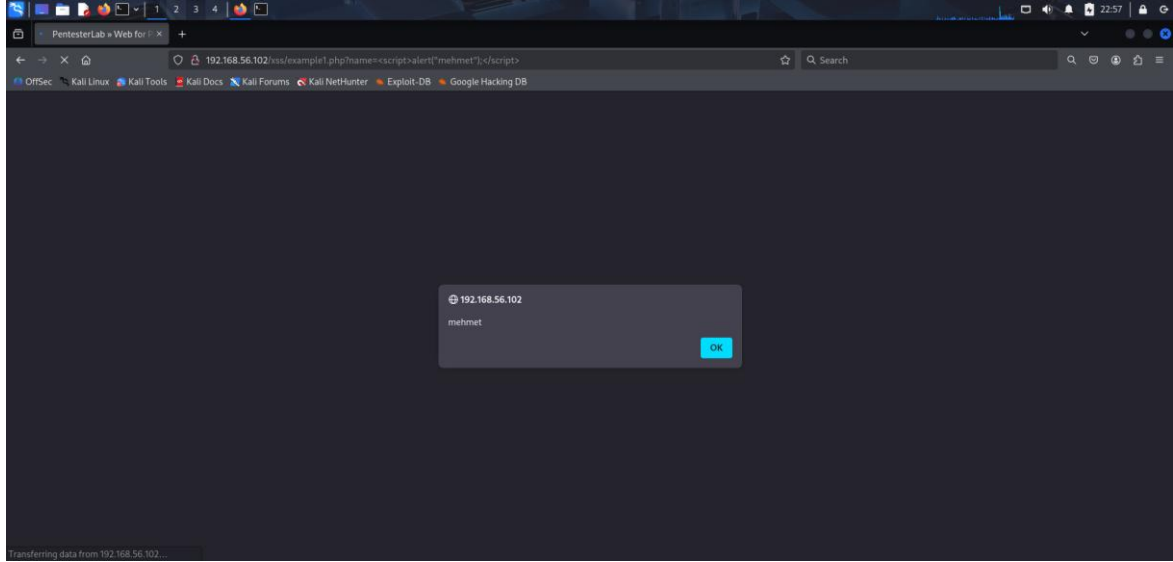


WEB FOR PENTESTER-1 ÇÖZÜMLERİ:

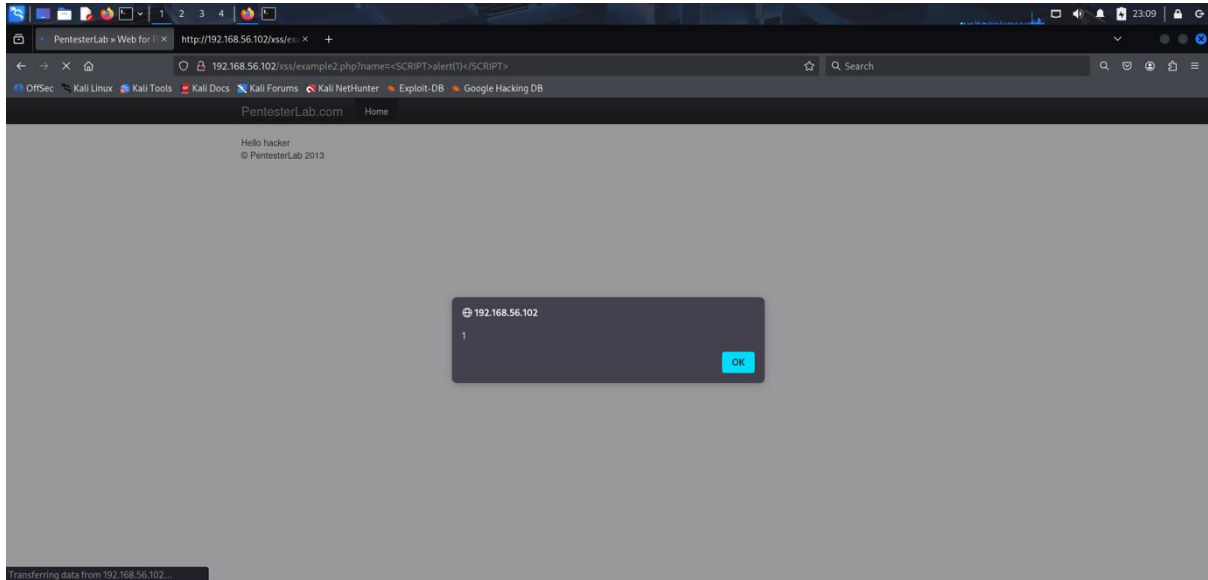
XSS 1. örnek: aklıma direk scriptin içine yazıp göndermek geliyor deniyorum



ve

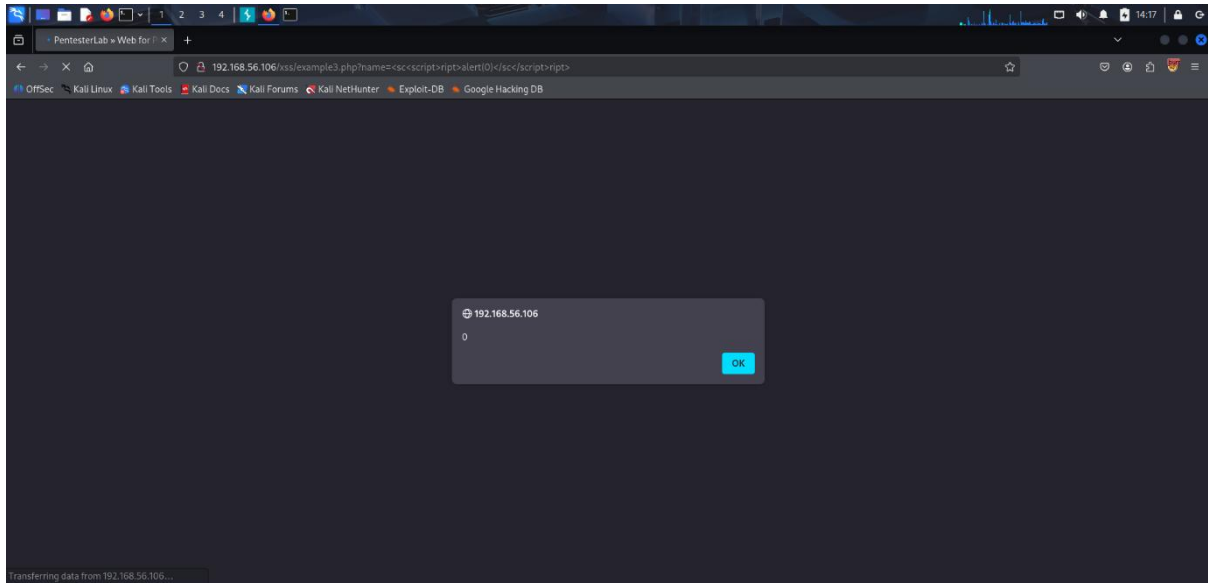
oluyor.

XSS örnek 2: aynı şeyi deniyorum olmuyor ve script in filtrelendiğini anlıyorum bu sefer farklı bir yol olarak büyük harflerle deniyorum ve oluyor.

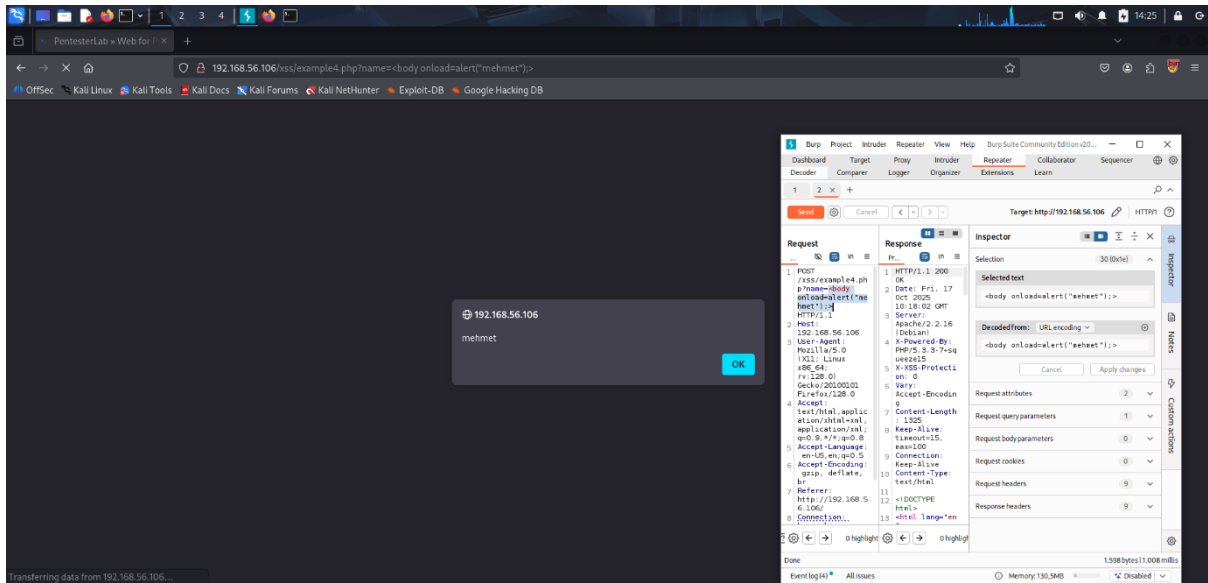


Kaldığım yerden xss 3. örnekten devam ediyorum.

