bilgileri toplanmaya çalışılır. Bu yaklaşım, saldırı sonrasında hedefin geriye dönüp trafik akışına baktığında kesin bir yargıya varamamasını sağlamayı amaçlar.

Aktif Keşif: Bu aşama, deneme uzmanının hedef sistemle doğrudan etkileşime geçtiği bilgi toplama sürecidir. Bu aşamada Nmap, FinalRecon gibi araçlar kullanılarak açık portlar, çalışan servisler, işletim sistemi bilgileri ve yapılandırma ayrıntıları belirlenir [50]. Ağ haritalama, sistem keşfi ve zafiyet taraması gibi işlemler içerdiğinden tespit edilme riski yüksektir. Ancak, sistemin mevcut güvenlik durumu hakkında en kapsamlı ve güncel bilgiler bu aşamada elde edilir

Elde edilen bilgiler ileride açık bir arka kapı bırakmak için ya da farklı deneme yaklaşımlarında kullanmak amacıyla önemlidir.

6.3 Tehdit Modelleme

Önceden toplanan bilgilerin ışığında, en uygun ve etkili deneme tekniklerini kapsayan bir yol haritası oluşturulur ve modelleme süreci başlatılır. Bu modelleme süreci; iş varlıkları,

iş süreçleri, potansiyel tehdit aktörleri ve bu aktörlerin bilgi seviyesi, erişim yetenekleri ve saldırı kapasiteleri gibi unsurları kapsamlı şekilde ele alır. STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege) gibi yöntemlerden yararlanılarak her bir tehdidin kategorik analizi yapılabilirken; DREAD (Damage, Reproducibility, Exploitability, Affected Users, and Discoverability) modeli kullanılarak tehditlerin hasar potansiyeli ve tekrar edilebilirliği gibi faktörlerle risk önceliği hesaplanabilir[51].

Sistem, her bir açıdan olası tehditleri belirlemeye çalışır ve bu tehditler, şirket yönetimiyle yapılan görüşmeler neticesinde belirlenen risk düzeylerine göre önceliklendirilir. Bu çerçevede; tehditlerin sıklığı, erişim türü, hedeflenen sistem bileşenleri ve uygulanan teknikler gibi parametreler dikkate alınır. Öncelikli tehditlerin belirlenmesinin ardından deneme uzmanları, bu tehditlere karşı etkili mücadele stratejileri geliştirerek sızma denemesi senaryolarını oluşturur. Deneme ekibi, saldırı vektörlerini, kullanılacak araçları ve süreç planlamasını belirleyerek denemesin uygulama çerçevesini tanımlar.

Bu yapılandırma, tehditlerin gerçekçi biçimde modellenmesini ve zafiyet analizinin yönünün belirlenmesini sağlar. Nihai olarak; potansiyel zayıflıklar, risk seviyelerine, etki derecelerine ve önem düzeylerine göre sıralanır ve değerlendirilir. Bu sistematik yaklaşım, sızma denemesinin teknik doğruluğunu artırmakla kalmaz, aynı zamanda kurumsal güvenlik önlemlerinin güçlendirilmesine de katkı sunar.

6.4 Zayıflık Analizi

Sistem ve uygulamalardaki zayıflıkların tespit edilmesi, siber güvenlik alanında kritik öneme sahip bir süreç olan zafiyet analizinin temelini oluşturur. Bu analiz, bir organizasyonun bilgi teknolojisi altyapısında potansiyel güvenlik açıklarını belirlemek ve bu açıkların etkilerini anlamak için önemlidir. Zafiyet değerlendirmesi, hedef bir sistemdeki zayıflıkları belirleme, analiz etme ve önceliklendirme sürecidir. Bu süreç; hedef keşfi, sistem taraması, sonuç analizi ve raporlama gibi aşamaları içerir. Amacı; yanlış yapılandırmalar, yazılım hataları ve saldırganlar tarafından istismar edilebilecek işlemsel açıklar gibi zayıflıkları ortaya çıkarmaktır[13].

Zafiyet analizinde, sızma denemesi ekibi, denemesin derinliğini ve kapsamını özenle belirler. Derinlik, her bir sistem veya uygulamanın kimlik doğrulama, erişim kontrolleri ve güvenlik önlemleri gibi kritik bileşenlerini detaylı bir şekilde incelemeyi içerir. Bu inceleme, sistemin savunma mekanizmalarının etkinliğini değerlendirmeyi amaçlar. Kapsam ise, denemesin hangi sistemlerin ve uygulamaların dahil edileceğini ve hangi ağ bölütlerinin deneme edileceğini belirler. Bu aşamada, organizasyonun iş ihtiyaçları, kritik varlıkları ve risk toleransı gibi faktörler dikkate alınır. Kapsamlı bir zafiyet analizi, organizasyonun tüm dijital varlıklarını koruma stratejilerini geliştirmesine ve siber saldırılara karşı daha hazırlıklı olmasına yardımcı olur.

Zafiyet analizi sürecinde, aktif denemeler, pasif gözlemlemeler, araştırmalar ve doğrulama adımları gibi farklı yöntemler kullanılır. Bu adımların tamamlanmasıyla, organizasyonun güvenlik zayıflıkları ve risk alanları hakkında kapsamlı bir anlayış elde edilir. Uzman bir deneme ekibi, deneyim ve bilgi birikimi sayesinde bazen daha önce bilinmeyen (zero-day) zayıflıkları bile tespit edebilir ve organizasyonun savunma stratejilerini güçlendirmesine yardımcı olabilir.

Aktif denemeler, uygulama arayüzü gibi üst seviyeden veya düşük seviyeli bir ağ bileşeniyle etkileşime geçebilir. Bu etkileşimler, manuel ve otomatik yöntemlerle gerçekleştirilebilir. Örneğin, aktif port taraması gibi işlemler, manuel olarak oldukça zaman alabilir; ancak, otomasyon ile yapılarak zaman tasarrufu sağlanır ve deneme uzmanının diğer alanlara daha fazla odaklanabilmesi mümkün hale gelir [52]. Otomatik port, servis veya direkt zafiyet tarayıcı araçlar mevcuttur. Bu araçların kullanılması zaman tasarrufu sağlayabilir, ancak otomatik sistemlerde hata payı bulunmaktadır. Bu nedenle, tespit edilen zayıflıklar, açık portlar veya servisler manuel olarak da doğrulanmalıdır. Bu doğrulama süreci, zafiyet analizinin daha güvenilir verilere dayanmasını sağlar ve saldırı vektörlerinin belirlenmesinde aktif bir rol oynar.

Pasif gözlemlerde, şirket içinden elde edilen dosyaların metadata analizi yapılabilir. Dosyanın oluşturan kişi, oluşturma tarihi, son kaydetme tarihi ve kullanıcının unvanı gibi bilgiler metadatayı oluşturur. Başka bir pasif veri elde etme yöntemi ise ağ trafiğinin dinlenmesidir. Bazen ağ paketlerin de veri sızıntıları oluşabilir ve bu sızıntılar dışarıya açık ve güvensiz bir şekilde yayılabilir. Bu veriler yeterince toplandığında, önemli bilgiler elde edilebilir.

Doğrulama aşamasında, sızma denemesi uzmanları genellikle aynı ana bilgisayarda birden fazla araç kullanarak elde ettikleri tekrarlanan belirli zayıflıkların mikro sorunlarına odaklanır. Ancak bu durum, deneme çıktısındaki istatistiksel sonuçları yanıltıcı bir şekilde etkileyebilir ve yanlış bir risk profili oluşturabilir. Belirli korelasyon, belirli bir tanımlanabilir soruna ilişkin bilgileri (zafiyet kimliği, Common Vulnerabilities and Exposures - CVE, satıcı dizinleme numaraları vb.) mikro faktörlerle (makine adı, IP, tam alan adı, MAC Adresi vb.) gruplandırır. Örneğin, aynı zafiyeti CVE numarasına göre gruplandırarak, farklı araçlardan gelen bulguları birleştirebiliriz. Böylece zayıflıklar tekrara uğramaz ve farklı araçlardan faydalanarak daha doğru sonuçlar elde edilir.

Araştırma aşamasında ise tespit edilen zayıflıkların kaynağı araştırılır ve bir açık kaynak yazılım veya aracından kaynaklanıp kaynaklanmadığı incelenir. Ayrıca, zafiyetin ne kadar kolay ulaşılabilir ve kullanılabilir olduğu değerlendirilir. Genellikle, bulunan zayıflıklar bir sürecin açığı olabileceğinden dolayı hızla çözüme kavuşturulabilir.

6.5 İstismar

İstismar aşaması, sızma denemesi sürecinin en kritik ve zorlu adımlarından biridir. Bu aşamada, sızma denemesi ekibi, hedef sistemde tespit edilen güvenlik açıklarını kullanarak yetkisiz erişim elde etmeye veya sistemin normal işleyişini aksatmaya çalışır. Sızma denemesi ekibinin amacı, gerçek dünyadaki saldırganlar gibi davranarak, hedef sistemin

güvenlik önlemlerini aşmak ve hedef sistem üzerinde kontrol sağlamaktır.

İstismar süreci, yalnızca teknik bilgi ve becerileri değil, aynı zamanda hedef sistemi analiz etme, uygun saldırı tekniklerini seçme ve saldırıları dikkatlice planlama yeteneğini de gerektirir. Bu nedenle, bu aşama genellikle diğer aşamalardan daha fazla zaman alır ve titizlikle yürütülmesi gerekir. İstismar aşaması sırasında, sızma denemesi ekibi OWASP İlk On Zafiyet tablosunda da yer alan başlıklar kapsamında aşağıda bir kısmı belirtilen çeşitli gerçek dünya saldırı tekniklerini kullanır [48]:

I. SQL Enjeksiyonu (SQL Injection)
II. Ortadaki Adam Saldırısı (Man-in-the-middle (MITM))
III. Hizmet Reddi Saldırısı (Denial of Service Attack- DoS)
IV. Parola Saldırısı (Password Attack)
V. Siteler Arası Betik Çalıştırma Saldırısı (Cross-site scripting - XSS)

Bu saldırı türleri, Bölüm 7 'de detaylı şekilde incelenmiştir.

İstismar aşamasında, saldırı türlerinden birini veya birkaçını hedef sistem üzerinde denemek, mevcut güvenlik açıklarının ne kadar ciddi olduğunu ortaya koymak açısından önemlidir. Ancak bu aşamada, saldırıların dikkatli bir şekilde yürütülmesi gerekmektedir; çünkü hedef sistem üzerinde gerçekleştirilen herhangi bir saldırı, sistemin çökmesine veya işlevselliğinin bozulmasına neden olabilir. Sızma denemesi ekipleri, sistemin güvenlik mekanizmalarını atlatmaya çalışırken genellikle şu tür engellerle karşılaşırlar:

1. Güvenlik yazılımları: Çoğu sistem anti-virüs yazılımı, saldırı tespit sistemleri (Intrusion Detection Systems - IDS), saldırı önleme sistemleri (Intrusion Prevention System - IPS) ve web güvenlik duvarları (WAF) ile korunur. Sızma denemesi ekipleri, bu yazılımları aşarak saldırılarını gerçekleştirmek zorundadır.

2. Şifreleme ve kod karartma teknikleri: Modern sistemler veri güvenliğini sağlamak için şifreleme ve kod karartma (obfuscation) tekniklerini kullanır. Bu yöntemler, sızma denemesi ekibinin şifrelenmiş verileri çözmesini ya da karartılmış kodları analiz etmesini zorlaştırır.

3. Gelişmiş güvenlik mekanizmaları: Sistemlerde kullanılan beyaz liste ve kara liste stratejileri, saldırıların engellenmesi için önemli bir güvenlik önlemidir. Beyaz listeleme, yalnızca belirlenen güvenli kaynaklardan gelen işlemlere izin verirken, kara listeleme bilinen zararlı kaynakları engeller. Sızma denemesi ekibi, bu güvenlik önlemlerini aşmak için daha karmaşık saldırı teknikleri geliştirmek zorundadır.

4. Saldırı tespit ve önleme sistemleri: IDS ve IPS sistemleri, ağ trafiğini analiz ederek şüpheli davranışları tespit eder ve saldırıları engeller. Bu tür sistemlerin etkin olduğu ortamlarda sızma denemesi ekipleri, saldırılarını bu sistemlerden gizlemek zorundadır.

