



YAVUZLAR BİLET SATIN ALMA UYGULAMASI

PENTEST RAPORU

İsim: Mustafa Talha DOĞAN

Takım: Yavuzlar 16

İncelediği Proje: <https://github.com/taro544/bilet-satin-alma>

NOT:

Pentest sürecinde uygulama analiz edilirken, uygulamanın temel Satın Al işlevinin kritik bir veritabanı hatası nedeniyle çalışmadığı tespit edilmiştir.

Tespit Edilen Hata: Satın alma sırasında hata oluştı: SQLSTATE[HY000]: General error: 1 table tickets has no column named company_name

Hatanın Kök Nedeni: Analiz sonucunda, cart.php dosyasındaki checkout kod bloğunun tickets tablosuna yeni bir kayıt eklerken, veritabanı şemasında bulunmayan birden fazla sütuna veri yazmaya çalıştığı görülmüştür. Bu sütunlar: company_name, departure_city, arrival_city, departure_time ve arrival_time'dır. Bu durum, uygulama kodu ile veritabanı şeması arasında bir uyumsuzluk olduğunu göstermektedir.

Testin Devamlılığı İçin Uygulanan Eylem: Raporun ilerleyen bölümlerinde belgelenen ödeme akışı zafiyetlerini ortaya çıkarabilmek ve uygulamanın tam bir güvenlik analizini tamamlayabilmek amacıyla, hedef sistemin veritabanı şemasına tarafimca müdahale edilmiştir. tickets tablosuna aşağıdaki SQL komutları uygulanarak eksik sütunlar eklenmiştir:

SQL Komutları:

```
ALTER TABLE tickets ADD COLUMN company_name TEXT;
```

```
ALTER TABLE tickets ADD COLUMN departure_city TEXT;
```

```
ALTER TABLE tickets ADD COLUMN arrival_city TEXT;
```

```
ALTER TABLE tickets ADD COLUMN departure_time DATETIME;
```

```
ALTER TABLE tickets ADD COLUMN arrival_time DATETIME;
```

UYARI:

Bu belge, Yavuzlar Web Güvenliği Eğitim Programı kapsamında, tamamen eğitim ve öğrenim amacıyla hazırlanmış bir penetrasyon testi sonuç raporudur. İçerik, testin yapıldığı anı yansıtmakta olup, hatalar veya eksik değerlendirmeler içerebilir.

Bu rapor, Mustafa Talha DOĞAN (Yavuzlar 16) tarafından, <https://github.com/taro544/bilet-satin-alma> adresinde kaynak kodları bulunan Bilet Satın Alma Platformu projesine karşı, görev tanımında belirtilen kurallar çerçevesinde yazılmıştır. Burada yapılanların herhangi bir yasal yükümlülüğü bulunmamaktadır.

İçindekiler

Yönetici Özeti	4
Raporlama ve Metodoloji.....	5
KAPSAM	5
TEST METODOLOJİSİ	5
RİSK DEĞERLENDİRMESİ.....	5
ZAFİYET BULGULARI VE DETAYLARI.....	7
BULGU-01	7
Açık erişilebilir SQLite veritabanı dosyası-database.sqlite.....	7
BULGU-02	9
Kupon Doğrulama Eksikliğinden Kaynaklanan İş Mantığı Zafiyeti	9
BULGU-03	12
Siteler Arası İstek Sahteciliği (CSRF)	12
BULGU-04	16
Zayıf Kimlik Doğrulama.....	16

Yönetici Özeti

Bu rapor, Yavuzlar Web Güvenliği Eğitim Programı kapsamında, github.com/taro544/bilet-satin-alma adresinde kaynak kodları bulunan Bilet Satın Alma Platformu projesine yönelik 26.10.2025-28.10.2025 tarihleri arasında gerçekleştirilen beyaz kutu (white-box) penetrasyon testi bulgularını özetlemektedir.

Testler sonucunda, uygulamanın güvenlik olgunluğunun düşük seviyede olduğu ve canlı bir sistemde kullanılması durumunda kritik riskler barındırdığı tespit edilmiştir. Uygulamanın, modern web uygulamaları için temel kabul edilen çok sayıda güvenlik kontrolünden yoksun olduğu görülmüştür.