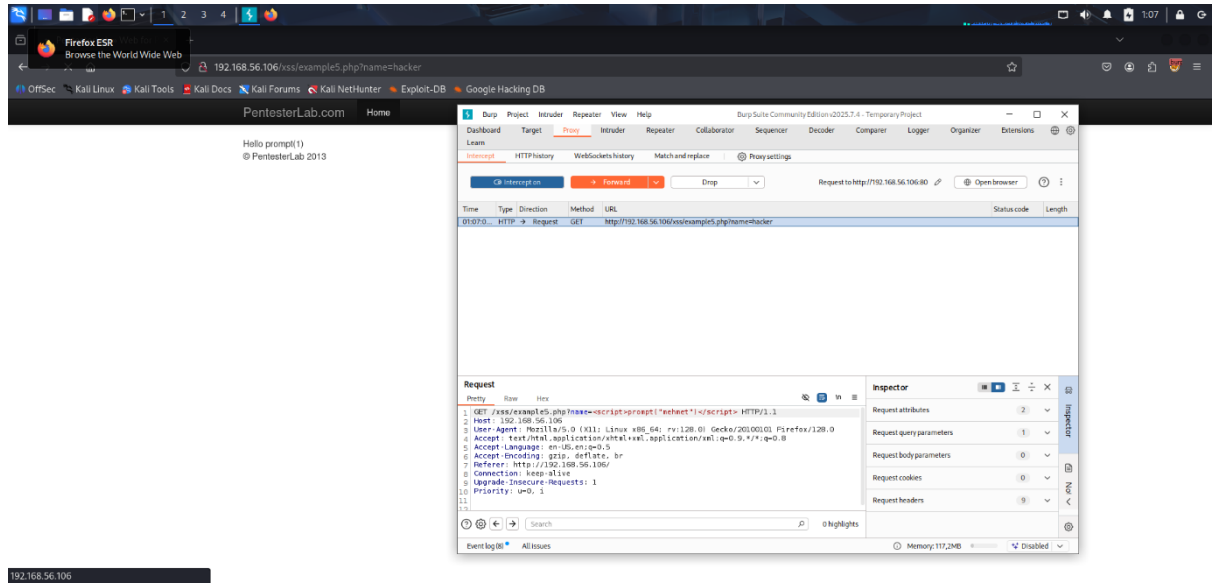
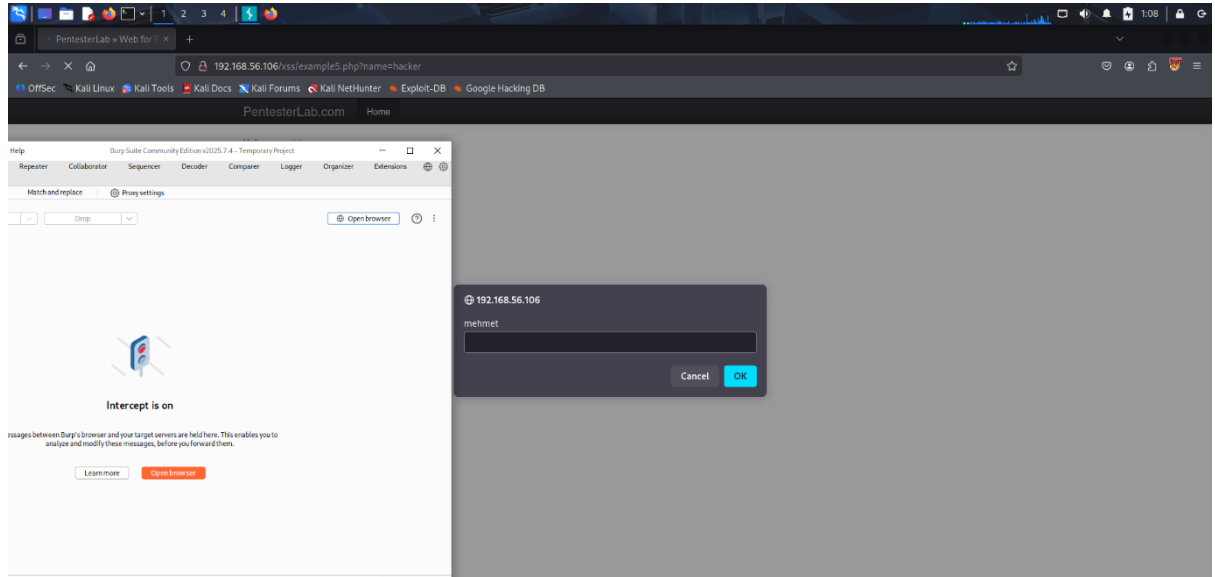
-Bir önceki payloadı denediğimde iş yapmıyor ve ekrana alert(0) yazıyor kaynak kodu inceliyorum. Dışarıdaki harfler birleşip script kodunu oluşturuyor. Dışardaki tüm scriptleri siliyor ama iki kere yapamıyor işte zafiyet burda.



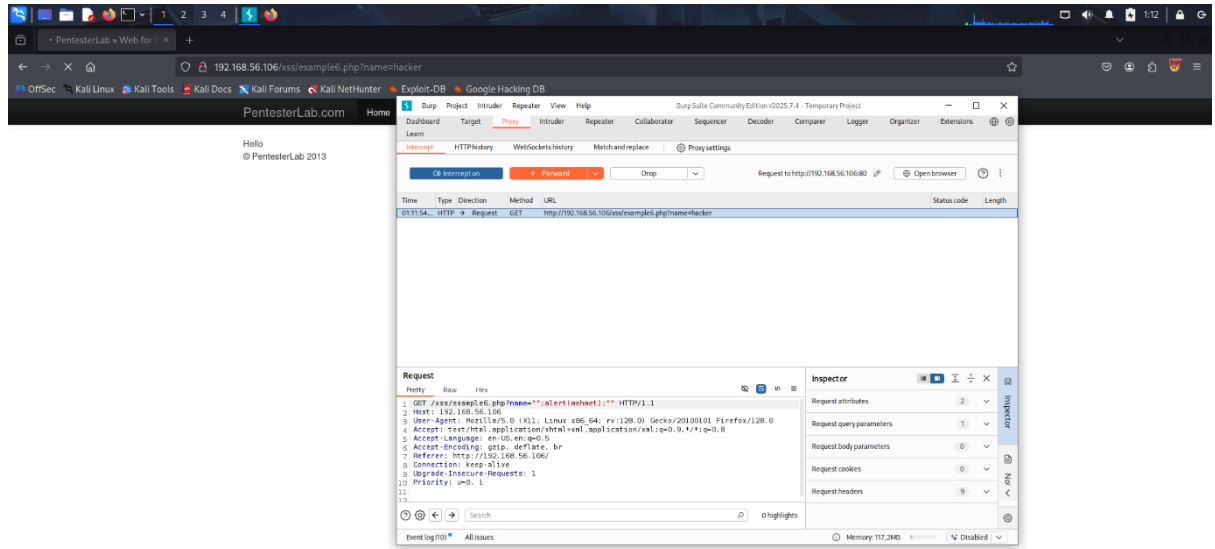
XSS 4. örnek: Burada kaynak koda script kelimesinin arandığının ve bulunca die kodunun çalıştırıldığını görüyoruz yani artık script ile işimiz yok.Bu yüzden onload deniyorum ve çalışıyor.



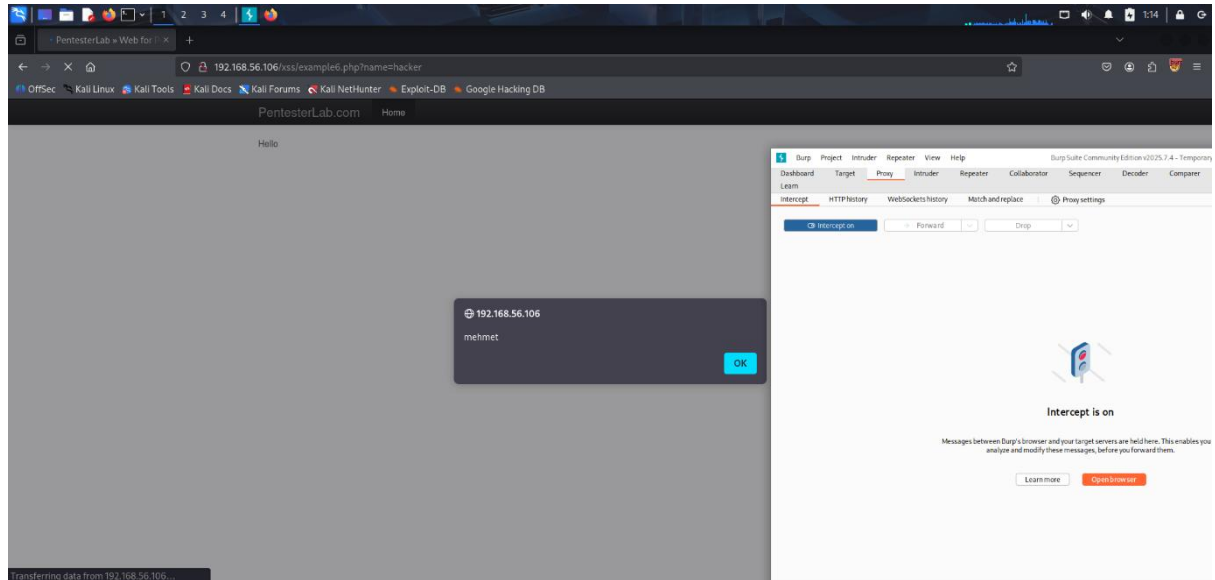
XSS ÖRNEK 5: Kaynak koda bakınca script kodunun çalışınca die fonksiyonunun devreye girdiğini görüyoruz. Bu yüzden scriptten farklı bir payload kullanmam gerekiyor burada onload benim işimi çözüyor.



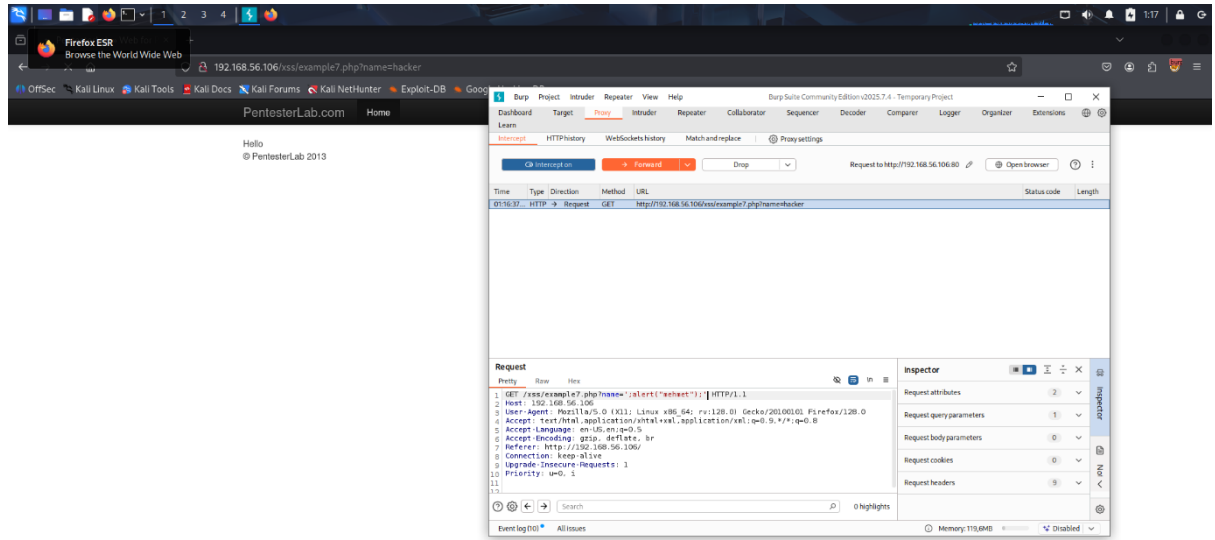
Xss örnek 6: kaynak kodda bizim değişkenimizi başka değişkene atadığını görüyorum önce scriptle değişkeni kapatıp promptla yazdırıyorum.



