**OWASP LAB ÇÖZÜMLEMELERİ**

OWASP (Open Web Application Security Project) listesinde yer alan Injection, Broken Access Control ve SSRF (Server-Side Request Forgery) zaafiyetlerinin çözümlemeleri ve açıklamaları.

**A. Injection**

**Açıklama:** Injection zaafiyetleri, bir saldırganın veri girişi üzerinden SQL, OS komutları veya başka komut dillerini çalıştırmasına olanak tanır. Bu, sistem üzerinde yetkisiz işlem yapılmasına veya veri sızdırılmasına neden olabilir.

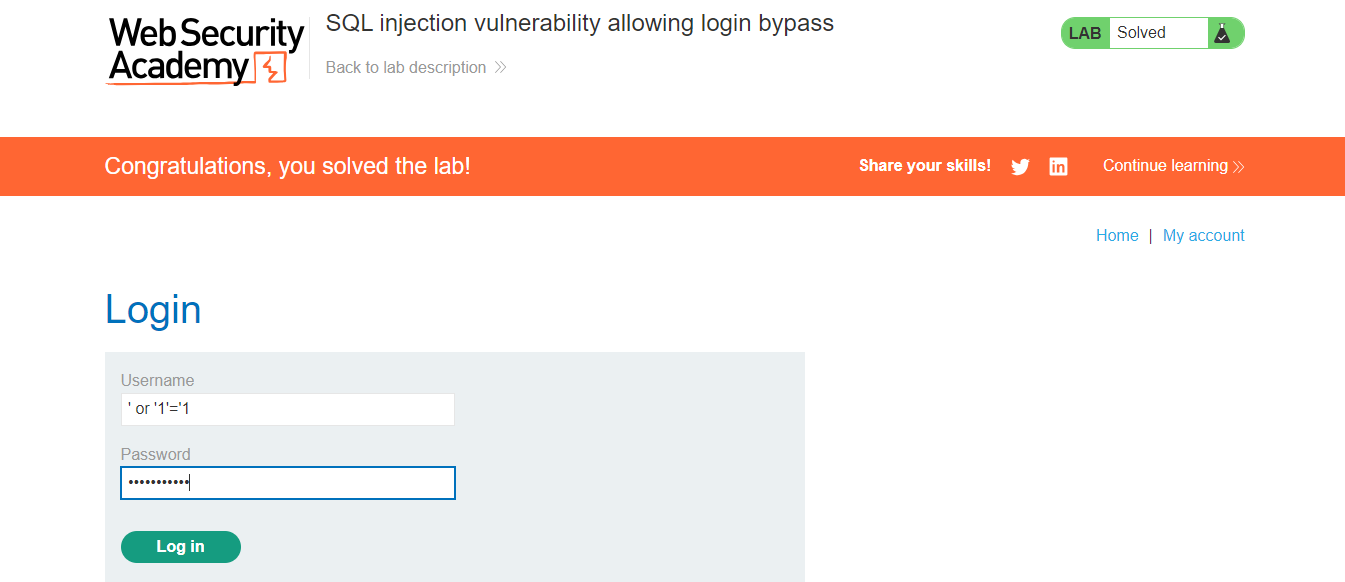
1. **LAB:** **SQL injection vulnerability allowing login bypass**

Bu laboratuvar, oturum açma işlevinde bir SQL güvenlik açığı içeriyor.

Laboratuvarı çözmek için uygulamada yönetici kullanıcı olarak oturum açan bir SQL enjeksiyon saldırısı gerçekleştirerek yönetici girişi yapmamız isteniyor.

1. **Adım:**

Siteye or açığı kullanarak girmeyi deniyoruz:



1. **Adım:**

Görüldüğü Gibi Sisteme Administrator Kullanıcı yani Yönetici Olarak Giriş Yapmış Bulunmaktayız:

****

1. **LAB : SQL injection vulnerability in WHERE clause allowing retrieval of hidden data**

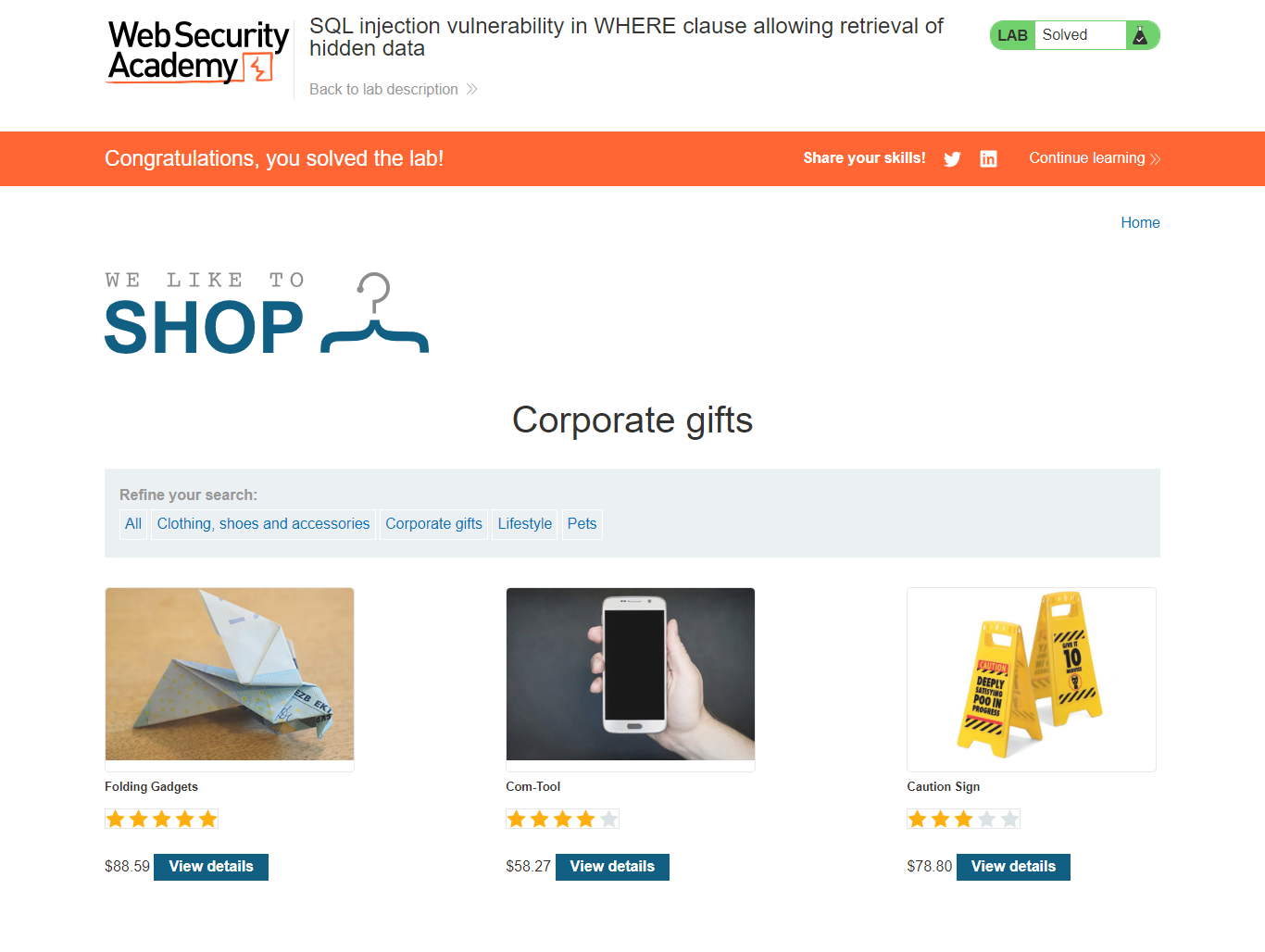
Bu laboratuvar, ürün kategorisi filtresinde bir [SQL enjeksiyonu güvenlik açığı içerir. Kullanıcı bir kategori seçtiğinde, uygulama aşağıdaki gibi bir SQL sorgusu gerçekleştirir:](https://portswigger.net/web-security/sql-injection)