Tespit edilen zayıflıklar, uygulamanın veri gizliliğini, bütünlüğünü ve finansal güvenilirliğini doğrudan tehdit etmektedir. Başlıca kritik bulgular şunlardır:

- **Kritik Veri Sızıntısı (Bulgu-01):** Uygulamanın tüm veritabanı (`database.sqlite`), internet üzerinden herhangi bir kimlik doğrulaması olmaksızın doğrudan indirilebilir durumdadır. Bu durum, tüm kullanıcıların ve yöneticilerin kişisel bilgilerinin ve parola karmalarının (hash) çalınmasına yol açmaktadır.
- **Finansal Kayıp Riski (Bulgu-02):** Fiyatlandırma mekanizmasında, saldırganların bir firmaya ait indirim kuponunu başka bir firma için kullanmasına olanak tanıyan kritik bir iş mantığı hatası bulunmaktadır. Bu, doğrudan gelir kaybına neden olabilir.
- **Hesap Ele Geçirme ve Veri Manipülasyonu (Bulgu-03 & 04):** Uygulama, Siteler Arası İstek Sahteciliği (CSRF) ve Kaba Kuvvet (Brute Force) saldırılarına karşı savunmasızdır. Bu zayıflıklar birleştirildiğinde, kötü niyetli kişilerin:
 - Kullanıcı ve firma hesaplarını ele geçirmesine,
 - Firma sahiplerinin haberi olmadan seferlerini silmesine, fiyatlarını manipüle etmesine veya sahte seferler eklemesine,
 - Kullanıcıların haberi olmadan adlarına bilet satın almasına olanak tanımaktadır.

Uygulamanın temel işlevlerini doğrudan etkileyen bu yüksek ve kritik seviyeli zayıflıklar, sistemin güvenliğini sağlamak amacıyla öncelikli olarak giderilmesi şiddetle tavsiye edilmektedir.

Raporlama ve Metodoloji

KAPSAM

Test, <http://localhost:8080> üzerinde çalışan uygulamanın tamamını kapsamaktadır. DoS/DDoS saldırıları, sosyal mühendislik ve fiziksel güvenlik testleri kapsam dışıdır.

TEST METODOLOJİSİ

Bu sizma testi, <https://github.com/taro544/bilet-satin-alma> adresindeki uygulamanın kaynak kodlarına erişim sağlanarak gerçekleştirilmiştir.

RİSK DEĞERLENDİRMESİ

CVSS Nedir ve Nasıl Çalışır

Bu rapordaki bulguların teknik ciddiyetini ölçmek için, endüstri standarı olan CVSS (Common Vulnerability Scoring System) esas alınmıştır. CVSS, her bir zayıflığın şiddetini 0.0 ile 10.0 arasında sayısal bir puanla ifade eden açık kaynaklı bir sistemdir. Bu puanlama, bir zayıflığın ne kadar kolay istismar edilebileceğini ve başarılı bir saldırının etkisini objektif bir şekilde ölçer.

Bu değerlendirmede CVSS v3.1 standarı kullanılmıştır. Hesaplamalar, [Chandanbn CVSS Calculator](#) aracı üzerinden yapılmıştır. CVSS v3.1, her zayıflığı sekiz temel metrik (Attack Vector, Attack Complexity, Privileges Required, User Interaction, Scope, Confidentiality, Integrity, Availability) üzerinden değerlendirir ve bu metriklerin kombinasyonu sonucunda Base Score üretilir.

Balbix CVSS rehberine göre, bu puan yalnızca bir zayıflığın şiddet derecesini temsil eder; organizasyonel risk düzeyi ise sistemin önemi, varlık değeri, tehdit ortamı ve mevcut güvenlik önlemleri gibi çevresel faktörlere göre değişebilir. Dolayısıyla CVSS, teknik ciddiyet ölçümünde referans olarak alınmış; ancak nihai risk değerlendirmesi, sistem bağlamında yorumlanmıştır.

Referanslar

<https://www.balbix.com/insights/understanding-cvss-scores/#:~:text=CVSS%20scoring%20assigns%20a%20number,characteristics%20may%20change%20over%20time.>
<https://chandanbn.github.io/cvss/>

CVSS Puan Aralıkları ve Risk Seviyeleri

Seviye	CVSS Puan Aralığı	Açıklama ve Proje Özelinde Örnekler
KRİTİK (Critical)	9.0 – 10.0	Sunucu Yanıyor Uzaktan, herhangi bir ayrıcalık veya kullanıcı etkileşimi gerektirmeden istismar edilebilen zayıflıklardır. Sistemin tamamının ele geçirilmesine (RCE), veritabanı sızmasına veya geri dönenlemez veri kaybına yol açabilir.
YÜKSEK (High)	7.0 - 8.9	Kilitli Kapıları Açılmak Genellikle bir kullanıcının oturumunu ele geçirmeye veya başka bir kullanıcının/firmanın verilerini görmeye veya değiştirmeye (IDOR) olanak tanıyan zayıflıklardır.
ORTA	4.0 - 6.9	Kapıyı Aralıklı Bırakmak Saldırganın işini kolaylaştıran, sisteme veya kullanıcılarla dair bilgi sızdırılan veya en iyi güvenlik pratiklerine (best practices) uyulmamasından kaynaklanan zayıflıklardır. Genellikle tek başına yıkıcı değildir ancak başka bir zayıfle birleştirilerek (chained) kullanılır.
DÜŞÜK	0.1 - 3.9	Kozmetik Güvenlik Etkisi düşük, istismarı çok zor olan veya uygulanmanın doğrudan güvenliğini (gizlilik, bütünlük, erişilebilirlik) etkilememeyen yapılandırma eksiklikleri ve en iyi pratiklerin (best practices) ihlalleri.
NONE	0.0	Aktif bir güvenlik riski oluşturmaz; yalnızca bilgilendirme amaçlıdır.