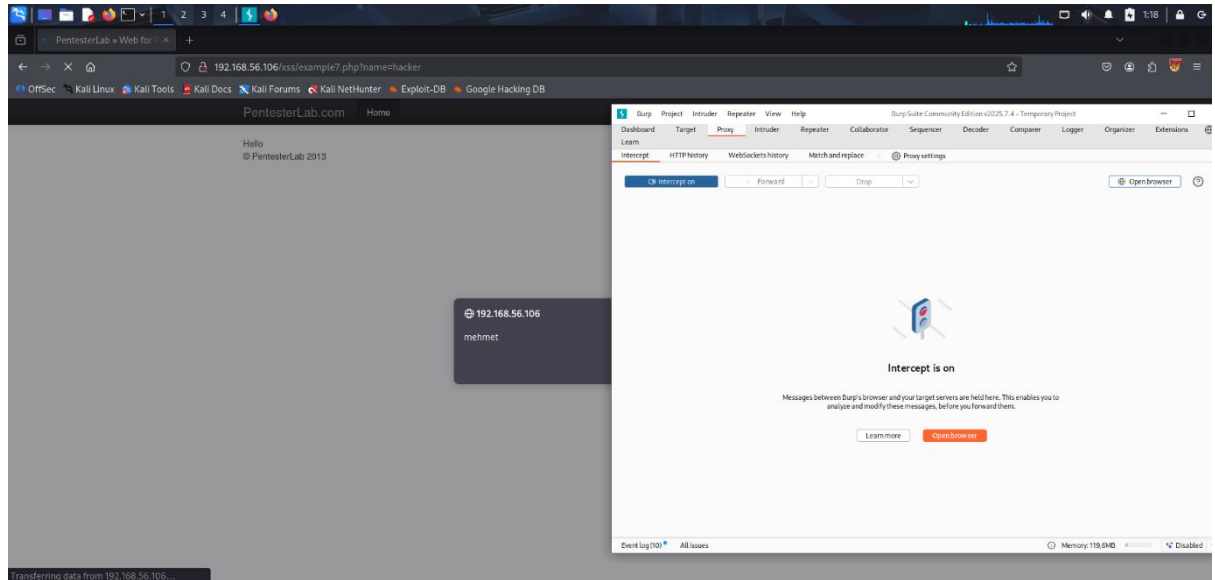
192.168.56.106



Xss örnek 7: kaynak kodda gelen parametre scriptin içine yazılıyor o yüzden script ile değişken atamasına atlatmamıza gerek yok onun yerine farklı yoldan atlatıyoruz.

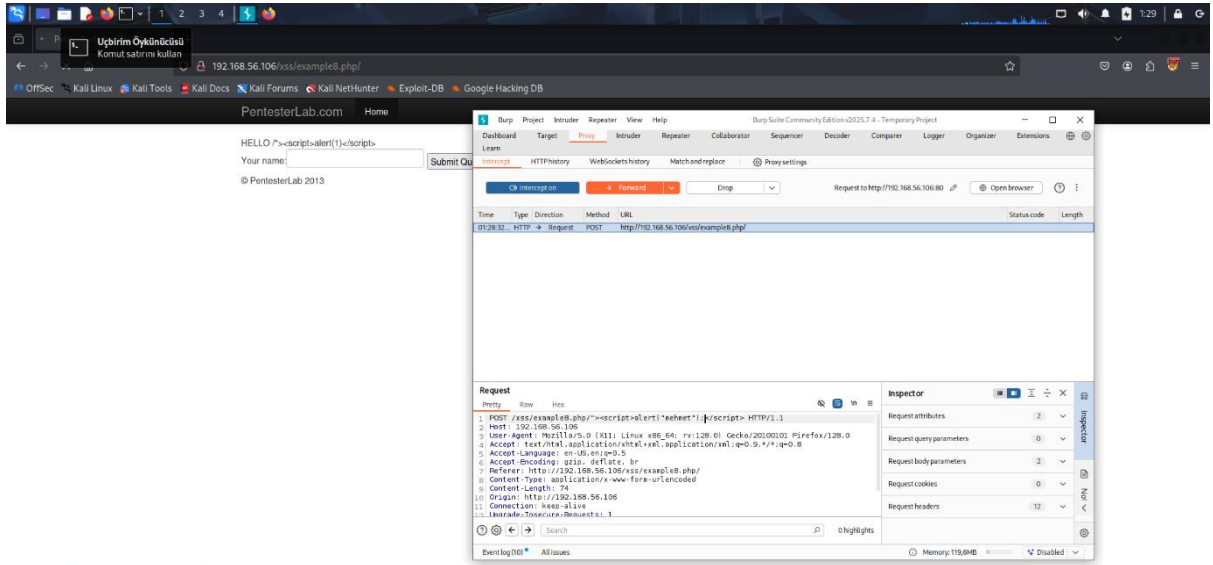
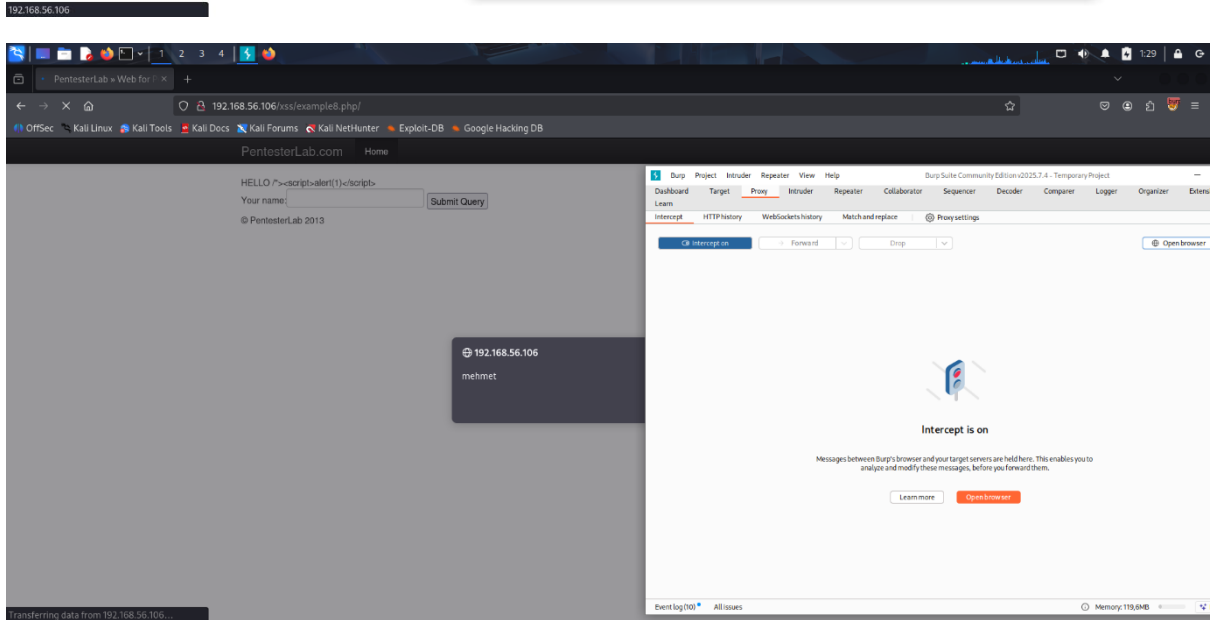


192.168.56.106



Xss örnek 8:htmlentities fonksiyonu kullanılıyor yani encode yapılıyor php_self fonksiyonunun gelen değeri url adresine post ettiğini öğreniyorum ve doğru payload ile sorunu

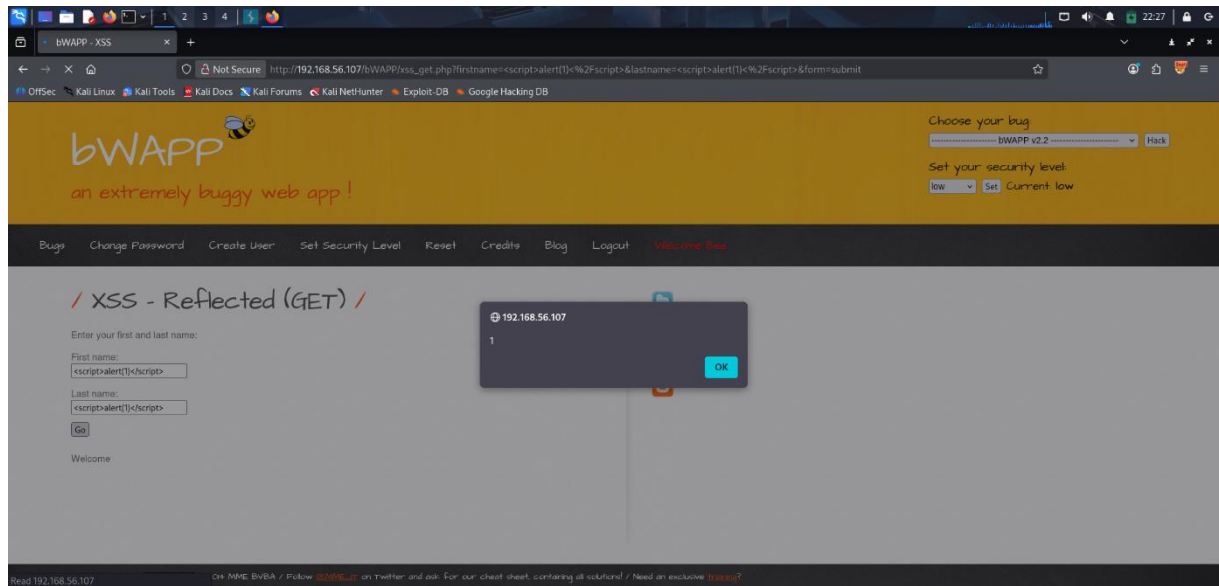
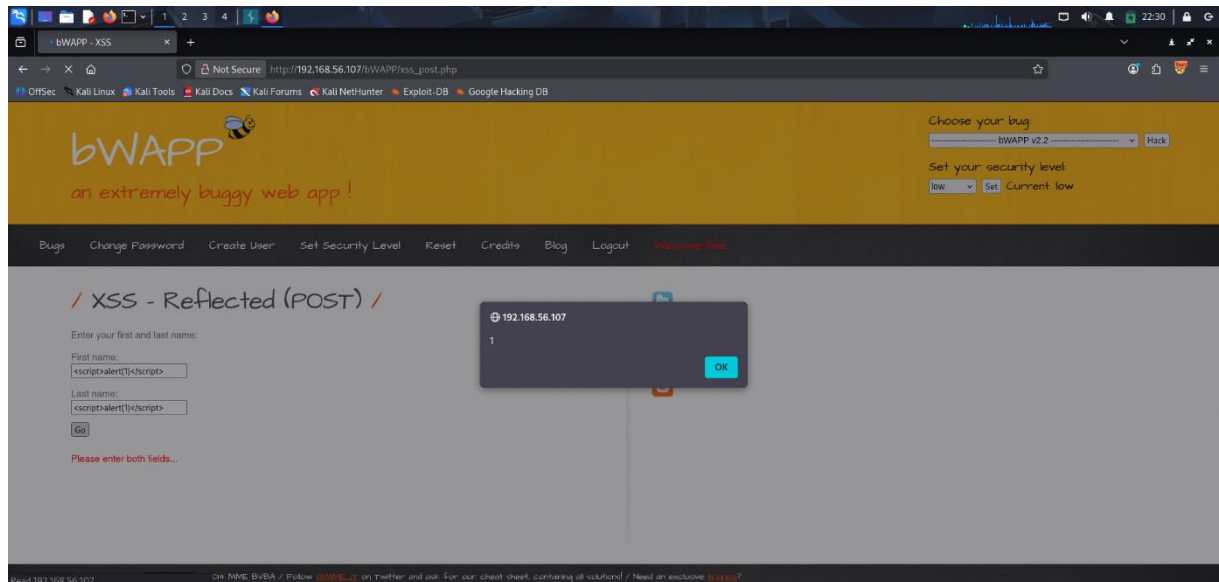
çözüyorum.



