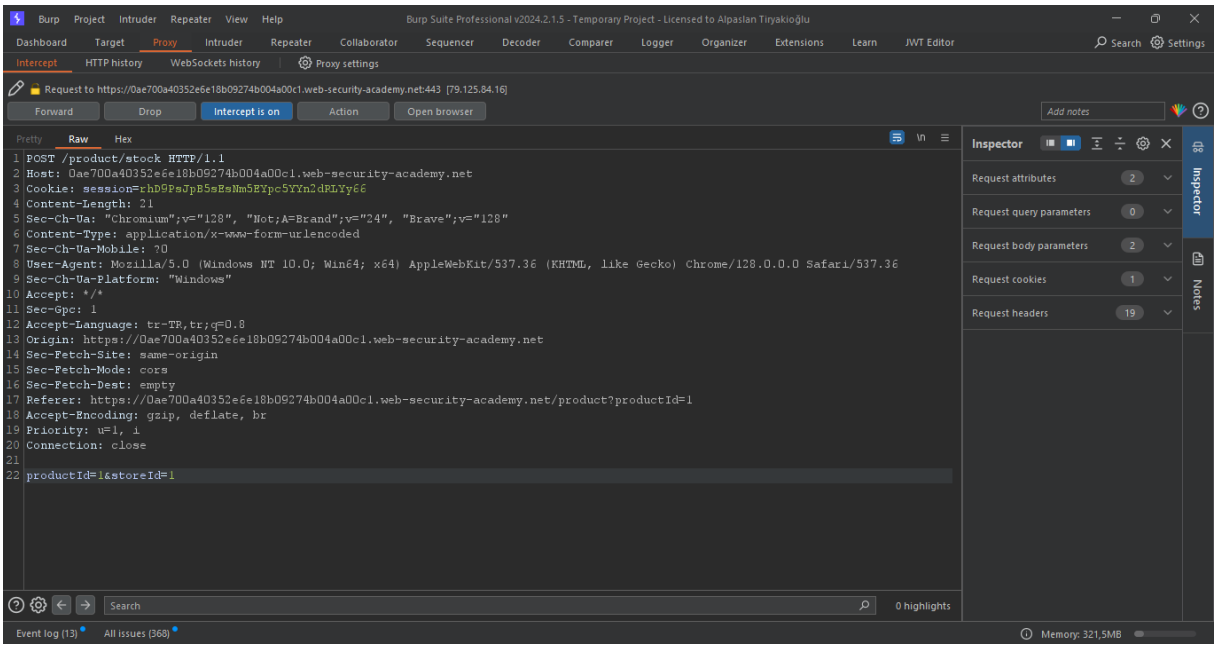
Injections

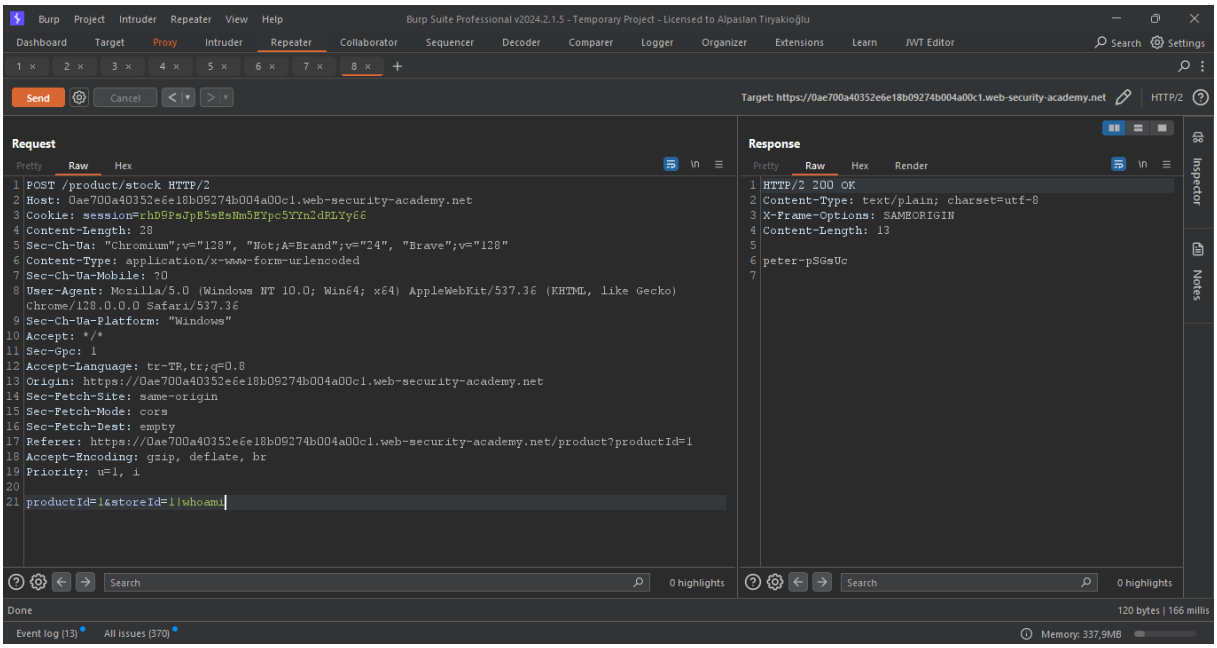
LAB_1: OS command injection, simple case

Bu laboratuvarda ürün stok kontrolünde OS komut enjeksiyonu bulunmaktadır. Uygulama, kullanıcı tarafından sağlanan ürün ve mağaza kimliklerini içeren bir