

IDOR (Insecure Direct Object Reference) Hakkında Her şey | MDISEC Neler Anlattı #3



İLKER YILMAZ

7 min read · 6 days ago



GİRİŞ

IDOR meselesine giriş yapmadan önce bir web uygulamasının temelde nasıl çalıştığını anlamalıyız öncelikle. Bir web uygulamasında siber güvenlik açısından riskleri değerlendireceksek burada en önemli nokta input'lardır. Yani bir web uygulamasının çalışması için bizden aldığı direktiflerdir.

IDOR meselesine giriş yapmadan önce bir web uygulamasının temelde nasıl çalıştığını anlamalıyız öncelikle. Bir web uygulamasında siber güvenlik açısından riskleri değerlendireceksek burada en önemli nokta input'lardır. Yani bir web uygulamasının çalışması için bizden aldığı direktiflerdir.

Şöyle bir web uygulaması düşünelim; bu web uygulaması sadece üzerinde çalıştığı işletim sisteminin saatini ekrana yazıyor. Ne yaparsak yapalım sadece ekrana bu saati yazdırıyor. Yani bizden herhangi bir bilgi ya da direktif almamakta. Böyle bir web uygulamasına takdir edersiniz ki pek bir saldırı bulunmamaktadır. Ancak günümüz web uygulamalarının çoğu bunun aksi yönünde oldukça fazla input alan uygulamalardır. Application security konuştuğumuzda ise en önemli konulardan biri bu input'lardır.

Web uygulamasının çalışması için kullanıcıya bir takım bilgileri vermesi gerekmektedir. Örneğin bir e-ticaret sitesinde adres bilgimizi kaydettiğimizi düşünelim. Birden fazla adres de yazabildiğimizi varsayalım. Ayrıca siparişimizi oluştururken de menüden adresimizi seçebilmekteyiz. Yani uygulamada sipariş tamamlama ekranına gelindiğinde, veritabanında kayıtlı olan adreslerimiz ile ilgili bilgiler alınarak karşımıza getirilmektedir.

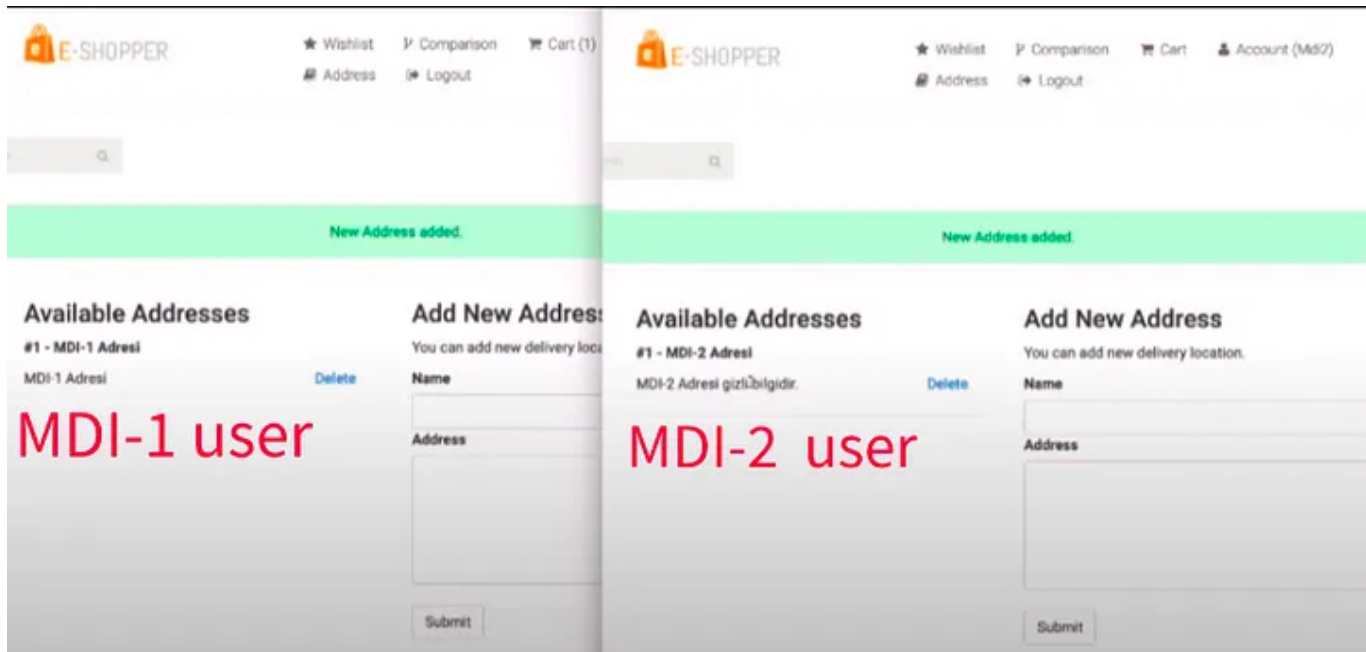
IDOR (Insecure Direct Object Reference) Ne Demektir ?

Bir web uygulaması çalışırken kullanıcıdan aldığı bilgiler ile veritabanındaki birtakım verilere erişim sağlayıp bu veriyi okuyarak kullanıcıya gösterme, güncelleme, silme ya da değiştirme gibi işlemler yapmaktadır. Ancak uygulamanın bu işlemleri yapabilmesi için birtakım kurallar bulunmaktadır.

Örneğin bu kural setlerinden biri şöyle olabilir; başka bir kullanıcının adresini görememeliyiz. Adreslerim kısmına geldiğimizde sadece kendi adreslerimizi görebilmekteyiz ve başka bir kullanıcının adresini görememekteyiz. Eğer siz bu kuralı atlatıp başkasının adresini görebilirseniz bu bizim için IDOR kategorisine giren zafiyet tipimize bir örnek olmuş olur.

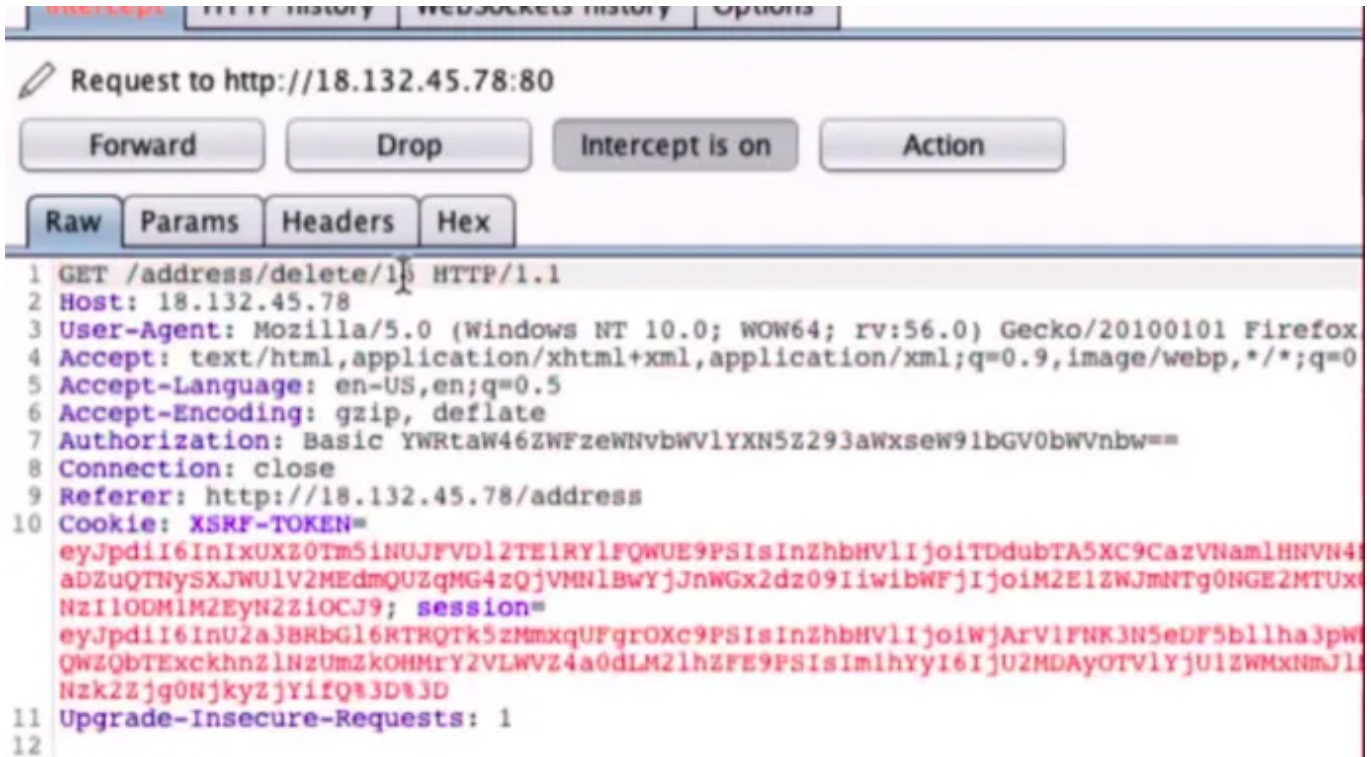
Bir örnekle açıkladığımız ifadeleri ayrıntılı bir şekilde ele alalım;

Buradaki zafiyetli web uygulamamızda bazı denemeler yaparak ilerleyelim. İki adet kullanıcı oluşturalım ve bu kullanıcıların adreslerini ekleyelim.



MDI-1 Kullanıcısı için adresini silmek istediğimizde şöyle bir request ile karşılaşmaktayız. Buradaki request yakalama ve manipüle etme işlemini de

Burp Suite ile sağlamaktayız. Bu araç sayesinde tarayıcımızda gerçekleşen tüm işlemlerin ayrıntılarını görebilmekteyiz.