BWAPP ÇÖZÜMLERİ:

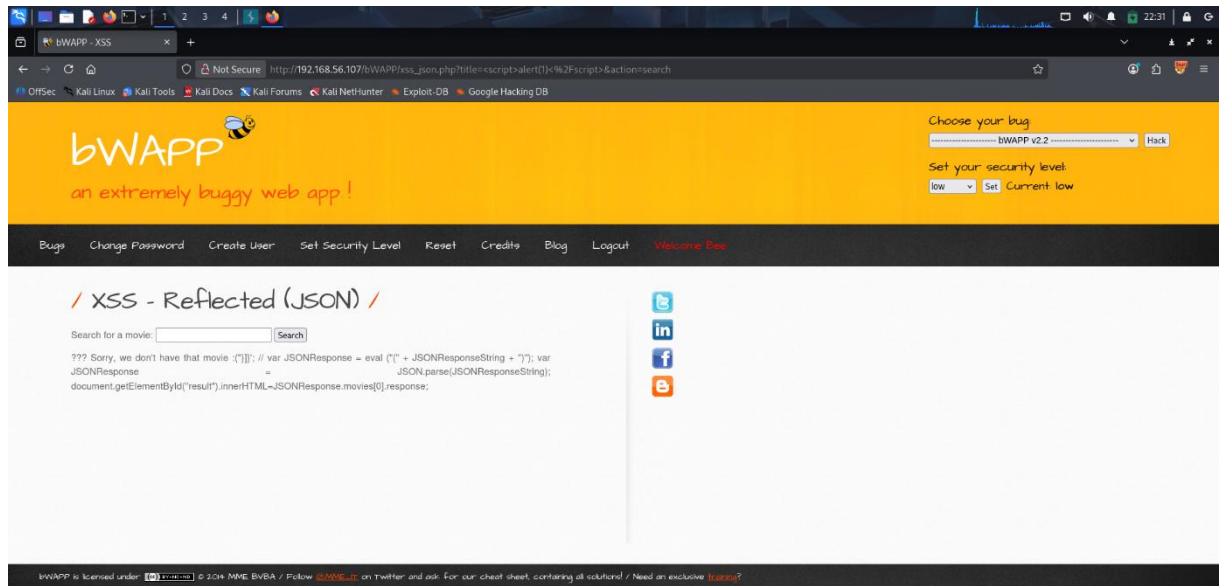
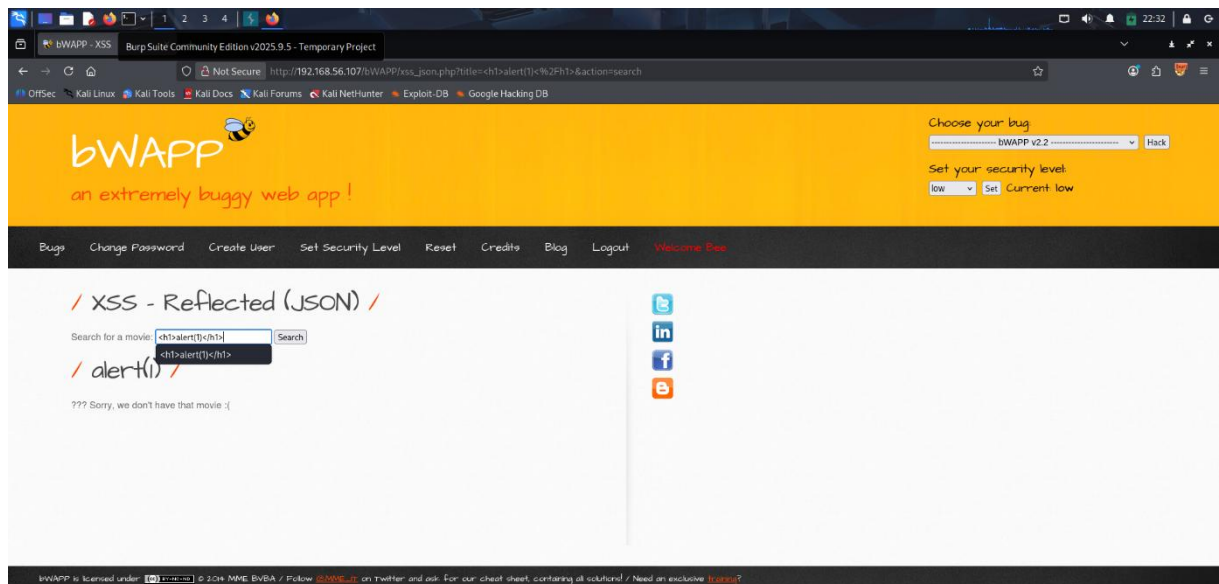
XSS REFLECTED(GET/POST):

İki örnekte de klasik olan xss payloadımızı yani `<script>alert(1)</script>` i şifre ve username kısmına yazınca pop up alıyoruz.



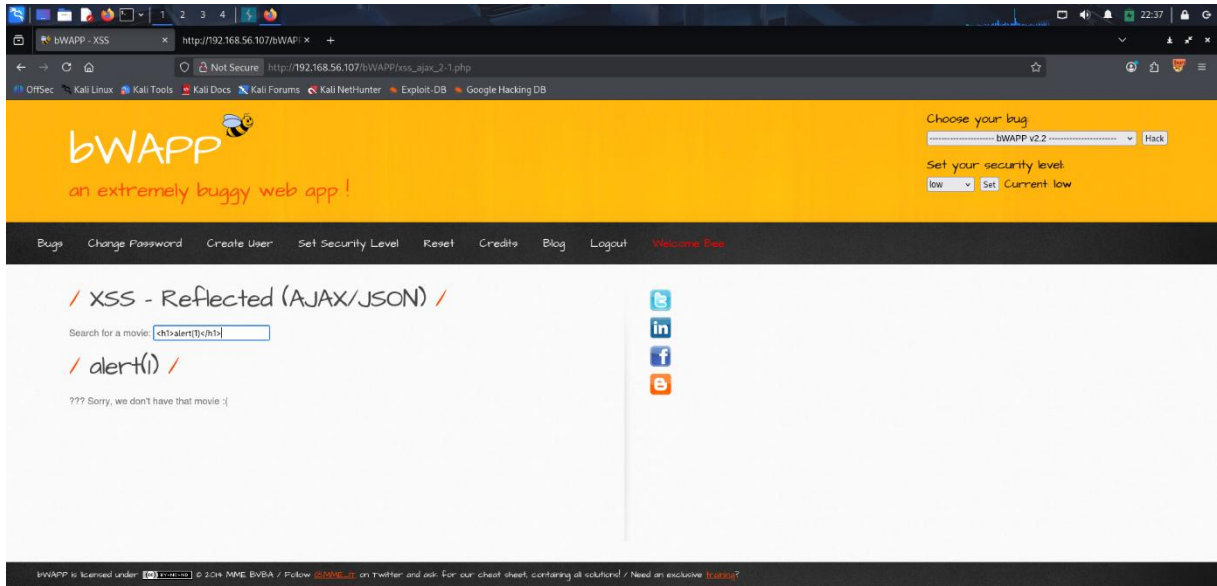
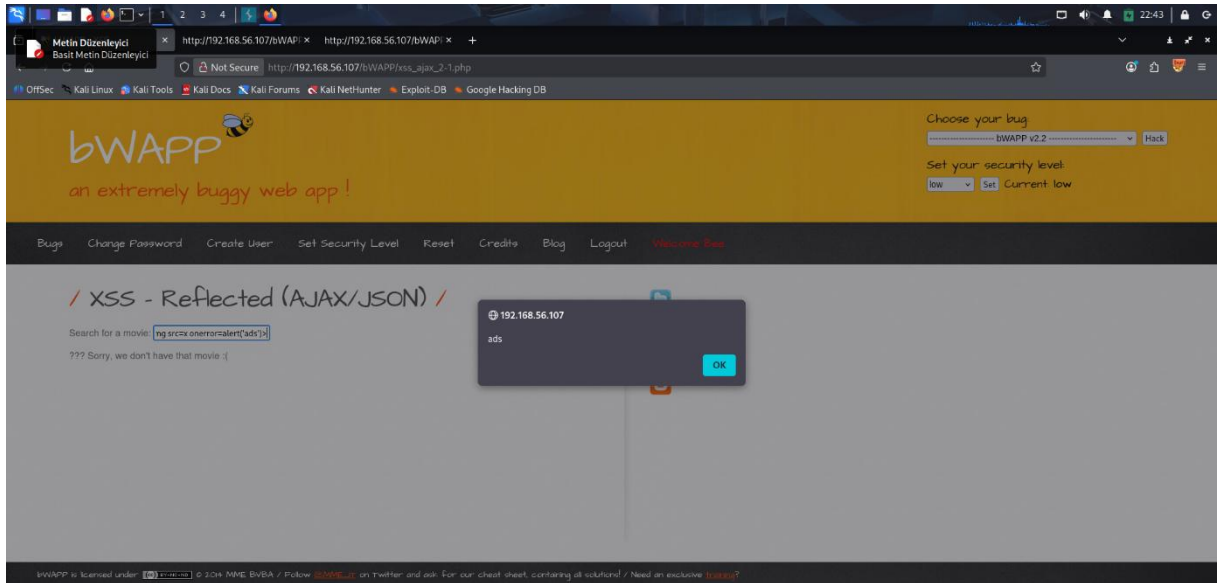
XSS REFLECTED(JSON):

Burada `<script>` payloadımız çalışıyor h1 payloadı ile ekrana yazdırabilir veya onerror payloadı ile pop up alabiliriz.