shell komutunu çalıştırır ve yanıtında komutun çıktısını döndürür. Laboratuvarı çözmek için *whoami* komutunu çalıştırıp son kullanıcı ismini bulmamız gerekiyor.

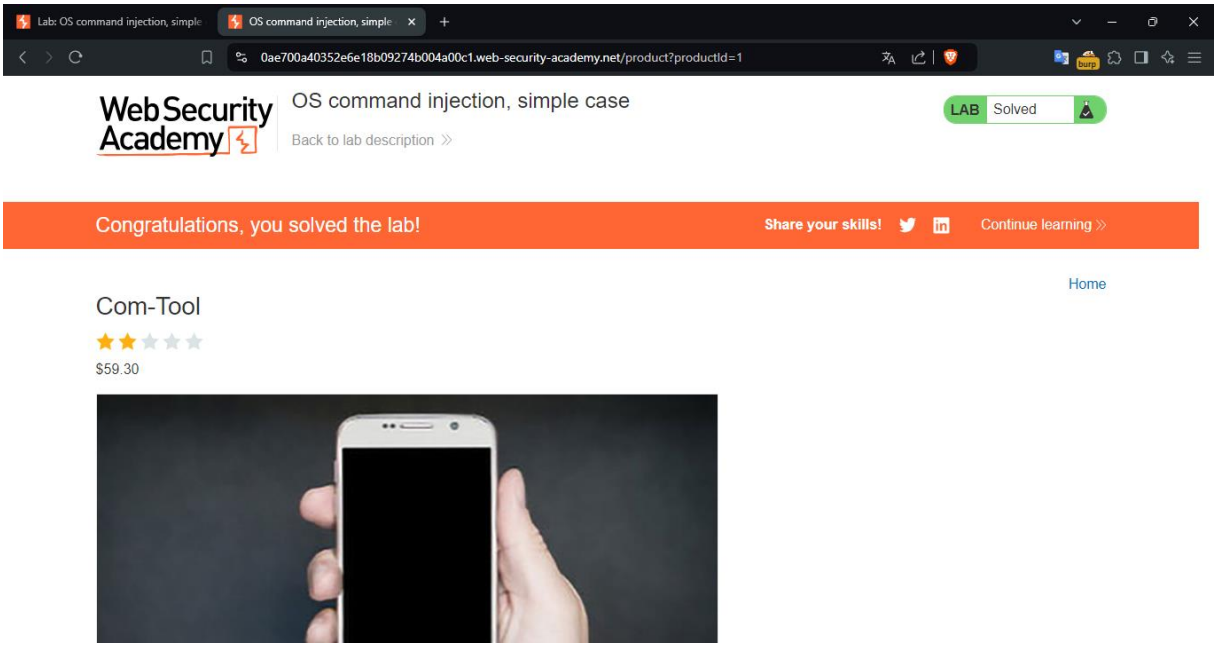
Laboratuvara eriştikten sonra bahsi geçen fonksiyon için bir ürüne giriş yapıyorum.