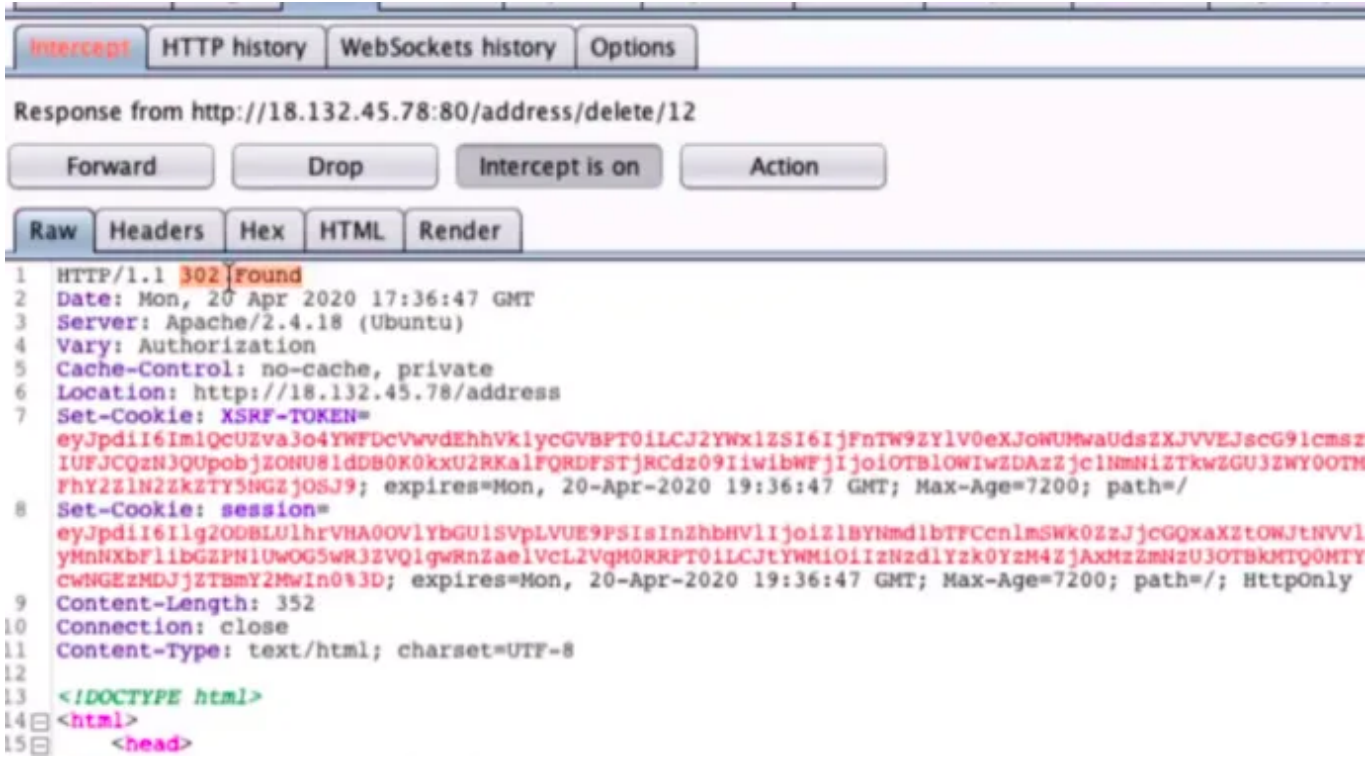
Buradaki request'i inceleyecek olursak; address isimli bir controller'ımızın olduğunu, delete isimli bir fonksiyonumuzun var olduğunu görebilmekteyiz. 15 değeri, silme işlemini gerçekleştiren fonksiyona bir function parametresi olarak iletilmektedir. Burada 15 değeri de veritabanında ilgili adresin ifade ettiği id değeri olarak kabul edilebilir.

```
//Veritabanı yapımızı bu şekilde düşünelim
Addressler //adresler tabosu
```

```
id AUTO INC //bu tabloya ait alanlar
user_id
title
adres_bilgisi
sehir_id
```

12 | MDI-2 Adresi | ?
15 | MDI-1 Adresi | dsfdfdsfs

Yakaladığımız bu request'te 15 değeri yerine 12 yazarsak ne olacağını inceleyelim;



Burada gördüğünüz üzere 302 Found kodu ile cevap verildi. Uygulama arayüzüne geri geldiğimizde de bizleri bu şekilde bir mesaj karşılamaktadır;

Authorization failure

Addresses

si

i bilgidir.

Delete

Add New Address

You can add new delivery location.

Name

Uygulamamızda yer alan ifadede ise 'Authorization Failure' mesajı yer almaktadır. Bu mesajı görüyor olmanız veriyi silemediğiniz anlamına gelmez. Bu sebeple veriyi silip silmediğimizi tekrar kontrol etmeliyiz. Hesaptaki adresleri kontrol ettiğimizde silme işleminin gerçekleştirilemediğini görmekteyiz.

Available Addresses

#1 - MDI-1 Adresi

MDI-1 Adresi

Delete

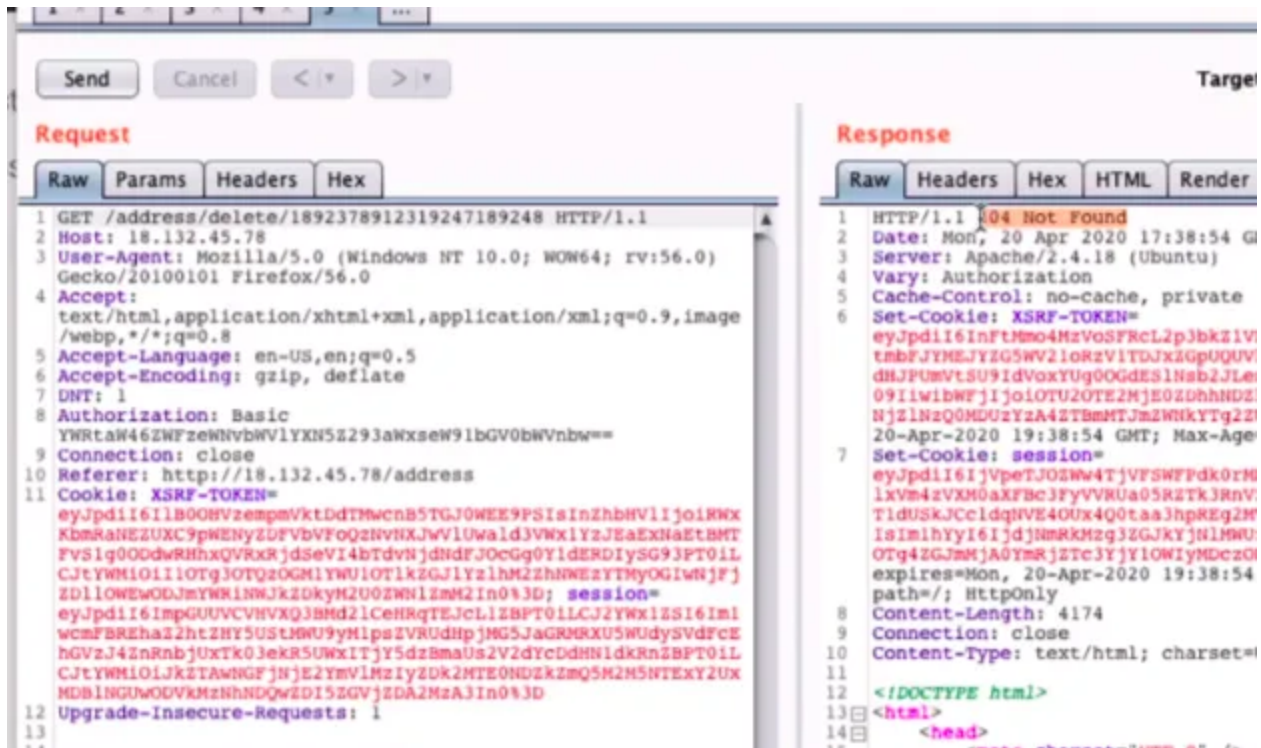
Add New Address

You can add new delivery location.

Name

Address

Peki eğer veritabanında bulunmayan bir id değeri girersek ne olur?



```
//uygulamanın davranışlarına göre oluşturduğumuz tahmini kod yapısı
```

```
class AddressController extend Controller {
  public function delete($address_id){
    if(!AddressModel->chechaddress($address_id)){
      redirect('/',404)
    }

    AddressModel->deleteAddress($address_id);
  }
}
```

Yukarıda yaptığımız işlemler neticesinde 15 değerini girince bu adresi silebiliyor olsaydık IDOR zafiyetinden bahsedebilirdik.

Yani IDOR ne demektir?