SELECT \* FROM products WHERE category = 'Gifts' AND released = 1

Laboratuvarı çözmek için, sitenin bir veya daha fazla yayınlanmamış ürünü görüntülemesini sağlayan bir SQL enjeksiyon saldırısı gerçekleştirmemiz isteniyor.

1. Adım:

Herhangi bir kategoriye giriyoruz öncelikle:



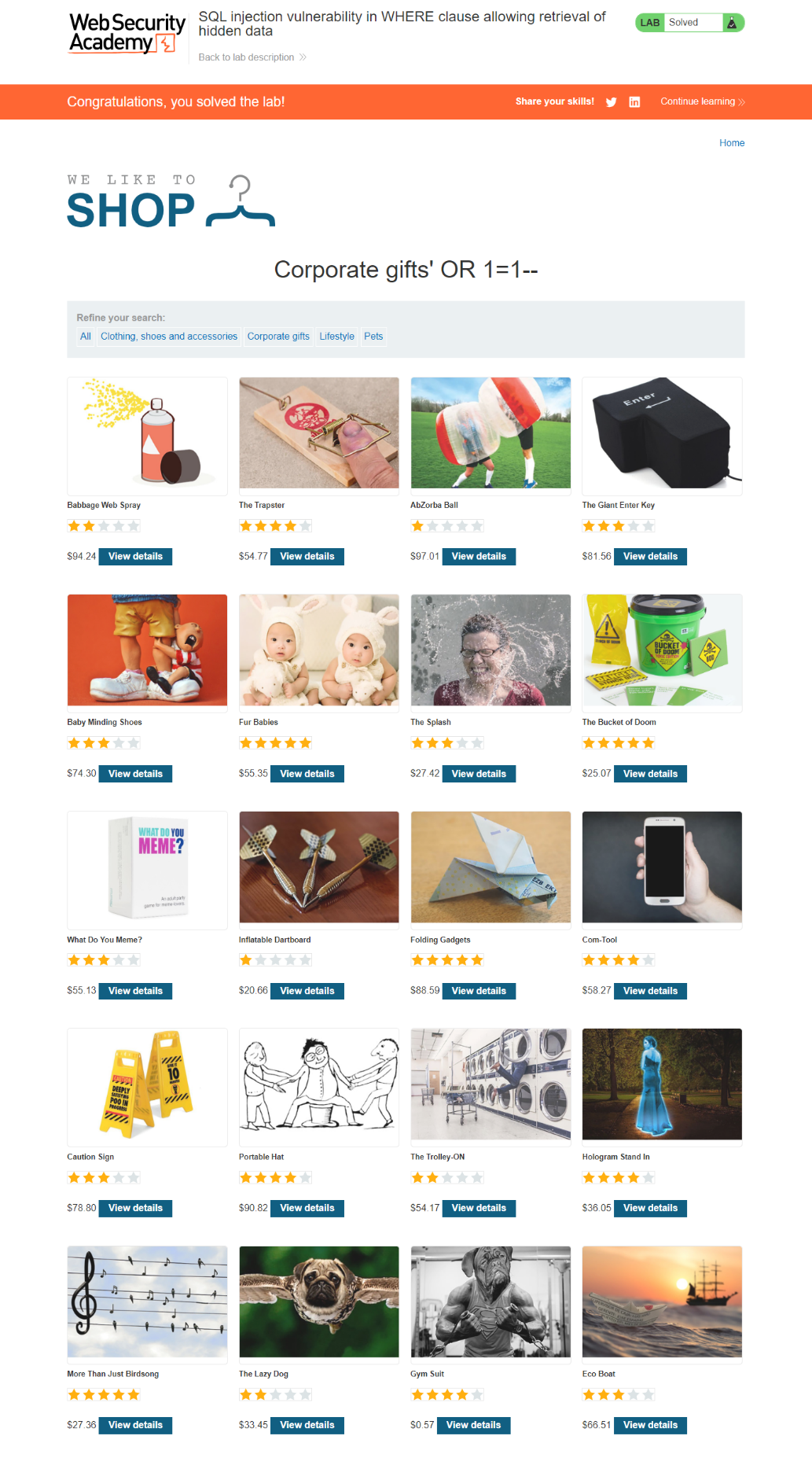
1. Adım:

Url Sonuna SQL kodu ekleyerek sonuca bakacağız SQL kodu : ' OR 1=1—



1. Adım

Ve Sonuç Olarak Gizli ve Normal Tüm Ürünleri Görmüş Bulunmaktayız :



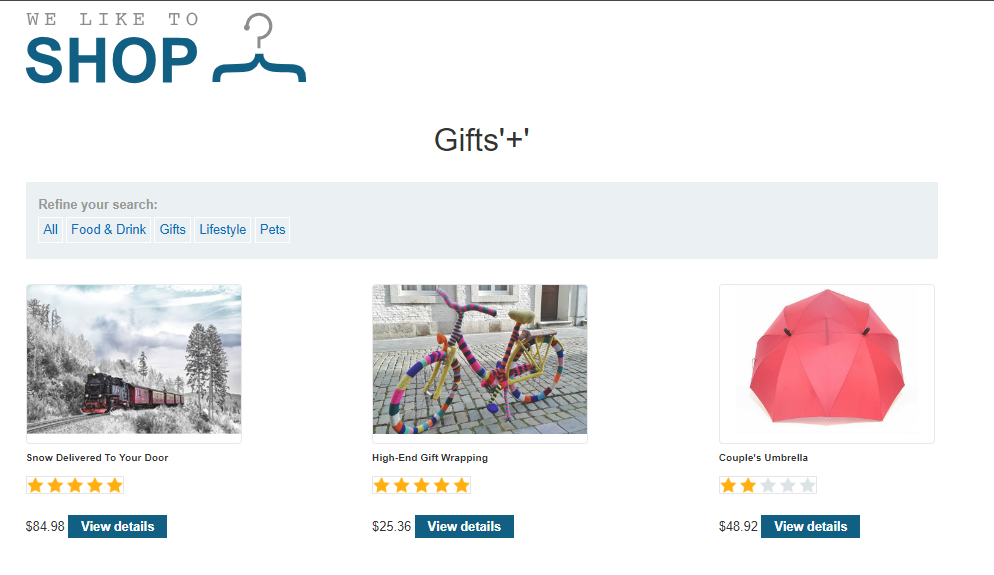
1. **LAB : Detecting NoSQL injection**

[Bu laboratuvar için ürün kategorisi filtresi bir MongoDB NoSQL veritabanı tarafından desteklenmektedir. NoSQL enjeksiyonuna](https://portswigger.net/web-security/nosql-injection) karşı savunmasızdır .