Buradaki iki değere de bir komut yazıyorum. Tabi öncesinde repeater aracına gönderiyorum.



Komut enjeksiyonunda http paketinde giden değer sunucu tarafında bir betikte doğrudan çalıştırıldığından bu zafiyet ortaya çıkar. Bizde bunu istismar etmek için burada göründüğü gibi düzenlemeler yaparız. |, ||, &, && gibi operatörler ile komutlar birbirlerine bağlanır. Bunları deneyerek bulabiliriz. Ben görsel kalabalığı olmaması için doğru operatörü ve doğru parametreyi buldum. Sonuç olarak dönen değeri görüyoruz.



Web sayfa tarafında da laboratuvarın çözüldüğünü görebiliriz.

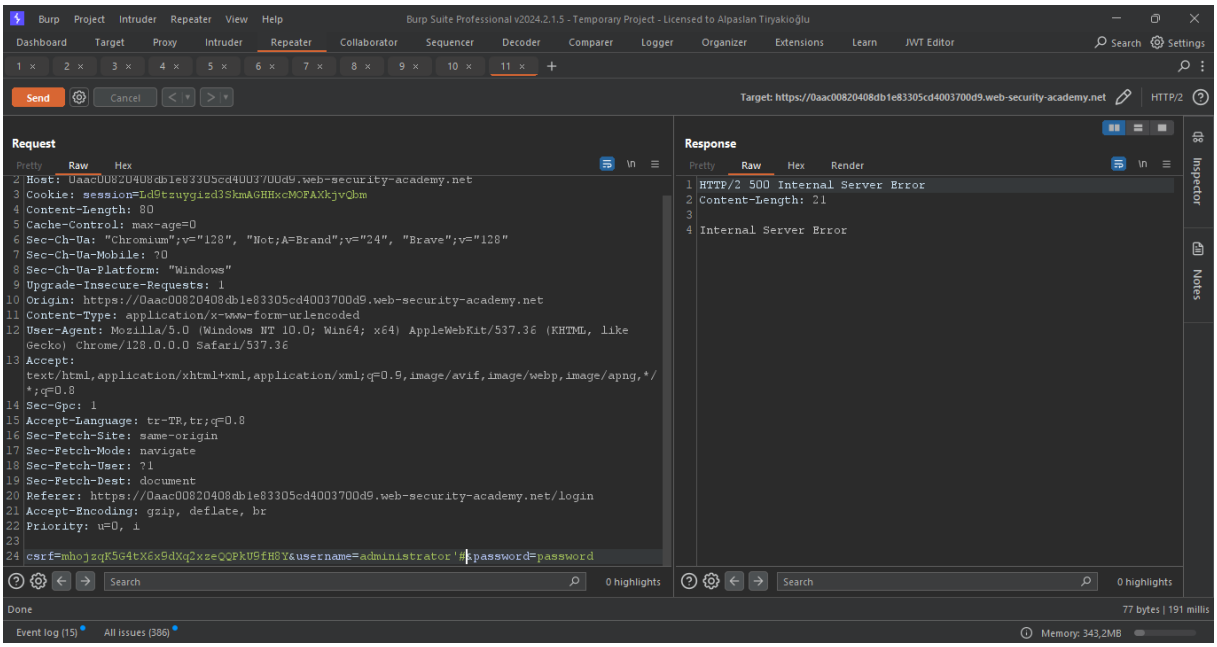
LAB_2: SQL injection vulnerability allowing login bypass