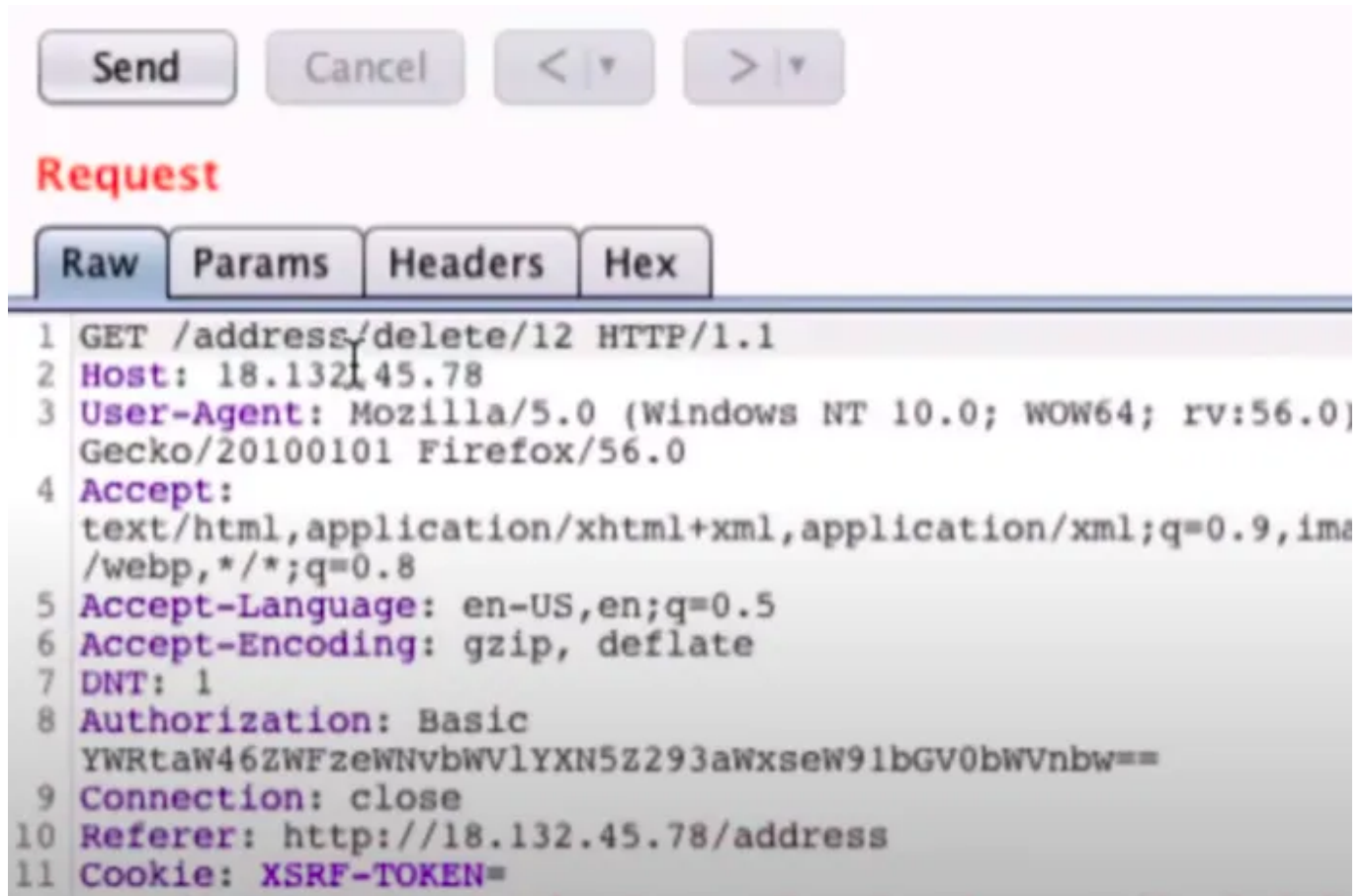
Bir obje referansına güvensiz bir şekilde doğrudan (direct) erişim ile ilgili bir husustur.

Bu aşamada birbiriyle çok karıştırılan Missing Function Level Access ile IDOR zafiyetinin farklarını ele alalım;

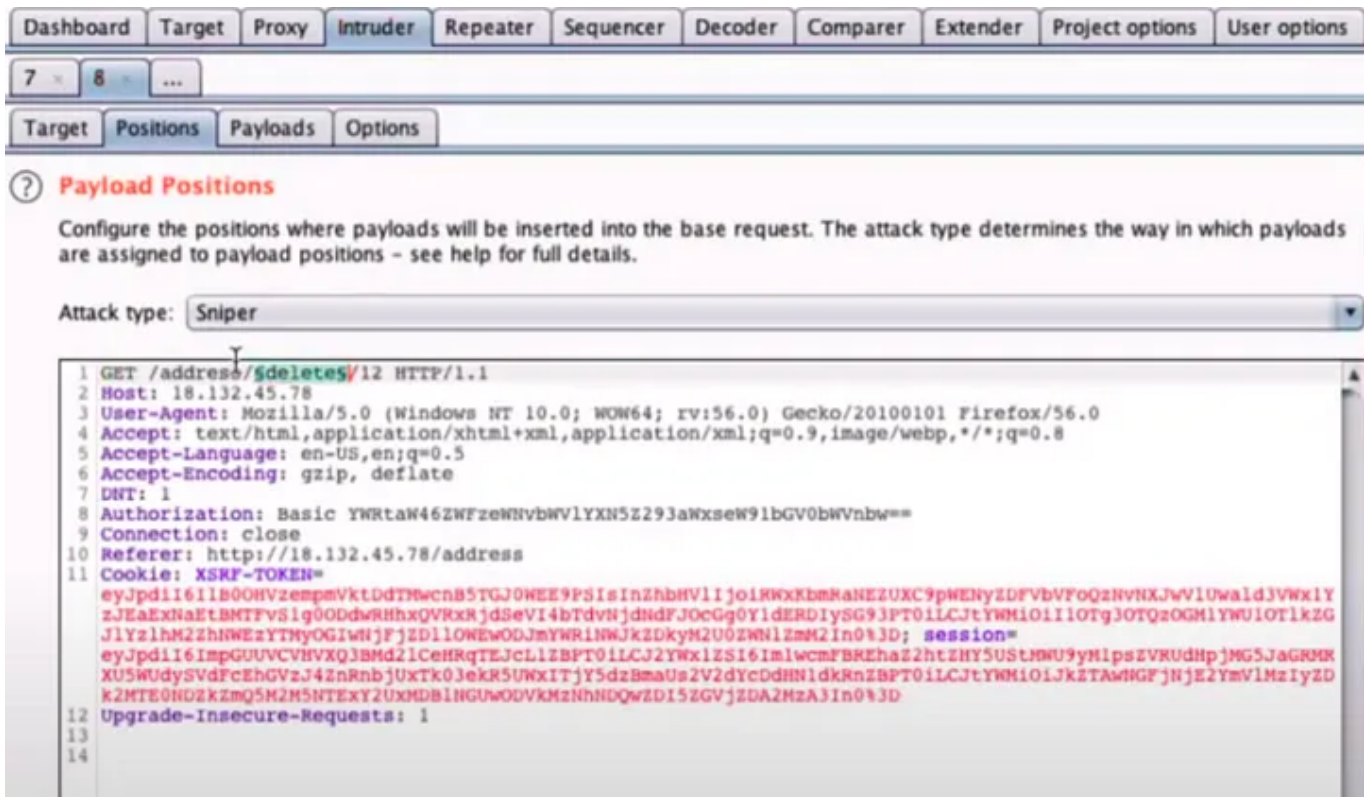
Insecure Direct Object Reference (Veriye Yetkisiz Erişim)

Missing Function Level Access Control (Fonksiyon Seviyesinde Yetki Kontrolü Eksikliği)

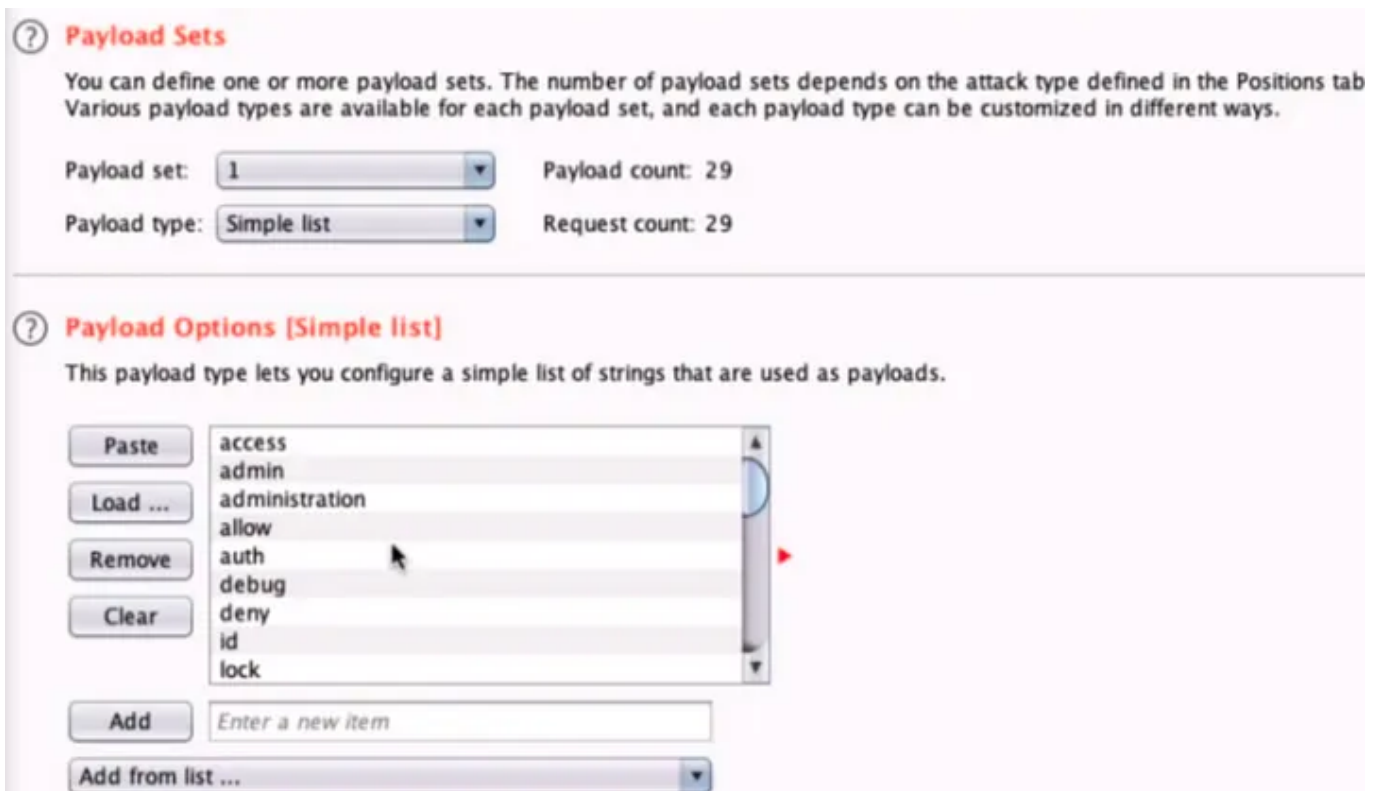
Şöyle bir request üzerinden ilerleyelim;