ZAFİYET BULGULARI VE DETAYLARI

BULGU-01

Bulgu Adı		
Açık erişilebilir SQLite veritabanı dosyası-database.sqlite		
Bulgu Kodu		
DBS_DISCLOSURE_01		
Önem Derecesi	CVSS	Vektor String
High	7.5 -- High	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N
Erişim Noktası	Kullanıcı Profili	
HTTP	Ziyaretçi (anonim)	

Bulgunun Tespit Edildiği Bileşen/Bileşenler
http://localhost:8080/db/database.sqlite
web root içinde doğrudan erişilebilir SQLite dosyası

Zafiyetin Etkisi
Confidentiality (C): High — Veritabanı dosyasının indirilmesiyle kullanıcı adları, e-posta adresleri, parola hash'leri, sipariş ve işlem kayıtları gibi hassas bilgiler açığa çıkabilir.
Olası kötüye kullanım senaryoları:
<ul style="list-style-type: none">Parola hash'lerinin elde edilmesi halinde offline brute-force/wordlist saldırılarıyla hesaplar ele geçirilebilir; zayıf hash/salt varsa risk artar.Elde edilen kullanıcı adları ile credential stuffing veya brute-force saldırıları yapılabilir. Bunun sonucu yetkisiz işlemler ve hesap ele geçirme olabilir.
Integrity / Availability:
Bu tespit, dosyanın okunmasına izin verir; doğrudan veri değişikliği veya hizmet kesintisi yaratmaz. Ancak yukarıdaki ikincil senaryolar (ele geçirilmiş hesaplar) dolaylı integrity/availability etkilerine sebep olabilir.

Zaafiyetin Açıklaması
Uygulama dizininde bulunan db/database.sqlite dosyasına web sunucusu üzerinden kimlik doğrulama olmadan erişilebilmektedir. Herhangi bir anonim kullanıcı tarayıcı veya curl/wget ile dosyayı indirip içeriğini inceleyebilir; bu da tüm veritabanı içeriğinin açığa çıkmasına neden olur.

Tetiklenme / Reproduce Adımları

Tarayıcıda `http://localhost:8080/db/database.sqlite` adresine git. Dosya doğrudan indirilecektir. Veya terminalden `curl -o database.sqlite http://<hedefhost>:8080/db/database.sqlite` komutunu çalıştırırsan dosya indirilecektir.

```
[kali㉿kali)-[~] $ curl -I http://192.168.84.1:8080/db/database.sqlite
HTTP/1.1 200 OK
Host: 192.168.84.1:8080
Date: Tue, 28 Oct 2025 06:22:57 GMT
Connection: close
Content-Length: 65536

[kali㉿kali)-[~] $ curl -O http://192.168.84.1:8080/db/database.sqlite
% Total    % Received % Xferd  Average Speed   Time     Time      Time  Current
          Dload  Upload   Total   Spent    Left  Speed
100 65536  100 65536     0      0  9256k      0 --:--:-- --:--:-- --:--:-- 10.4M

[kali㉿kali)-[~] $ sqlite3 database.sqlite ".tables"
cart           coupons        orders         trips        users
companies      notifications    tickets    user_coupons
```

Çözüm Önerileri

Dosyaları Web Kök Dizininden Taşıma: Veri tabanı gibi hassas dosyalar asla web sunucusunun doğrudan erişebileceği bir dizinde barındırılmamalıdır. Bu dosyalar, web kök dizininin bir üst seviyesine proje köküne taşınmalıdır. PHP betikleri bu dosyalara include '`../config.php`' gibi dosya sistemi yollarıyla erişmeye devam edebilir, ancak dışarıdan bir kullanıcı URL ile erişemez.