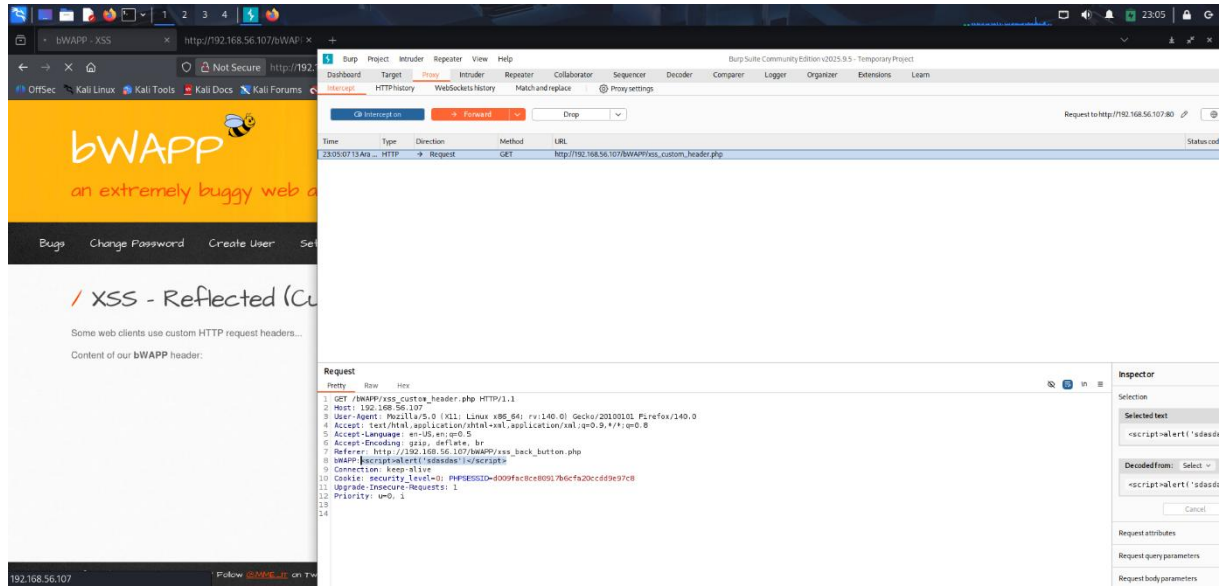
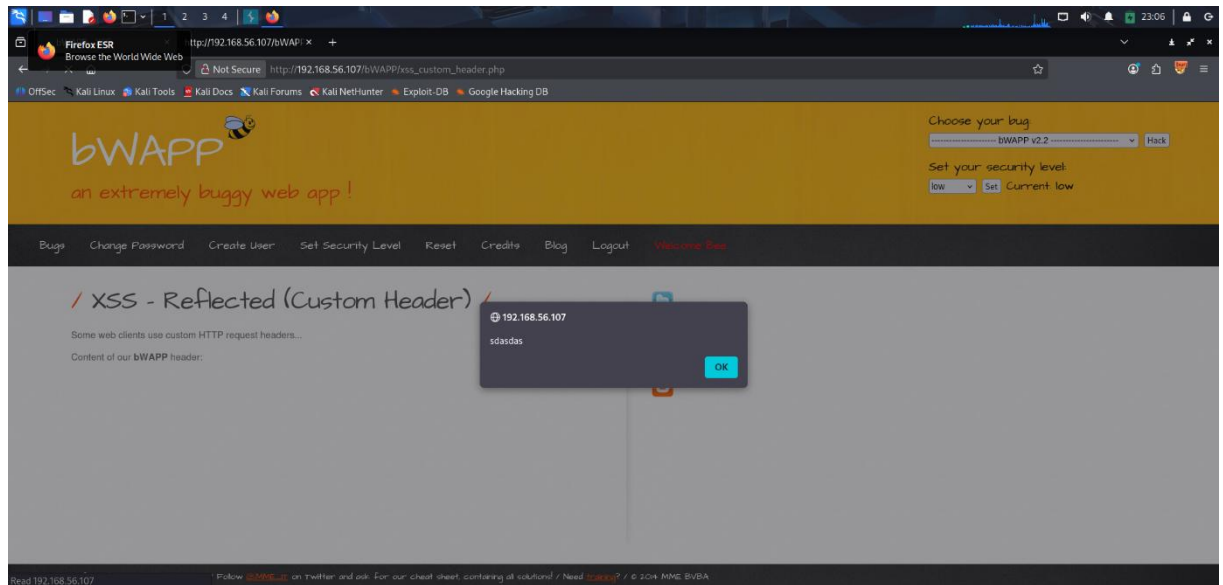
XSS REFLECTED(AJAX/JSON):

Bir önceki zafiyet ile aynı nitelikleri taşıyor.



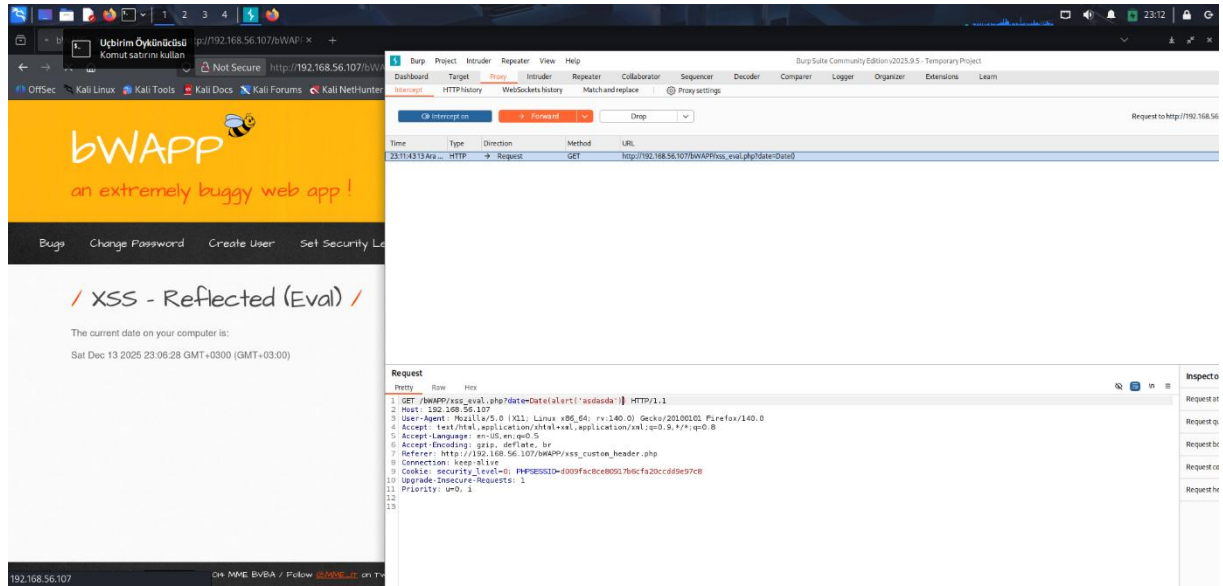
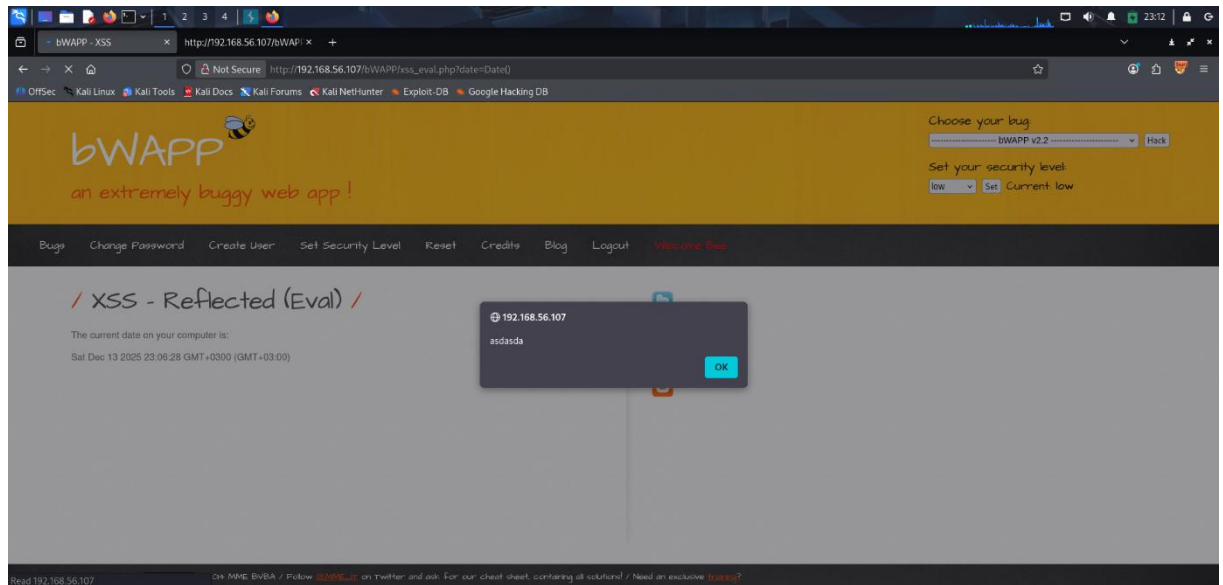
XSS REFLECTED (CUSTOM HEADER):

Bu labda ise ana ekranda yapabileceğimiz bir şey yok o yüzden önce burp suite intercept açıp sayfayı yeniliyoruz ve referer kısmına alertimizi yazdırıyoruz.