Laboratuvar sorununu çözmek için, sitenin yayınlanmayan ürünleri görmemizi sağlayan bir NoSQL enjeksiyon saldırısı yapmamız istenmekte.

1. **Adım:**

Örnek Bir Kodu Şifreleyerek Kategori Url Sonuna Ekliyoruz Ve Hata Olup Olmadığına Bakıyoruz örnek kod ('+'):

****

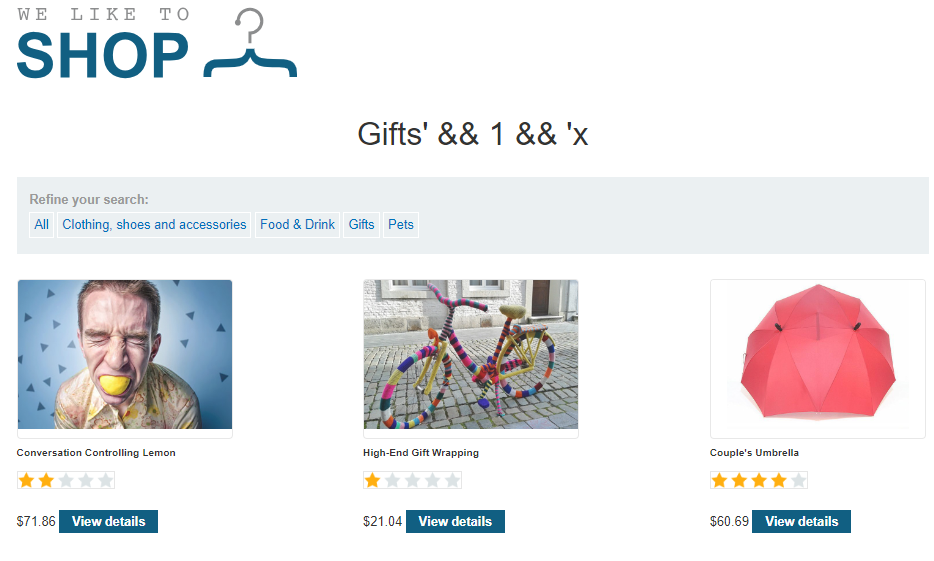
1. **Adım:**

Şimdi Sırada Şifrelenmiş Boolean Kodu Enjekte Ederek Yanlış Bir Paremetre İle Ürün Almamamız Gerek Onu Kontrol Edeceğiz Kod(Gifts' && 0 && 'x):



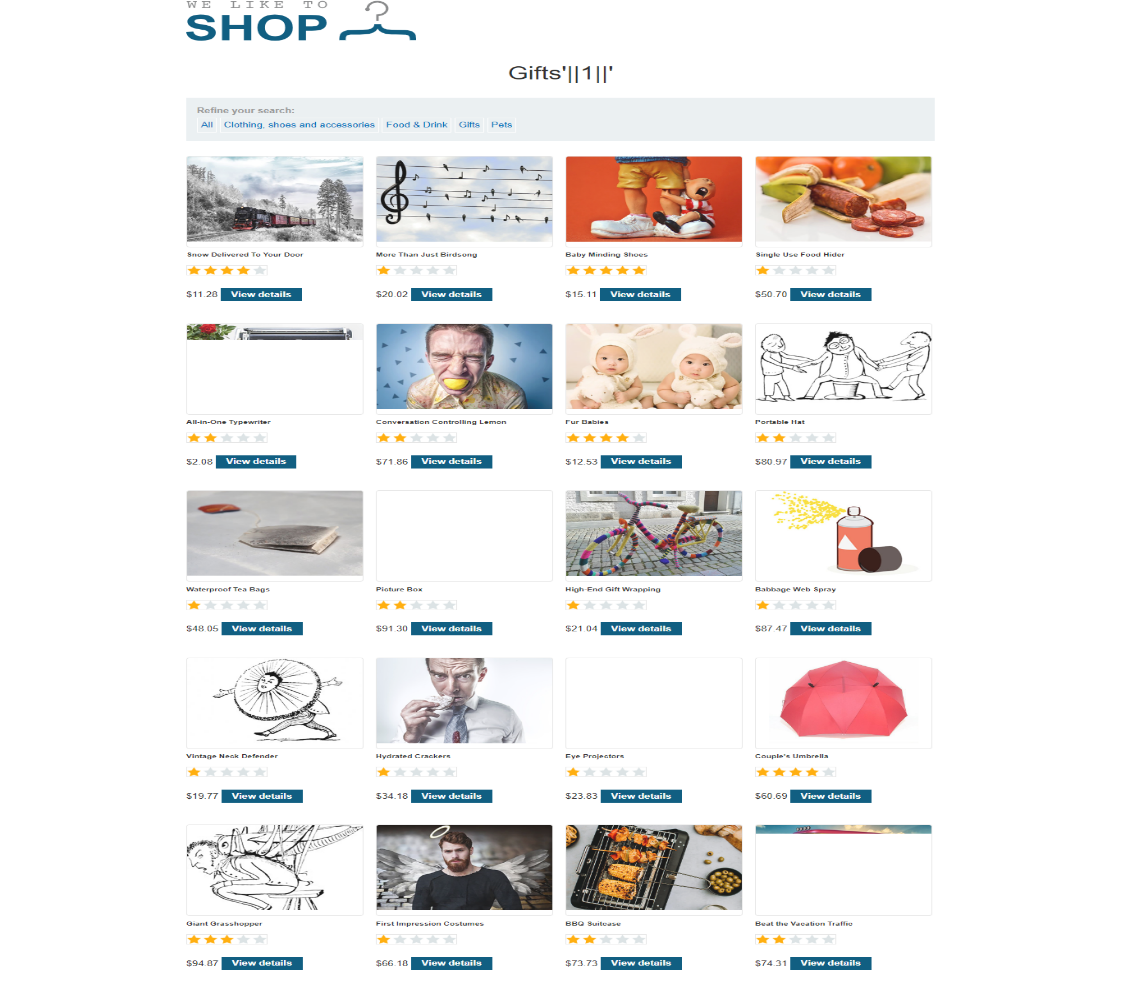
1. **Adım**

Şimdi Doğru Koşul Ekleyerek Ürünleri Listelemeye Çalışacağız Kod(Gifts' && 1 && 'x):



1. **Adım**

Şimdide Herzaman Doğru Döndüren Şifrelenmiş Bir Boolean Kodunu Enjekte Ederek Tüm Yayınlanan ve Yayınlanmamış Ürünleri Görüyoruz Kod('||1||'):



**B.** [**Broken Access Control**](https://owasp.org/Top10/A01_2021-Broken_Access_Control/)

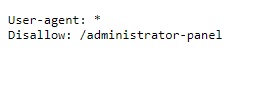
**Açıklama:** [Broken Access Control](https://owasp.org/Top10/A01_2021-Broken_Access_Control/), yetkisiz kullanıcıların erişmemeleri gereken verilere erişmesine bunları değiştirmesine veya silmesine olanak tanıyan kritik bir güvenlik açığıdır.