Sunucu Yapılandırması ile Erişimi Engelleme: Eğer dosya yapısını değiştirmek mimari nedenlerle mümkün değilse, web sunucusu (örn: Apache) bu dosyalara ve dizinlere yapılan tüm web isteklerini engelleyecek şekilde yapılandırılmalıdır. Bu, projenin ana dizinine eklenecek bir `.htaccess` dosyası ile sağlanabilir

Referanslar

<https://cwe.mitre.org/data/definitions/219.html>

<https://www.php.net/manual/en/security.php>

BULGU-02

Bulgu Adı		
Kupon Doğrulama Eksikliğinden Kaynaklanan İş Mantığı Zafiyeti		
Bulgu Kodu		
LOGIC_VOUCHER_01		
Önem Derecesi	CVSS	Vektor String
Medium	4.3--Medium	CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:L/A:N
Erişim Noktası	Kullanıcı Profili	
HTTP	User (Kullanıcı)	

Bulgunun Tespit Edildiği Bileşen/Bileşenler
http://localhost:5005/cart.php

Zafiyetin Etkisi
Integrity (I): High — Zafiyet, uygulamanın temel iş mantığını fiyatlandırmayı manipüle eder. Saldırgan, normalde yetkisi olmayan indirimleri sepetine uygulayarak siparişin son fiyatını yasa dışı bir şekilde değiştirebilir.
Olası kötüye kullanım senaryoları:
<ul style="list-style-type: none">Sadece belirli bir firmaya (A Firması) tanımlanmış yüksek değerli bir indirim kuponu, bu zafiyet kullanılarak sistemdeki diğer tüm firmalardan (B, C, D Firmaları) bilet almak için kullanılabilir.

Zaafiyetin Açıklaması
Uygulamanın sepet yönetim mantığı, bir kupon uygulandıktan sonra sepet içeriği değiştiğinde, uygulanan kuponun yeni sepet içeriğiyle yeniden kontrol edilmesi gerçekleştirmemektedir.
Kuponun geçerliliği, sepete eklendiği <i>an</i> bir kez kontrol edilmekte, ancak sepet güncellendiğinde bu onaylanmış indirim durumu sepetten kaldırılmamaktadır. Bu durum, bir firmaya (A Firması) ait kuponun, o firma sepetten çıkarıldıkten sonra bile B Firması'nın biletine uygulanmasına olanak tanır. Bu, CWE-840: Business Logic Vulnerability (İş Mantığı Zafiyeti) olarak sınıflandırılır.

Tetiklenme / Reproduce Adımları

Oturum açmış bir kullanıcı olarak, A Firması'na ait bir biletin sepete ekle.

Sepet sayfasında (cart.php), sadece A Firması'na özel olan bir kupon kodunu gir ve uygula.

İndirimin sepete başarıyla uygulandığı gözlemlen.

Şimdi, A Firması'na ait biletin sepetten kaldırır.

B Firması'na ait bir biletin sepete ekle.

Sepet özeti kontrol et. Bir önceki kupondan kaynaklanan indirim hala aktif olduğu ve B Firması'nın bilet fiyatına uygulandığı görülür.

Sepetim

Metro Turizm
Konya → Ankara
30.10.2025 15:20 → 19:20
Bilet Fiyatı: 200.00 ₺
Mevcut Koltuk: 40
Seçilen Koltuklar: 5
Bilet Sayısı: Toplam: 200.00 ₺
Koltuk seçimi biletlerde miktar değiştirilemez. Koltuk değiştirmek için ürünü kaldırıp tekrar seçin.
Kaldır

Sepeti Güncelle

İndirim Kuponu
 Uygula

Sepet Özeti
Ara Toplam: 200.00 ₺
Toplam Tutar: 200.00 ₺
Bakiyeniz: 200.00 ₺
Satin Al

Sepetim

Kupon kodu uygulandı! %50 indirim (Firma kupon).

Metro Turizm
Konya → Ankara
30.10.2025 15:20 → 19:20
Bilet Fiyatı: 200.00 ₺
Mevcut Koltuk: 40
Seçilen Koltuklar: 5
Bilet Sayısı: Toplam: 200.00 ₺
Koltuk seçimi biletlerde miktar değiştirilemez. Koltuk değiştirmek için ürünü kaldırıp tekrar seçin.
Kaldır

Sepeti Güncelle

İndirim Kuponu
Kupon uygulandı: **METRO1** (%50 indirim)
Kaldır

Sepet Özeti
Ara Toplam: 200.00 ₺
İndirim (%50) - 100.00 ₺
Toplam Tutar: **100.00 ₺**
Bakiyeniz: 200.00 ₺
Satin Al

The screenshot shows a travel booking website interface. At the top, there's a header with a logo and the word "Sepetim". Below it is a "Varan Turizm" section containing flight details: Istanbul → Ankara, Date: 30.10.2025 23:07 → 09:06, Ticket Price: 150.00 ₺, Current Seats: 40, Selected Seats: 38. It also shows a ticket count of 1 and a total amount of 150.00 ₺. A note says: "Koltuk seçimi biletlerde miktar değiştirilemez. Koltuk değiştirmek için ürünü kaldırıp tekrar seçin." Below this is a "Kaldır" (Delete) button. A blue "Sepeti Güncelle" (Update Cart) button is at the bottom of this section. Below the cart section is a "İndirim Kuponu" (Discount Coupon) section with a green bar indicating a coupon is applied: "Kupon uygulandı: METRO1 (%50 indirim)". There's a "Kaldır" button next to it. The bottom part of the screenshot shows a "Sepet Özeti" (Cart Summary) box. It lists the total amount as 150.00 ₺, the discount as %50 (-75.00 ₺), the final total as 75.00 ₺, and the balance as 200.00 ₺. A large green "Satın Al" (Buy Now) button is at the bottom.