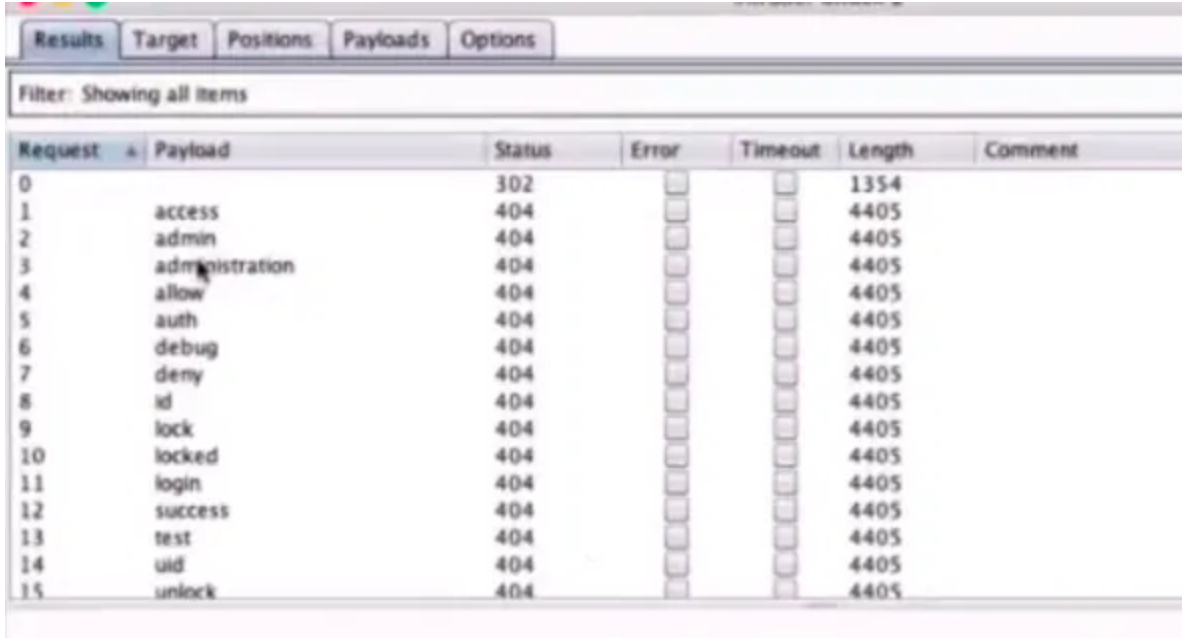
Buradaki request'i Burp Suite ile Intruder kısmına ilettiğimizde burada 'delete' olarak ifade edilen kısma farklı ifadeler koyarak sistemin bize ne gösterdiğini kontrol edelim. Çünkü bu web uygulamasında 'delete' fonksiyonu gibi başka fonksiyonlar da kullanılmış olabilir.



Intruder aracılığıyla ‘delete’ ifadesi yerine başka ifadeler yazarak sistemin ne cevap verdiğini test edebiliriz. Burada fieldname’ler deneyerek ilerleyebiliriz.



İşte yaptığımız işlemler neticesinde aldığımız sonuçlar;



Request	Payload	Status	Error	Timeout	Length	Comment
0		302			1354	
1	access	404			4405	
2	admin	404			4405	
3	administration	404			4405	
4	allow	404			4405	
5	auth	404			4405	
6	debug	404			4405	
7	deny	404			4405	
8	id	404			4405	
9	lock	404			4405	
10	locked	404			4405	
11	login	404			4405	
12	success	404			4405	
13	test	404			4405	
14	uid	404			4405	
15	unlock	404			4405	

Bu sonuçları incelediğimizde 'edit' ifadesi için alınan değerin uzunluğu diğerlerinden farklı bir uzunluktadır. Bu yüzden sistemde kontrol edip bu ifadenin bize hangi sonuçları getirdiğini görmemiz gerekmektedir;



Burada veri olmasına rağmen intruder kısmında neden 404 verildiğini düşünmemiz gerekmektedir. Burada ilk deneme yapılırken 'delete' ifadesi ile adres silinmektedir ve diğer request'lerde böyle bir ifade olmadığı için hata alınmaktadır.

ResultsTargetPositionsPayloadsOptions

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length
19	secret	404	<input type="checkbox"/>	<input type="checkbox"/>	4405
20	visible	404	<input type="checkbox"/>	<input type="checkbox"/>	4405
21	hidden	404	<input type="checkbox"/>	<input type="checkbox"/>	4405
22	hide	404	<input type="checkbox"/>	<input type="checkbox"/>	4405
23	show	404	<input type="checkbox"/>	<input type="checkbox"/>	4405
24	source	404	<input type="checkbox"/>	<input type="checkbox"/>	4405
25	backdoor	404	<input type="checkbox"/>	<input type="checkbox"/>	4405
26	root	404	<input type="checkbox"/>	<input type="checkbox"/>	4405
27	content	404	<input type="checkbox"/>	<input type="checkbox"/>	4405
28	update	404	<input type="checkbox"/>	<input type="checkbox"/>	4405
29	edit	404	<input type="checkbox"/>	<input type="checkbox"/>	5144

RequestResponse

RawParamsHeadersHex

```

1 GET /address/edit/15 HTTP/1.1
2 Host: 18.132.45.78
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:56.0) Gecko/20100101 Firefox/56.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
  
```

Dolayısıyla burada ilk deneme için ‘delete’ ifadesi yerine karşılığı olmayan bir ifade yazarak verinin silinmesini engellemeliyiz.

Target
Positions
Payloads
Options

Payload Positions

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions – see help for full details.