1. **LAB : Unprotected admin functionality**

Bu laboratuvarın korumasız bir yönetici paneli var. Zaafiyetleri Bulup Yönetici Paneline Girmemizi ve Carlos Kullanıcısın silerek laboratuvarı çözmemizi İstiyorlar.

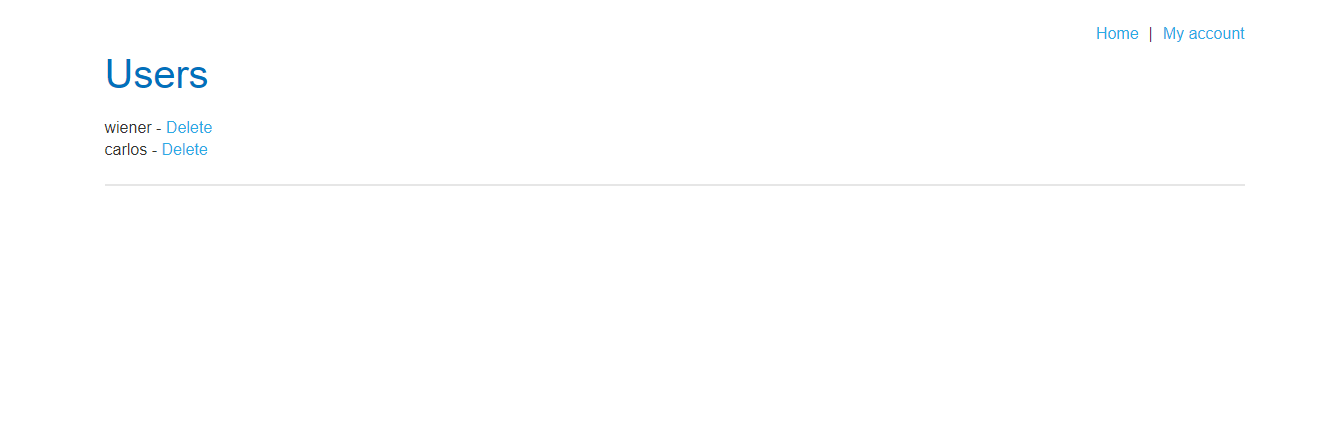
1. **Adım:**

İlk Olarak Robots.txt Dosyasını Kontrol Ediyoruz Yönetici Panelinin Yolunu Öğrenmek İçin:



1. **Adım:**

Yönetici Paneli Url’ sine Giriyoruz ve Hiçbir Login veya Güvenlik Önlemi Olmadığını Görüyoruz. Bu Sayede Kolayca Carlos Kullanıcısını Siliyoruz:

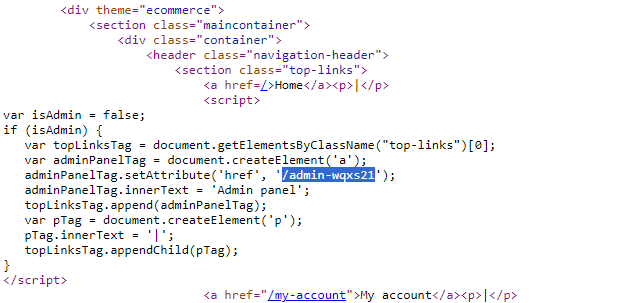


1. **LAB : Unprotected admin functionality with unpredictable URL**

Bu Labda Yönetici panelinin URL'sini içeren bazı JavaScript Kodları Bulunmakta Onları Bulup Yönetici Paneli URL’ sini Öğrenerek Carlos Kullanıcısını Silmemiz İstenmekte.

1. **Adım:**

İlk Olarak Tarayıcıda Sayfa Kaynağını Görüntülemeye Giriyoruz Kısa Yolu CTRL + U, Ve Kodlarda Javascript Kodları İçin Yönetici Paneli Urlsini Buluyoruz:

****

1. **Adım:**

Yönetici Paneli Url’ sine Gidiyoruz ve Bir Güvenlik Önlemi Olmadığını Farkediyoruz Bu Sayede Carlos Kullanıcısını Kolayca Siliyoruz:

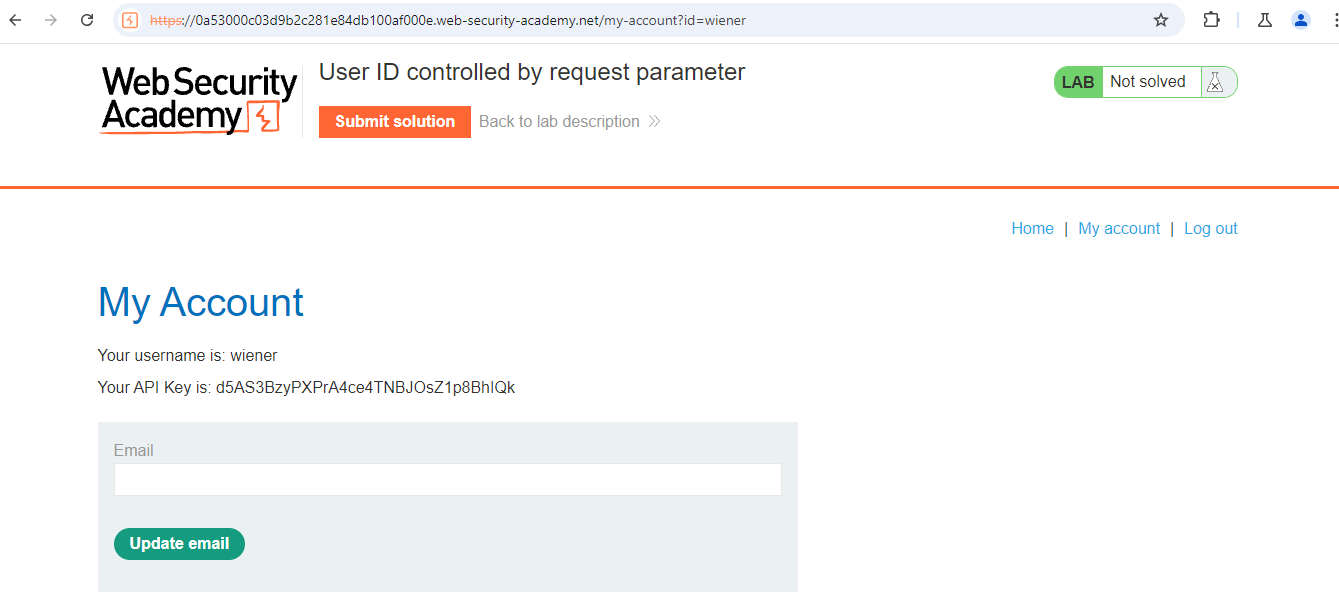


1. **Lab: User ID controlled by request parameter**

Verilen Kullanıcı Verileri İle Giriş Yaparak Url Üzerinden Başka Kullanıcı Hesaplarının Api Bilgisine Ulaşmamız İsteniyor.