Çözüm Önerileri

cart.php (veya sepeti yöneten herhangi bir sunucu tarafı betik), sepete her ürün eklendiğinde VEYA çıkarıldığında çalışacak bir fonksiyon içermelidir.

Bu fonksiyon, sepette halihazırda uygulanmış bir kupon (`$_SESSION['coupon']` gibi) varsa, bu kuponun mevcut sepet içeriğiyle (tutar, firma, sefer ID'si vb.) olan uygunluğunu veritabanından yeniden doğrulamalıdır.

Eğer kupon artık sepetle uyumlu değilse, kupon otomatik olarak sepetten (session'dan) kaldırılmalı ve kullanıcıya Sepet içeriğiniz değiştiği için X kuponu kaldırıldı. gibi bir mesaj gösterilmelidir.

Referanslar

<https://cwe.mitre.org/data/definitions/840.html>

BULGU-03

Bulgu Adı		
Siteler Arası İstek Sahteciliği (CSRF)		
Bulgu Kodu		
CSRF_ALL_FORMS_01		
Önem Derecesi	CVSS	Vektor String
High	8.1-- High	CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:H/A:H
Erişim Noktası	Kullanıcı Profili	
HTTP	Oturumu açık user— Oturumu açık company/ Oturumu açık admin	

Bulgunun Tespit Edildiği Bileşen/Bileşenler
http://localhost:5005/cart.php
http://localhost:8080/cancel_ticket.php
http://localhost:8080/trip.php
http://localhost:8080/edit_trip.php
http://localhost:8080/new_trip.php
http://localhost:8080/delete_trip.php
http://localhost:8080/admin/manage_companies.php
http://localhost:8080/admin/manage_coupons.php
http://localhost:8080/admin/add_company.php
http://localhost:8080/admin/manage_company_admins.php
http://localhost:8080/admin/manage_users.php
http://localhost:8080/admin/edit_company.php

Zafiyetin Etkisi

Integrity (I): High — Saldırgan, kurbanın yetkileri dahilinde uygulamadaki hemen hemen her veriyi (sefer bilgileri, firma detayları, kuponlar, kullanıcı hesapları) kurbanın haberi olmadan değiştirebilir veya silebilir. Fiyatları manipüle edebilir, sahte içerik (firma, admin, kupon) ekleyebilir.

Availability (A): High — Saldırgan, kritik verilerin (tüm firmalar, tüm seferler) silinmesini tetikleyerek uygulamanın veya belirli bölümlerinin kullanılamaz hale gelmesine neden olabilir. Örneğin, tüm firmaların silinmesi, platformun işlevsiz kalmasına yol açar.

Olası kötüye kullanım senaryoları:

- Oturumu açık bir admin kaldırılarak sistemdeki tüm firmaların silinmesi.
- Oturumu açık bir admin kaldırılarak meşru bir firmaya saldırının kontrolünde yeni bir company admini eklenmesi (Yetki Yükseltme).
- Oturumu açık bir company admini kaldırılarak tüm seferlerinin silinmesi veya fiyatlarının "1 TL" olarak güncellenmesi.
- Oturumu açık bir company admini kaldırılarak firmaya ait tüm indirim kuponlarının silinmesi veya "%100 indirimli" sahte kuponlar eklenmesi.
- Oturumu açık bir user kaldırılarak sepetindeki ürünlerin haberi olmadan satın alınması veya tüm biletlerinin iptal edilerek para iadesi alındırılması.

Zaafiyetin Açıklaması

Uygulama genelinde, kullanıcının oturumunda durum değişikliğine neden olan (veri ekleme, silme, güncelleme vb.) POST isteklerini işleyen formlar, bu isteklerin gerçekten kullanıcının o anki oturumu sırasında bilinçli olarak mı yapıldığını yoksa harici bir kaynaktan mı tetiklendiğini doğrulamak için kullanılan Anti-CSRF token mekanizmasını içermemektedir. Sunucu tarafı kodları, bir isteğin meşruluğunu sadece istekle birlikte gelen oturum cerezinin (örn: PHPSESSID) geçerliliğine bakarak kontrol etmektedir. Bu durum, bir saldırının, oturumu açık bir kurbanı kendi hazırladığı zararlı bir web sayfasına yönlendirerek, kurbanın tarayıcısının arka planda bu savunmasız formlara (kurbanın oturum cereziyle birlikte) otomatik olarak istek göndermesini sağlamasına olanak tanır. Sunucu, istekle birlikte geçerli oturum cerezini gördüğü için, isteğin kaynağını sorgulamadan işlemi kurban adına gerçekleştirir.