Attack type: Sniper

```

1 GET /address/${invalidmethodnames}/16 HTTP/1.1
2 Host: 18.132.45.78
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:56.0) Gecko/20100101 Firefox/56.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Authorization: Basic YWRtaW46ZWZzeWNVbWVlYXN5Z293aWxsZW91bGV0bWVnbw==
8 Connection: close
9 Referer: http://18.132.45.78/address
10 Cookie: XSRF-TOKEN=
$eyJpdii6Im9zNnpiaitENE02UHVVTSS3bUlRYOE9PSisInzhbMVliJoiaXcvDhsR0FJVWVwQmYxdVpCL3FFADZ6ZmNYeXUyMEFSMUNm
dFVlRlNjZnBOT0ZBbkZ4WGRVNWJQNSFHTLBOMjERdit3QmJaYnEkVklFVUxudEkyTjB0dz09IiwibWFiIjoIYmQ0MThjYzZkMTY0YzgzOH
Zz4MDixNRURlNmQ4OWQ2NTZkOGVhbnUwNmI0OGFkOWE2YmQXNjU0OTE0YjMxZGVlOSJ9$; session=
$eyJpdii6Im9zNnpiaitENE02UHVVTSS3bUlRYOE9PSisInzhbMVliJoiaXcvDhsR0FJVWVwQmYxdVpCL3FFADZ6ZmNYeXUyMEFSMUNm
dFVlRlNjZnBOT0ZBbkZ4WGRVNWJQNSFHTLBOMjERdit3QmJaYnEkVklFVUxudEkyTjB0dz09IiwibWFiIjoIYmQ0MThjYzZkMTY0YzgzOH
Zz4MDixNRURlNmQ4OWQ2NTZkOGVhbnUwNmI0OGFkOWE2YmQXNjU0OTE0YjMxZGVlOSJ9$; session=
$eyJpdii6Im9zNnpiaitENE02UHVVTSS3bUlRYOE9PSisInzhbMVliJoiaXcvDhsR0FJVWVwQmYxdVpCL3FFADZ6ZmNYeXUyMEFSMUNm
dFVlRlNjZnBOT0ZBbkZ4WGRVNWJQNSFHTLBOMjERdit3QmJaYnEkVklFVUxudEkyTjB0dz09IiwibWFiIjoIYmQ0MThjYzZkMTY0YzgzOH
Zz4MDixNRURlNmQ4OWQ2NTZkOGVhbnUwNmI0OGFkOWE2YmQXNjU0OTE0YjMxZGVlOSJ9$; session=
$eyJpdii6Im9zNnpiaitENE02UHVVTSS3bUlRYOE9PSisInzhbMVliJoiaXcvDhsR0FJVWVwQmYxdVpCL3FFADZ6ZmNYeXUyMEFSMUNm
dFVlRlNjZnBOT0ZBbkZ4WGRVNWJQNSFHTLBOMjERdit3QmJaYnEkVklFVUxudEkyTjB0dz09IiwibWFiIjoIYmQ0MThjYzZkMTY0YzgzOH
Zz4MDixNRURlNmQ4OWQ2NTZkOGVhbnUwNmI0OGFkOWE2YmQXNjU0OTE0YjMxZGVlOSJ9$; session=
$eyJpdii6Im9zNnpiaitENE02UHVVTSS3bUlRYOE9PSisInzhbMVliJoiaXcvDhsR0FJVWVwQmYxdVpCL3FFADZ6ZmNYeXUyMEFSMUNm
dFVlRlNjZnBOT0ZBbkZ4WGRVNWJQNSFHTLBOMjERdit3QmJaYnEkVklFVUxudEkyTjB0dz09IiwibWFiIjoIYmQ0MThjYzZkMTY0YzgzOH
Zz4MDixNRURlNmQ4OWQ2NTZkOGVhbnUwNmI0OGFkOWE2YmQXNjU0OTE0YjMxZGVlOSJ9$; session=
$eyJpdii6Im9zNnpiaitENE02UHVVTSS3bUlRYOE9PSisInzhbMVliJoiaXcvDhsR0FJVWVwQmYxdVpCL3FFADZ6ZmNYeXUyMEFSMUNm
dFVlRlNjZnBOT0ZBbkZ4WGRVNWJQNSFHTLBOMjERdit3QmJaYnEkVklFVUxudEkyTjB0dz09IiwibWFiIjoIYmQ0MThjYzZkMTY0YzgzOH
Zz4MDixNRURlNmQ4OWQ2NTZkOGVhbnUwNmI0OGFkOWE2YmQXNjU0OTE0YjMxZGVlOSJ9$; session=
$eyJpdii6Im9zNnpiaitENE02UHVVTSS3bUlRYOE9PSisInzhbMVliJoiaXcvDhsR0FJVWVwQmYxdVpCL3FFADZ6ZmNYeXUyMEFSMUNm
dFVlRlNjZnBOT0ZBbkZ4WGRVNWJQNSFHTLBOMjERdit3QmJaYnEkVklFVUxudEkyTjB0dz09IiwibWFiIjoIYmQ0MThjYzZkMTY0YzgzOH
Zz4MDixNRURlNmQ4OWQ2NTZkOGVhbnUwNmI0OGFkOWE2YmQXNjU0OTE0YjMxZGVlOSJ9$; session=
$eyJpdii6Im9zNnpiaitENE02UHVVTSS3bUlRYOE9PSisInzhbMVliJoiaXcvDhsR0FJVWVwQmYxdVpCL3FFADZ6ZmNYeXUyMEFSMUNm
dFVlRlNjZnBOT0ZBbkZ4WGRVNWJQNSFHTLBOMjERdit3QmJaYnEkVklFVUxudEkyTjB0dz09IiwibWFiIjoIYmQ0MThjYzZkMTY0YzgzOH
Zz4MDixNRURlNmQ4OWQ2NTZkOGVhbnUwNmI0OGFkOWE2YmQXNjU0OTE0YjMxZGVlOSJ9$; session=
$eyJpdii6Im9zNnpiaitENE02UHVVTSS3bUlRYOE9PSisInzhbMVliJoiaXcvDhsR0FJVWVwQmYxdVpCL3FFADZ6ZmNYeXUyMEFSMUNm
dFVlRlNjZnBOT0ZBbkZ4WGRVNWJQNSFHTLBOMjERdit3QmJaYnEkVklFVUxudEkyTjB0dz09IiwibWFiIjoIYmQ0MThjYzZkMTY0YzgzOH
Zz4MDixNRURlNmQ4OWQ2NTZkOGVhbnUwNmI0OGFkOWE2YmQXNjU0OTE0YjMxZGVlOSJ9$; session=
$eyJpdii6Im9zNnpiaitENE02UHVVTSS3bUlRYOE9PSisInzhbMVliJoiaXcvDhsR0FJVWVwQmYxdVpCL3FFADZ6ZmNYeXUyMEFSMUNm
dFVlRlNjZnBOT0ZBbkZ4WGRVNWJQNSFHTLBOMjERdit3QmJaYnEkVklFVUxudEkyTjB0dz09IiwibWFiIjoIYmQ0MThjYzZkMTY0YzgzOH
Zz4MDixNRURlNmQ4OWQ2NTZkOGVhbnUwNmI0OGFkOWE2YmQXNjU0OTE0YjMxZGVlOSJ9$; session=
$eyJpdii6Im9zNnpiaitENE02UHVVTSS3bUlRYOE9PSisInzhbMVliJoiaXcvDhsR0FJVWVwQmYxdVpCL3FFADZ6ZmNYeXUyMEFSMUNm
dFVlRlNjZnBOT0ZBbkZ4WGRVNWJQNSFHTLBOMjERdit3QmJaYnEkVklFVUxudEkyTjB0dz09IiwibWFiIjoIYmQ0MThjYzZkMTY0YzgzOH
Zz4MDixNRURlNmQ4OWQ2NTZkOGVhbnUwNmI0OGFkOWE2YmQXNjU0OTE0YjMxZGVlOSJ9$; session=
$eyJpdii6Im9zNnpiaitENE02UHVVTSS3bUlRYOE9PSisInzhbMVliJoiaXcvDhsR0FJVWVwQmYxdVpCL3FFADZ6ZmNYeXUyMEFSMUNm
dFVlRlNjZnBOT0ZBbkZ4WGRVNWJQNSFHTLBOMjERdit3QmJaYnEkVklFVUxudEkyTjB0dz09IiwibWFiIjoIYmQ0MThjYzZkMTY0YzgzOH
Zz4MDixNRURlNmQ4OWQ2NTZkOGVhbnUwNmI0OGFkOWE2YmQXNjU0OTE0YjMxZGVlOSJ9$; session=
$eyJpdii6Im9zNnpiaitENE02UHVVTSS3bUlRYOE9PSisInzhbMVliJoiaXcvDhsR0FJVWVwQmYxdVpCL3FFADZ6ZmNYeXUyMEFSMUNm
dFVlRlNjZnBOT0ZBbkZ4WGRVNWJQNSFHTLBOMjERdit3QmJaYnEkVklFVUxudEkyTjB0dz09IiwibWFiIjoIYmQ0MThjYzZkMTY0YzgzOH
Zz4MDixNRURlNmQ4OWQ2NTZkOGVhbnUwNmI0OGFkOWE2YmQXNjU0OTE0YjMxZGVlOSJ9$; session=
$eyJpdii6Im9zNnpiaitENE02UHVVTSS3bUlRYOE9PSisInzhbMVliJoiaXcvDhsR0FJVWVwQmYxdVpCL3FFADZ6ZmNYeXUyMEFSMUNm
dFVlRlNjZnBOT0ZBbkZ4WGRVNWJQNSFHTLBOMjERdit3QmJaYnEkVklFVUxudEkyTjB0dz09IiwibWFiIjoIYmQ0MThjYzZkMTY0YzgzOH
Zz4MDixNRURlNmQ4OWQ2NTZkOGVhbnUwNmI0OGFkOWE2YmQXNjU0OTE0YjMxZGVlOSJ9$; session=
$eyJpdii6Im9zNnpiaitENE02UHVVTSS3bUlRYOE9PSisInzhbMVliJoiaXcvDhsR0FJVWVwQmYxdVpCL3FFADZ6ZmNYeXUyMEFSMUNm
dFVlRlNjZnBOT0ZBbkZ4WGRVNWJQNSFHTLBOMjERdit3QmJaYnEkVklFVUxudEkyTjB0dz09IiwibWFiIjoIYmQ0MThjYzZkMTY0YzgzOH
Zz4MDixNRURlNmQ4OWQ2NTZkOGVhbnUwNmI0OGFkOWE2YmQXNjU0OTE0YjMxZGVlOSJ9$; session=
$eyJpdii6Im9zNnpiaitENE02UHVVTSS3bUlRYOE9PSisInzhbMVliJoiaXcvDhsR0FJVWVwQmYxdVpCL3FFADZ6ZmNYeXUyMEFSMUNm
dFVlRlNjZnBOT0ZBbkZ4WGRVNWJQNSFHTLBOMjERdit3QmJaYnEkVklFVUxudEkyTjB0dz09IiwibWFiIjoIYmQ0MThjYzZkMTY0YzgzOH
Zz4MDixNRURlNmQ4OWQ2NTZkOGVhbnUwNmI0OGFkOWE2YmQXNjU0OTE0YjMxZGVlOSJ9$; session=
$eyJpdii6Im9zNnpiaitENE02UHVVTSS3bUlRYOE9PSisInzhbMVliJoiaXcvDhsR0FJVWVwQmYxdVpCL3FFADZ6ZmNYeXUyMEFSMUNm
dFVlRlNjZnBOT0ZBbkZ4WGRVNWJQNSFHTLBOMjERdit3QmJaYnEkVklFVUxudEkyTjB0dz09IiwibWFiIjoIYmQ0MThjYzZkMTY0YzgzOH
Zz4MDixNRURlNmQ4OWQ2NTZkOGVhbnUwNmI0OGFkOWE2YmQXNjU0OTE0YjMxZGVlOSJ9$; session=
$eyJpdii6Im9zNnpiaitENE02UHVVTSS3bUlRYOE9PSisInzhbMVliJoiaXcvDhsR0FJVWVwQmYxdVpCL3FFADZ6ZmNYeXUyMEFSMUNm
dFVlRlNjZnBOT0ZBbkZ4WGRVNWJQNSFHTLBOMjERdit3QmJaYnEkVklFVUxudEkyTjB0dz09IiwibWFiIjoIYmQ0MThjYzZkMTY0YzgzOH
Zz4MDixNRURlNmQ4OWQ2NTZkOGVhbnUwNmI0OGFkOWE2YmQXNjU0OTE0YjMxZGVlOSJ9$; session=
$eyJpdii6Im9zNnpiaitENE02UHVVTSS3bUlRYOE
```

Artık bu şekilde request istekleri istediğimiz gibi gelmektedir. 'edit' field name'i için de 200 kodunun döndüğünü görebilmekteyiz;

Results	Target	Positions	Payloads	Options
---------	--------	-----------	----------	---------

Filter: Showing all items

Request	Position	Payload	Status	Error	Timeout	Length	Comment
24	1	source	404			4405	
25	1	backdoor	404			4405	
26	1	root	404			4405	
27	1	content	404			4405	
28	1	update	404			4405	
29	1	edit	200			11552	
30	2	access	404			4405	
31	2	admin	404			4405	
32	2	administration	404			4405	
33	2	allow	404			4405	
34	2	auth	404			4405	
35	2	debug	404			4405	
36	2	deny	404			4405	
37	2	id	404			4405	
38	2	lock	404			4405	
39	2	locked	404			4405	