Bu laboratuvarda bir SQLi zafiyeti bulunduğu belirtilmiştir. Çözüm için *administrator* kullanıcısı ile oturum açmamız isteniyor.

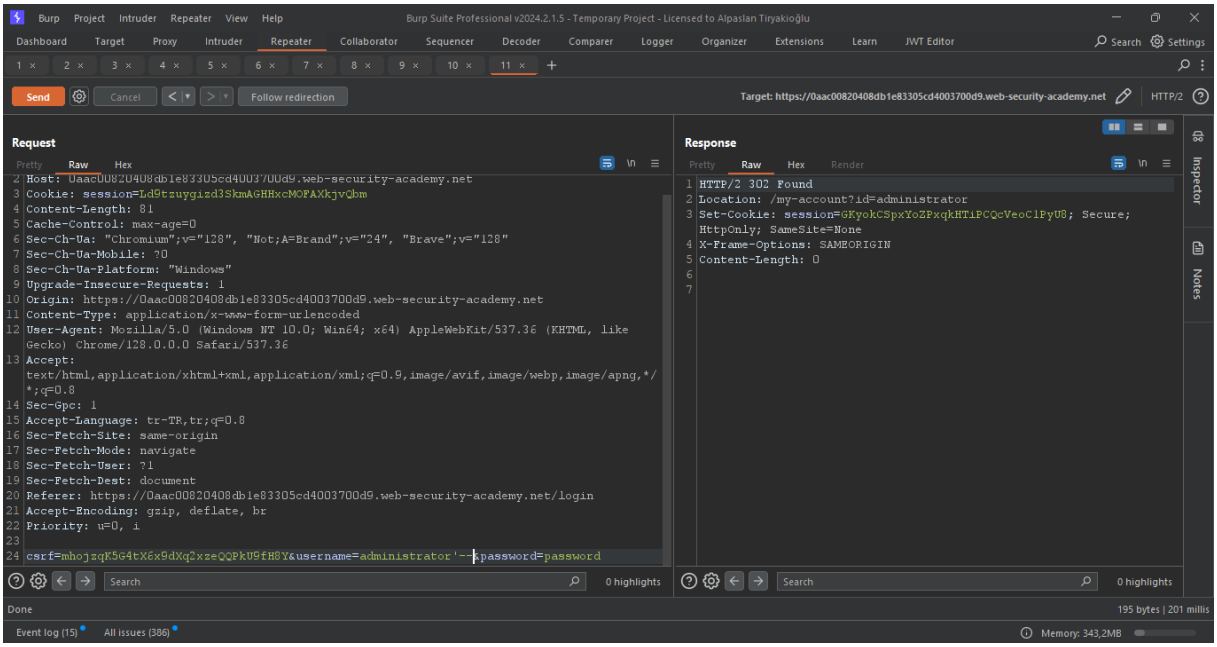
Buradaki isteğe baktığımızda akla ilk gelen şeylerden birisi kullanıcı adını girip sorgunun geri kalanını yorum satırı haline getirmek olabilir. Bunu denedikten sonra başka çözüm yollarını araştırabiliriz.

İlk başta # işareti ile yorum satırı yapmayı deniyoruz. Bu arada username kısmından çıkmak için ‘ işaretini koymayı unutmamız gerekir.