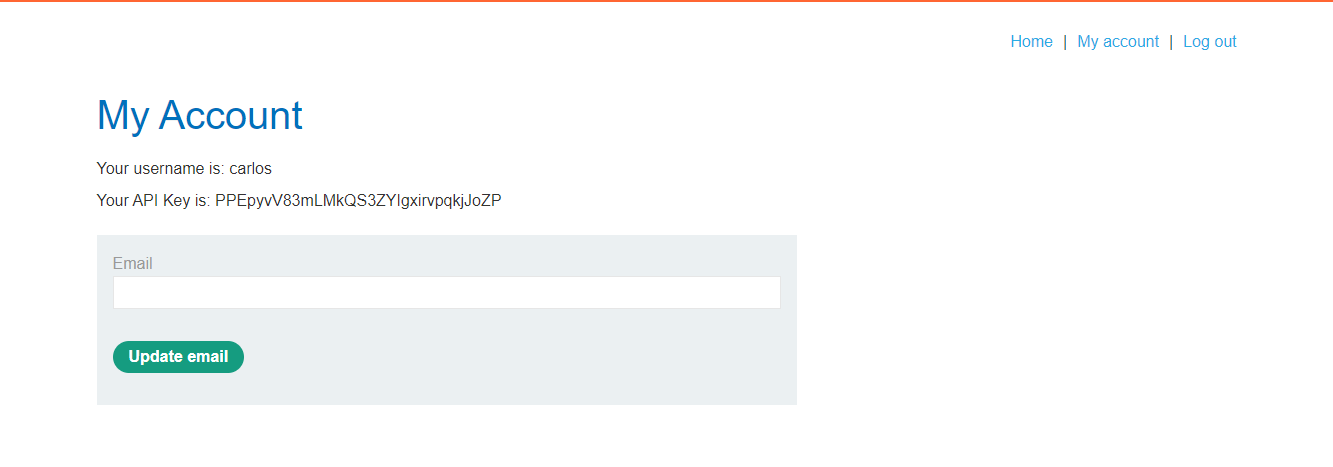
1. **Adım:**

Verilen Hesap bilgileriyle Sisteme Giriyoruz Ve Url Sonunda Get İle İd Parametresi İle Kullanıcı Adı Gönderildiğini Görüyoruz:



1. **Adım:**

Bu Paremetreyi İstenen Carlos Kullanıcı Adı İle Değiştiriyoruz ve Bu Sayede Carlos Kullanıcısının Api Bilgilerine Ulaşıyoruz:



**C. SSRF**

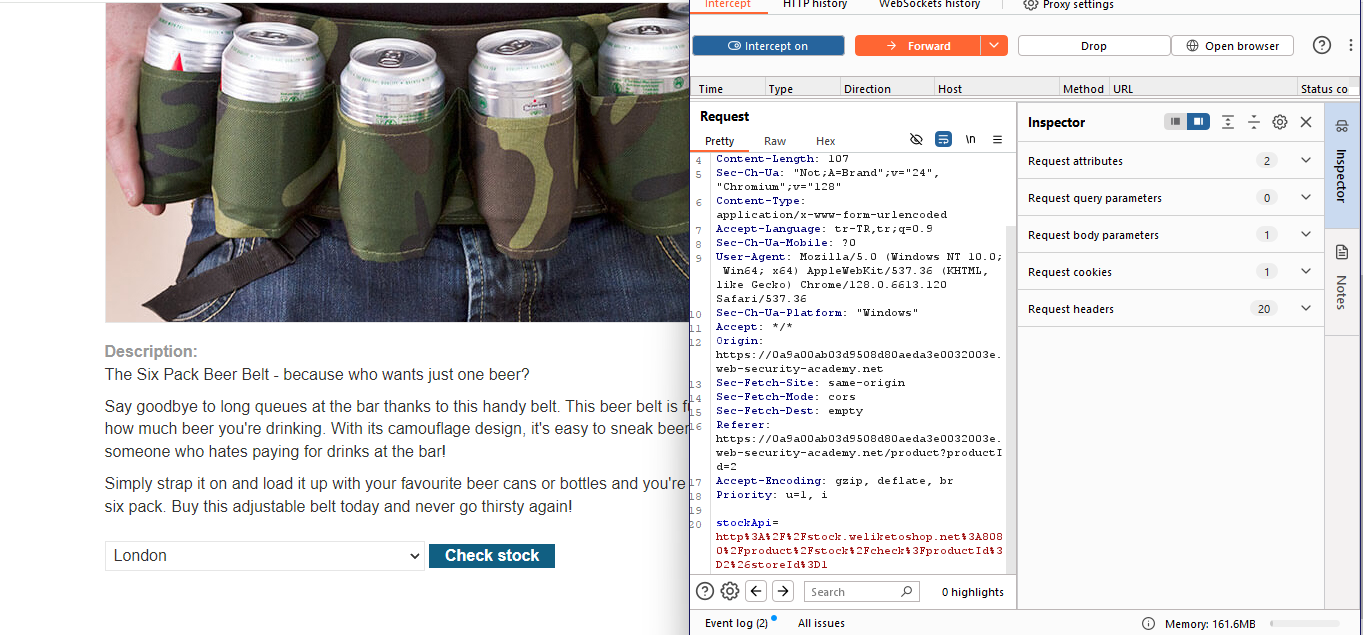
**Açıklama:** Server Side Request Forgery (SSRF) web uygulamalarının derinliklerinde yer alan güvenlik tehlikelerinden biridir. SSRF saldırısında saldırgan sızdığı sistemin sunucunun kendisi istek yapmış gibi gelen istekleri kendine yönlendirerek dış dünya ile etkileşime girmesini manipüle eder.

1. **LAB: Basic SSRF against the local server**

Bu lab da dahili bir sistemden veri getiren bir stok kontrolü özelliği var.

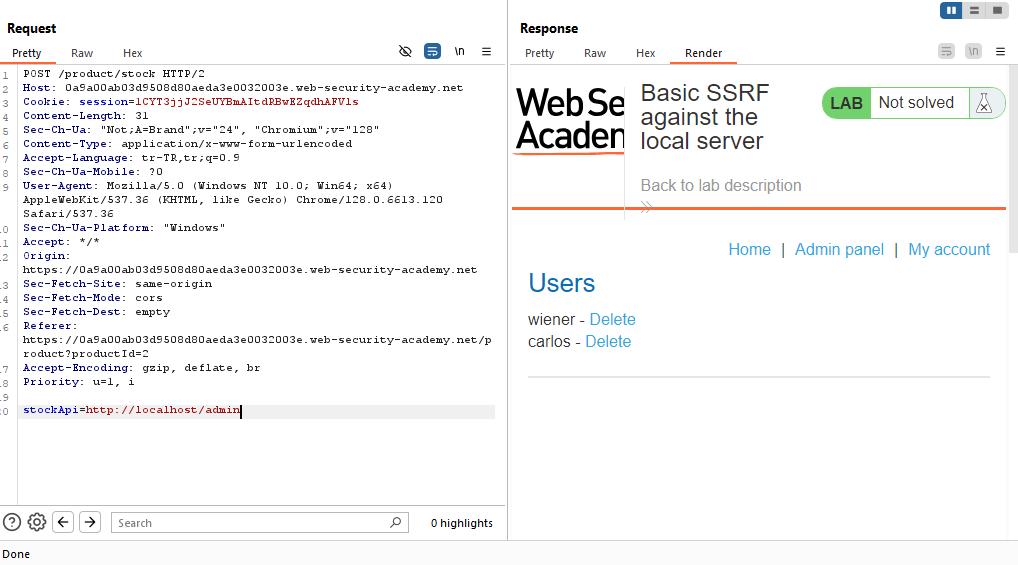
Lab’ı çözmek için http://localhost/admin adresindeki yönetici arayüzüne erişmek için stok kontrolü URL'sini değiştirerek carlos kullanıcısını silmemiz isteniyor.