Tetiklenme / Reproduce Adımları

User hesabıyla <http://localhost:8080> adresinde oturum açılır.

Sepete bir bilet eklenir ancak satın alma işlemi tamamlanmaz.

Saldırgan kötü niyetli bir içerik üretir.

Saldırgan, kurbanı herhangi bir yöntemle (e-posta, sosyal medya mesajı, başka bir web sitesindeki link/resim/reklam) kötü niyetli bir içeriğe tıklamaya veya o içeriğin yüklenmesini sağlamaya ikna eder. Aşağıdaki örnekteki gibi.

Özel Teklif!

Bedava tatil kazanmak için tıkla!

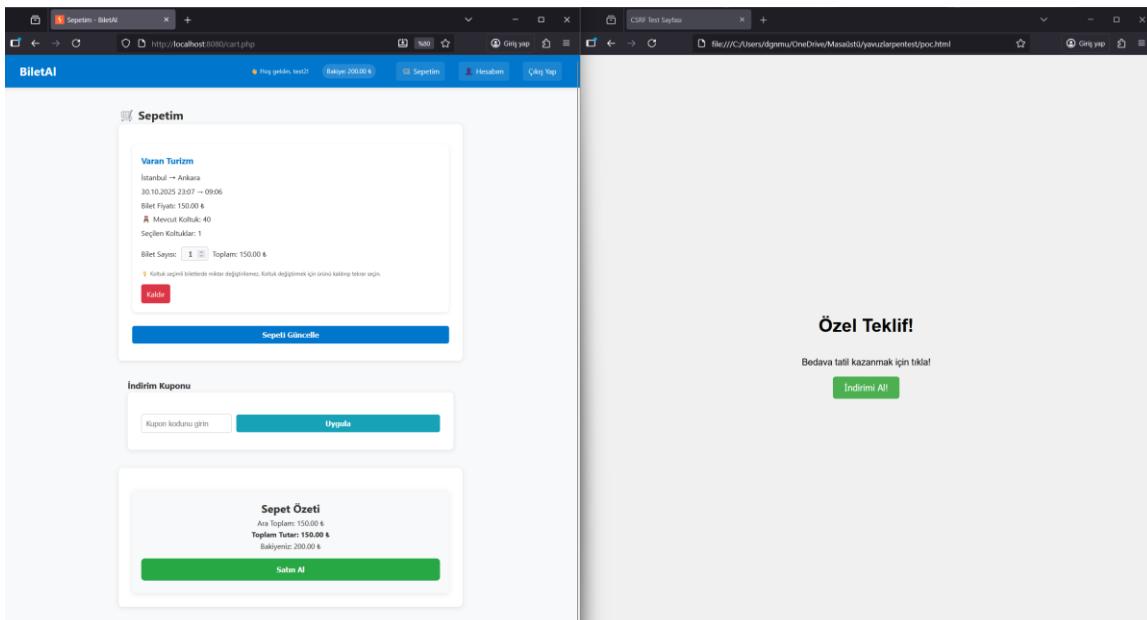
İndirimi Al!