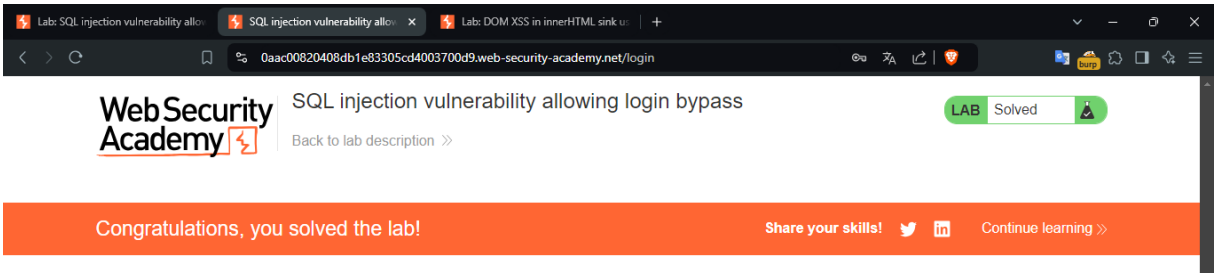
HTTP paketini yakalayıp repeater aracına gönderdim ve incelemeleri düzenlemeleri her işlemi buradan yağacağım.



Bu şekilde daha net görebiliriz. # işaretini kabul etmedi. Sunucu hatası ile karşılaştık. Bir diğer yöntem – işaretini koymak.



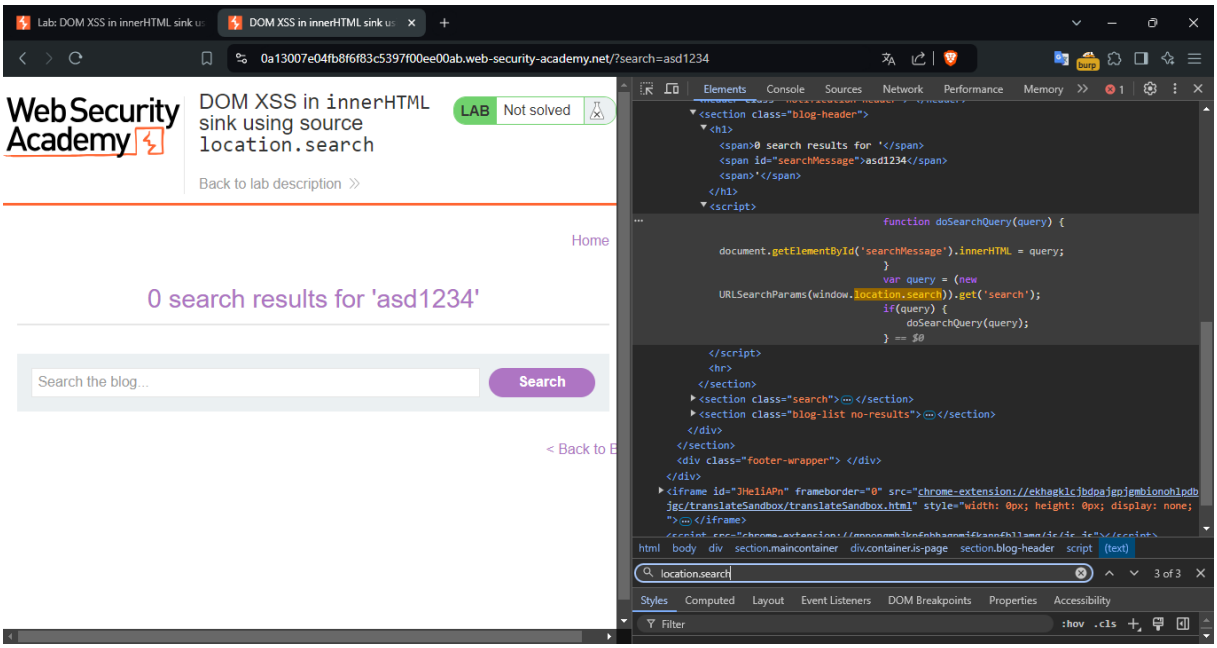
Bu payload ile yönlendirme almayı başardık. Follow redirection diyerek yönlendirmeye gidiyorum.