Request

Response

Raw	Params	Headers	Hex
-----	--------	---------	-----

1 GET /address/edit/16 HTTP/1.1

2 Host: 18.132.45.78

3 User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:56.0) Gecko/20100101 Firefox/56.0

4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8

5 Accept-Language: en-US,en;q=0.5

6 Accept-Encoding: gzip, deflate

7 Authorization: Basic YWRtaW46ZWZlZGVhbnVlYXN5Z293aWxseW91bGV0bWVnbw==

8 Connection: close

9 Referer: http://18.132.45.78/address

10 Cookie: XSRF-TOKEN=evJpd1l6Im92Zm9laiteENE02UHVVTS3bU1RY0E9PSisInZhbHVlIjoiaXcxV0hsR0FJVWVwQmYxdVpcLjFFaDZ6Zm9yeXUyMEF5aHUiImdFVlR3h1ZnB0T0ZBbkI

11

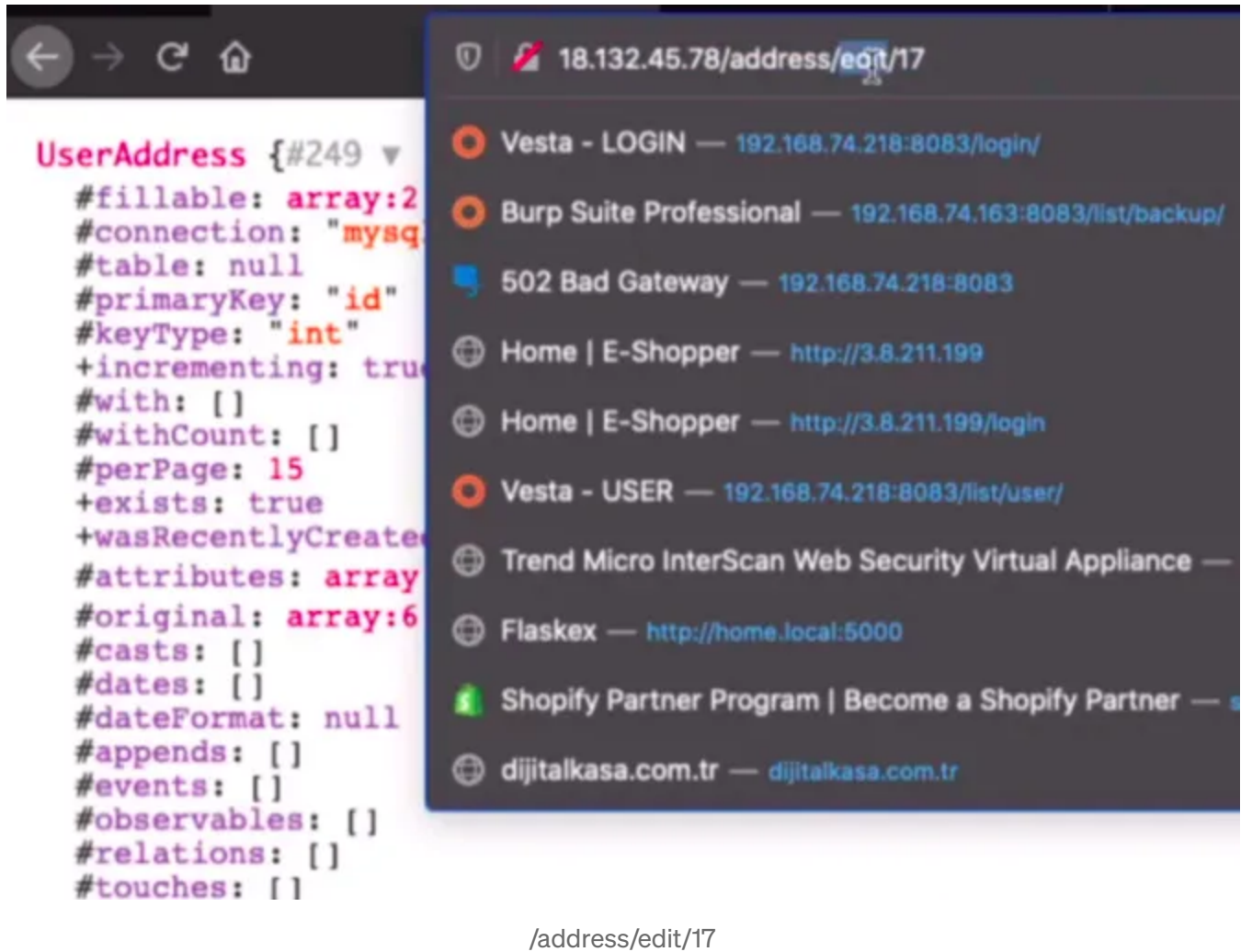
<

+

>

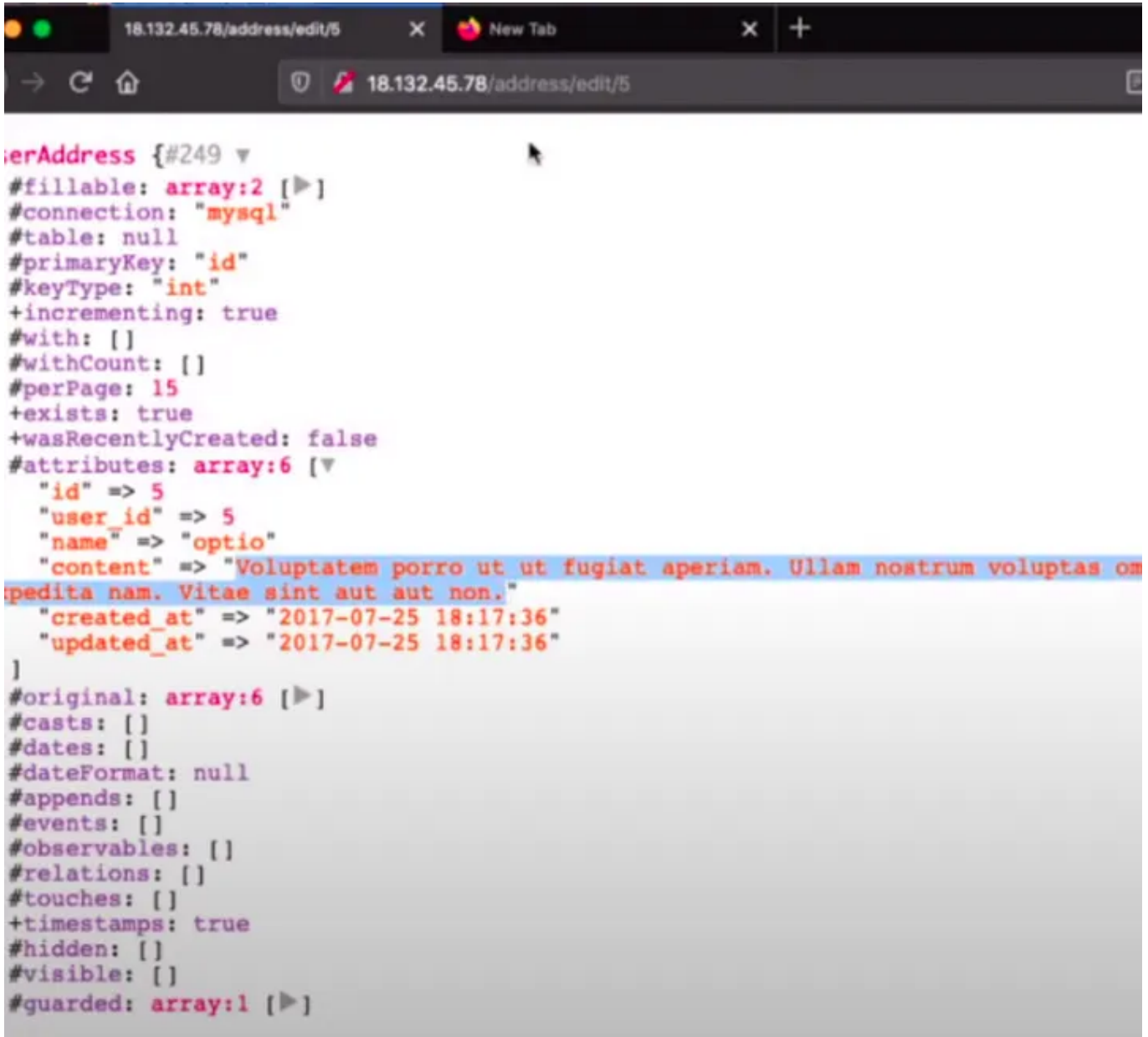
Type a search term

Buraya kadar olan kısımdan yapabileceğimiz bazı çıkarımlar bulunmaktadır. İncelediğimiz web uygulamasında bize 'edit' ile ilgili bir fonksiyon işlemi sunulmamasına rağmen bu fonksiyona erişebildik. Bu konu Missing Function Level Access Control zafiyeti olarak kabul edilmektedir.



/address/edit/17

IDOR zafiyeti ise başkasına ait verileri görebildiğimiz durumlarda geçerlidir. Örneğin 'id' değeri olarak 17 ifadesi yerine 5 ifadesini koyduğumuzda başkasına ait verileri görebildiğimiz için IDOR zafiyetinin varlığından da bahsedebiliriz.