1. **Adım:**

İlk Önce Bir Ürüne Girip Burp Suite İle Proxy Açıp Stok Butonuna Basıyoruz ve Gelen İsteği Yakalıyoruz:

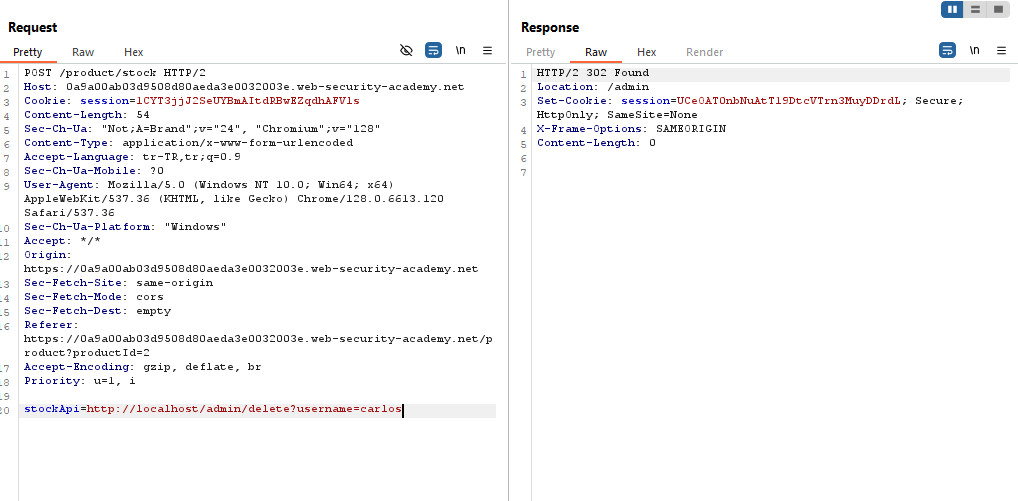
1. **Adım:**

Yakaladığımız İsteği Repeater’a Gönderiyoruz ve Stockapi Kısmını <http://localhost/admin> Olarak Değiştiriyoruz ve İstek Gönderiyoruz Bu Sayede Admin’e Erişmiş Oluyoruz:



1. **Adım:**

Gelen Çıktıdaki Carlos Kullanıcısını Silme Urlsini Yakalayıp Stockapi Verisine Yapıştırıyoruz Bu Sayede Admin Yetkisi İle Carlos Kullanıcısını Silmiş Oluyoruz:

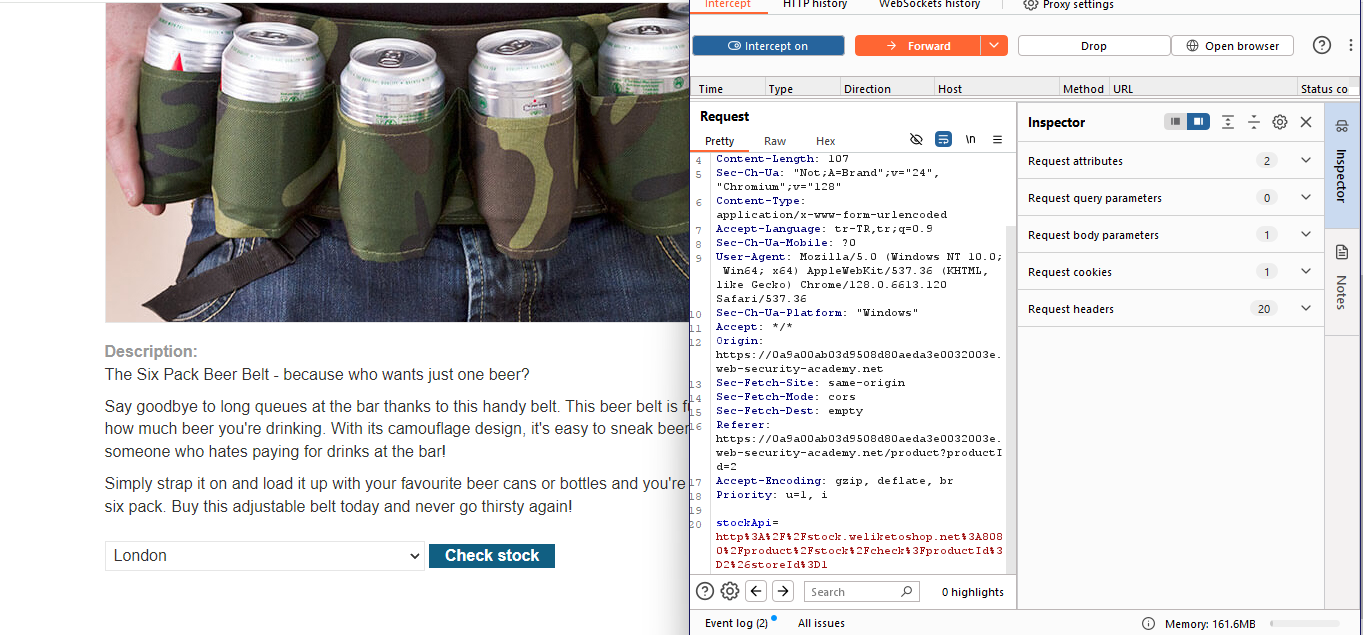


1. **LAB: SSRF with blacklist-based input filter**

Bu laboratuvarın dahili bir sistemden veri getiren bir stok kontrolü özelliği var.

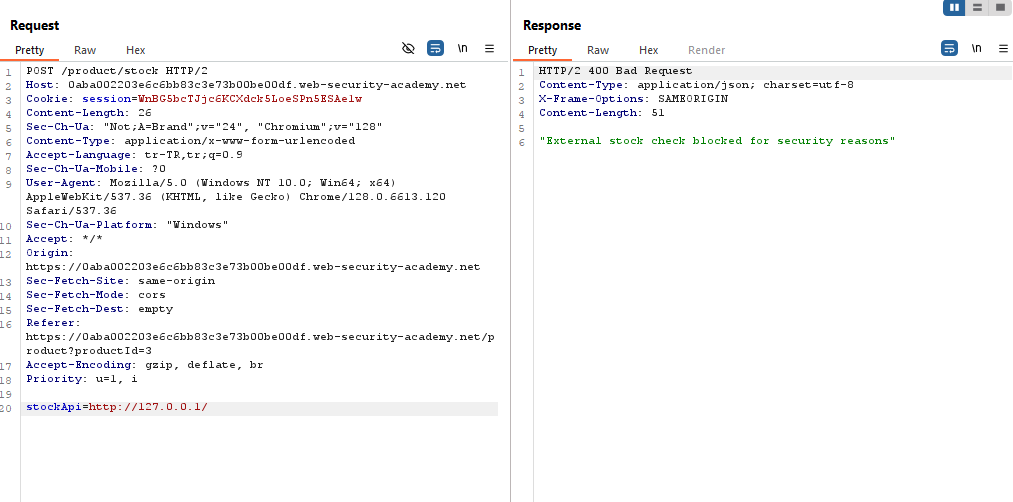
Laboratuvarı çözmek için, http://localhost/admin adresindeki yönetici arayüzüne erişmek için stok kontrolü URL'sini değiştireceğiz ve carlos kullanıcısını sileceğiz.