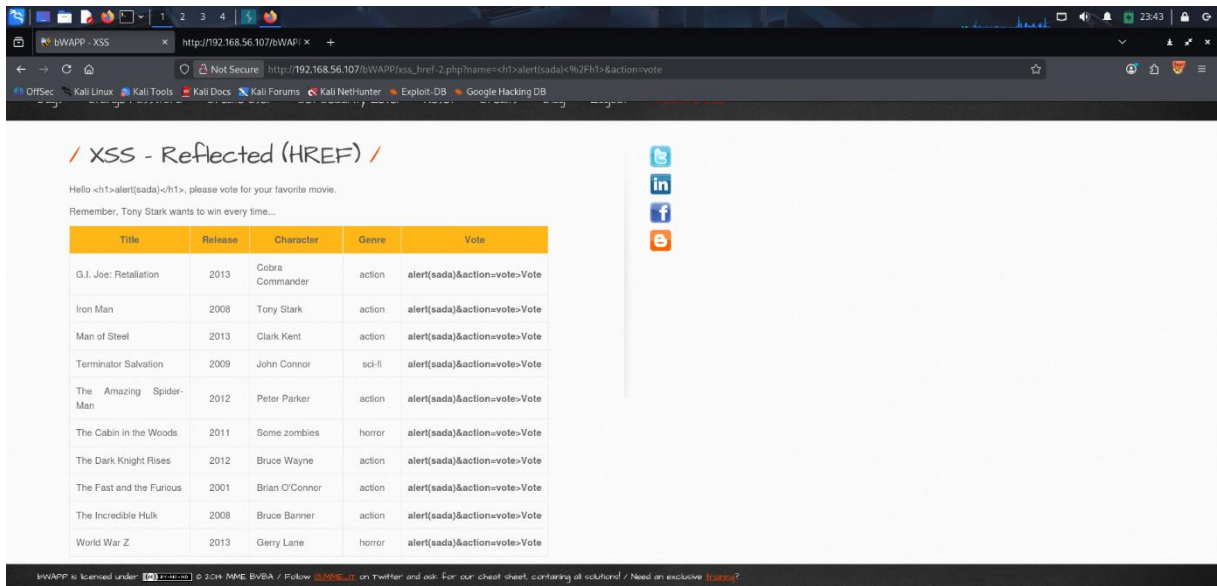
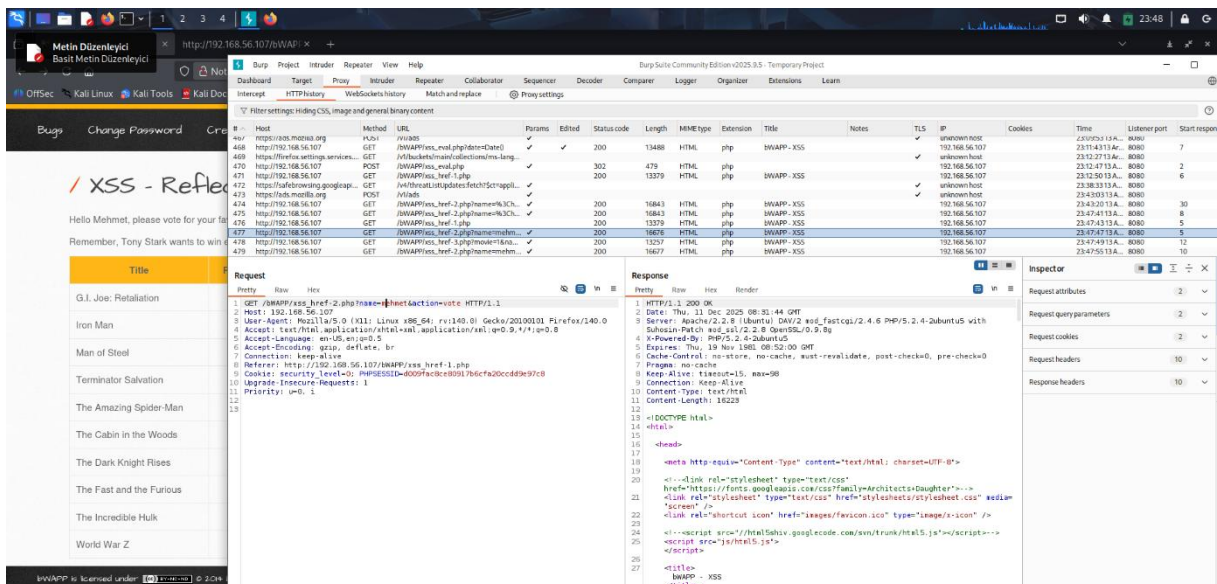
XSS REFLECTED (EVAL):

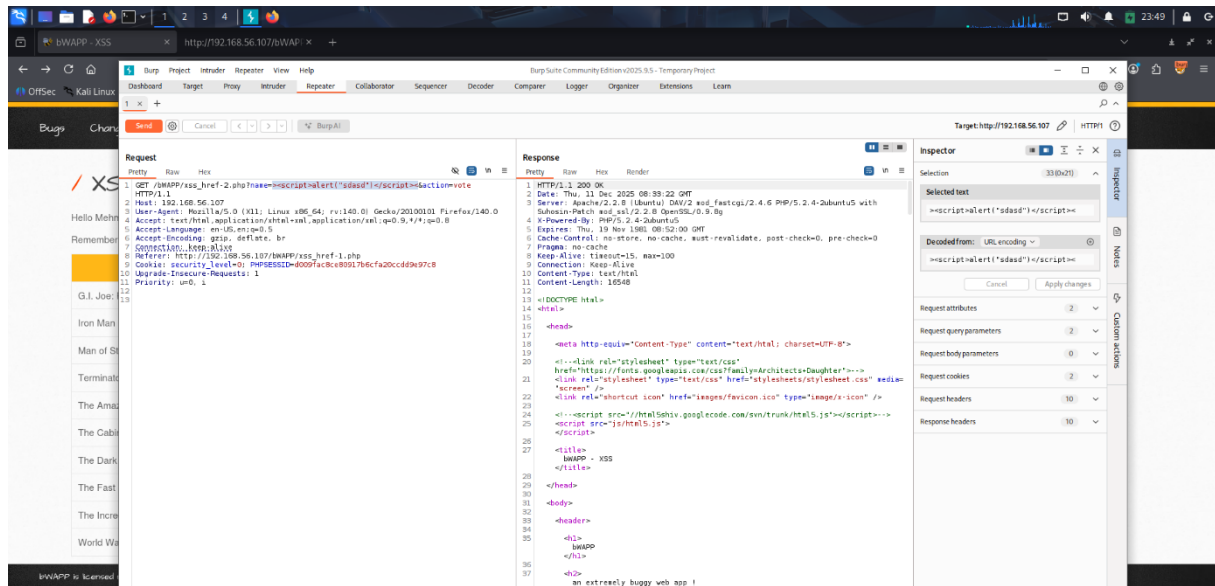
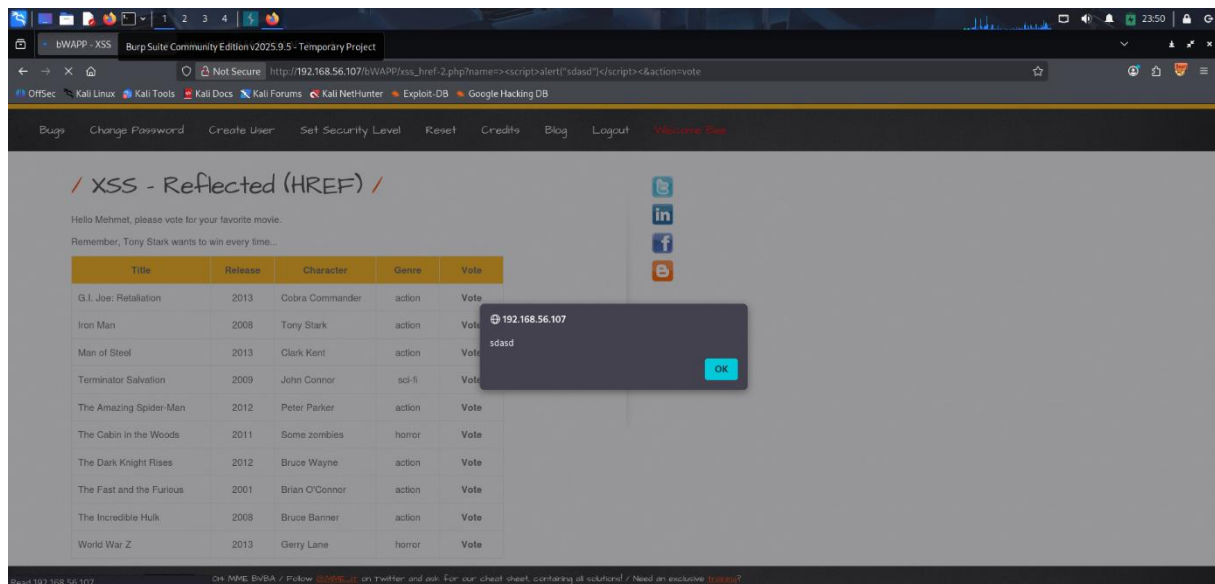
Burada ana sayfada tarihin yazıldığını görüyoruz ve aynı şekilde intercept açıp sayfa yeniliyoruz sonra tarih parametresine doğru payloadımızı giriyoruz.



XSS REFLECTED(HREF):

Bu labda ise bir kullanıcı ismi yazdığımızda bizi oylama sayfasına atıyor. Oylama yaptıktan sonra bunu burp http history de buluyoruz. İsim kısmına ise payload yazıyoruz Ama burada şöyle bir sorunumuz var script doğrudan çalışmıyor bu yüzden scripti kaçırıp çalıştırmamız gerekiyor. Bunun için ise payloada `<<script>` şeklinde başlıyoruz ve pop up ı çalıştırıyoruz.

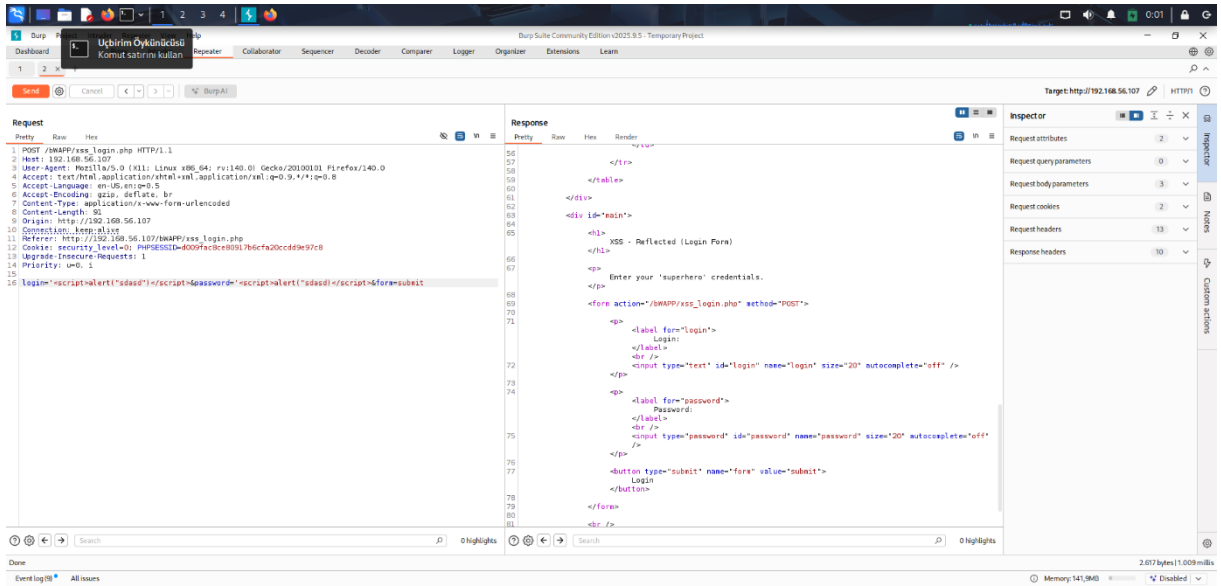
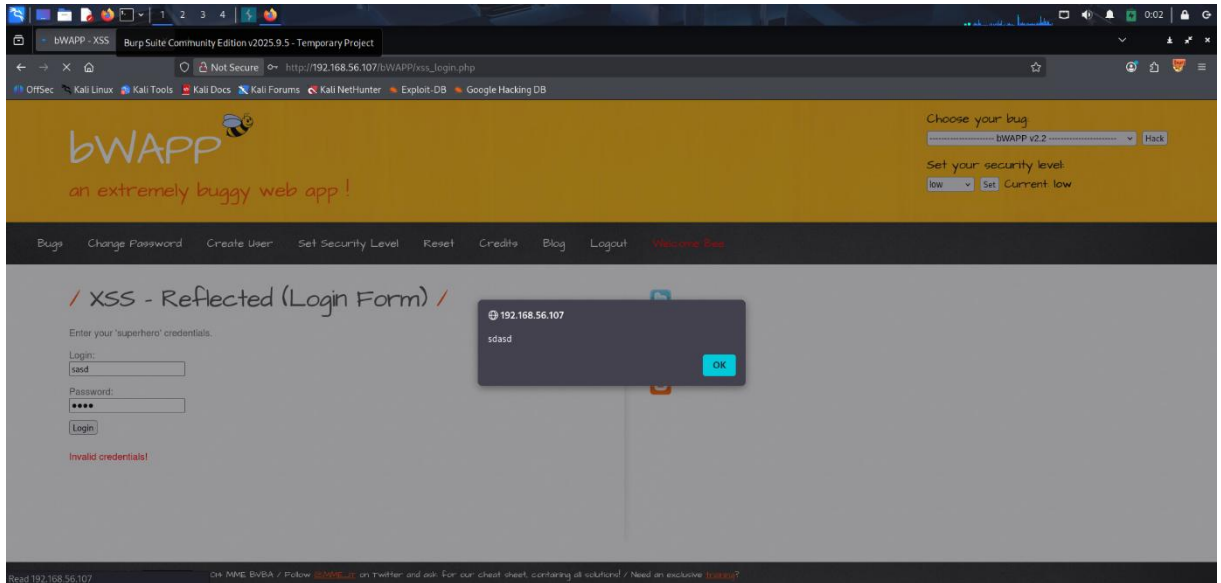