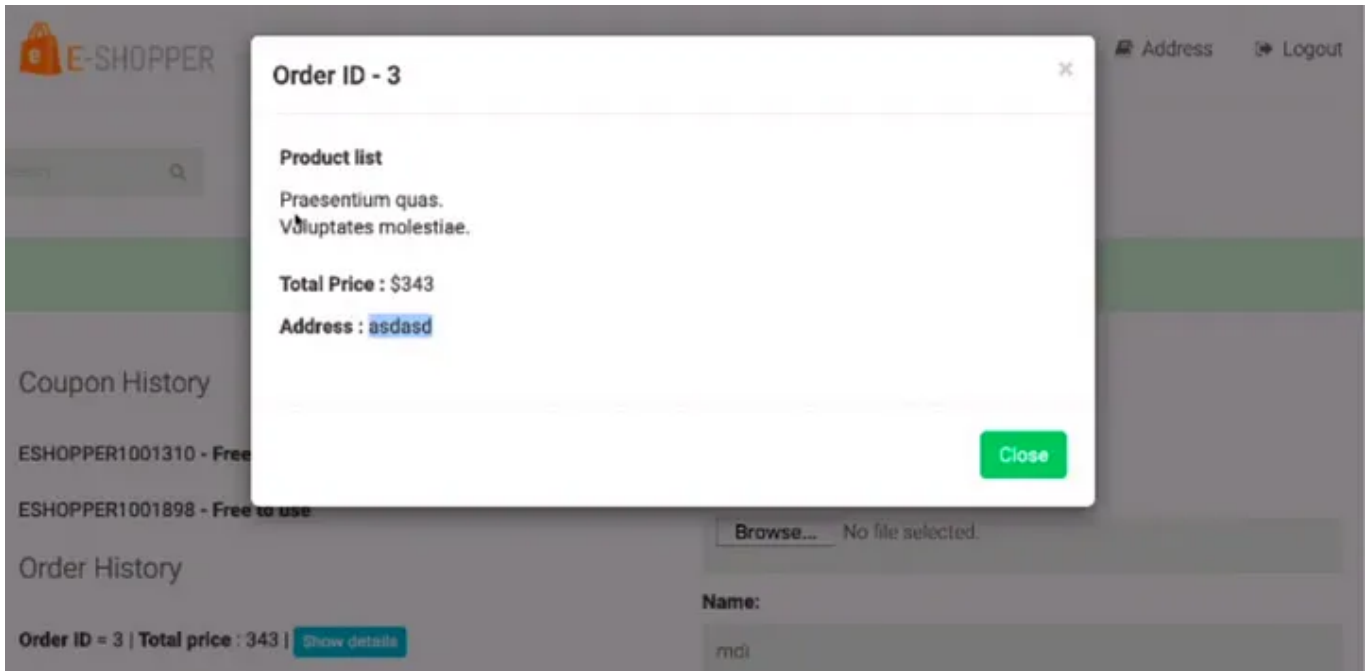


/address/edit/5

Bu farkı da açıkladıktan sonra web uygulaması için başka yerlerde IDOR zafiyeti olabilir mi ona bakalım?

Sipariş detayı sayfasına geldiğimizde adres bilgilerini görebilmekteyiz.

Burası adresin kullanıldığı bir başka yer. Herhangi bir ürünü sepete ekleyip onu alacağımız esnada adres listedi gelmektedir.

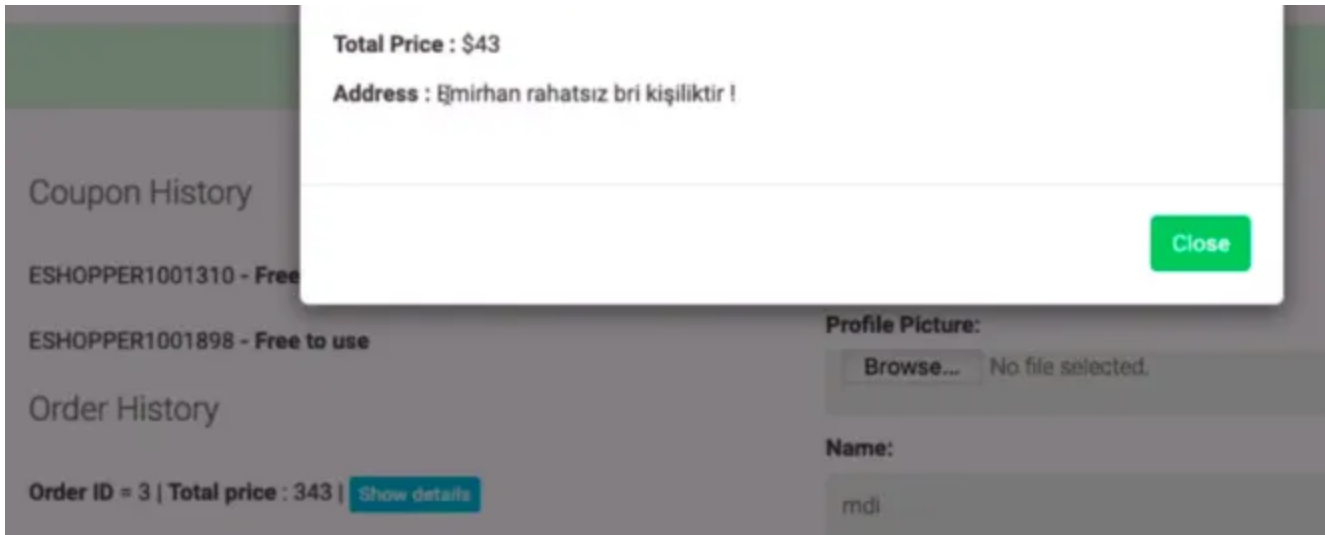


Peki buradaki adres ifadesi için 'id' değerini değiştirirsek başka bir kullanıcının adres bilgisine erişebilir miyiz?

ibHVlIjoiczJreWRSWHdiWkxaUkxITmVNT0tUaXNhekd
09IiwibWFjIjo1NjN1ZmZjOWIyNGIwYzYyNzA4MDE2MT

quantity%5B%5D=1&coupon=&address=17

Bu kısımdaki address değerini 17 yerine 18 yaparak başka bir kullanıcının adres bilgileri ile değiştirebilmekteyiz. Yani burada sipariş özelliğini kullanarak başka bir kullanıcının adres bilgisine erişebilmekteyiz.



Bu işlemin literatürdeki tam adı da **Second Order Insecure Direct Object Reference** şeklindedir. Adresi seçtiğimiz endpoint ile değiştirdiğimiz id değerini gördüğümüz kısım farklıdır çünkü. Olay tek bir request-response döngüsü içinde yaşanmamaktadır.

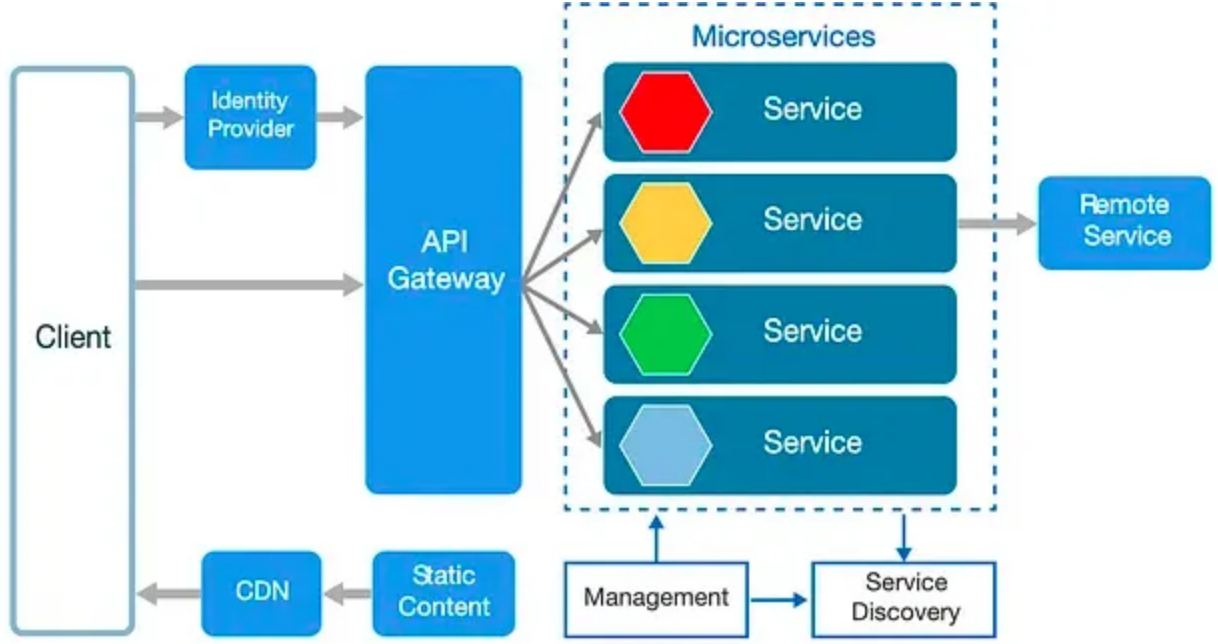
IDOR Neden Web Dünyasındaki En Zor Zafiyettir ?

Zafiyetler temel olarak Teknik zafiyetler ve Business Logic zafiyetler olmak üzere ikiye ayrılmaktadır. Burada Business Logic bir zafiyetten bahsetmekteyiz. Herhangi bir saldırı kodu yok. SQL Injection'da, XSS'te, SSRF'te herhangi bir payload görebilirsiniz ancak IDOR'da payload bulunmamakta. Örneğin id değeri 17 iken 15'e çevirdik. Code review'da da anlaşılıp bulunması son derece zor bir zafiyettir. Kaynak kod analizi araçlarının da bulması zor olan bir zafiyet türüdür.

Günümüzde IDOR'un En Çok Karşılaşılan ve Etkisi En Kritik Zafiyetlerden Olmasının Sebebi Nedir?

Bu konuda da developer'ların yani yazılım geliştiricilerin çok büyük yapılar içerisinde kaybolduğu gerçeği yer almaktadır. Bu yüzden de bu tür açıklar oldukça yaygındır.

Bu Tür Zafiyetler Nasıl ve Neden Ortaya Çıkmaktadır ? — Mikro servis Mimarilerine Bakış



<https://medium.com/@OlabodeAbesin/microservice-architecture-the-complete-guide-357bf7131cf1>

Günümüzde şöyle bir durum bulunmaktadır. Bir e-ticaret sitesi düşünelim. Burada user'ın bilgisi bulunmakta. User bilgisine erişmesi gereken farklı uygulamalar vardır. Bunun için user bilgilerini dönen bir service yazılır. Burada mimari düzgün bir yapıda oluşturulmalıdır. Kullanıcı yetkilendirme işlemlerinin nasıl uygulanacağı belirlenmelidir. Eğer uygun yapı oluşturulmazsa bu yanlış mimariler IDOR gibi zafiyetlere sebebiyet verebilmektedir.

Bir örnek verecek olursak buradaki yapıda farklı programlama dillerinin json'ı farklı yorumlamasından dolayı ortaya çıkan bir problem ile karşılaşmaktayız. API Gateway ile Microservice yapısındaki programlama dillerinin farklılığından dolayı bu tarz problemler ortaya çıkabilmektedir.

```
GET /user/ HTTP/1.1
Host: api.asdasd.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:56.0) Gecko/20100101 Firefox/56.0
Authorization: asdhajksdhasjkdahjsd

{
  'operation': 'get_user_info',
  'user_id': '5',
  'user_id': '6',
}

PYTHON obj.get('user_id') = 5

X

KENDİ
```

AuthMatrix

İncelediğimiz bir web uygulaması için yetki ve grup tanımlamalarını oluşturarak denemeler yapmak gerekmektedir. Tüm yetki şeması simüle edilerek hangi kullanıcının neyi görebildiğini öğrenebiliriz.

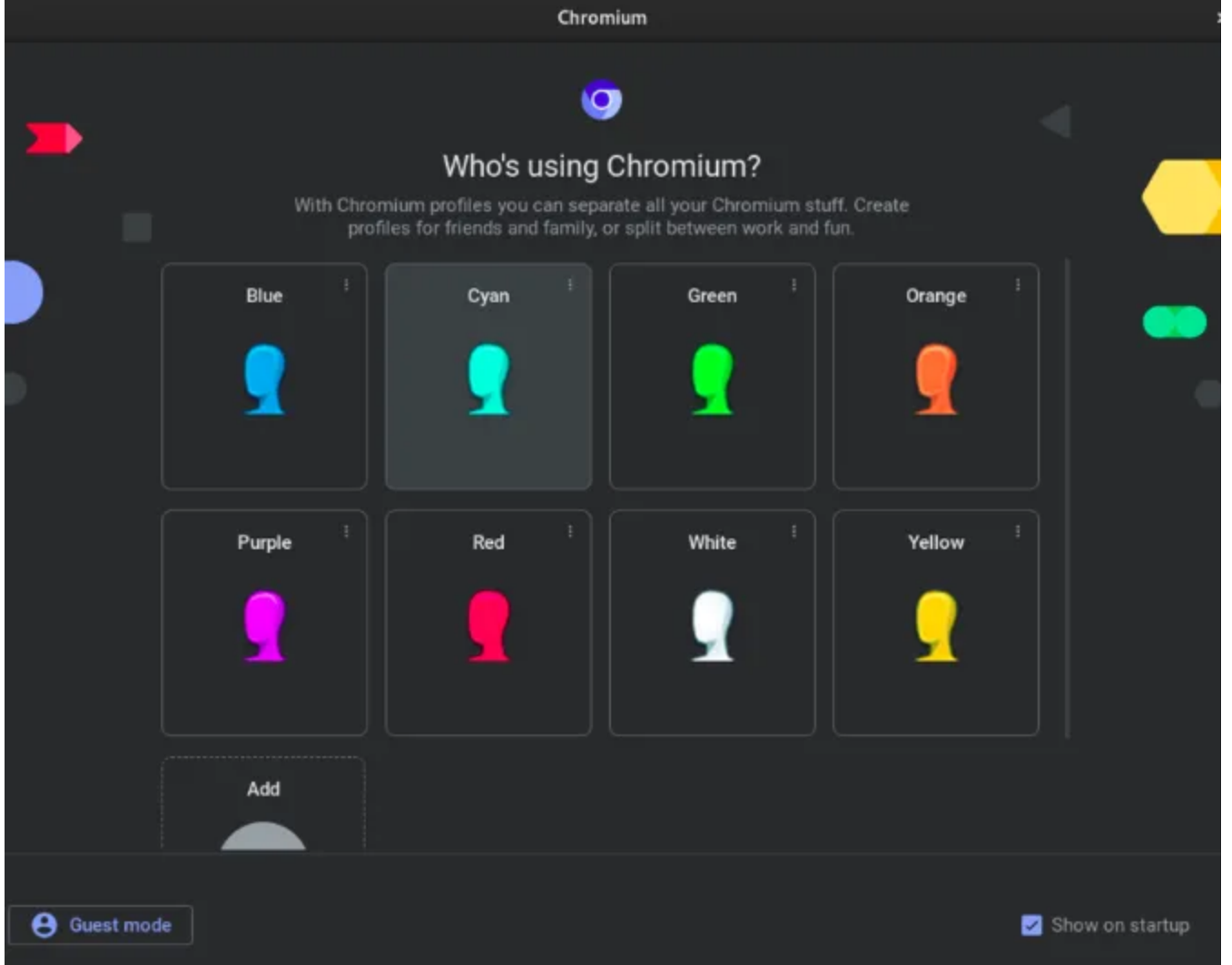
User Name	Cookies	Anony	Moderator	Admin	Reg
anony		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
mdi1		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
mdi2		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
moderator1		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
admin1		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

ID	Request Name	Response Regex	Anony	Mo...	Ad...	Reg...	Ano...	mdi...	mdi...	Use...	Mo...	ano...	mdi...
----	--------------	----------------	-------	-------	-------	--------	--------	--------	--------	--------	-------	--------	--------

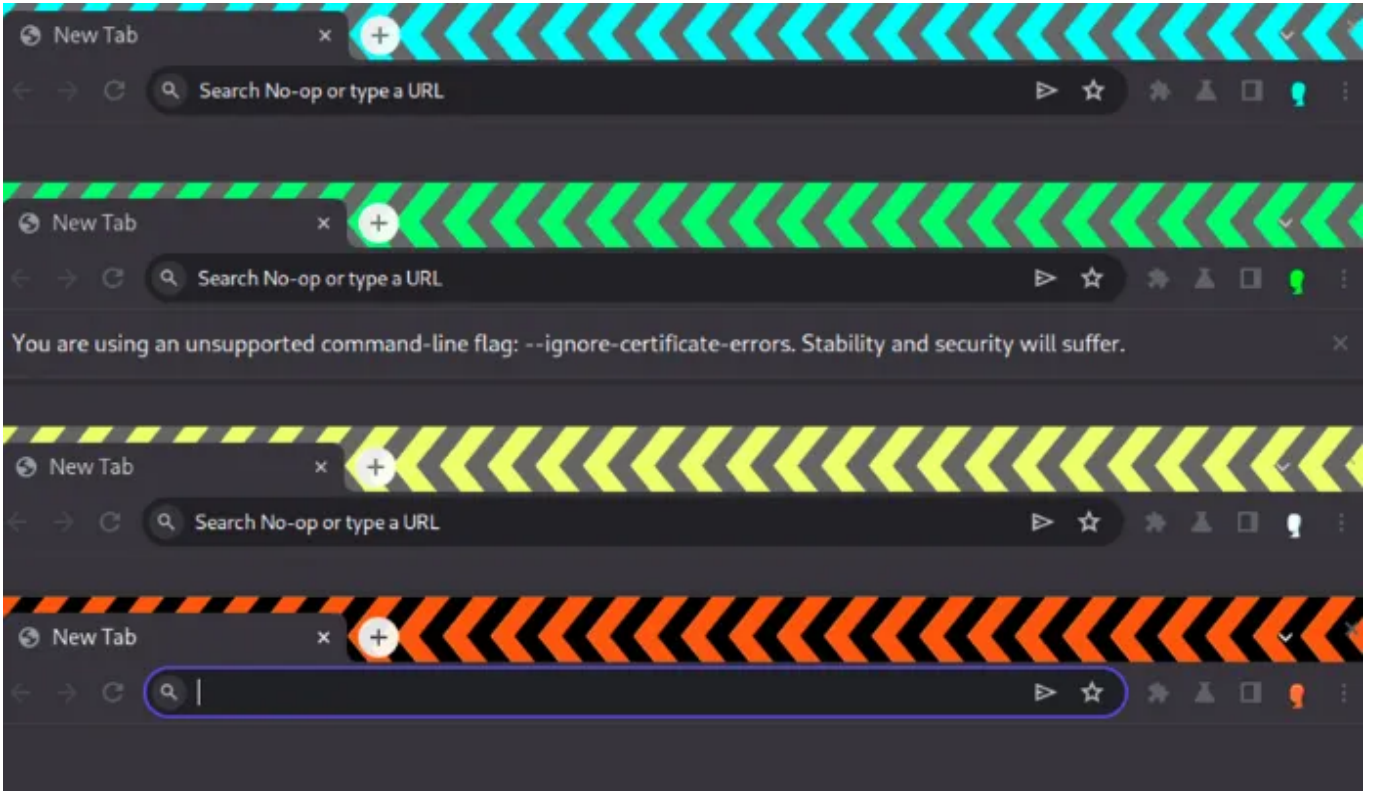
autochrome

Araştırdığımız sistemde birden fazla kullanıcı ekleyerek denemeler yapmamız gerektiği için kullanabileceğimiz bir diğer faydalı araç da autochrome aracıdır. Bu araç sayesinde istediğiniz kullanıcıları ekleyerek

her kullanıcı için ayrı bir pencerede işlemlerinizi yürütebilirsiniz. Bu sayede yaptığınız işlemler birbiriyle karışmayacaktır.



autochrome aracı için de buradaki bağlantıdan kurulumu yapabilirsiniz;
<https://github.com/nccgroup/autochrome>



Buraya kadar okuduğunuz için teşekkür ederim. Selametle ...

KAYNAKÇA

[Web Security 101 0x02 | IDOR Insecure Direct Object Reference Zafiyetleri Hakkında Her şey — Mehmet İnce — Youtube](#)

<https://medium.com/@aysebilgegunduz/everything-you-need-to-know-about-idor-insecure-direct-object-references-375f83e03a87s>