1. **Adım:**

İlk Olarak Bir Ürüne Girip Burp Suite İle Proxy Açıp Stok Butonuna Basıyoruz ve Gelen İsteği Yakalıyoruz:

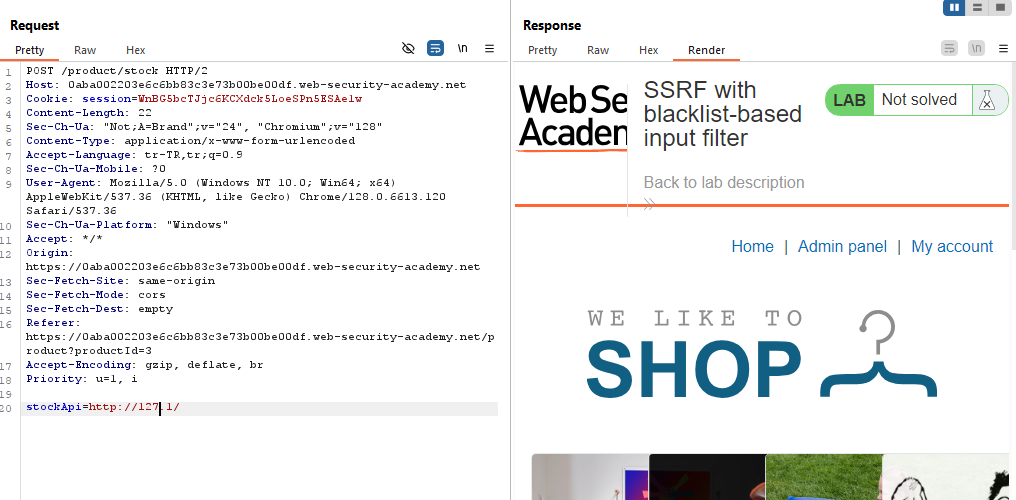
1. **Adım:**

Gelen İstekteki Stockapi Verisini 127.0.0.1 Yapıyoruz Fakat Blocklanmış sayfa ile karşılaşıyoruz:



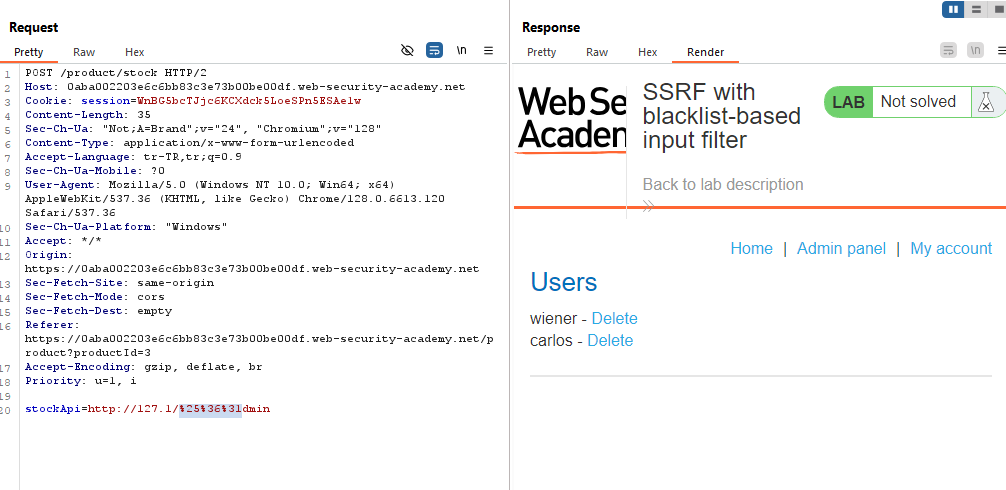
1. **Adım:**

Bu sefer İsteği 127.1 yaparak Girmeyi Deniyoruz ve Başarılı Bir Şekilde Giriş Yapıyoruz:



1. **Adım:**

Şimdi de 127.1/admin isteği atarak admin panele erişmeye çalışıyoruz ve başarılı bir şekilde erişiyoruz fakat istek gönderirken admin in a harfini 2 kez art arda url all chracter encode ile şifreleyerek gönderiyoruz:



1. **Adım:**

Gelen Responstaki Carlos Kullanıcısını Silme Urlsini Kopyalayıp istek Sonuna Ekliyoruz ve Bu Sayede Başarılı Şekilde Kullanıcıyı Siliyoruz:

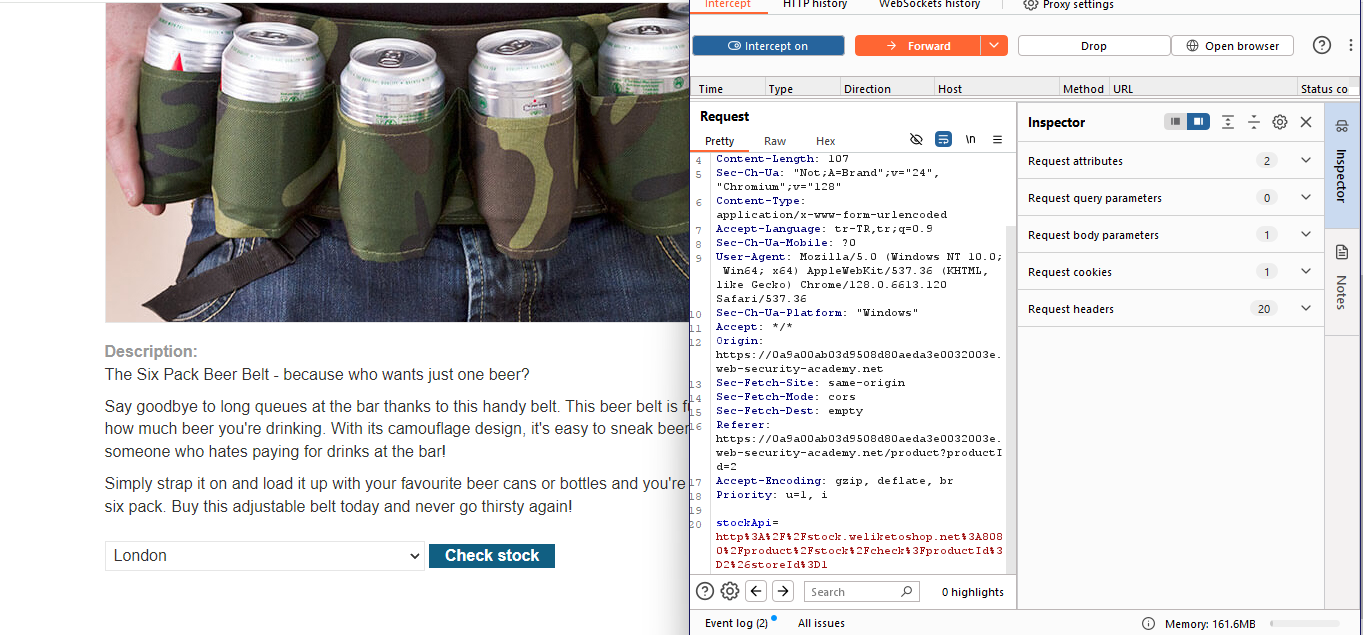


1. **LAB: Basic SSRF against another back-end system**

Laboratuvarı çözmek için, stok kontrol işlevini kullanarak dahili 192.168.0.X aralığı 8080 portunda bir yönetici arayüzü için tarama yapacağız ve ardından bulduğumuz ip üzerinden Carlos kullanıcısını sileceğiz.