XSS REFLECTED (LOGIN FORM):

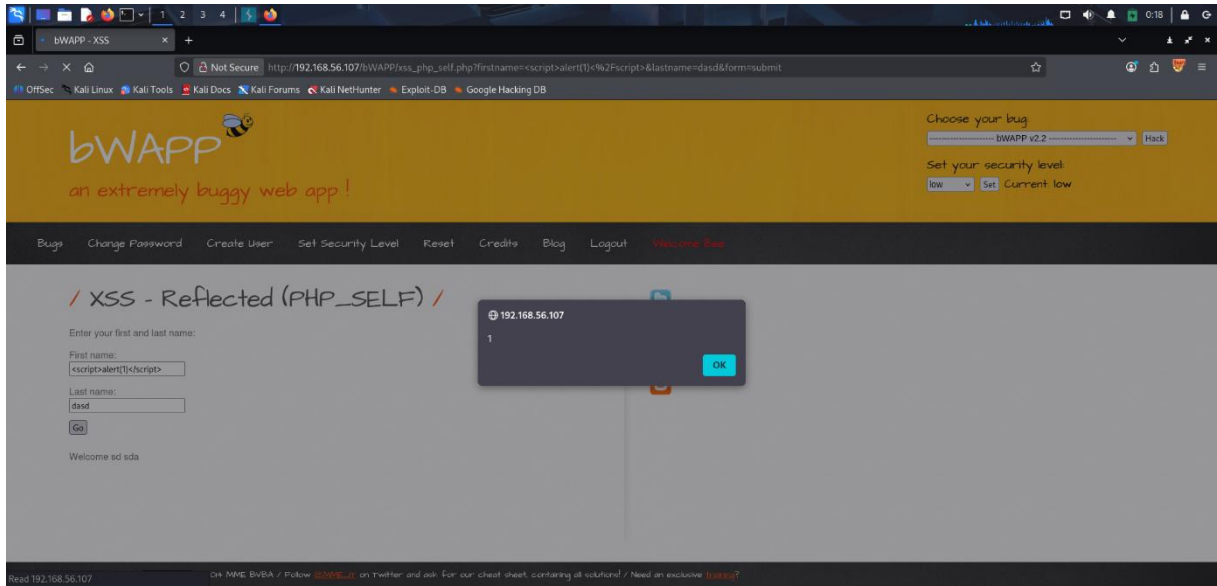
Ana ekranda login formunu görüyoruz. Buraya şifre ve kullanıcı adı giriyoruz sonrasında ise bu isteği http historyden alıp repeater a yolluyoruz burada kullanıcı adı ve şifremizin yerine `<script>` payloadımızı yerleştireceğiz. Tabi burada da sadece `<script>` şeklindeki payload yeterli olmuyor. Burada uygulama payloadımızı stringe çeviriyor bundan kaçmak için ise payloadımız şu hali alıyor

`'<script>alert("dasdas")</script>`



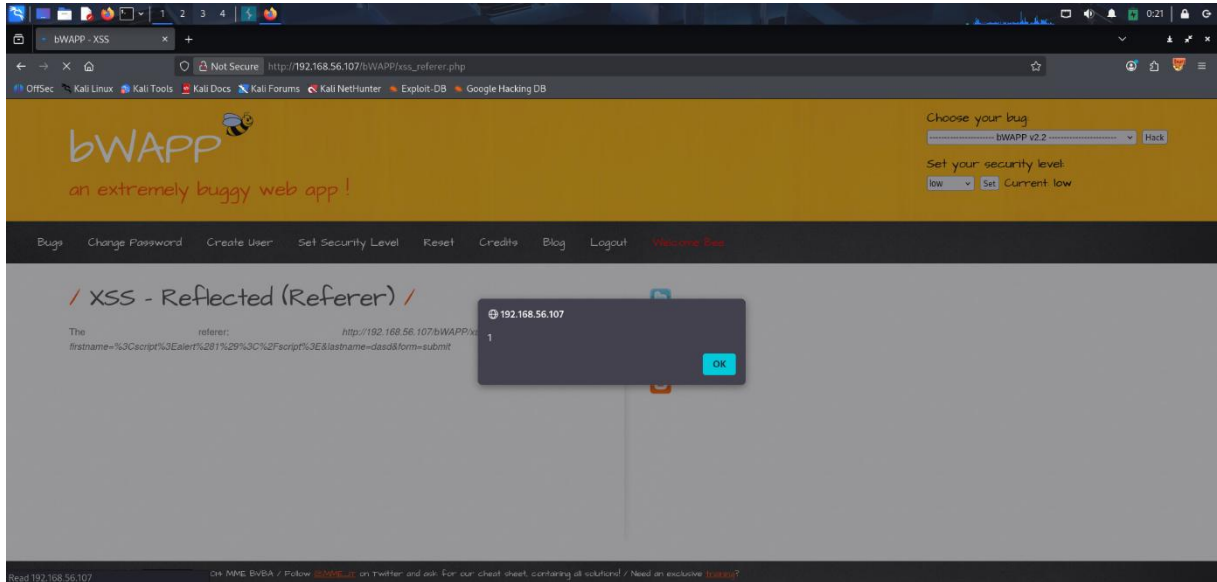
XSS REFLECTED(PHP_SELF):

Bu labda kullanıcı girdisi yerine klasik <script> payloadını deniyoruz yine ve direk pop up çalışıyor.



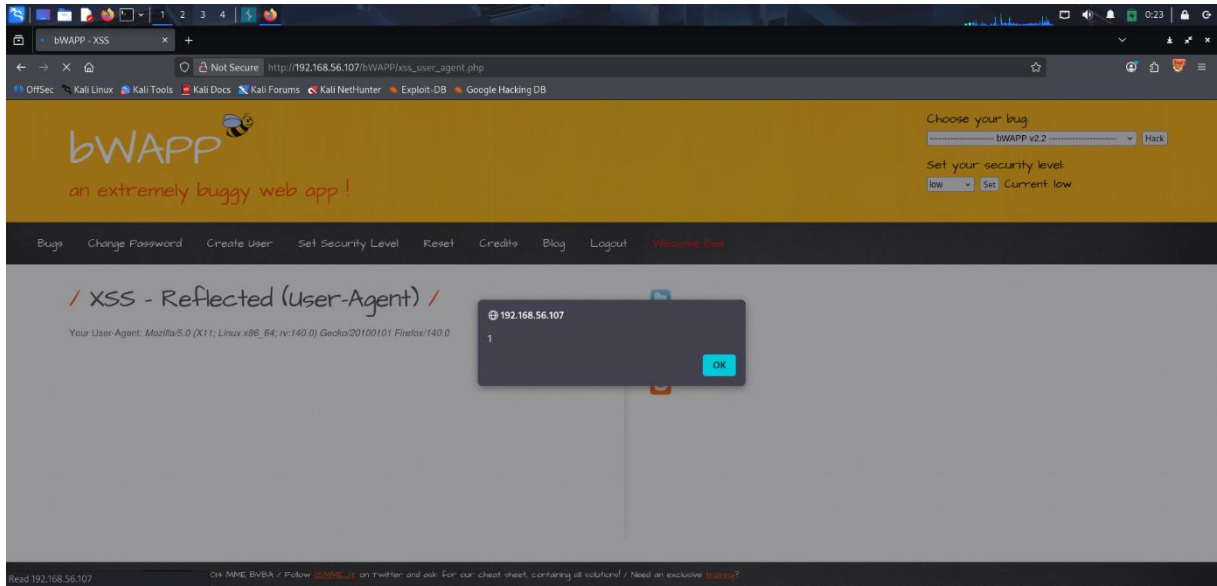
XSS REFLECTED(REFERER):

Burada da sayfayı yenileyip isteği burp te yakalıyoruz ve lab adından anlaşılacağı üzere referer kısmına `<script>alert(1)</script>` yazıp forward ettiğimizde pop-up çalışıyor.



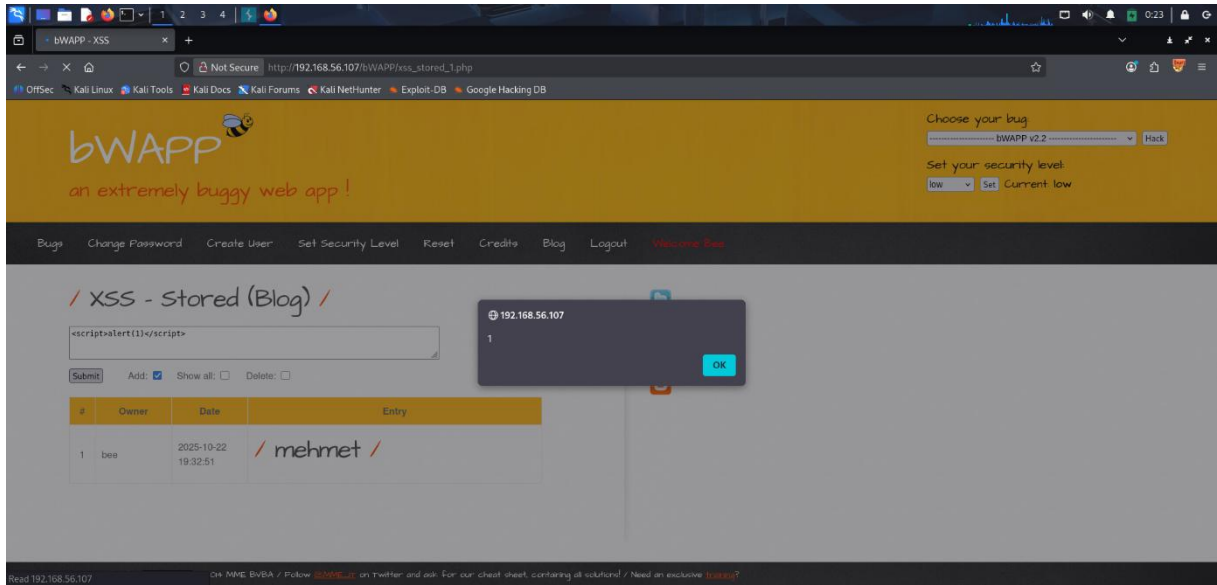
XSS REFLECTED(USER-AGENT):

Bir önceki zafiyetten hiçbir farkı yok sadece user agent kısmına payloadımızı yazıyoruz.



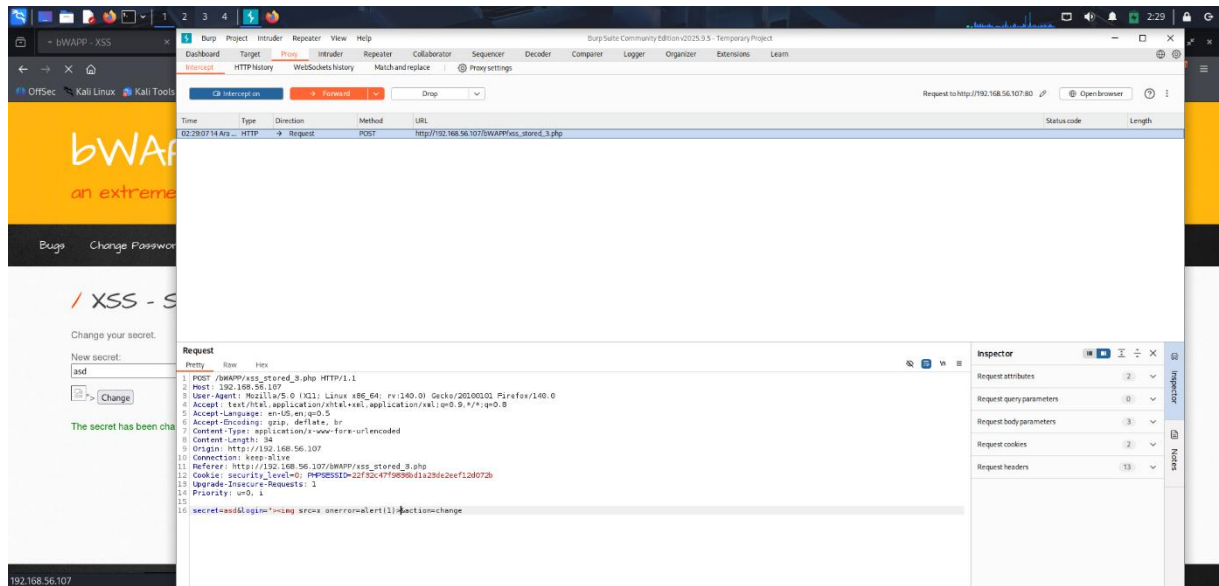
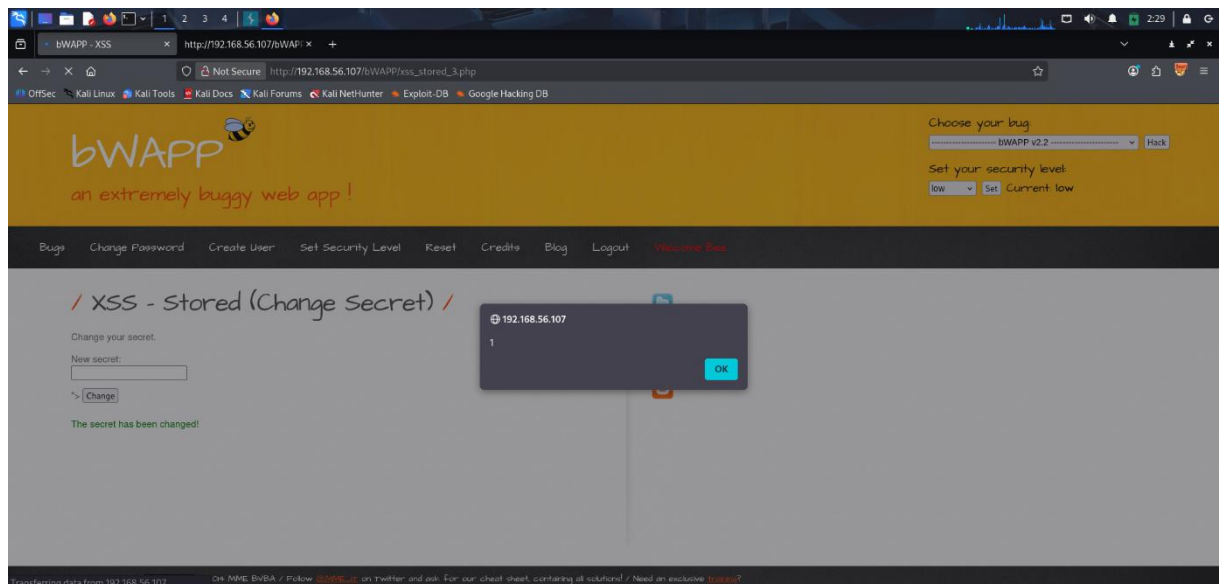
XSS STORED (BLOG):

Artık kalıcı xss kısmına geldik sadece burada da aynı payloadlar ile işlemlerimizi yapacağız.



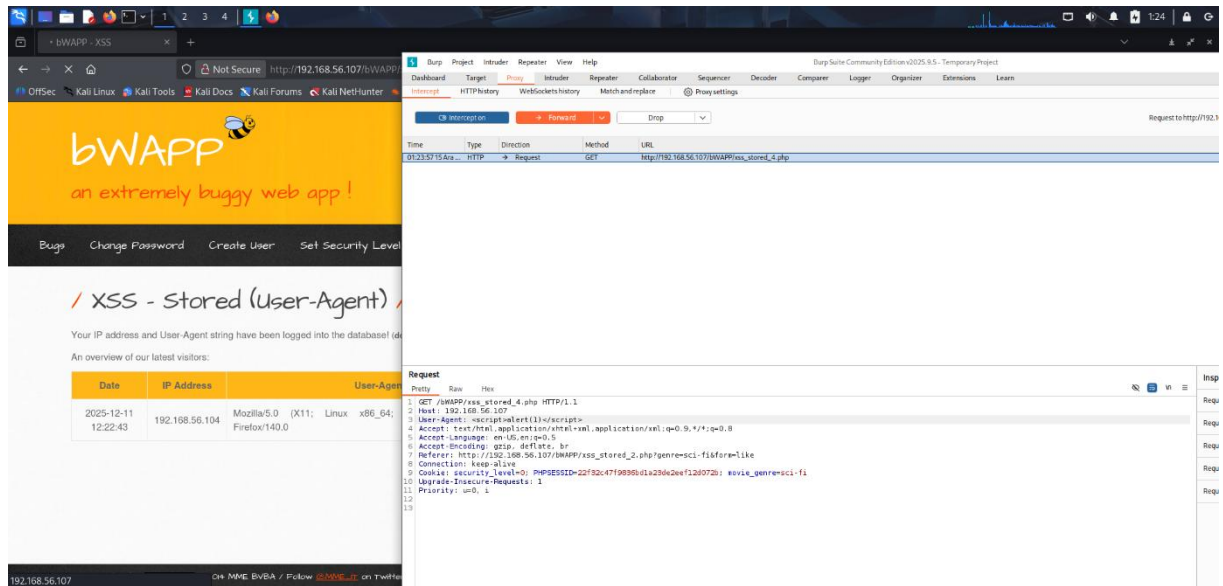
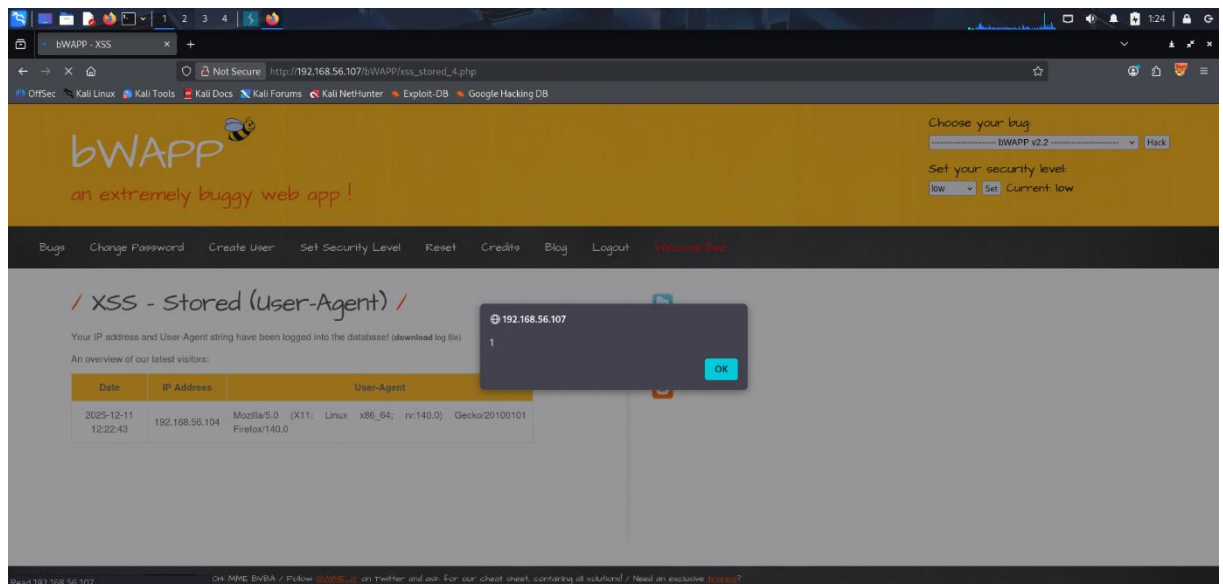
XSS STORED(CHANGE SECRET):

Burada lab bizden bir secret inputu girmemizi istiyor. Intercept açıp buisteğimizi yakalıyoruz ve login kısmına payloadlarımız deniyoruz <script> payloadı yine tek başına başarılı olmuyor. Sonunda doğru payloadı buluyoruz.



XSS STORED(USER AGENT):

İntercept ile sayfayı yenileyip user agent kısmına <script> payloadımızı yerleştiriyoruz.



XSS REFLECTED (AJAX XML):

Bu labda ise `<h1>alert(1)</h1>` payloadı kullanıyoruz ama sayfa bunu kabul etmiyoruz encode edince ise payloadımız çalışıyor.