5. Ağ bölütleme ve ileri düzey erişim denetimleri: Kurumlar genellikle ağlarını bölütlere ayırır ve her bölüt için ayrı erişim denetimleri uygularlar. Sızma denemesi ekipleri, bu önlemleri aşarak farklı ağlara sızmayı hedefler.

6. Gerçek zamanlı izleme ve alarmlar: Şüpheli faaliyetlerin tespit edilmesi durumunda yöneticilere anında bildirim gönderen gerçek zamanlı izleme ve alarm sistemleri kullanılır. Sızma denemesi ekipleri, bu alarmları devre dışı bırakmaya ya da farklı bir saldırı stratejisi geliştirmeye çalışır.

İstismar aşamasında bulunan zayıflıkların kritiklik düzeyi, sızma denemesinin genellikle en önemli sonuçlarından biridir; çünkü bu aşama, sistemin güvenlik seviyesinin ne derece zayıf olduğunu açık bir şekilde ortaya koyar. İstismar aşaması tamamlandığında, sızma denemesi ekibi ele geçirdiği sistemde kalıcı bir yer edinme aşamasına geçer. Bu süreçte, daha önce ele geçirilen sistemde, saldırıların devamlılığını sağlamak için çeşitli yöntemler kullanılır. Ayrıca, sistemde daha derinlere inmek ve ek hedefler keşfetmek amacıyla yatay hareket stratejileri devreye sokulabilir.

6.6 İstismar Sonrası

Sızma denemesinin istismar sonrası aşamasında, ele geçirilen bir makine üzerindeki kontrolü sürdürmek amaçlanır. Bu aşamada, sızma denemesi ekibi ağ arayüzleri, yönlendirme tabloları, Alan Adı Sistemi (Domain Name System - DNS) ayarları ve vekil sunucuları (proxy servers) gibi bileşenleri analiz ederek ek hedefler belirler ve uzun vadeli erişim için arka kapı programları yükler. Gerektiğinde, deneme sonrası sistemlerde izleri silmek için temizlik işlemi de yapılır.

Uzun vadeli erişim sağlama

1. Arka Kapı Yerleştirme: Arka kapılar, sistem yeniden başlatılsa ya da oturum kapatılsa bile saldırganın sisteme erişimini sürdürmesini sağlayan yazılım ya da yöntemlerdir. Bu, bir hizmeti manipüle etmek ya da zararlı bir yazılım yerleştirmek suretiyle yapılabilir. Arka kapılar, saldırganın ileriki aşamalarda sisteme daha hızlı ve fark edilmeden erişmesini sağlar.

2. Yetki Yükseltme: Sızma denemesi sırasında elde edilen yetkiler her zaman sistemdeki en yüksek yetkiler olmayabilir. İstismar sonrası aşamada saldırgan ya da sızma denemesi ekibi, yönetici (root) yetkilerini ele geçirmeye çalışır. Bu yetkiler, sistem üzerinde tam kontrol sağlamanın yanı sıra daha fazla hedefe ulaşmada da kullanılabilir.

3. Bilgi Çıkışı: Ele geçirilen sistemde ya da ağda önemli veriler bulunabilir. Bu aşamada, kritik verilere erişilip bu verilerin dışarıya sızdırılması deneme edilir. Bu veriler, kimlik bilgileri, müşteri bilgileri, ticari sırlar veya başka hassas bilgiler olabilir. Bilgi çıkışı, genellikle saldırganların nihai hedeflerinden biridir ve ciddi veri ihlallerine neden olabilir.

4. Yatay Hareket Kabiliyeti: İstismar sonrası süreçte, saldırganlar mevcut ele geçirdikleri sistemleri kullanarak başka sistemlere erişmeye çalışır. Yatay hareket, aynı ağ içinde ya da başka bir ağ bölütüne geçmek anlamına gelir. Bu süreç, deneme ekibinin ağın ne kadar derinine nüfuz edebileceğini ve hangi diğer cihazları hedef alabileceğini anlamak için önemlidir. Bu, birden fazla sistemi kontrol altına almayı ve en kritik verilere ulaşmayı sağlayabilir.

5.

İzlerin temizlenmesi

Gerçek dünya saldırılarında, saldırganlar genellikle sistemdeki izlerini temizleyerek kendilerini gizlerler. Bu süreç, sızma denemesi sırasında da denenebilir. Log dosyalarının silinmesi veya log manipülasyonu, sistemde yapılan işlemler hakkında iz bırakmamak için yaygın kullanılan yöntemlerdir. Ayrıca, saldırı sırasında kullanılan araçlar ve arka kapılar da kaldırılır ya da tespit edilmesi zor olacak şekilde gizlenir. Bu adım, saldırganın sistemde uzun süre fark edilmeden kalabilmesi için kritik öneme sahiptir. Ancak, sızma denemeleri tamamlandıktan sonra deneme edilen sistemlerin normal çalışma düzenine dönmesi için bu izlerin tamamen temizlenmesi gerekmektedir.

7. Log Manipülasyonu: Deneme sırasında kullanılan araçların ve yapılan işlemlerin sistemde iz bırakmaması için olay günlüklerinde değişiklik yapılabilir. Bu da saldırının tespit edilmesini zorlaştırır.
8. Zararlı Yazılım (Malware) veya Kod Bloğu (Script) Kaldırma: Sızma denemesi sırasında kullanılan zararlı yazılımlar veya komut dosyaları denemesin sonunda kaldırılmalıdır. Bu, sistemin normal çalışma durumuna geri döndüğünden emin olmak için kritik bir adımdır.
9. Ağ Trafiği İzleme ve Gizleme: Saldırının tespit edilmesini önlemek amacıyla ağ trafiğinde yapılan değişiklikler ve gizli bağlantılar da temizlenmelidir.

6.7 Raporlama

Raporlama aşaması, sızma denemesinin en önemli çıktılarından biridir ve gerçekleştirilen denemelerin sonuçlarının hem teknik ekip hem de yönetim seviyesine uygun biçimde sunulmasını amaçlar. Yapılan tüm deneme ve analizler tamamlandıktan sonra, tespit edilen zayıflıkların detaylandırıldığı kapsamlı bir teknik rapor hazırlanır. Teknik rapor, sızma denemesi süresince kullanılan yöntemleri, tespit edilen güvenlik açıklarının ciddiyetini ve her bir zafiyetin ortaya çıkarılmasında uygulanan teknikleri ayrıntılı bir şekilde ele alır. Ayrıca, sektörde yaygın olarak kullanılan diğer deneme yöntemlerine referanslar da bu rapor kapsamında sunulur. Bu şekilde, teknik ekip, hangi yöntemlerin ve araçların kullanıldığını net bir şekilde anlayarak, bulguların güvenilirliği ve geçerliliği hakkında daha kapsamlı bir değerlendirme yapabilir.

Yönetim seviyesine sunulan rapor ise, daha stratejik ve özetleyici bir yaklaşımla hazırlanır. Bu rapor, belirlenen zayıflıkların risk seviyeleri ve bu zayıflıkların kuruma veya sistemin işleyişine doğurabileceği potansiyel etkiler üzerine kapsamlı bir analiz sunar. Yönetim düzeyinde hazırlanan rapor, genellikle teknik detaylardan arındırılarak, güvenlik açıklarının iş sürekliliği, itibar kaybı ve mali zararlar gibi kritik konular üzerindeki etkilerine odaklanır. Bu değerlendirme, üst düzey yöneticilere, kurumun mevcut siber güvenlik durumunu anlamaları ve alınması gereken önlemler hakkında bilinçli kararlar verebilmeleri için gerekli olan bilgileri sağlar.

Her iki raporda da tespit edilen güvenlik açıklarına yönelik düzeltici ve önleyici öneriler sunulur. Teknik raporda, her bir zafiyetin nasıl giderilebileceği üzerine ayrıntılı teknik çözümler ve en iyi uygulama önerileri yer alırken, yönetim seviyesindeki rapor, bu güvenlik açıklarının kapatılmasının sağlayacağı işlevsel ve işlemsel faydaları vurgular. Özellikle, zafiyet yönetimine yönelik öneriler, siber güvenlik politikalarının iyileştirilmesine ve sistemlerin genel güvenlik duruşunun güçlendirilmesine yönelik somut adımlar içerir.

Bu raporlama süreci, sızma denemesinin bütünsel bir çerçevede değerlendirilebilmesi açısından kritik bir öneme sahiptir. Raporlar hem teknik hem de yönetim seviyesinde, kurumun güvenlik açıklarını kapatma yönündeki eylem planlarını şekillendirmekle kalmaz, aynı zamanda kurumun uzun vadeli siber güvenlik stratejilerine katkı sağlar. Sonuç olarak, bu sistematik ve hedefe yönelik raporlama yaklaşımı, kurumun genel siber güvenlik duruşunu güçlendirmek için gerekli olan stratejik kararların alınmasını kolaylaştırır ve potansiyel tehditlere karşı daha dirençli bir savunma mekanizması oluşturulmasına zemin hazırlar.

7 İstismar Saldırıları

Bu bölümde sistem de bulunan zayıflıklara nasıl saldırı denemelerinin yapıldığını göreceğiz. Sızma denemesi süreçlerinde OWASP'ta yaygın olarak karşılaşılan bazı istismar tiplerinin bahsedilecektir.

7.1 SQL Enjeksiyonu

Saldırgan, SQL enjeksiyonu gerçekleştirmek için, normal bir SQL komutunun anlamını değiştirmek amacıyla kodun içine anahtar kelimeler veya operatörler yerleştirdiğinde, bir SQL saldırısı gerçekleşmiş olur. Kötü niyetli SQL ifadeleri, güvenlik açığı bulunan bir uygulamada, farklı giriş mekanizmaları aracılığıyla uygulamaya dahil edilebilir [53] .

Çerezler aracılığıyla enjeksiyon

SQL enjeksiyonu, çerezler aracılığıyla da gerçekleştirilebilir. Çerezler, istemci tarafında yerel bir dosyada saklanır ve kullanıcının oturum durumunu koruyarak web uygulamalarında kullanılır. Kötü niyetli bir saldırgan, çerezlere zararlı kodlar enjekte edebilir. Eğer web uygulaması, kullanıcı oturumlarını çerezler aracılığıyla izliyorsa, saldırgan bu çerezlere zararlı SQL kodlarını gömebilir ve böylece sunucuya gönderilen SQL sorgularını manipüle edebilir.

Sunucu değişkenleri üzerinden enjeksiyon

Sunucu değişkenleri, Hipermetin Transfer Protokolü (Hypertext Transfer Protocol-HTTP), ağ başlıkları ve çevresel değişkenler (bir kullanıcının oturum bilgileri, geçerli dizin yolu vb.) gibi bir dizi veriyi kapsar. Web uygulamaları, bu sunucu değişkenlerini kullanıcı etkinliğini izlemek ve tarama desenlerini analiz etmek gibi çeşitli amaçlar için kullanır. Ancak, bu değişkenler uygun temizlenmeden bir veri tabanında saklanırsa, bu bir SQL enjeksiyonu zafiyeti yaratabilir. Kötü niyetli aktörler, HTTP ve ağ başlıklarında saklanan değerleri manipüle ederek, SQL enjeksiyon saldırılarını doğrudan başlıklara yerleştirebilirler. Sonuç olarak, veri tabanı sunucu değişkenlerini kaydetmeye çalıştığında, başlıklardaki enjekte edilmiş SQL kodu çalıştırılabilir ve potansiyel güvenlik ihlallerine yol açabilir.

İkinci dereceden enjeksiyon

İkinci dereceden enjeksiyonlar, saldırganların bir sistem veya veri tabanına kötü niyetli girdiler ekleyerek, bu girdiler daha sonra kullanıldığında dolaylı olarak bir SQL enjeksiyon saldırısını tetiklemesidir. Bu tür saldırının amacı, tipik bir (yani birinci dereceden) enjeksiyon saldırısından önemli ölçüde farklılık gösterir. İkinci dereceden enjeksiyonlar, kötü niyetli girişin veri tabanına ilk gönderildiği anda saldırının gerçekleşmesini sağlamaya odaklanmaz. Bunun yerine, saldırganlar, girdinin daha sonradan hangi amaçla kullanılacağını bilerek ve saldırılarını bu kullanım sırasında gerçekleşecek şekilde planlarlar.