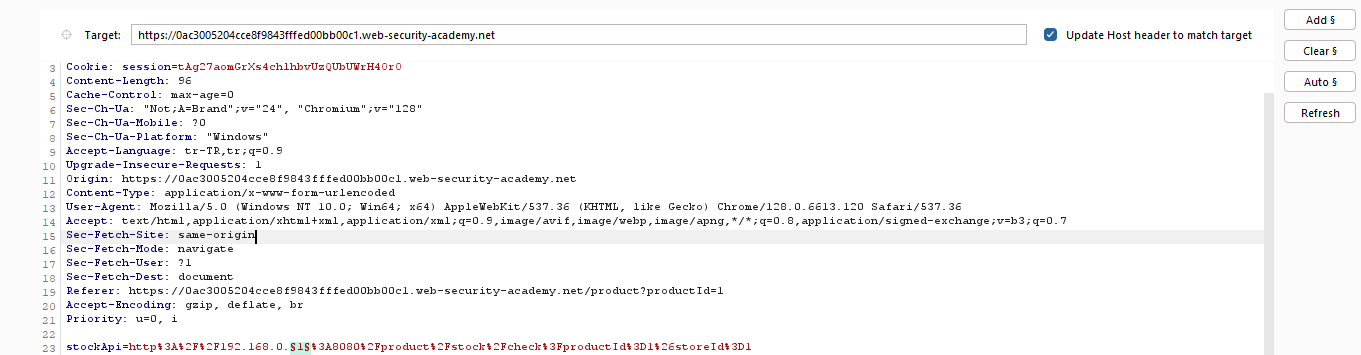
1. **Adım:**

İlk Olarak Bir Ürüne Girip Burp Suite İle Proxy Açıp Stok Butonuna Basıyoruz ve Gelen İsteği Yakalıyoruz:



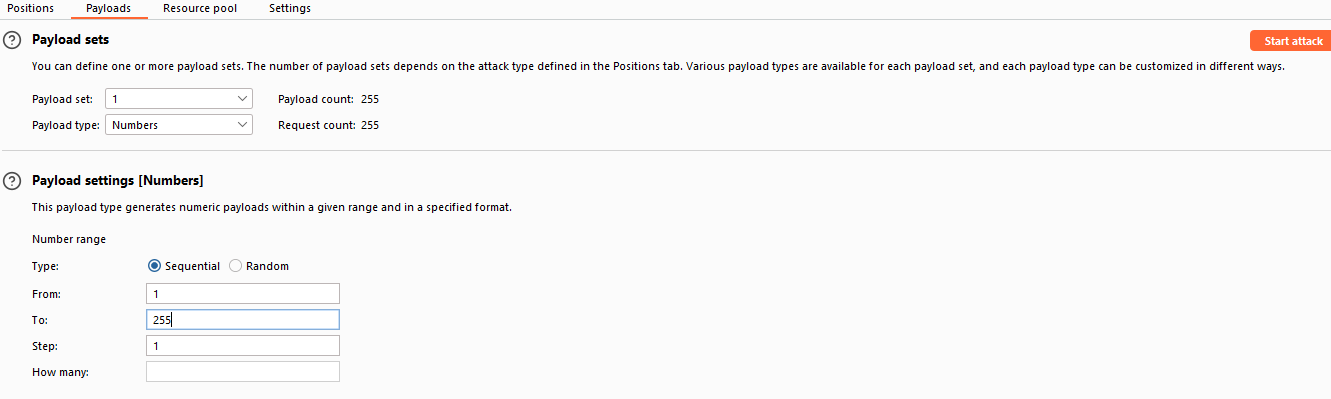
1. **Adım:**

Bu İsteği İntrudere Gönderip ip sonundaki 1 numarısını “Ekle §” ye tıklayarak vurguluyoruz:



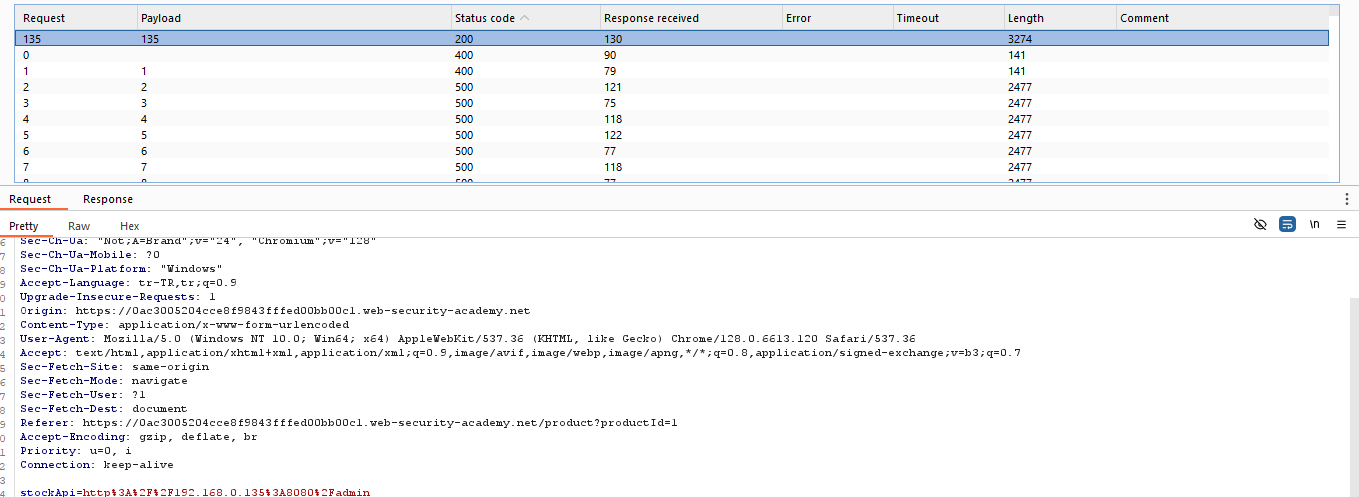
1. **Adım:**

Ardından Payloads Kısmına Geçerek Payload Tipini Numbers Olarak Seçiyoruz ve From Verisini 1 To Verisini 255 Giriyoruz ve Atağı Başlatıyoruz:



1. **Adım:**

Atak Bittikten Sonra Status Code si 200 Olan İsteği Buluyoruz ve O İsteği İncelediğimizde Admin Panele Eriştiğimizi Görüyoruz. Sonrasında Bu İsteği Repeatera Gönderiyoruz:



1. **Adım:**

Gelen Responstaki Kullanıcı Silme Urlsini repeater ile gönderiyoruz ve bu sayede kullanıcıyı başarılı bir şekilde silmiş oluyoruz:

