qwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnm

|  |
| --- |
| HTML INJECTION(HTML ENJEKSİYONU)  Güvenli Web Yazılımı Geliştirme  Volkan Berkay Demir  2020220014 |

İÇİNDEKİLER

* HTML INJECTION NEDİR ?
* HTML INJECTION NEDEN OLUR ?
* HTML INJCETION KAÇA AYRILIR ?
* HTML INJECTION ÖRNEĞİ HANGİ MAKİNADA YAPILDI ?
* HTML INJECTION ÖRNEĞİ
* HTML INJECTION NASIL ÖNLENİR ?
* KAYNAKÇA

HTML INJECTION (HTML ENJEKSİYONU) NEDİR ?

Html injection web sitelerinde bulunan yetersiz önlem sonucunda saldırganın siteyi html kodları ile suistimal etmesi ile yapılan, OWASP TOP 10 de yer alan bir saldırı vektörüdür.

Bu açık ile saldırgan; kullanıcı adı , soyadı, yaşı gibi kişisel bilgileri, eğer site E-Ticaret gibi kredi kartı veya kimlik gibi önemli bilgileri barındırıyor ise saldırgan bunlara erişim elde edebilir , çalabilir.

Peki Neden Olur Bu Injection ?

* Development (Uygulama Geliştiricileri) yazdığı kodlar üzerinde yeterli özeni göstermemesi,
* Sitenin PHP sürümünde bulunmuş açıklar,
* Kullanılan add-ons (eklenti) üzerinde bulunan açıklar,
* Form , atama butonu gibi son kullanıcıların giriş(input) işlemleri yapılabileceği alanlardaki denetimin yeterli şekilde yapılmaması,
* Kullanıcının , input alanlarında “Özek Karakterleri” kullanamaması için black list yapılmaması,

Gibi durumlar sebebiyle bu zafiyet ortaya çıkmaktadır.

HTML INJECTION iki başlık altında incelenebilir.

* Yansıtılmış(Reflected)

Yansıtılmış enjeksiyon da genel olarak tek bir kullanıcı hedef alınır veya phishing (oltalama) saldırısı ile birden çok kullanıcı hedef alınır. Bu enjeksiyon türünde sitenin veri tabanında herhangi bir kayıt gerçekleşmez. Çünkü sunucu ve istemci arasında bir veri kayıdı yoktur.

* Depolanmış(Stored)

Depolanmış enjeksiyon da uygulama veya site üzerinde bulunan bir yorum sayfası , iletişim formu , arama motoru , sunucu-istemci arasında değişiklikler gerçekleşebilir. Yansıtılmış enjeksiyonun aksine veri tabanında kayıt gerçekleşir.

En sık rastlanan Depolanmış enjeksiyon türlerinden biri ise “Kayıt Ol/Giriş Yap” alanlarına PHP ve HTML kodları entekre edilerek “Oturum Süresi Doldu, Yeniden Giriş Yapın” gibi Pop-up çıktısı vererek kullanıcıyı sahte bir forma aktarmasıdır.

Saldırgan bu saldırıları yaptıktan sonra sunucuyu burpsuite , Netacad , veya WhireShark gibi toollar ile sniffing’e (koklama) alarak kullanıcı verilerini elde edebilir.

Gelin bu açığı daha iyi anlayabilmek için beraber sömürelim.

* İlk olarak iki tane makinaya ihtiyacımız var. Bunlar bee-box ve Kali Linux

Bee-box içerisinde bir çok açık barındıran, yasal bir şekilde açık sömürebileceğimiz

bir makina. Kali Linux ise içerisinde bir çok tool barındıran bir işletim sistemi.

Bee-box link = <https://sourceforge.net/projects/bwapp/postdownload>

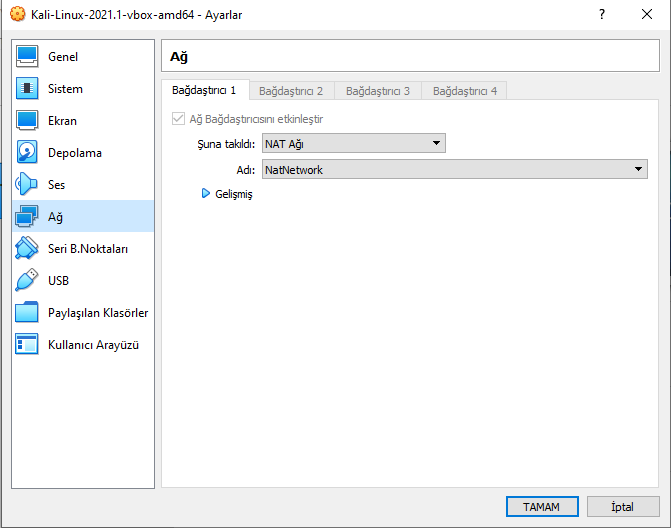
Bee-box kurulum videosu = <https://www.youtube.com/watch?v=syBbcK9PrA0&t=196s>

Kali – Linux link = <https://www.offensive-security.com/kali-linux-vm-vmware-virtualbox-image-download/>

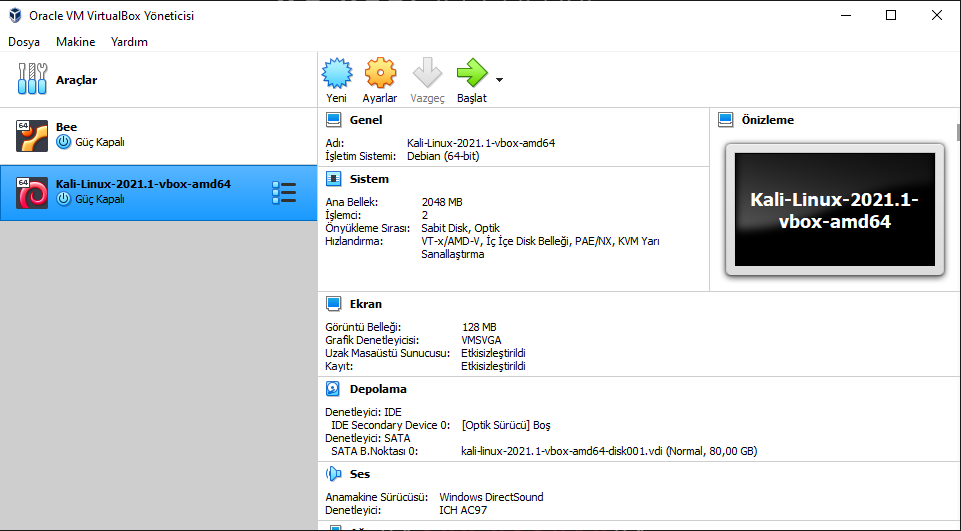
Kali – Linux kurulum videosu = <https://www.youtube.com/watch?v=m1Um-lSr50k>

Biz bu gün Reflected açığını sömüreceğiz.

* İlk olarak Kalimizi ve bee-box ımızın nat ayarlarını yapıyoruz. Bu sayede kali ile bee-box a bağlanabileceğiz.



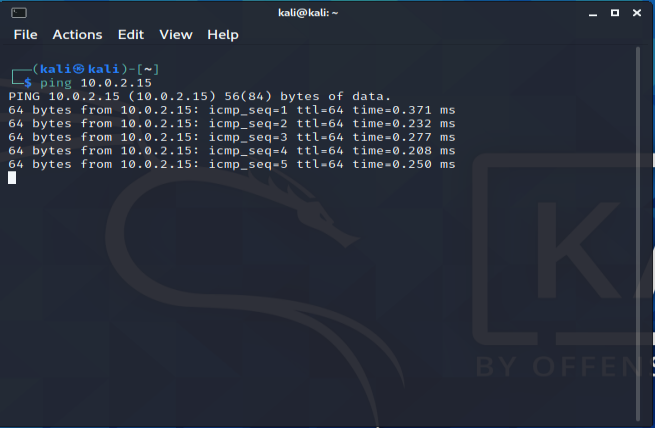
* Daha sonra Kalimizi ve bee-boxımızı çalıştırıyoruz.



* Şimdi ise bee-box ın terminalini açarak “ifconfig” komutunu çalıştırıyoruz. Bu sayede bee-box ın ip sini elde etmiş oluyoruz.



* Kalimize geçip terminali açarak “ping (bee-box ip)” komudunu çalıştırarak bee-box ile ilişki kurup kuramadığımızdan emin oluyoruz.

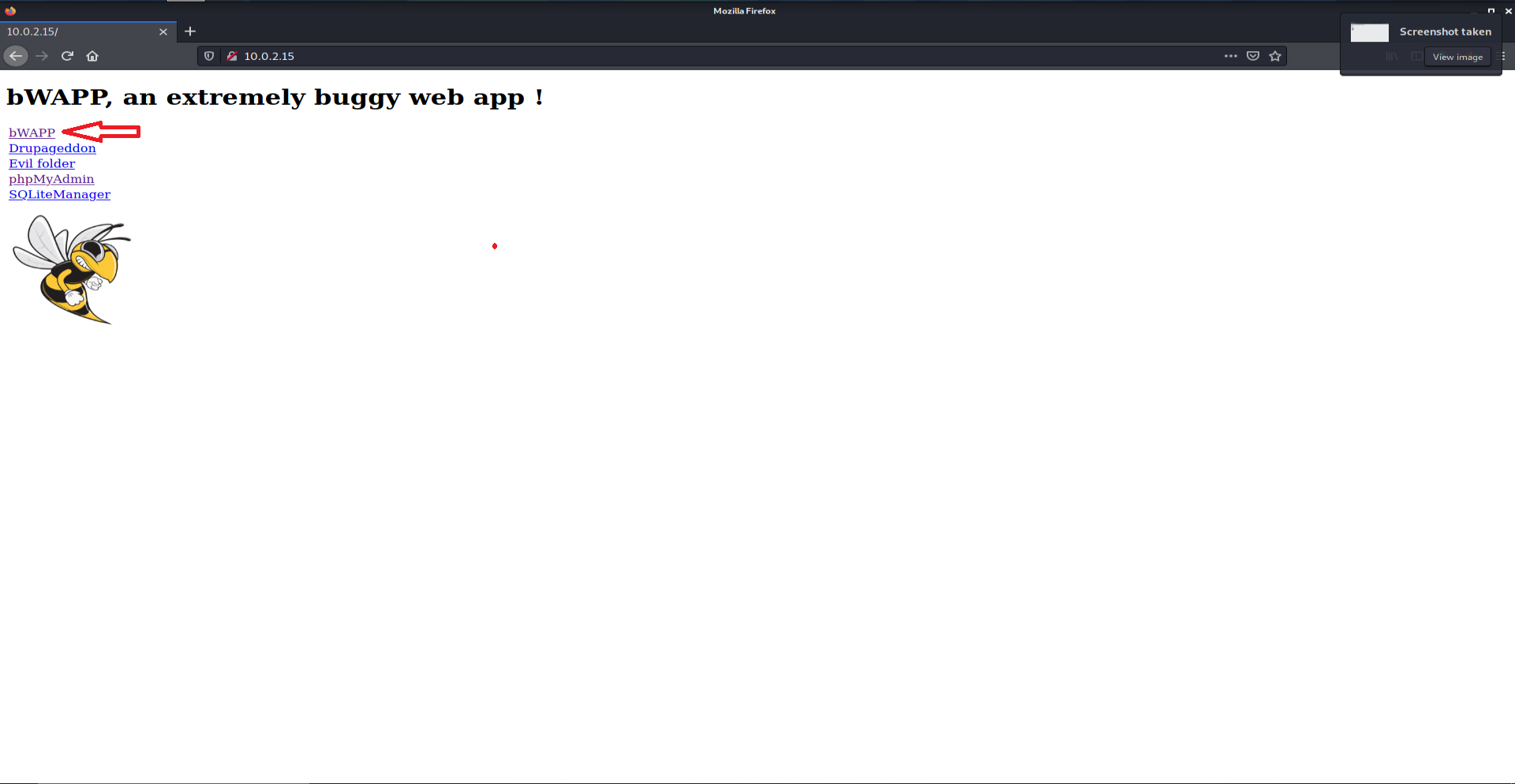


Yukarıdaki çıktıyı aldıysanız bağlanabiliriz.

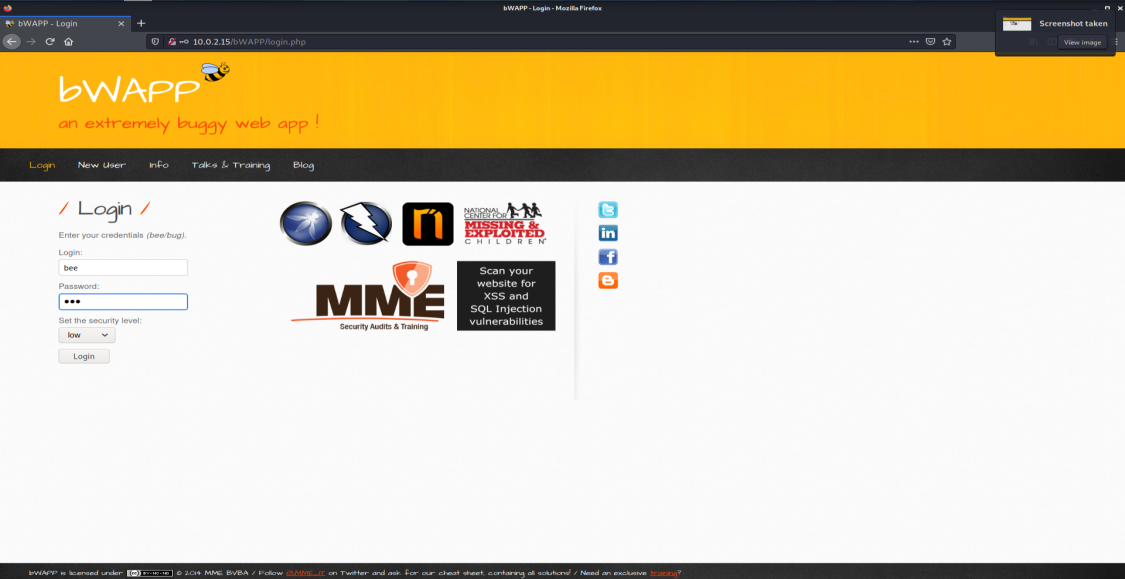
* Bee-box ın ip adresini herhangi bir internet tarayacısının arama motoru kısmına yazıyoruz

5_arama.png

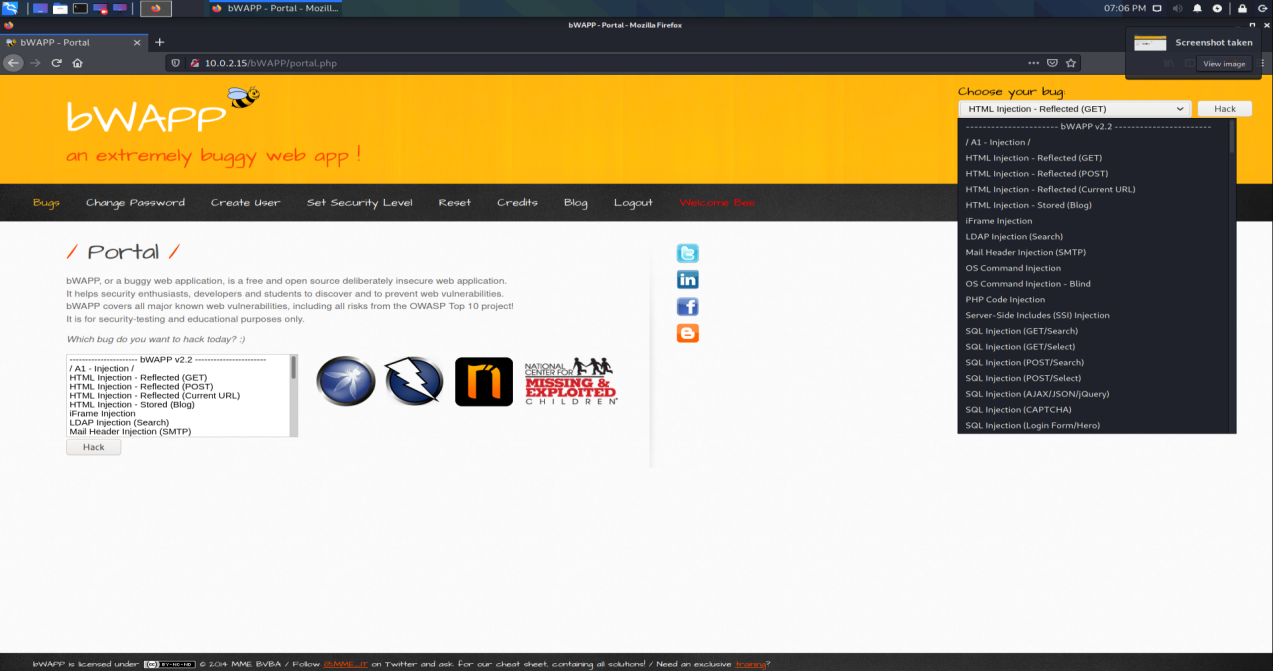
* Ve bee-box ın menüsüne geldik biz en üst seçenekteki bwapp e gireceğiz.



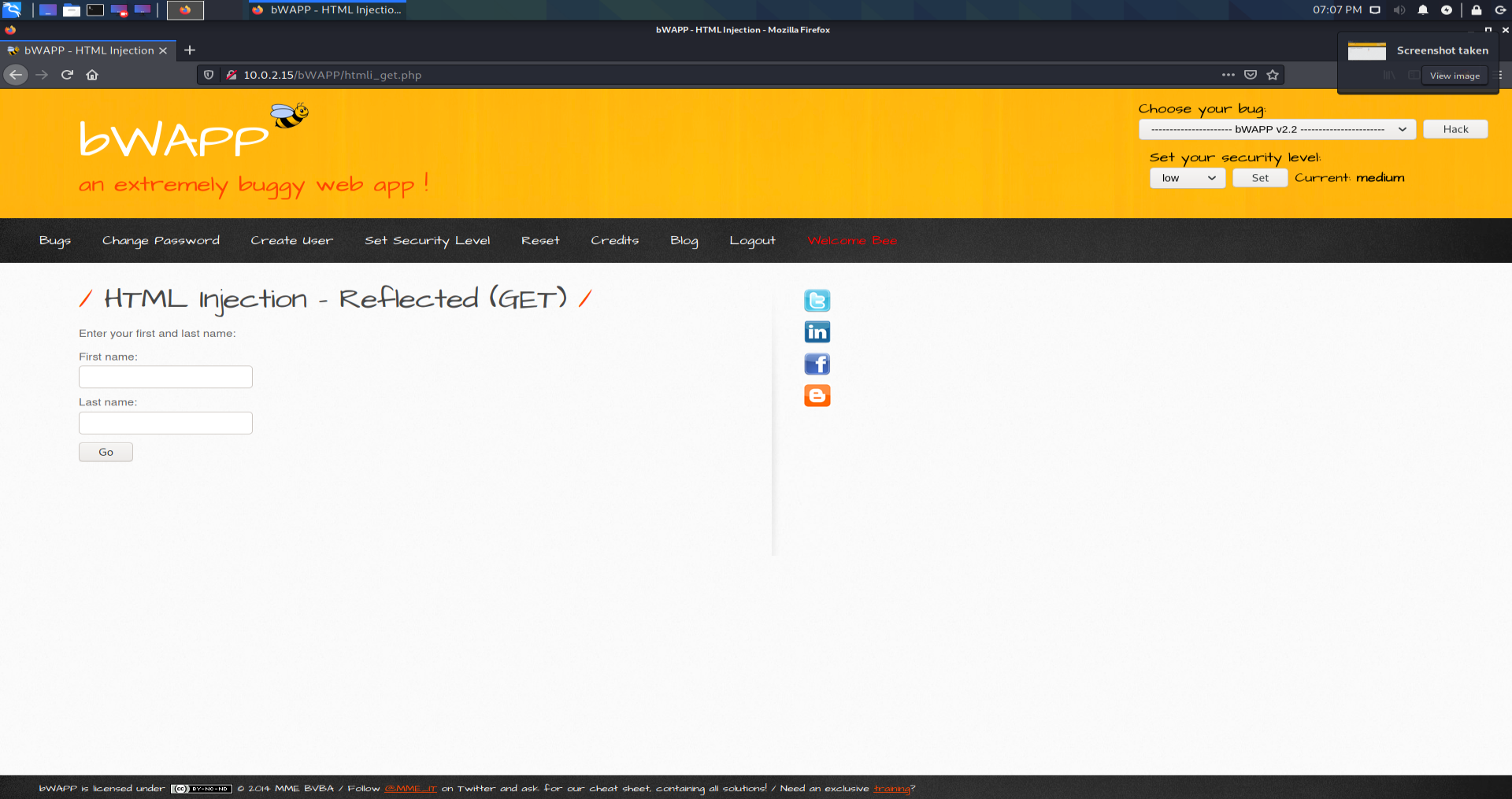
* En üste geldikten sonra soldaki giriş kısmına default olan “bee” kullanıcı adı ve “bug” prolasını giriyoruz



* Şimdi sağ üstteki sekmeden A1 bölümündeki “HTML INJECTION – REFLECTED (GET)” açığını medium ayarda seçiyoruz



* Seçtikten Sonra Hack butonuna basarak arayüzümüze geliyoruz



* Şimdi ilk olarak input yaparak sistemin ne kadar güvenli olduğuna bakalım.

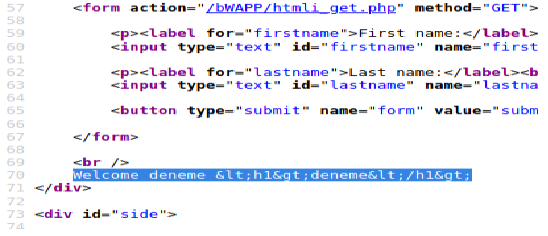
Ben First name kısmına “deneme”, Last name kısmına ise “<h1>deneme</h1>” yazdım, “<h1>deneme</h1>” kodu html dilinde kısa bir text çıktısı alabilmek için kullanıyor.



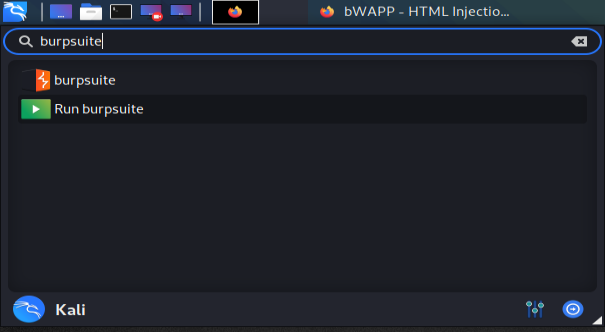
* Go butonuna basarak çalıştıralım ve çıktımıza baklım



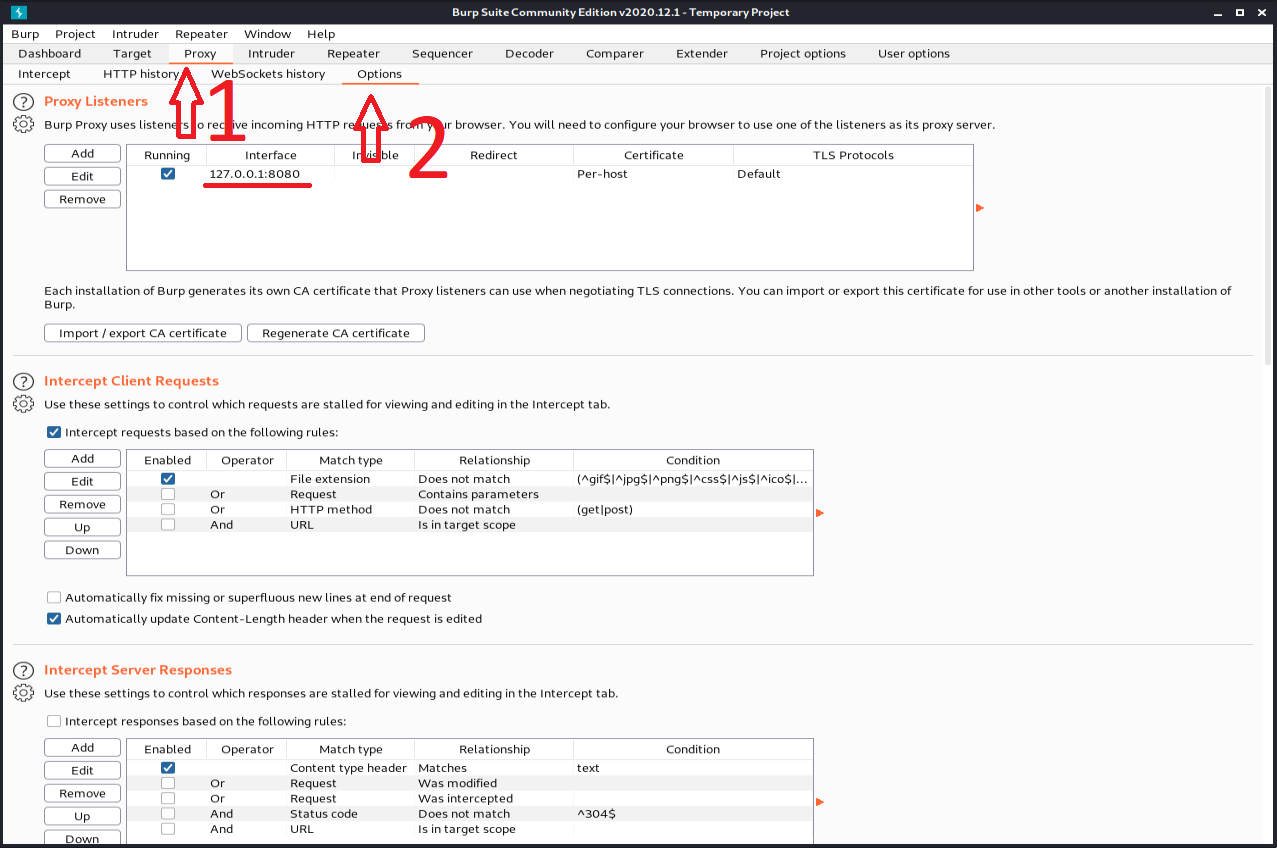
* Alt kısımda “Welcome deneme <h1>deneme</h1>” çıktısı aldık. Biz kodu direkt “deneme” çıktısı olarak istiyorduk. Sayfa kaynağına bakalım nedeni neymiş.



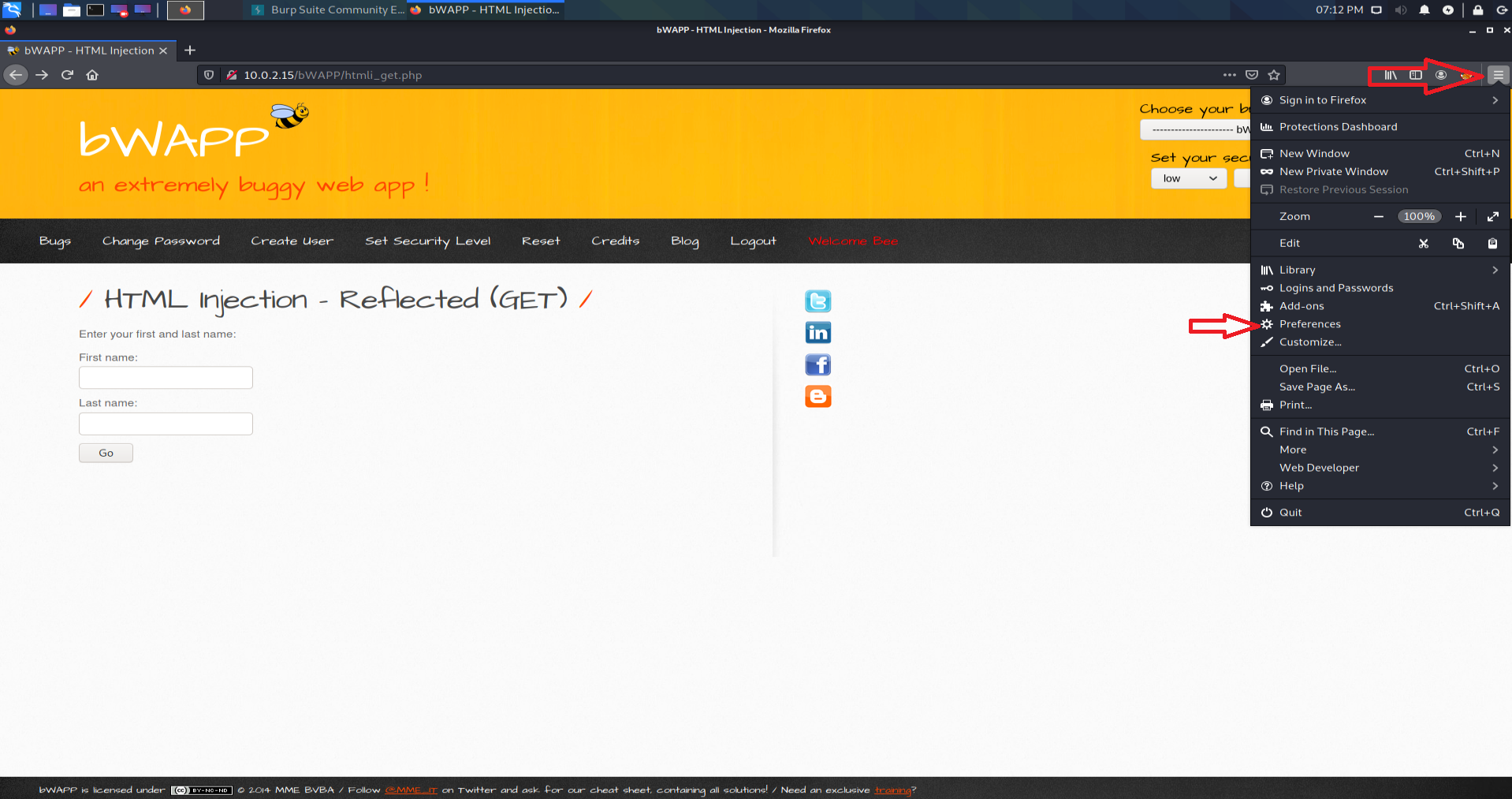
* Gördüğünüz gibi girdiğimiz “<h1>deneme</h1>” kodu “&lt; h1&gt;deneme&lt;/h1&gt;” şeklinde gözüküyor bunun nedeni html karakterleri black list e alınmış. Bu sayede “<” işareti “&lt;” , “>” işareti ise “&gt;” şeklinde sunucuya istek olarak gidiyor.
* O zaman sunucuya giden paketini yakalayıp üzerinde oynama yapalım. Kali makinamızda sol üstde ki arama kısmına gelerek “BurpSuite” adlı toolu açıyoruz. BurpSuite nedir dersek, kısaca tek kişilik ordu diyebiliriz.



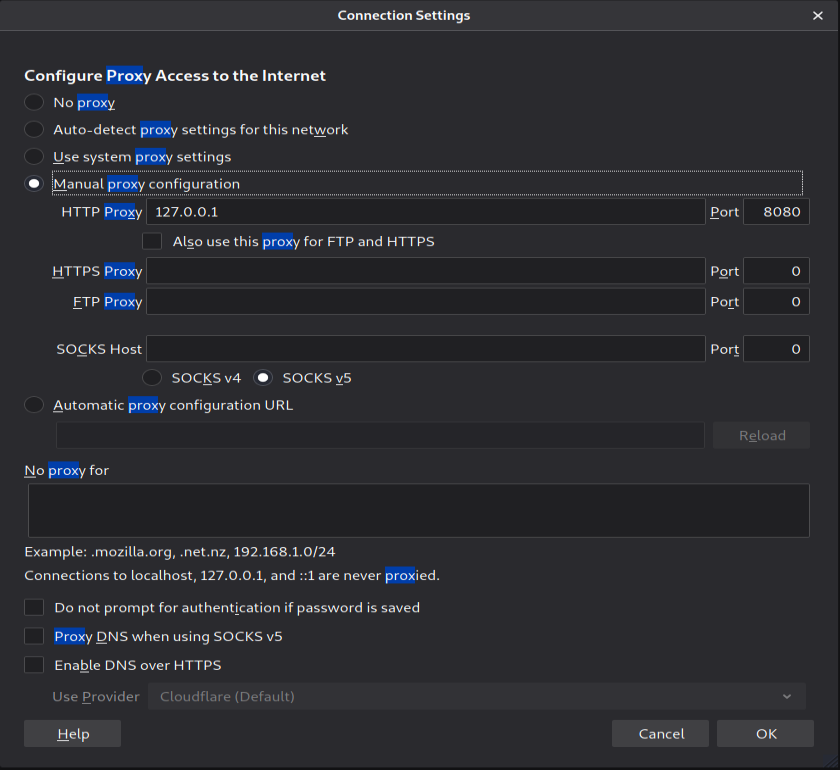
* BurpSuite i açtıktan sonra yukarıdan Proxy kısmına girip options kısmına tıklıyoruz.



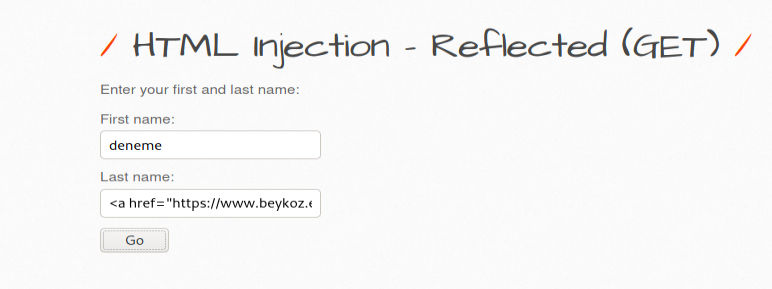
* Altı çizili kısımdaki 127.0.0.1:8080 kısmı BurpSite in default proxy ayarlarıdır. Proxy sayesinde işlem yaptığımız sayfadaki paketleri yakalayabileceğiz.
* Şimdi ise Web tarayıcımızın ayarlarına girip sağ üstteki üç çizgiye tıklayarak “Preferences” e tıklıyoruz.



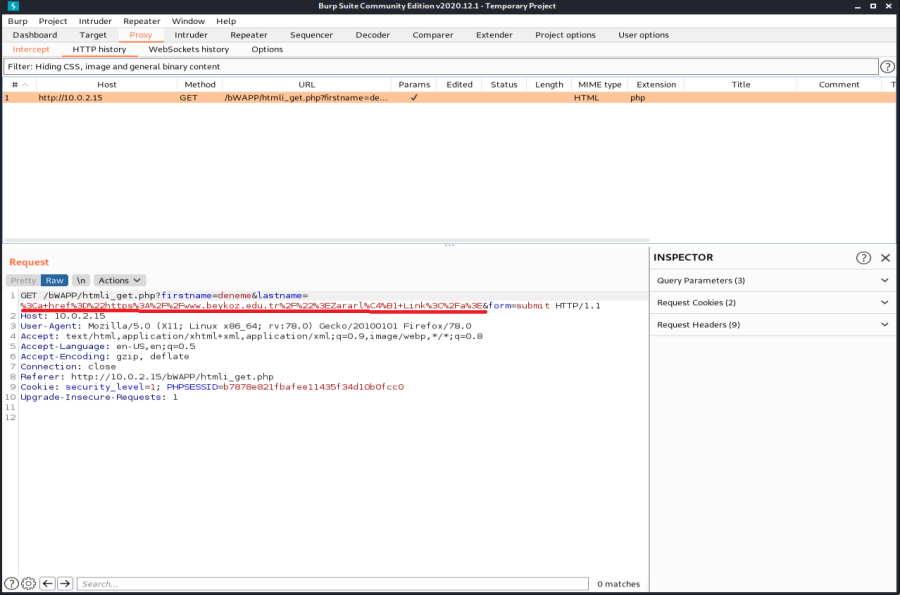
* Girdikten sonra arama kısmına Proxy yazarak BurpSuite in default proxy ayarlarını giriyoruz.