Şekil-5'te görülen bir parola güncelleme SQL sorgusu üzerinde nasıl bir enjeksiyon örneği yapılabileceğini aşağıda verilmektedir.

```
queryString="UPDATE users SET password='" + newPassword + "' WHERE
    userName='" + userName + "' AND password='" + oldPassword + "'"
```

Şekil- 5: Parola güncelleme sorgusu

Burada newpasword yerine istediğimiz değeri (ör:newpwd) , oldpassword yerinede rastgele bir (ör: oldpwd) değer ve username yerine "admin' --" yazılırsa SQL sorgusu Şekil-6'deki haline dönüşür.

```
UPDATE users SET password='newpwd'
WHERE userName= 'admin'--' AND password='oldpwd'
```

Şekil- 6: Enjeksiyon sorgusu

"--" operatöründen sonraki alanlar yorum satırı haline gelir ve parola kısmı etkisiz hale gelir.

Bu SQL çalıştığı zaman admin kullanıcısının istediğimiz ve SQL sorgusuna yazdığımız parolasıyla girebiliriz.

7.2 Aradaki Adam Saldırısı

Aradaki adam saldırısı (MITM), iki taraf arasında gerçekleşen iletişim trafiğine izinsiz şekilde müdahale edilmesini içeren kritik bir siber saldırı türüdür. Saldırgan, iletişim kuran tarafların arasına girerek veri akışını gizlice dinleyebilir, değiştirebilir veya yönlendirebilir. Bu saldırılar, özellikle şifrelenmemiş bağlantılarda veya zayıf güvenlik önlemlerine sahip ağlarda kolaylıkla gerçekleştirilebilmektedir.

MITM saldırıları hem yazılım hem de fiziksel yöntemlerle uygulanabilmektedir. Yazılım teknikler arasında ARP dinzime (spoofing) ve DNS dinzimesi gibi ağ katmanı manipülasyonları öne çıkarken, fiziksel yöntemlerde ise ağ trafiğinin pasif olarak izlenmesi için TAP (Test Access Point) cihazları veya ağ anahtarlarında yer alan iskele aynalama (port mirroring) özellikleri kullanılmaktadır[54]. Bu yöntemler sayesinde, saldırganlar ağ trafiğini doğrudan manipüle etmeksizin izleyebilir ve veri toplama işlemini ağ performansını etkilemeden gerçekleştirebilirler. Özellikle TAP'ler, tüm trafik akışını kayıpsız ve kesintisiz şekilde sunarak iskele aynalamaya kıyasla daha güvenilir veri erişimi sağlamaktadır [55]. Diğer bir yöntem ise saldırganın cihazını meşru bir erişim noktası gibi yapılandırarak kullanıcıları sahte bir Wi-Fi ağına yönlendirmesiyle gerçekleştirilen ve şeytan ikizi (evil-twin) olarak adlandırılan yöntem, MITM saldırılarında sıkça kullanılan etkili bir tekniktir [56]. Bu senaryoda kullanıcı, saldırganın erişim noktasına bağlanarak tüm trafiğini onun üzerinden geçirir.

Aşağıda MITM saldırısına ait tipik bir topoloji örneği sunulmuştur ve yukarıda bahsedilen saldırı vektörlerinin nasıl gerçekleştirilebileceği aktarılmıştır.

[KULLANICI] <-----> [SALDIRGAN] <-----> [SUNUCU]

Adım 1: Kullanıcı sunucuya bağlanmak ister.
Adım 2: Trafik saldırganın cihazına yönlendirilir.
Adım 3: Saldırgan veriyi okuyabilir, değiştirebilir.
Adım 4: Saldırgan veriyi sunucuya iletir.
Adım 5: Sunucudan dönen veri tekrar saldırgana gelir.
Adım 6: Manipüle edilmiş veya izlenmiş veri kullanıcıya ulaşır.

Bu tür saldırılar sonucunda kişisel bilgiler, oturum verileri, finansal bilgiler veya kurumsal veriler ele geçirilebilir. Ayrıca saldırganlar, kimlik doğrulama sürecine müdahale edebilir, kullanıcıyı sahte web sayfalarına yönlendirerek kötü amaçlı yazılımlar bulaştırabilir ya da kullanıcı davranışlarını manipüle edebilir.

MITM saldırıları genellikle şu aşamalardan oluşur:

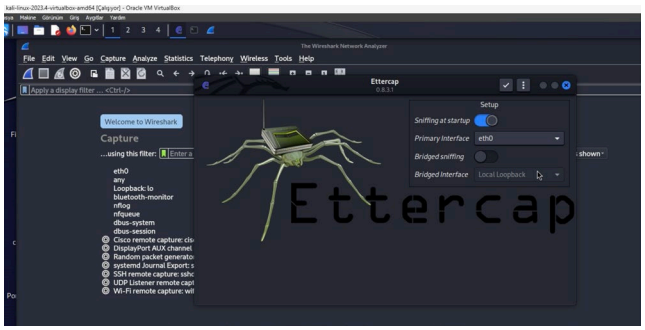
- Yakalama (Intercepting): Saldırgan, iki taraf arasındaki iletişimi kesmek veya yönlendirmek için bir yöntem bulur. Bu, genellikle ARP sahteciliği, DNS zehirlemesi veya Güvenli Soket Katmanı (Secure Sockets Layer - SSL) / Taşıma Katmanı Güvenliği (Transport Layer Security - TLS) zayıflıklari üzerinden yapılır. Örneğin, ARP sahteciliği ile saldırgan, yerel ağdaki cihazlara kendi MAC adresini yönlendirir ve böylece iki taraf arasındaki trafiği kendine çeker.
- Dinleme (Eavesdropping): Saldırgan, iki cihaz arasındaki trafiği başarılı bir şekilde üzerine aldıktan sonra, bu trafiği izleyebilir. Şifrelenmemiş veriler saldırgan tarafından kolayca okunabilir ve hassas bilgiler elde edilebilir. Şifreli trafiğin ise SSL/TLS protokollerindeki güvenlik açıkları kullanılarak çözülmesi mümkündür.
- Manipülasyon (Manipulation): Trafiğin saldırganın eline geçmesiyle birlikte, saldırgan iletişimdeki veri paketlerini değiştirebilir, sahte veri gönderebilir veya iletişim akışını bozar. Örneğin, kullanıcının girdiği banka bilgilerinin değiştirilebilmesi ya da bir kullanıcıyı kimlik avı sitesine yönlendirme gibi durumlar bu aşamada gerçekleşir.

MITM saldırıları, farklı yöntemler kullanılarak gerçekleştirilebilir ancak en yaygın olarak ARP sahteciliği yapılarak gerçekleştirilir [57]. Bu yöntem, ARP protokolünün zayıf güvenliği ve saldırının basit bir şekilde uygulanabilmesi nedeniyle tercih edilmektedir. ARP, MAC adresi ile IP adresi arasında bir eşleme yaparak çalışır ve iki tür mesaj kullanır: istek ve yanıt. Bu iletişim, kaynak ve hedef olmak üzere iki cihaz arasında gerçekleşir. ARP'de herhangi bir güvenlik önlemi bulunmadığı için, sahte bir ARP yanıtı bir cihaz tarafından alınabilir ve istekte bulunulmamış olsa bile bu yanıttaki bilgi cihazın belleğine kaydedilebilir. Bu durum, MITM saldırılarının kolayca gerçekleştirilmesine zemin hazırlamaktadır.

MITM saldırısı örneği

Uygulamada bir iOS cihazın modem olarak davrandığı ve Windows işletim sistemine sahip iki bilgisayarın bağlı olduğu bir ağ ortamında ARP sahteciliği kullanılarak bir MITM saldırısı gerçekleştirilmiştir.

Bu saldırının gerçekleştirilmesin de kullanılan araç Şekil- 7'de görüldüğü gibi Ettercap, Linux tabanlı sistemlerde yaygın olarak kullanılan bir ağ analiz ve saldırı aracıdır. Ettercap başlatıldığında, ilk olarak hangi ağ arabiriminin kullanılacağına karar verilmesi gerekmektedir. Genellikle, kullanılan ağ arayüzü "*eth0*" veya "*wlan0*" olacaktır. Çıkan pencereden uygun ağ arabirimi seçilir ve onaylanır (örneğin, Şekil- 7'de görüldüğü gibi *eth0* seçilir). Böylece tarama (unified sniffing) başlatılır. MITM saldırısının başarılı olması için hedef cihazları belirlemeniz gerekmektedir.



Şekil- 7: Ettercap açılışı

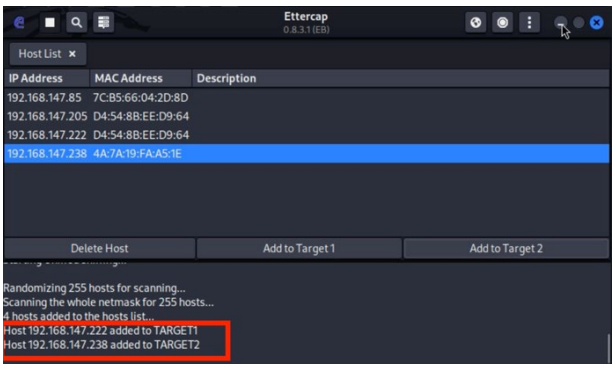
Ettercap, ağdaki tüm cihazları tarayabilir ve bunları IP adresleriyle birlikte listeler. Ettercap arabiriminde menüden Şekil- 8'de olduğu gibi arama sembolüne tıklanarak tarama başlatılır ve listeyi yanındaki butona tıklayarak görüntüleyebiliriz.

Şekil- 8: Tarama başlatılması

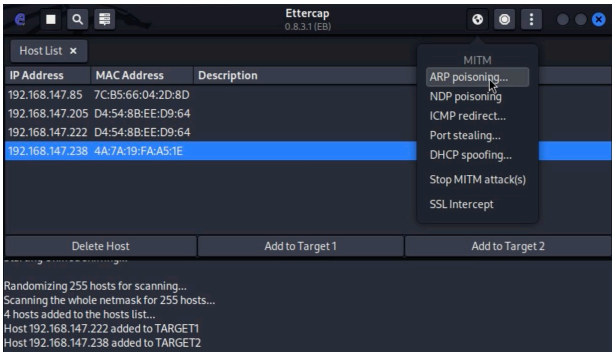
Şekil- 9'de görüldüğü üzere saldırı düzenlenmek istenen cihazlar belirlenir. Örneğin, modem (router) ve kurban cihaz seçilebilir. Ardından ARP sahteciliği kullanılarak MITM saldırısını başlatılacaktır. Bu saldırı, iki cihaz (örneğin, bir kullanıcı cihazı ve modem) arasındaki trafiğin saldırgan üzerinden geçirilmesini sağlar.

Şekil- 10'da görüldüğü gibi menüde yer alan MITM > ARP Poisoning seçeneği izlenerek, "*Sniff remote connections*" seçeneği işaretlenip onaylandığında, iki cihaz arasındaki tüm trafiğin dinlenmesi sağlanmaktadır. Bu işlem, ağ üzerinde gerçekleşen veri iletişimini izlemeye olanak tanımaktadır.

MITM saldırısı ile iki cihaz arasındaki trafik başarılı bir şekilde yönlendirilip kontrol altına alındıktan sonra, ağdaki veri trafiğinin detaylı olarak izlenmesi ve analiz edilmesi amacıyla Wireshark kullanılmıştır.



Şekil- 9: Ağ listesinden hedef cihazların seçimi



Şekil- 10: ARP sahteciliğinin başlatılması

Wireshark, ağ üzerindeki veri paketlerini yakalayarak bu paketlerin içeriğini ayrıntılı şekilde inceleme imkânı sunan güçlü bir ağ analiz aracıdır. Saldırı esnasında Wireshark aracılığıyla ağ trafiği dinlenmiş ve iki cihaz arasında gerçekleşen tüm veri paketleri detaylı olarak incelenmiştir. Özellikle ARP sahteciliği işlemi sonucunda, iki cihaz arasındaki veri alışverişi Wireshark yardımıyla yakalanmış ve analiz edilmiştir.