Yönlendirmeden sonra laboratuvarın çözüme kavuştuğunu görebiliriz.

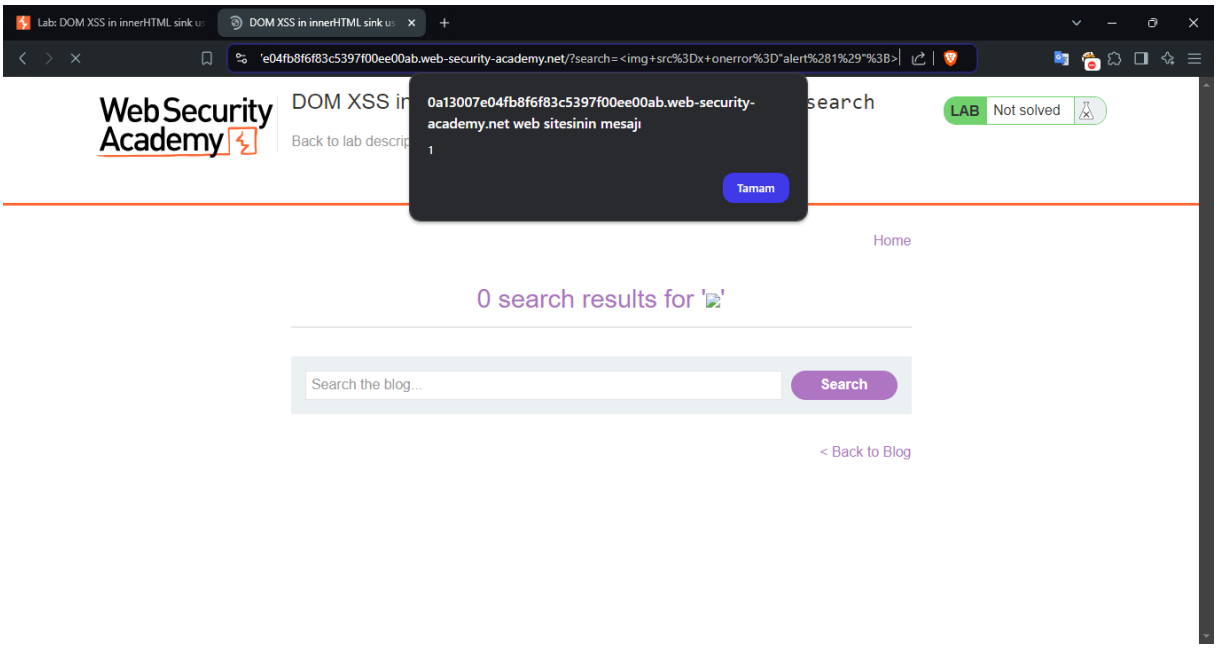
LAB_3: DOM XSS in *innerHTML* sink using source *location.search*

Bu laboratuvarda blog arama fonksiyonunda DOM Based XSS zafiyeti bulunmaktadır. location.search kısmındaki ögeyi alıp bir div içeriini değiştiren innerHTML ataması çalışmaktadır. Çözüm için alert fonksiyonunun çağırılması istenmekte. Hemen laboratuvara giriş yapalım.



Direkt olarak bahsi geçen alana gittim. HTML kodu içerisindeki JS kodunu buldum. Burayı inceledikten sonra işleyişi anladık. Artık burada uygun bir payload bulmamız gerekmekte.

Yaptığım denemelerde <script> tagını kabul etmediğini tespit ettim. Bu yüzden farklı bir tag deneyeceğiz. İlk akla gelen iframe, img gibi etiketler. Bunları deneyebiliriz.



Deneme yanılma yöntemi ile bulduğum 3 payload'da işlemi başarıyla gerçekleştirdim.