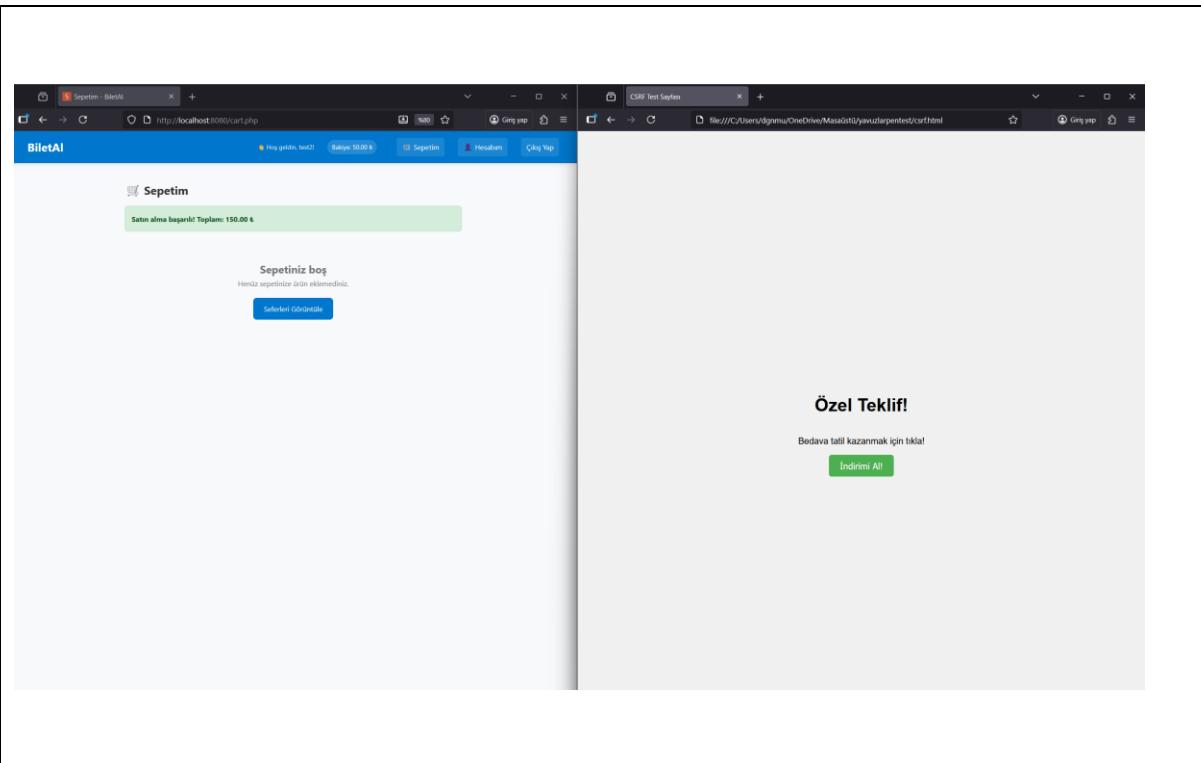
kaynak kodu temel olarak şu kısım önemli:

```
<form name="csrfForm" action="http://localhost  
:8080/cart.php" method="POST" target=  
"hiddenFrame">  
    <input type="hidden" name="checkout" value="" />  
</form>
```

Eğer kullanıcı butona tıklarsa gizli form otomatik olarak kurbanın tarayıcısından <http://localhost:8080/cart.php> adresine checkout= parametresiyle gönderilir.

Kurban, cart.php sayfasına geri dönüp yenilediğinde "Satın alma başarılı!" mesajını ve sepetinin boşaldığını görür. account.php sayfasında ise bakiyesinin düştüğü ve biletin hesabına eklendiği doğrulanır. Kurban, sadece "Tebrikler" diyen bir sayfaya tıklayarak haberi olmadan para harcamış olur.





Çözüm Önerileri

Tüm state-changing (durum değiştiren - Ekleme, Silme, Güncelleme, Satın Alma) formlara Anti-CSRF Token (Synchronizer Token Pattern) koruması eklenmelidir.

Token Oluşturma

Kullanıcı oturum açtığında, config.php veya merkezi bir betikte, kriptografik olarak güvenli, benzersiz ve tahmin edilemez bir token üretilmelidir. Bu token, \$_SESSION['csrf_token'] içinde saklanmalıdır.

Örnek

```
if (empty($_SESSION['csrf_token'])) { $_SESSION['csrf_token'] = bin2hex(random_bytes(32)); }
```

Token'ı Forma Ekleme:

Değişiklik yapan tüm HTML formlarına bu token gizli bir input alanı olarak eklenmelidir.

Örnek

```
<form action="delete_trip.php" method="POST"><input type="hidden" name="csrf_token" value="<?php echo htmlspecialchars($_SESSION['csrf_token']); ?>"><button type="submit">Seferi Sil</button></form>
```

Token'ı Doğrulama

Sunucu tarafında POST isteği işlenmeden *hemen önce*, formdan gelen token ile session'da saklanan token'in eşleşip eşleşmediği kontrol edilmelidir.

Referanslar

<https://cwe.mitre.org/data/definitions/352.html>

https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html

BULGU-04

Bulgu Adı		
Zayıf Kimlik Doğrulama		
Bulgu Kodu		
AUTHN_BRUTEFORCE_01		
Önem Derecesi	CVSS	Vektor String
Critical	9.8-- Critical	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
Erişim Noktası	Kullanıcı Profili	
HTTP	Ziyaretçi(Anonim)	

Bulgunun Tespit Edildiği Bileşen/Bileşenler
http://localhost:8080/login.php

Zafiyetin Etkisi
Bu zafiyet, uygulamanın kimlik doğrulama mekanizmasındaki temel bir eksiklikten kaynaklanır ve sistemin tamamen ele geçirilmesine yol açabilir.
Confidentiality (H), Integrity (H), Availability (H): Zafiyet, saldırganın admin@example.com gibi bilinen bir yönetici veya firma hesabını hedef alarak parola kırma denemesi yapmasına olanak tanır. Başarılı bir saldırı sonucunda:
<ul style="list-style-type: none">• Sistem Ele Geçirilir: Saldırgan, yönetici yetkileriyle oturum açar.• Tüm Veriler Sızdırılır: Tüm kullanıcıların kişisel bilgileri, bilet geçmişleri ve firma bilgileri çalınabilir.• Tüm Veriler Manipüle Edilebilir: Sefer fiyatları değiştirilebilir, biletler silinebilir, sahte yönetici hesapları oluşturulabilir.• Sistem Durdurulabilir: Tüm verilerin (kullanıcılar, seferler) silinmesi yoluyla uygulama tüm kullanıcılar için erişilemez hale getirilebilir.

Zaafiyetin Açıklaması

Login.php betiği, bir kullanıcının (veya bir IP adresinin) belirli bir zaman diliminde kaç kez hatalı parola denemesi yaptığıni kısıtlayan bir mekanizmaya veya hesap kilitleme politikasına sahip değildir.

Bir saldırıcı, admin@example.com gibi bilinen bir hesaba karşı, yaygın şifre listeleri kullanarak saniyede binlerce otomatik parola denemesi (Brute Force Saldırısı) gerçekleştirebilir. PHP'nin password_verify fonksiyonu güvenli bir karma (hash) karşılaşması yapsa da, sunucu tarafında bir deneme kısıtlaması olmadığından bu tür otomatik saldırılara karşı tek başına bir koruma sağlamaz.

Tetiklenme / Reproduce Adımları

Bir saldırıcı, hedef olarak admin@example.com gibi geçerli bir kullanıcı adı belirler.

Kalideki ffuf komutu ile hazır bir seclist olan rockyou.txt kullanılarak brüte force yapılır.

Komut

```
ffuf -u http://192.168.84.1:8080/login.php -w /usr/share/wordlists/rockyou.txt -X POST -d "email=admin%40admin.com&password=FUZZ" -H "Content-Type: application/x-www-form-urlencoded" -fs 721
```

```
(kali㉿kali)-[~]
$ ffuf -u http://192.168.84.1:8080/login.php -w /usr/share/wordlists/rockyou.txt -X POST -d "email=admin%40admin.com&password=FUZZ" -H "Content-Type: application/x-www-form-urlencoded" -fs 721

[!] Starting attack [POST] [http://192.168.84.1:8080/login.php] [v2.1.0-dev]
[!] Threads: 40, Timeout: 10s, Threads per host: 1, Filter: Response size: 721, Follow redirects: false, Calibration: false
[!] Headers: Content-Type: application/x-www-form-urlencoded
[!] Data: email=admin%40admin.com&password=FUZZ
[!] Wordlist: /usr/share/wordlists/rockyou.txt
[!] Method: POST
[!] URL: http://192.168.84.1:8080/login.php
[!] Fuzz: FUZZ
[!] Header: Content-Type: application/x-www-form-urlencoded
[!] Data: email=admin%40admin.com&password=FUZZ
[!] Follow redirects: false
[!] Calibration: false
[!] Timeout: 10
[!] Threads: 40
[!] Matcher: Response status: 200-299,301,302,307,401,403,405,500
[!] Filter: Response size: 721

password [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 156ms]
```

Sunucudan hiçbir kısıtlama gelmediği için saldırı, doğru parola bulunana kadar kesintisiz devam eder.

Eğer doğru parola bulunursa ffuf toolu ekrana şifreyi basar.

Çözüm Önerileri

Uygulamaya kademeli bir savunma mekanizması eklenmelidir.

Hesap Kilitleme (Account Lockout)

Veritabanındaki users tablosuna failed_login_attempts (INT) ve last_failed_login (TIMESTAMP) gibi iki yeni sütun eklenmelidir.

- Kullanıcı şifreyi yanlış girdiğinde failed_login_attempts sayacı 1 artırılır.
- Sayac 5'e (veya 10'a) ulaştığında, hesap kilitlenmeli ve kullanıcıya parola sıfırlama e-postası gönderilmelidir. Bu, hedefli saldırıları tamamen durdurur.

Hız Kısıtlama (Rate Limiting)

Belirli bir IP adresinden, kısa bir süre içinde (örn. 1 dakika) login.php'ye gelen POST isteklerinin sayısı sınırlanmalıdır.

- Örn: Bir IP, 1 dakika içinde 20'den fazla hatalı giriş denemesi yaparsa, o IP 15 dakika boyunca engellenmeli ve HTTP 429 Too Many Requests yanıtı döndürülmelidir. Bu, yaygın (noisy) brute-force saldırılarını yavaşlatır.

CAPTCHA: 3 hatalı giriş denemesinden sonra, kullanıcıya bir CAPTCHA gösterilmelidir. Bu, otomatik bot saldırılarını büyük ölçüde engeller.

Referanslar

<https://cwe.mitre.org/data/definitions/307.html>

[https://owasp.org/Top10/A07_2021-
Identification and Authentication Failures/](https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/)