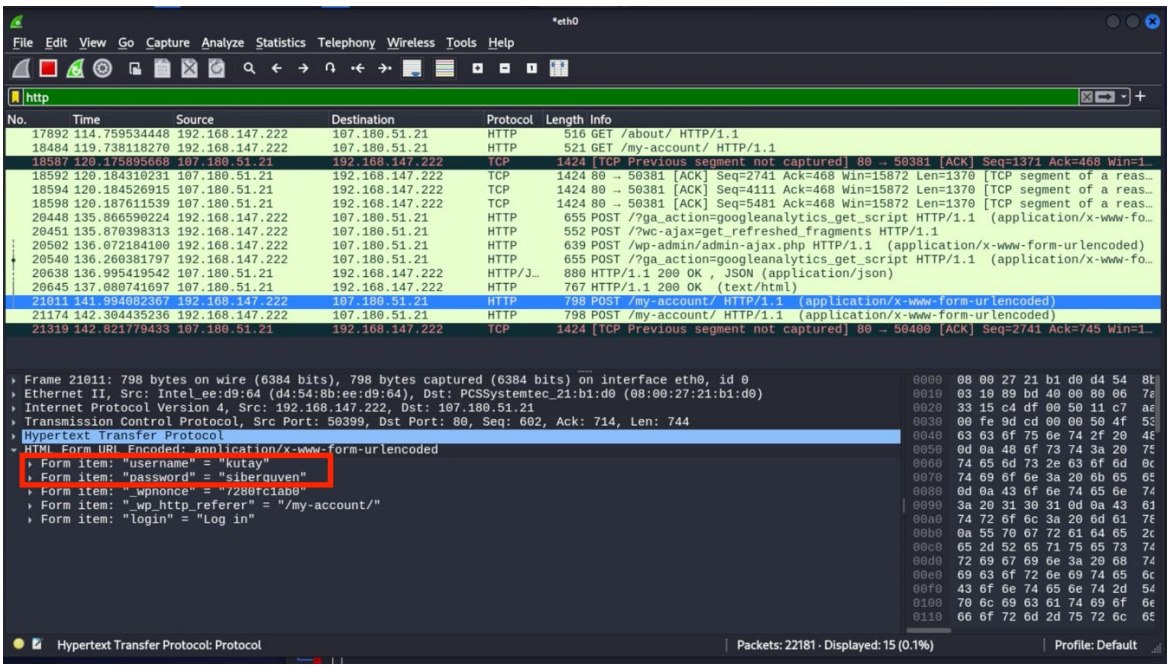
MITM saldırısı yapıldığı sırada mağdur aşağıdaki işlemleri yapmıştır:

- Bir web sayfasına girmiş.
- Kullanıcı adı ve parolasını doğru bir şekilde girmiş.
- Oturumunu açmıştır.

Saldırganın eşzamanlı hamleleri aşağıdaki gibidir:

- MITM saldırısı başarıyla yapılır.
- Herhangi bir ağ izleme aracı açılır (Örneğin Wireshark).
- Saldırganın trafiğini takip etmek için ilgili protokol filtresi uygulanır.
- Gelen ve giden istekler detaylıca incelenerek trafik dinlenir.

Saldırgan, Şekil- 11'de görüldüğü gibi kullanıcının kullanıcı adı ve parolasını yazıp gönderdiği "HTTP POST" isteğini yakalayarak kullanıcının bilgilerini ele geçirmiştir.



Şekil- 11: ARP sahteciliği ile trafiğin dinlenmesi

Bu uygulamadan da anlaşılacağı üzere, kolayca gerçekleştirilebilen ARP sahteciliği saldırıları, bireyler ve kurumlar açısından ciddi güvenlik riskleri taşımakta ve önemli zararlara yol açabilmektedir. MITM saldırılarının risklerini en aza indirmek amacıyla, uç cihazlarda çalışan hafif IDS-IPS sistemleri kullanılarak saldırılar başarıyla tespit edilebilmektedir. Bu yaklaşım, özellikle kaynak kısıtlı ortamlarda etkili bir önlem olarak önerilmektedir [58].

7.3 Hizmet Reddi Saldırısı

Hizmet Reddi (Denial of Service - DDoS) veya birden fazla makine/bilgisayar kullanan DoS saldırısı türü olan Dağıtılmış Hizmet Reddi (Distributed Denial of Service - DDoS) saldırıları, web sunucuları için önemli bir tehdit oluşturmaktadır. Birçok ele geçirilmiş sistem, hedefe aşırı trafik göndererek normal hizmet işleyişini kesintiye uğratır [58]. Bu saldırılar genellikle uygulama katmanında HTTP istekleriyle gerçekleştirilir, bu da kötü niyetli trafiğin normal kullanıcı taleplerini taklit etmesi nedeniyle tespit edilmesini zorlaştırır. (D)DoS saldırısının amacı, bant genişliği, Merkezi İşlem Birimi (Central Processing Unit - CPU) ve bellek gibi sunucu kaynaklarını aşırı yükleyerek gerçek kullanıcıların hizmete erişmesini engellemektir. Bu bölümde tek bir makine tarafından tek bir hedefe hizmet reddi (DoS) saldırı örneği çalışması yapılacaktır.

DoS saldırısı örneği

Yapılan uygulamada, hedef sistemin DoS saldırısına maruz bırakılarak hizmet kesintisine uğratılması için HPing3 aracı kullanılmıştır. İlk aşamada, hedef sistemin domain adresine ping gönderilerek IP adresi elde edilmiştir. Bu işlem, terminale aşağıdaki komutun girilmesiyle gerçekleştirilmiştir:

```
ping hedefdomain.com
```

Bu komut ile hedef domeyne İnternet Kontrol Mesajlaşma Protokolü Yankı İsteği (Internet Control Message Protocol (ICMP) echo request) gönderilmiş ve dönüş yanıtlarından hedef IP adresi elde edilmiştir. Elde edilen IP adresi, sonraki aşamalarda DoS saldırısında kullanılmak üzere kaydedilmiştir.

DoS saldırısının bir parçası olarak İletim Kontrol Protokolü Senkron (Transmission Control Protocol Synchronous – TCP

SYN) saldırısı gerçekleştirilmiştir. Bu saldırı türü, hedef sistemin TCP bağlantılarını aşırı yükleyerek hizmetlerin kesintiye uğratılmasını amaçlamaktadır. HPing3 aracı kullanılarak sahte TCP SYN paketleri hedefe gönderilmiş ve bu şekilde sistemin kaynaklarının tüketilmesi sağlanmıştır. Saldırının başlatılması için aşağıdaki komut kullanılmıştır:

```
sudo hping3 -S --flood -V -p 80 hedefIP
```

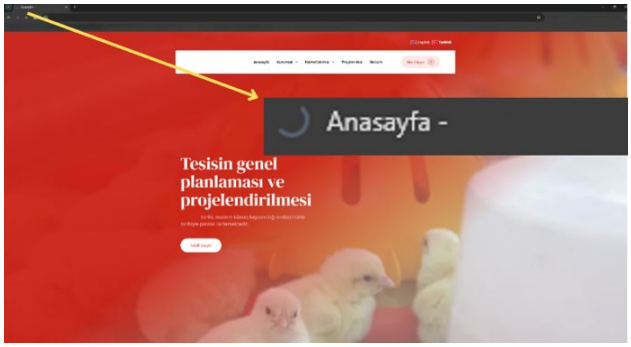
Bu komutta:

- -S bayrağı, TCP SYN paketlerinin gönderilmesini sağlamaktadır.
- --flood parametresi, paketlerin sürekli ve hızlı bir şekilde gönderilmesine olanak tanımaktadır.
- -V seçeneği, saldırı sürecindeki bilgilerin ayrıntılı olarak gösterilmesini sağlamaktadır.
- -p 80 parametresi, hedefin 80 numaralı portuna saldırı yapılacağını belirtmektedir; bu port, genellikle web sunucuları tarafından HTTP hizmetleri için kullanılır.
- hedefIP ise, ping komutuyla elde edilen IP adresini temsil etmektedir.

Saldırı esnasında hedefe sürekli TCP SYN paketleri gönderilmiş ve bu işlem sonucunda hedefin TCP bağlantıları yoğun bir şekilde aşırı yüklenmiştir. Bu yüklenme sonucunda, Şekil-12'de görüldüğü gibi hedef web sayfasının erişilemez/tepkisiz hale geldiği gözlemlenmiştir. Normal kullanıcılar, bu saldırı esnasında hizmetlere erişimde zorluk yaşamış ve web sayfası yüklenememiştir.

7.4 Parola Saldırısı

Parola saldırıları, sistemin kimlik doğrulama bileşenlerini hedef alarak kullanıcı hesaplarına yetkisiz erişim sağlamayı amaçlayan saldırı türleridir. En yaygın yöntemlerden biri olan kaba kuvvet (brute force) saldırısında, parola kombinasyonları sistematik olarak denenir.



Şekil-12: DoS saldırısı sırasında hedef web sayfasının tepkisi

Bu tür saldırılar, parolanın uzunluğu ve karmaşıklığına bağlı olarak zaman ve işlem gücü açısından maliyetli hale gelebilir. Alternatif olarak kullanılan sözlük saldırısı ise, daha önceden bilinen veya sık kullanılan parolaların listesinden yararlanarak çok daha hızlı sonuç verebilir. Kullanıcıların yaygın parolaları tercih etmesi, bu yöntemi özellikle etkili kılmaktadır[59]. Yapılan denemeler, işlemci gücü açısından düşük ila orta seviye tek bir GPU'nun bile birkaç gün içinde parolaların %95'inden fazlasını kırabildiğini, daha özel bir sistemin ise en güçlü %0,5'lik dilim dışındaki tüm parolaları çözebildiğini göstermiştir [59].

Bu tür saldırıların etkinliğini artıran başlıca unsurlardan biri, saldırganların kimlik doğrulama sistemlerini çeşitli otomasyon araçları aracılığıyla sistematik olarak deneme edebilmesidir. Parola tabanlı kimlik doğrulama mekanizmalarına yönelik saldırılarda yaygın olarak kullanılan araçlar arasında Hydra, John the Ripper, Medusa, Cain & Abel ve Burp Suite Intruder öne çıkmaktadır [60]. Bu araçlar, genellikle sözlük tabanlı saldırılar ve kaba kuvvet saldırıları (brute force attacks) yoluyla kullanıcı adı ve parola kombinasyonlarını otomatik olarak denemekte; bu sayede zayıf veya yetersiz güvenlik yapılarını tespit etmede yüksek başarı oranı sergilemektedir. Bu çalışmada, HTTP protokolü üzerinden çalışan parola giriş noktalarına yönelik olarak Burp Suite yazılımının Intruder modülü kullanılmış ve kaba kuvvet saldırısı denemeleri gerçekleştirilmiştir.

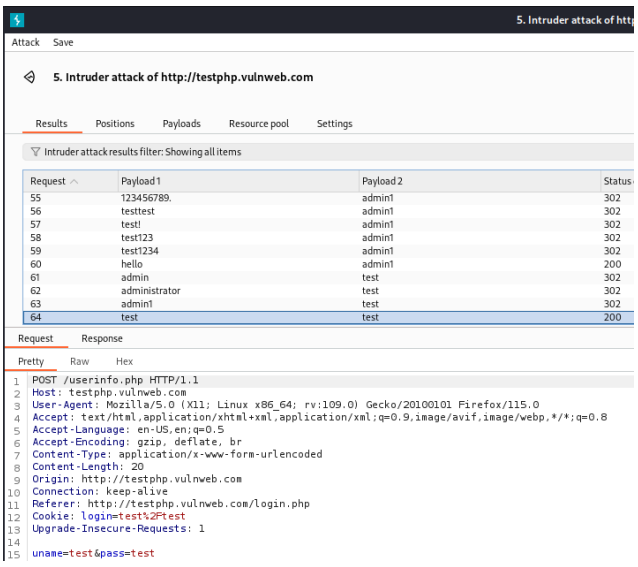
Parola saldırısı örneği

Bu bölümde, kaba kuvvet yöntemi kullanılarak bir parola saldırısının nasıl gerçekleştirildiği anlatılacaktır. Görseller ile süreç detaylandırılarak saldırının işleyişi ve sonuçları açıklanacaktır. Doğru giriş bilgilerini elde etmek amacıyla, birçok olası parola ve kullanıcı adı kombinasyonunun sistematik bir şekilde denendiği bir kaba kuvvet saldırısı gerçekleştirilecektir. Bu saldırı yöntemi, özellikle zayıf parola sistemlerine karşı oldukça etkilidir.

Bu saldırı senaryosunda, Burp Suite'in Intruder modülü kullanılarak hedef sisteme ardışık istekler gönderilmektedir. Intruder modülü, saldırganın belirli bir parametreyi (örneğin, kullanıcı adı ve parola alanlarını) belirleyip bu parametre üzerinde çok sayıda deneme yapmasına olanak tanır. Burp Suite, farklı kullanıcı adı ve parola kombinasyonlarını otomatik olarak deneyerek, bir başarı durumu elde edene kadar sürekli istek göndermektedir. Her deneme için sunucudan alınan yanıtlar kaydedilmekte ve bu yanıtlar üzerinden saldırının başarı durumu değerlendirilmektedir.

Şekil- 13 ve Şekil- 14'te görsellerinde, Burp Suite kullanılarak gerçekleştirilen kaba kuvvet saldırısının sırasıyla istek ve yanıt bilgileri gözlemlenmektedir. Bu adımda, farklı kullanıcı adı ve parola kombinasyonlarını kullanılarak hedef sistemin giriş formuna ardışık istekler gönderilmektedir. İsteklerin her birinde farklı "kullanıcı adı" ve "parola" değerleri denenmektedir (örneğin, admin/deneme, deneme/deneme123 gibi). Gönderilen isteklerin yanıtları incelendiğinde, hedef sunucu her kombinasyona karşı bir yanıt kodu döndürmektedir. Yanıtların 302 ve 200 durum kodları dikkate alınmalıdır. 302 durumu, yönlendirme anlamına gelir ve genellikle hatalı giriş bilgilerine karşı sistemin giriş sayfasına geri yönlendirdiği anlamına gelir. 200 durum kodu ise, başarılı bir girişin gerçekleştiğini ve oturumun açıldığını gösterir.

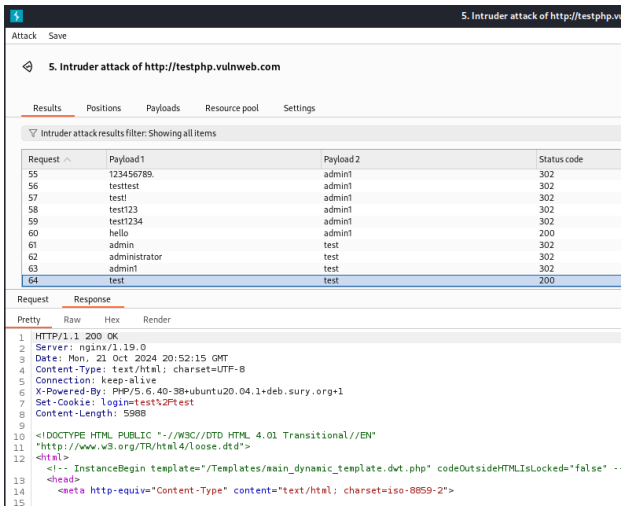
Şekil- 14'te yanıt analizine göre, denenen kombinasyonlardan biri (deneme/deneme) başarıya ulaşmış ve hedef sistemden 200 OK yanıtı alınmıştır. Bu durum, doğru kullanıcı adı ve parola kombinasyonunun bulunduğu anlamına gelir.



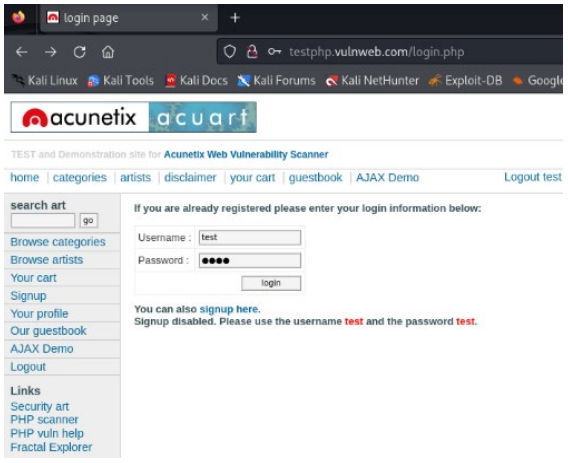
Şekil- 13: Parola saldırısı istek detayı

Şekil- 15-a'da soldaki resimde, Şekil- 14'te görülen 200 OK yanıtı ile doğrulanan kullanıcı adı ve parola kombinasyonu (deneme/deneme) kullanılarak giriş yapılmıştır. Bu noktada, saldırgan, doğru giriş bilgilerini bularak hedef sistemde oturum açmıştır. Bu işlem, saldırının başarıya ulaştığını göstermektedir. Kullanıcı adı ve parola girildikten sonra, Şekil- 15-b'de resimde oturum açıldığına dair bir sayfa görüntülenmektedir. Bu sayfa, başarılı girişin kanıtıdır ve kullanıcının sisteme giriş yaptığını göstermektedir. Giriş ekranında kullanıcı bilgileri ve diğer oturum detayları yer almaktadır. Saldırganın sistemde oturum açtıktan sonra kullanıcı profili bilgilerine eriştiği gözlemlenmektedir.

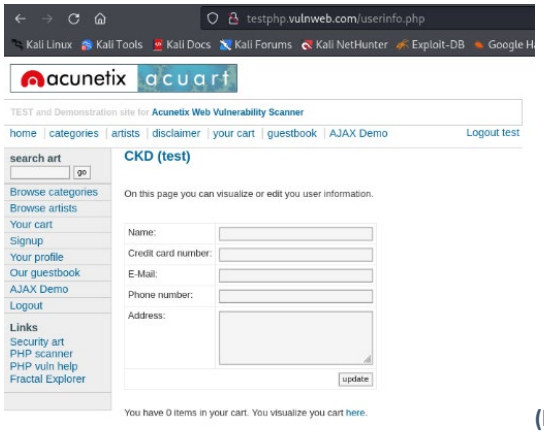
Kullanıcıya ait e-posta adresi, adres bilgileri ve diğer kişisel bilgilerin yer aldığı bu sayfa, saldırının başarılı bir şekilde sonuçlandığını ve sistemdeki kullanıcı verilerine ulaşıldığını göstermektedir. Bu aşamada, saldırganın hedef sistemdeki hassas bilgilere erişimi olduğu ve bunları görüntüleyebildiği anlaşılmaktadır. Kaba kuvvet saldırısı sonucunda sistemin zayıf parola koruma mekanizması aşılmış ve yetkisiz erişim sağlanmıştır.



Şekil- 14: Parola saldırısı yanıt detayı



(a)



(b)

Şekil- 15: Doğru kullanıcı bilgileri ile giriş

7.5 Siteler Arası Betik Çalıştırma Saldırısı

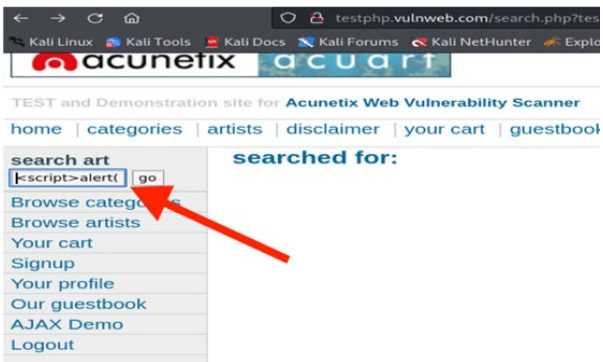
Siteler arası betik çalıştırma saldırısı (Cross-Site Scripting - XSS), saldırganın bir web uygulamasına kötü niyetli komut dosyaları enjekte ettiği bir tür kod enjeksiyonu saldırısıdır. Bu komut dosyaları, genellikle kurbanın haberi olmadan, kurbanın tarayıcısında çalıştırılır ve çerezlerin çalınması, kullanıcı oturumlarının ele geçirilmesi, kullanıcıların kötü amaçlı sitelere yönlendirilmesi veya kötü amaçlı yazılım yayılması gibi çeşitli zararlı sonuçlara yol açar [61]. XSS saldırıları, web uygulamalarındaki yetersiz girdi doğrulama açıklarını istismar eder ve saldırganın içeriği manipüle etmesine olanak tanır. XSS saldırılarının üç ana türü vardır: Yansıtmalı (kalıcı olmayan), depolanmış (kalıcı) ve belge nesne modeli (Document Object Model - DOM) tabanlı XSS[62]. Yansıtmalı XSS, kötü niyetli komut dosyalarının bir web sunucusu üzerinden yansıtılmasıyla gerçekleşirken, depolanmış XSS, saldırganın sunucuya enjekte ettiği ve web sayfasına erişen kullanıcılar tarafından çalıştırılan kodu içerir. DOM tabanlı XSS ise istemci tarafı kodundaki güvenlik açıklarından kaynaklanır. XSS saldırılarını etkili bir şekilde önlemek için en yaygın kullanılan yöntemler arasında, Content Security Policy (CSP) uygulanması, katı girdi doğrulama ve çıktı kodlaması, ayrıca HttpOnly ve Secure niteliklerine sahip çerezlerin kullanılması yer almaktadır[62].

XSS saldırı örneği

Burada XSS saldırısının gerçekleştirilme süreci anlatılmaktadır. XSS, web uygulamalarında güvenlik açıklarından faydalanarak kullanıcı tarayıcılarında zararlı komutlar çalıştırmayı amaçlayan bir saldırı türüdür. Bu tür saldırılar, genellikle yetersiz girdi doğrulama mekanizmalarıyla savunmasız hale gelmiş web sitelerinde yapılır ve saldırganların zararlı kodu hedef siteye enjekte etmesine olanak tanır.

Şekil- 16'te sunulan görselde, kalıcı olmayan (*reflected*) bir XSS saldırısının başlangıç adımı görülmektedir. Burada, bir web uygulamasının arama veya form alanına, JavaScript kodu enjekte edilmiştir. Enjekte edilen kod aşağıdaki gibidir:

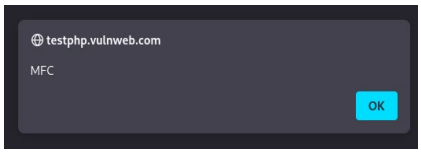
```
<script>alert("MFC")</script>
```



Şekil- 16: Arama alanına JavaScript kodunun enjekte edilmesi

Bu kod, web uygulamasının girdi alanına gönderildiğinde, tarayıcıya bu komutun çalıştırılmasını emreder. Web uygulamasının bu girdiyi doğrulamadan doğrudan kabul etmesi ve tarayıcıya iletmesi, güvenlik açığının varlığını gösterir. Özellikle, sunucu tarafında girdi doğrulama

mekanizmaları yetersiz olduğu zaman, bu tür saldırılar başarılı olur. Şekil- 17'da sunulan görselde, gönderilen XSS kodunun tarayıcıda başarılı bir şekilde çalıştığı gözlemlenmektedir. Tarayıcı, bu enjekte edilen JavaScript kodunu algılar ve komutu çalıştırarak bir uyarı penceresi (popup) oluşturur. Bu durumda, ekranda "MFC" içeren bir uyarı mesajı görüntülenir. Bu işlem, saldırganın tarayıcı üzerinde zararlı kod çalıştırabileceğini ve potansiyel olarak daha ciddi saldırılar gerçekleştirebileceğini kanıtlamaktadır.



Şekil- 17: Başarılı XSS saldırısı çıktısı

8 Saldırı Türleri ile Sızma Denemesi Ölçünlerinin İlişkisi

Sızma denemeleri, güvenlik zayıflıklarını tespit etmek ve sistemleri olası saldırılara karşı daha dayanıklı hale getirmek için kullanılan yöntemsel yaklaşımlardır. Farklı sızma denemesi ölçünleri, belirli saldırı türlerine yönelik tespit ve analiz mekanizmaları sunarak, güvenlik açıklarının giderilmesine katkı sağlamaktadır. Ancak her yöntem, tüm saldırı türlerini kapsamamakta ve farklı odak noktalarına sahip olabilmektedir.

Çizelge 4, yaygın sızma denemesi ölçünlerinin SQL enjeksiyonu, Ortadaki Adam Saldırısı (MITM), Hizmet Reddi Saldırısı (DoS), Parola Saldırıları ve Siteler Arası Betik Çalıştırma (XSS) gibi saldırılarla ilişkisini ortaya koymaktadır. Bu karşılaştırma, farklı ölçünlerin hangi saldırıları ele aldığı ve hangi alanlarda daha etkin olduğu konusunda rehberlik sunmaktadır.

Çizelge 4: Sızma denemesi ölçünlerinin saldırı türleriyle ilişkisi

Saldırı Türü	NIST	OWASP	PTES	OSSTMM	ISSAF
SQL Enjeksiyonu	✓	✓	✓	X	✓
Ortadaki Adam Saldırısı (MITM)	✓	X	✓	✓	✓
Hizmet Reddi Saldırısı (DoS)	✓	X	✓	✓	X
Parola Saldırısı	✓	✓	✓	✓	✓
Siteler Arası Betik Çalıştırma (XSS)	X	✓	✓	X	✓

Saldırı türleri ve sızma denemesi ölçünlerinin ilişkilendirilmesiyle sonucunda, farklı ölçünlerin çeşitli güvenlik açıklarını ele alma kapasiteleri aşağıda maddeler halinde sunulmuştur:

- OWASP, özellikle web uygulamalarına odaklandığından, SQL Enjeksiyonu ve XSS gibi uygulama katmanındaki saldırıları kapsamlı şekilde ele almaktadır. Ancak ağ seviyesindeki saldırılara (MITM, DoS) karşı doğrudan bir yöntem sunmamaktadır [15], [18], [40].
- NIST, daha geniş bir güvenlik perspektifi sunarak hem ağ hem de uygulama güvenliği denemelerini içeren bir yaklaşım benimsemektedir ancak OWASP kadar web güvenliği detaylandırılmamıştır[18], [28].
- PTES, kapsamlı bir saldırı benzetim yöntemi sunarak tehdit modellemelerinin daha sistematik ve derinlemesine gerçekleştirilmesine olanak tanımaktadır. Güvenlik açıklarının nasıl kötüye kullanılabileceğini ve sistem üzerindeki potansiyel etkilerini vurgular [5], [17], [18].
- OSSTMM, MITM ve DoS gibi ağ seviyesindeki tehditlere karşı daha kapsamlı analizler sunarken, uygulama güvenliği konusunda daha sınırlı kalmaktadır[17], [18], [42].
- ISSAF, çok yönlü bir deneme çerçevesi sunduğundan, saldırı türlerinin büyük bir kısmını ele almaktadır, ancak modern tehditlere uyum sağlama açısından güncel değildir [15], [46].

Bu karşılaştırma, farklı sızma denemesi yöntemlerinin güçlü ve zayıf yönlerini değerlendirmek ve hangi tehdit türleri için hangi ölçünlerin daha uygun olduğunu belirlemek adına önemli bir yol gösterici niteliğinde olarak görülebilir.

9 Vaka Analizi: Örnek Sızma Denemesi Senaryosu

Bu vaka çalışmasında, orta ölçekli bir kurumunun web tabanlı bilgi sistemine yönelik gerçekleştirilen kontrollü ve izinli bir sızma denemesi sürecinin benzetimi yapılmıştır. Senaryo, makalenin önceki bölümlerinde detaylı biçimde ele alınan saldırı türlerini içerecek şekilde yapılandırılmış ve deneme zinciri, bilgi toplama, XSS, oturum ele geçirme, SQL enjeksiyonu, parola saldırısı ve log suppression (kısmi DoS) adımlarını içermiştir. Her bir adım, OWASP Top 10 zafiyet kategorileri ve CVE kayıtları temel alınarak değerlendirilmiştir.

Senaryonun sonunda, tespit edilen güvenlik açıkları için etki ve olasılık puanlamasına dayalı risk analizi gerçekleştirilmiş ve bu analizler ışığında hem teknik ekipler hem de yöneticiler için örnek bir raporlama çıktısı hazırlanmıştır. Bu yapı, sızma denemelerinin yalnızca tespit süreciyle sınırlı kalmayıp, kurumsal karar süreçlerini destekleyen ölçülebilir çıktılar üretmesi yönünden önemli bir örnek teşkil etmektedir.

9.1 Senaryo Akışı ve Aşamaları

Adım 1: Bilgi Toplama

Sızma denemesi uzmanı, hedef sistemin genel mimarisi ve servisleri hakkında bilgi edinmek amacıyla Nmap ve Shodan araçlarını kullanmıştır. Açık portların tespiti sonrası HTTP yanıt başlıklarında web sunucusunun (Apache) versiyon bilgisi gibi meta verilerin dışarıya açık olduğu görülmüştür. Bu durum, sistemin yapılandırma seviyesindeki zaaflarını yansıtmakta olup OWASP A05:2021 -Security Misconfiguration kapsamında değerlendirilir.

Adım 2: Komut Enjeksiyonu Tabanlı Oturum Taklidi Hazırlığı (XSS benzeri)

Uzman, kullanıcı başvuru formunda girdi doğrulamasının yapılmadığını tespit etmiş ve form alanına kötü amaçlı bir JavaScript kodu enjekte etmiştir. Enjekte edilen kodun çalışması sonucunda oturum açmış bir yöneticinin oturum çerezi, deneme uzmanının kontrolündeki dış bir sunucuya yönlendirilmiştir. Bu davranış, özellikle istemci etkileşimiyle tetiklenen komut yürütme açıklıklarına benzemekte olup, gerçek dünyada benzer etkiler CVE-2022-30190 (Follina) gibi açıklarla gözlenmiştir. Bu açık, kullanıcı girdisi üzerinden uzaktan komut yürütme ile sistem kontrolünün sağlanmasına olanak tanımaktadır.

Adım 3: Oturum Ele Geçirme (Session Hijacking)

Elde edilen çerez bilgileri, uzman tarafından tarayıcıda oturum oluşturmak amacıyla manuel olarak kullanılmış ve bu sayede sistemin yönetim paneline doğrudan erişim sağlanmıştır. Bu işlem, sistemin oturum yönetimi kurulumunda güvenli bayrak (secure flag) ve HttpOnly parametrelerinin eksikliğinden kaynaklanmaktadır. Zafiyet, OWASP A07:2021 – Identification and Authentication Failures başlığı altında değerlendirilir.

Adım 4: SQL Enjeksiyonu ile Yetkisiz Veri Erişimi

Yönetici panelindeki arama fonksiyonunda, kullanıcı girişleri sunucu tarafında filtrelenmediği için "OR '1'='1" gibi klasik SQL enjeksiyon ifadesiyle veri tabanına doğrudan sorgular gönderilebilmiştir. Bu zafiyet sayesinde sistemdeki kullanıcı bilgileri dışarıya sızdırılmıştır. Bu açık, OWASP A01:2021 – Broken Access Control kapsamında sınıflandırılabilir ve CVE-2022-24124 gibi bilinen SQL enjeksiyonu vakalarıyla benzerlik göstermektedir.

Adım 5: Kaba Kuvvet Yöntemi ile Hesap Ele Geçirme

Elde edilen kullanıcı adı listeleri, uzman tarafından Burp Suite Intruder aracıyla parola kombinasyonları denenerek kaba kuvvet saldırılarına maruz bırakılmıştır. “admin: admin123” gibi varsayılan parolalar hâlen sistemde aktif olduğu için giriş başarılı olmuştur. Bu zafiyet, OWASP A07:2021 kapsamında tekrar değerlendirilir.

Adım 6: Günlük Kayıtlarının Silinmesi – Kısmi Hizmet Engelleme (Log Suppression)

Sızma denemesi uzmanı, denemesin sonunda sistemdeki olay günlüklerini (auth.log, iislogs, vb.) hedefli şekilde silerek, olay sonrası analiz süreçlerinin engellenmesine ilişkin benzetimini yapmıştır. Bu işlem, hizmetin kendisini doğrudan aksatmasa da, güvenlik denetim ve olay müdahale süreçlerini geçici olarak işlevsiz hale getirerek uygulama katmanında kısmi DoS etkisi yaratmaktadır. Bu tarz senaryolar, NIST SP 800-61 Rev.2 tarafından tanımlanan olay yönetimi ilkelerine aykırılık teşkil eder [63].

9.2 Zayıflıkların Uluslararası Ölçünlerle Risk Değerlendirmesi

Bu bölümde, gerçekleştirilen sızma denemesi senaryosunda tespit edilen güvenlik açıkları; uluslararası kabul görmüş güvenlik ölçünleri olan OWASP Top 10:2021 [3], Common Vulnerabilities and Exposures (CVE) [64] kayıtları ve Common Vulnerability Scoring System (CVSS v3.1) [65] esas alınarak değerlendirilmiştir. Her bir zafiyet, etki ve olasılık düzeylerine göre puanlanmış ve bu iki değerin çarpımıyla risk skoru hesaplanmıştır. CVSS temel puanları, CVE verisi olan açıklar için doğrudan NIST Ulusal Zafiyet Veri tabanı (NVD) [66] üzerinden alınmıştır. CVE tanımı bulunmayan açıklarda ise benzer zafiyet türlerinin ortalama risk düzeyleri baz alınarak tahmini değerleme yapılmıştır.

Çizelge 5, tespit edilen her bir zafiyeti; OWASP kategorisi, varsa CVE kodu, CVSS skoru ve etki-olasılık değerleriyle birlikte sunmakta ve zayıflıkların renkli sınıflandırma yöntemiyle risk düzeylerine göre önceliklendirilmesini sağlamaktadır.

9.3 Raporlama Örneği

Aşağıda sunulan rapor bölümü, çalışmada tanımlanan senaryo temelinde oluşturulmuş, teknik bulgularla yönetim düzeyinde karar destek sağlayacak biçimde yapılandırılmıştır.

Teknik Rapor Özeti

Deneme sürecinde belirlenen temel zayıflıklara ilişkin açıklayıcı bilgileri içeren teknik rapor özeti, aşağıdaki Çizelge 6’da sunulmuştur.

Çizelge 6: Sızma denemesi teknik rapor özeti

No	Zafiyet Tanımı	Açıklama	Risk Seviyesi
1	Apache HTTP başlıklarında versiyon	Server yanıt başlıklarında sunucu sürüm bilgisi dışa sızmakta	Orta
2	XSS benzeri komut enjeksiyonu (Follina)	Kullanıcı girdisi kontrol edilmeden işlendi, dış sunucuya veri sızdırma gerçekleşti	Yüksek
3	Oturum yönetiminde güvenlik açığı	HttpOnly veya Secure cookie bayrakları eksik, oturum ele geçirme mümkün	Orta
4	SQL Enjeksiyonu	Arama fonksiyonunda filtrelenmemiş girişle veri tabanına doğrudan erişim sağlandı	Yüksek
5	Zayıf parola kullanımı	Admin kullanıcı için varsayılan parola aktif, kaba kuvvet ile giriş mümkün	Yüksek
6	Log silme ve iz temizleme	Yetkisiz kullanıcı tarafından sistem loglarının silinmesi teknik olarak mümkün	Orta

Çizelge 5: Risk değerlendirmesi

	Zafiyet Tanımı	OWASP Kategorisi	CVE Kodu	CVSS v3.1 Skoru	Etki (1–5)	Olasılık (1–5)	Risk Skoru	Risk Düzeyi
1	HTTP başlıklarında sürüm bilgisi açık	A05:2021 – Misconfiguration	CVE-2014-0226	5.0	2	5	10	Orta
2	Follina (komut enjeksiyonu ile XSS)	A03:2021 – Injection	CVE-2022-30190	7.8	5	4	20	Yüksek
3	Oturum güvenliği eksikliği	A07:2021 – ID/Auth Failures	CVE-2015-2080	6.8	4	3	12	Orta
4	SQL Enjeksiyonu	A01:2021 – Broken Access Ctrl	CVE-2022-24124	7.5	5	5	25	Yüksek
5	Zayıf parola / brute-force açığı	A07:2021 – ID/Auth Failures	CWE-521	~6.5	4	4	16	Yüksek
6	Log silme (adli bilişim engelleme)	A09:2021 – Security Logging and Monitoring Failures	CWE-778	~6.0	3	3	9	Orta
1–5: Düşük , 6–14: Orta , 15–19: Yüksek , 20–25: Kritik								

Önerilen Güvenlik İyileştirmeleri:

- Web sunucusu yapılandırmasında ServerTokens ve ServerSignature kapatılmalı.
- Tüm kullanıcı giriş alanlarında sunucu taraflı girdi doğrulama uygulanmalı.
- Oturum yönetiminde HttpOnly, Secure ve SameSite cookie bayrakları etkinleştirilmeli.
- SQL sorguları parametreli hale getirilerek ORM tabanlı yapı kullanılmalı.
- Parola politikaları güncellenerek kompleksite ve zorunlu değişiklik periyodu tanımlanmalı.
- Loglama mekanizmaları merkezi bir SIEM altyapısına bağlanmalı ve loglar imzalanarak saklanmalı.

Yönetim Özeti

Gerçekleştirilen sızma denemesi sonucunda, hedef sistemin dışa açık bileşenlerinde toplam 6 adet kritik ve orta seviyeli güvenlik zafiyeti tespit edilmiştir. Bu açıklardan ikisi doğrudan yetkisiz veri erişimi ve veri sızıntısı ile sonuçlanabilecek nitelikte olup, özellikle kişisel veri içeren sistemlerde KVKK veya GDPR gibi regülasyonlara aykırılık riski oluşturmaktadır. Oturum güvenliğindeki zayıflık, kullanıcı hesaplarının ele geçirilmesine yol açabilir ve kurumsal kaynaklara sızılmasını mümkün kılar.

Parola yönetimi ve log izleme eksiklikleri, kurumun hem önleyici güvenlik (preventive controls) hem de olay tespit ve müdahale (detection & response) süreçlerinde zayıflık yarattığını göstermektedir. Bu durum, sistemde gerçekleşen saldırıların geç tespit edilmesine veya hiç fark edilmemesine neden olabilir.

Tespit edilen zayıflıkların kuruma potansiyel etkileri şunlardır:

- İtibar Kaybı: Veri sızıntısı sonrası kamuoyunda güven kaybı ve basın etkisi.
- Yasal Riskler: Kişisel veri ihlali durumunda KVKK kapsamında yaptırım ve para cezaları.
- Finansal Kayıplar: Hizmet kesintisi, sistem geri yükleme ve kriz iletişimi maliyetleri.
- İş Sürekliliği Riski: Kritik sistemlerin saldırıya uğraması sonucu işlemsel duraksamalar.

Deneme süreci, uluslararası ölçünlere uygun şekilde OWASP Top 10, CVE/CWE sistemleri ve CVSS v3.1 risk derecelendirmesi çerçevesinde gerçekleştirilmiştir. Bulgular, teknik ekiplerin müdahale önceliklerini belirlemesine ve yönetim kadrosunun kurumsal bilgi güvenliği stratejilerini yeniden yapılandırmasına rehberlik etmektedir.

10 Sonuç

Bu çalışmada, farklı sızma denemesi yöntemleri karşılaştırılarak OWASP, OSSTMM, PTES, ISSAF ve NIST gibi yaygın ölçünlerin siber güvenlik alanındaki rollerine dair kapsamlı bir değerlendirme sunulmuştur. Literatürde genellikle yalnızca belirli bir yönteme odaklanan sınırlı sayıdaki çalışmanın aksine, bu çalışmada farklı ölçünlerin güçlü ve zayıf yönleri sistematik olarak analiz edilmiş ve karşılaştırmalı bir çerçeve oluşturulmuştur.

Çalışmanın önemli katkılarından biri, sızma denemesi yöntemlerinin yaygın saldırı türleriyle eşleştirilerek teknik düzeyde ilişkilendirilmesidir. Buna ek olarak, senaryo tabanlı örnek bir sızma denemesi uygulaması benzetimi yapılarak, gerçekçi bir saldırı zinciri üzerinden güvenlik açıkları belirlenmiş ve her bir açıklık CVSS puanlarına göre risk analizi ile değerlendirilmiştir. Elde edilen bulgular ayrıca teknik ve yönetsel düzeyde yapılandırılmış bir rapor örneği ile desteklenmiştir.

Analizler, OWASP'ın uygulama güvenliği denemelerinde, NIST'in ağ güvenliğinde, OSSTMM'nin kapsamlı güvenlik değerlendirmelerinde ve PTES'in saldırı benzetimlerinde öne çıktığını ortaya koymuştur. ISSAF yöntemi ise kapsamlı yapısına rağmen güncellenmemesi nedeniyle güncel uygulamalarda etkisini kaybetmektedir. Her yöntemin farklı odak alanları bulunsa da özellikle kurumsal düzeyde risk yönetimi ve otomasyon süreçlerinin entegrasyonu açısından yöntemler arası birlikte çalışabilirlik önem kazanmaktadır.

Sonuç olarak bu çalışma, Türkçe literatürde sınırlı olarak ele alınan yöntem karşılaştırmalarına katkı sağlamakta; uygulamalı bir vaka analizi, teknik değerlendirme ve risk temelli raporlama içeriğiyle sızma denemesi süreçlerinin etkin kullanımına yönelik çok boyutlu bir yol haritası sunmaktadır.

Gelecek çalışmaların, sızma denemesi süreçlerine yapay zekâ destekli otomatik denemelerin entegrasyonu ve güvenlik denemelerinin daha sistematik hale getirilmesi üzerine yoğunlaşabileceği düşünülmektedir.

11 Kaynakça

[1] Admass, W. S., Y. Y. Munaye, ve A. A. Diro, *"Cyber security: State of the art, challenges and future directions"*, Cyber Security and Applications, c. 2, Oca. 2024, doi: 10.1016/J.CSA.2023.100031.

[2] Nurse, J. R. C., N. Williams, E. Collins, N. Panteli, J. Blythe, ve B. Koppelman, *"Remote Working Pre- and Post-COVID-19: An Analysis of New Threats and Risks to Security and Privacy"*, Communications in Computer and Information Science, c. 1421, ss. , pp. 583-590, 2021, doi: 10.1007/978-3-030-78645-8_74.

[3] OWASP, "OWASP Top Ten", OWASP Foundation. Erişim: 22 Mart 2024. [Çevrimiçi]. Erişim adresi: https://owasp.org/www-project-top-ten/

[4] Karen, S. ve O. Angela, *"NIST - Technical Guide to Information Security Testing and Assessment Recommendations"*, Nist Special Publication, c. 800, ss. , pp. 1-80, 2008, doi: 10.6028/NIST.SP.800-115.

[5] PTES, "The Penetration Testing Execution Standard", Pentest Standard. Erişim: 05 Şubat 2025. [Çevrimiçi]. Erişim adresi: http://www.pentest-standard.org/index.php/Main_Page

[6] Positive Technologies, "Web Applications vulnerabilities and threats: statistics for 2019", Global Ptsecurity. Erişim: 23 Şubat 2025. [Çevrimiçi]. Erişim adresi: https://global.ptsecurity.com/analytics/web-vulnerabilities-2020

[7] SiteLock, "Cybersecurity Statistics Report 2022", SiteLock. Erişim: 20 Şubat 2025. [Çevrimiçi]. Erişim adresi: https://www.sitelock.com/resources/security-report/

[8] Goel, J. N. ve B. M. Mehtre, *"Vulnerability Assessment & Penetration Testing as a Cyber Defence Technology"*, içinde *Procedia Computer Science*, Elsevier, 2015, ss. , pp. 710-715. doi: 10.1016/j.procs.2015.07.458.

[9] "Penetration Testing Market Size, Share | Growth Report [2032]", Fortune Business Insights. Erişim: 02 Şubat 2025. [Çevrimiçi]. Erişim adresi: https://www.fortunebusinessinsights.com/penetration-testing-market-108434

[10] Sari, A., *"Turkish national cyber-firewall to mitigate countrywide cyber-attacks"*, Computers and Electrical Engineering, c. 73, ss., pp. 128-144, Oca. 2019, doi: 10.1016/j.compeleceng.2018.11.008.

[11] ThinkTech, "ThinkTech - Siber Tehdit Durum Raporu (Ocak - Mart 2024)", Savunma Teknolojileri Mühendislik A.Ş. Erişim: 03 Şubat 2025. [Çevrimiçi]. Erişim adresi: https://thinktech.stm.com.tr/tr/siber-tehdit-durum-raporu-ocak-mart-2024

[12] Neciyev, S. ve B. Pazarbaşi, *"Siber Güvenlik, Siber Savaş Alanında Seçili Anahtar Kelimeler ile İlgili Araştırmaların Bibliyometrik Analizi"*, Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım ve Teknoloji, c. 12, sy 1, ss. , pp. 57-79, Mar. 2024, doi: 10.29109/GUJSC.1378921.

[13] Shah, S. ve B. M. Mehtre, *"An overview of vulnerability assessment and penetration testing techniques"*, J Comput Virol Hack Tech, c. 11, ss. , pp. 27-49, 2015, doi: 10.1007/s11416-014-0231-x.

[14] Bertoglio, D. D. ve A. F. Zorzo, *"Overview and open issues on penetration test"*, Journal of the Brazilian Computer Society, c. 23, sy 1, ss. , pp. 1-16, Ara. 2017, doi: 10.1186/S13173-017-0051-1/FIGURES/6.

[15] Adam, H. M., Widyawan, ve G. D. Putra, *"A Review of Penetration Testing Frameworks, Tools, and Application Areas"*, içinde *Proceedings - 2023 IEEE 7th International Conference on Information Technology, Information Systems and Electrical Engineering, ICITISEE 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, ss. , pp. 319-324. doi: 10.1109/ICITISEE58992.2023.10404397.

[16] Shanley, A., *"Penetration Testing Frameworks and methodologies: A comparison and evaluation"*, Edith Cowan University, 2016. Erişim: 11 Şubat 2025. [Çevrimiçi]. Erişim adresi: https://ro.ecu.edu.au/theses_hons/1553

[17] Sarker, K. U., F. Yunus, ve A. Deraman, *"Penetration Taxonomy: A Systematic Review on the Penetration Process, Framework, Standards, Tools, and Scoring Methods"*, 01 Temmuz 2023, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/su151310471.

[18] Haq, I. U. ve T. A. Khan, *"Penetration Frameworks and Development Issues in Secure Mobile Application Development: A Systematic Literature Review"*, IEEE Access, c. 9, ss. , pp. 87806-87825, 2021, doi: 10.1109/ACCESS.2021.3088229.

[19] Wen, S.-F., A. Shukla, ve · Basel Katt, *"Artificial intelligence for system security assurance: A systematic literature review"*, International Journal of Information Security 2024 24:1, c. 24, sy 1, ss. , pp. 1-42, Ara. 2024, doi: 10.1007/S10207-024-00959-0.

[20] Vural, Y. ve Sağıroğlu,Ş. *"Kurumsal Bilgi Güvenliğinde Güvenlik Denemeleri ve Öneriler"*, J. Fac. Eng. Arch. Gazi Univ., c. 26, sy 1, ss. , pp. 89-103, 2011.

[21] Yılmaz, E. N., Gönen, S., Şanoğlu, S., Karacayılmaz,G. ve Özbirinci, Ö.,*"Endüstri 4.0'ın Gelişim Sürecinde Unutulan Bileşen: Siber Güvenlik"*, Düzce Üniversitesi Bilim ve Teknoloji Dergisi, c. 9, sy 4, ss. , pp. 1142-1158, Tem. 2021, doi: 10.29130/dubited.905340.

[22] Tulgar, M., Halim Zaim, A., Ali Aydın, M.ve Tarihi,G. *"Ulusal Bilgi ve İletişim Güvenliği Rehberi: IoT Güvenliği İçin Bir Uygulama Örneği"*, İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi, c. 21, sy 42, ss. , pp. 353-382, Ara. 2022, doi: 10.55071/TICARETFBD.1141795.

[23] Of, M., *"Siber Güvenlik Üzerine Bir Araştırma: Yazılım Güvenliği"*, Bayburt Üniversitesi Fen Bilimleri Dergisi, c. 2, sy 2, ss. , pp. 254-260, Ara. 2019, Erişim: 06 Şubat 2025. [Çevrimiçi]. Erişim adresi: https://dergipark.org.tr/en/pub/bufbd/issue/50962/653374

[24] Aydoğdu, D. ve M. S. Gündüz, *"Web Uygulama Güvenliği Açıklıkları ve Güvenlik Çözümleri Üzerine Bir Araştırma"*, Uluslararası Bilgi Güvenliği Mühendisliği Dergisi, c. 2, sy 1, ss. , pp. 1-7, Haz. 2016, doi: 10.18640/UBGMD.56836.

[25] Yalçınkaya, M. A. ve E. Küçüksille, *"Web Uygulama Sızma Denemelerinde Kapsam Genişletme İşlemi İçin Yöntem Geliştirilmesi ve Uygulanması"*, Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi, c. 25, sy 1, ss. , pp. 16-27, Nis. 2021, doi: 10.19113/sdufenbed.661867.

[26] Senturk, Z. ve E. Irmak, *"Windows Aktif Dizin Etki Alanı Servisi ve Kurumsal Ağ Güvenliği: PowerShell Erişiminin Analizi ve Önlemler"*, Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım ve Teknoloji, Eyl. 2024, doi: 10.29109/gujsc.1447924.

[27] Kestane, A. ve G. Kurt, *"Bulut güvenlik denetimi: Bulut siber güvenlik uygulamalarında iç denetim"*, Ömer Halisdemir Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, c. 17, sy

3, ss. , pp. 667-690, Tem. 2024, doi: 10.25287/ohuiibf.1482734.

[28] Scarfone, K., M. Souppaya, A. Cody, ve A. Orebaugh, *"Guideline on network security testing - Special Publication 800-115"*, National Institute of Standards, 2008, Erişim: 13 Şubat 2025. [Çevrimiçi]. Erişim adresi: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf

[29] Geer, D. ve J. Harthorne, *"Penetrationt: A duet"*, 18th Annual Computer Security Applications Conference, ACSAC, c. 2002-January, ss. , pp. 185-195, 2002, doi: 10.1109/CSAC.2002.1176290.

[30] Leeuw, K. ve J. Bergstra, *The history of information security : a comprehensive handbook*. Elsevier, 2007. Erişim: 16 Mart 2024. [Çevrimiçi]. Erişim adresi: https://books.google.com.tr/books/about/The_History_of_Information_Security.html?id=pQBrsonDp6cC&redir_esc=y

[31] Tabibian, O. R., *Internet Scanner Finds Security Holes*. PC Magazine, 1996. Erişim: 16 Mart 2024. [Çevrimiçi]. Erişim adresi: https://books.google.com/books?id=LYp7r6OrMdIC&pg=PA536

[32] Raju, R., *"A Literature Survey on System Security and Network Vulnerability Assessment"*, International Journal of Scientific Research in Engineering and Management, c. 08, sy 04, ss. , pp. 1-5, Nis. 2024, doi: 10.55041/IJSREM29695.

[33] Dehghantanha, A., A. Yazdinejad, ve R. M. Parizi, *"Autonomous Cybersecurity: Evolving Challenges, Emerging Opportunities, and Future Research Trajectories"*, AutonomousCyber 2024 - Proceedings of the Workshop on Autonomous Cybersecurity, Co-Located with: CCS 2024, c. 10, sy 24, ss. , pp. 1-10, Kas. 2024, doi: 10.1145/3689933.3690832.

[34] Roesch, M., "Snort - Network Intrusion Detection & Prevention System", Snort. Erişim: 17 Şubat 2025. [Çevrimiçi]. Erişim adresi: https://www.snort.org/

[35] Alkhurayyif, Y. ve Y. Saad Almarshdy, *"Adopting Automated Penetration Testing Tools"*, Journal of Information Security and Cybercrimes Research, c. 7, sy 1, ss. , pp. 51-66, Haz. 2024, doi: 10.26735/rjjt2453.

[36] Khan, M. E. ve F. Khan, *"A Comparative Study of White Box, Black Box and Grey Box Testing Techniques"*, (IJACSA) International Journal of Advanced Computer Science and Applications, c. 3, sy 6, 2012, [Çevrimiçi]. Erişim adresi: www.ijacsa.thesai.org

[37] Reed, K. ve T. Murnane, *"On the Effectiveness of Mutation Analysis as a Black Box Testing Technique"*, içinde *Proceedings of the Australian Software Engineering Conference, ASWEC*, IEEE Computer Society, 2001. doi: 10.1109/ASWEC.2001.948492.

[38] Sarker, K. U., F. Yunus, ve A. Deraman, *"Penetration Taxonomy: A Systematic Review on the Penetration Process, Framework, Standards, Tools, and Scoring Methods"*, Sustainability 2023, Vol. 15, Page 10471, c. 15, sy 13, s. , pp. 10471, Tem. 2023, doi: 10.3390/SU151310471.

[39] Saxena, R. ve M. Singh, *"Gray Box Testing: Proactive Methodology for the Future Design of Test Cases to Reduce Overall System Cost"*, c. 1, sy 8, ss. , pp. 62-66, 2014, Erişim: 04 Mayıs 2025. [Çevrimiçi]. Erişim adresi: http://www.krishisanskriti.org/jbaer.html

[40] OWASP, "OWASP Web Security Testing Guide", OWASP Foundation. Erişim: 12 Şubat 2025. [Çevrimiçi]. Erişim adresi: https://owasp.org/www-project-web-security-testing-guide/

[41] Vieira, M. ve B. Sousa, *"Evaluation Of Static Analysis Tools In Detecting OWASP Top 10 Vulnerabilities"*, Master Thesis, Technology of the University of Coimbra, 2024. Erişim: 12 Şubat 2025. [Çevrimiçi]. Erişim adresi: https://hdl.handle.net/10316/118126

[42] Herzog, P., *"The Open Source Security Testing Methodology Manual (Version 3.0)"*, Institute for Security and Open Methodologies (ISECOM), 2010.

[43] Prandini, M. ve M. Ramilli, *"Towards a practical and effective security testing methodology"*, içinde *IEEE Symposium on Computers and Communications (ISCC)*, 2010, ss. , pp. 320-325. doi: 10.1109/ISCC.2010.5546813.

[44] Abu-Dabaseh, F. ve E. Alshammari, *"Automated Penetration Testing: An Overview"*, Academy and Industry Research Collaboration Center (AIRCC), ss. , pp. 121-129, 2018, doi: 10.5121/csit.2018.80610.

[45] Ceylan, M. F., *"Makine öğrenmesi ile adres çözümleme protokolü sahteciliğinin tespiti"*, Master's thesis, Balıkesir Üniversitesi, Fen Bilimleri Enstitüsü, 2024. Erişim: 22 Şubat 2025. [Çevrimiçi]. Erişim adresi: https://dspace.balikesir.edu.tr/xmlui/handle/20.500.12462/15943

[46] OISSG, "Information Systems Security Assessment Framework (ISSAF) draft 0.2", 2005. [Çevrimiçi]. Erişim adresi: https://untrustednetwork.net/files/issaf0.2.1.pdf

[47] Wack, J., M. Souppaya, ve M. Tracy, *"Guideline on Network Security Testing - Special Publication 800-42"*, National Institute of Standards, Eki. 2003, doi: 10.6028/NIST.SP.800-42.

[48] Chu, G., *"Automation of Penetration Testing"*, Doctoral dissertation, The University of Liverpool, 2021.

[49] Lachkov, P., L. Tawalbeh, ve S. Bhatt, *"Vulnerability Assessment for Applications Security Through Penetration Simulation and Testing"*, Journal of Web Engineering, c. 21, sy 7, ss. , pp. 2187-2208, 2022, doi: 10.13052/JWE1540-9589.2178.

[50] Laxmi Kowta, A. S., K. Bhowmick, J. R. Kaur, ve N. Jeyanthi, *"Analysis and overview of information gathering tools for pentesting"*, 2021 International Conference on Computer Communication and Informatics, ICCCI 2021, c. 2021-January, Oca. 2021, doi: 10.1109/ICCCI50826.2021.9457015.

[51] Zografopoulos, I., J. Ospina, X. Liu, ve C. Konstantinou, *"Cyber-Physical Energy Systems Security: Threat Modeling, Risk Assessment, Resources, Metrics, and Case Studies"*, IEEE Access, c. 9, ss. , pp. 29775-29818, 2021, doi: 10.1109/ACCESS.2021.3058403.

[52] Altulaihan, E. A., A. Alismail, ve M. Frikha, *"A Survey on Web Application Penetration Testing"*, Electronics (Switzerland), c. 12, sy 5, Mar. 2023, doi: 10.3390/ELECTRONICS12051229.

[53] Halfond, W. G. J., J. Viegas, ve A. Orso, *"A Classification of SQL Injection Attacks and Countermeasures"*, ISSSE, 2006.

[54] Lin, H., Z. Yan, Y. Chen, ve L. Zhang, *"A Survey on Network Security-Related Data Collection Technologies"*, IEEE Access, c. 6, ss. , pp. 18345-18365, Mar. 2018, doi: 10.1109/ACCESS.2018.2817921.

[55] Parry, J., D. Hunter, K. Radke, ve C. Fidge, *"A network forensics tool for precise data packet capture and replay in cyber-physical systems"*, ACM International Conference Proceeding Series, c. 01-05-February-2016, Şub. 2016, doi: 10.1145/2843043.2843047;TOPIC:TOPIC:CONFERENCE-COLLECTIONS>AUS-CSW;JOURNAL:JOURNAL:ACMOTHERCONFERENCES;CTYPE:STRING:BOOK.

[56] Wang, L. ve A. M. Wyglinski, *"Detection of man-in-The-middle attacks using physical layer wireless security techniques"*,

Wireless Communications and Mobile Computing, c. 16, sy 4, ss, pp. 408-426, Mar. 2016, doi: 10.1002/WCM.2527;CTYPE:STRING:JOURNAL.

[57] Ylli, E. ve J. Fejzaj, *"Man in the Middle: Attack and Protection"*, içinde *RTA-CSIT*, 2021, ss. , pp. 198-204.

[58] Jaafar, G. A., S. M. Abdullah, ve S. Ismail, *"Review of Recent Detection Methods for HTTP DDoS Attack"*, 2019, *Hindawi Limited*. doi: 10.1155/2019/1283472.

[59] Bosnjak, L., J. Sres, ve B. Brumen, *"Brute-force and dictionary attack on hashed real-world passwords"*, içinde *41st International Convention on Information and Communication Technology*, Institute of Electrical and Electronics Engineers Inc., Haz. 2018, ss. , pp. 1161-1166. doi: 10.23919/MIPRO.2018.8400211.

[60] Islam, S., *"Security Auditing Tools: A Comparative Study"*, International Journal of Computing Sciences Research, c. 5, sy 1, ss., pp. 407-425, Oca. 2021, doi: 10.25147/ijcsr.2017.001.1.49.

[61] Sarmah, U., D. K. Bhattacharyya, ve J. K. Kalita, *"A survey of detection methods for XSS attacks"*, Journal of Network and Computer Applications, c. 118, ss. , pp. 113-143, Eyl. 2018, doi: 10.1016/j.jnca.2018.06.004.

[62] Rodríguez, G. E., J. G. Torres, P. Flores, ve D. E. Benavides, *"Cross-site scripting (XSS) attacks and mitigation: A survey"*, Computer Networks, c. 166, s. , pp. 106960, Oca. 2020, doi: 10.1016/J.COMNET.2019.106960.

[63] Cichonski, P., T. Millar, T. Grance, ve K. Scarfone, *NIST - Computer Security Incident Handling Guide*. 2025. doi: 10.6028/NIST.SP.800-61r3.

[64] "CVE - Common Vulnerabilities and Exposures". Erişim: 02 Mayıs 2025. [Çevrimiçi]. Erişim adresi: https://cve.mitre.org/

[65] Maris, A., A. Kundu, ve A. Yoon, *Common Vulnerability Scoring System version 3.1 Specification Document Revision 1*. 2024. Erişim: 02 Mayıs 2025. [Çevrimiçi]. Erişim adresi: https://www.first.org/cvss/

[66] National Institute of Standards and Technology (NIST), "National Vulnerability Database (NVD)". Erişim: 02 Mayıs 2025. [Çevrimiçi]. Erişim adresi: https://nvd.nist.gov/