* Şimdi ise bee-box sayfasına geri dönelim. First name kısmına yeniden “deneme”, Last name kısmına bu sefer bir link kodu ekleyeceğim. Bu link okulumuz Beykoz Üniversitesinin sitesidir , fakat saldırgan birisi bunu zararlı bir link olarak da ekleyebilirdi.
* Benim girdiğim kod “<a href="https://www.beykoz.edu.tr/">Zararlı Link</a>” budur.
* Html de “<a>” tagi link eklemek için kullanılır, “href” kısmına link yazılır. “Zararlı Link” yazan yer ise üzerine tıkladığımızda Üniversite sitesine gönderecek olan yazıdır.

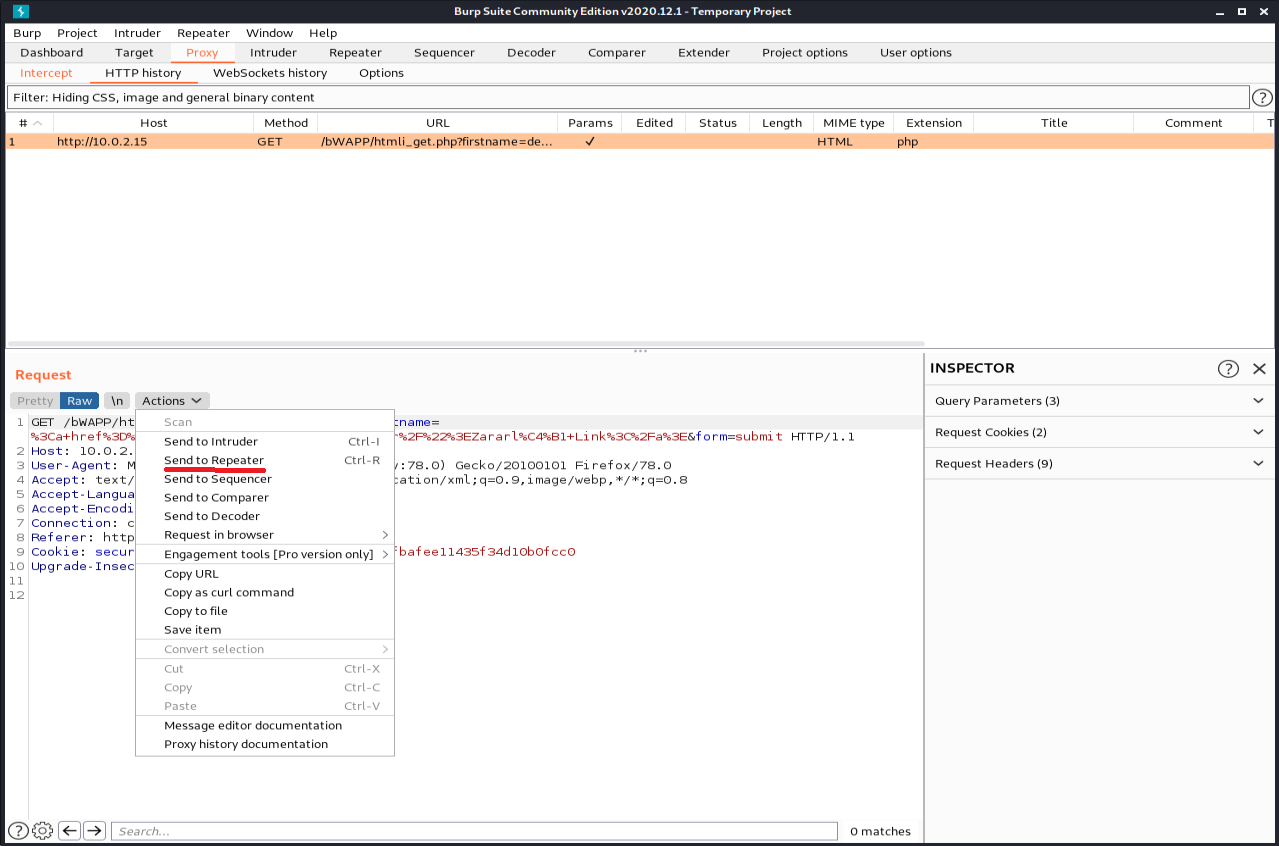


* Go butonuna bastıktan sonra BurpSuite paketimizi yakaladı.



* Altı çizili koda bakarsanız bizim yolladığımız

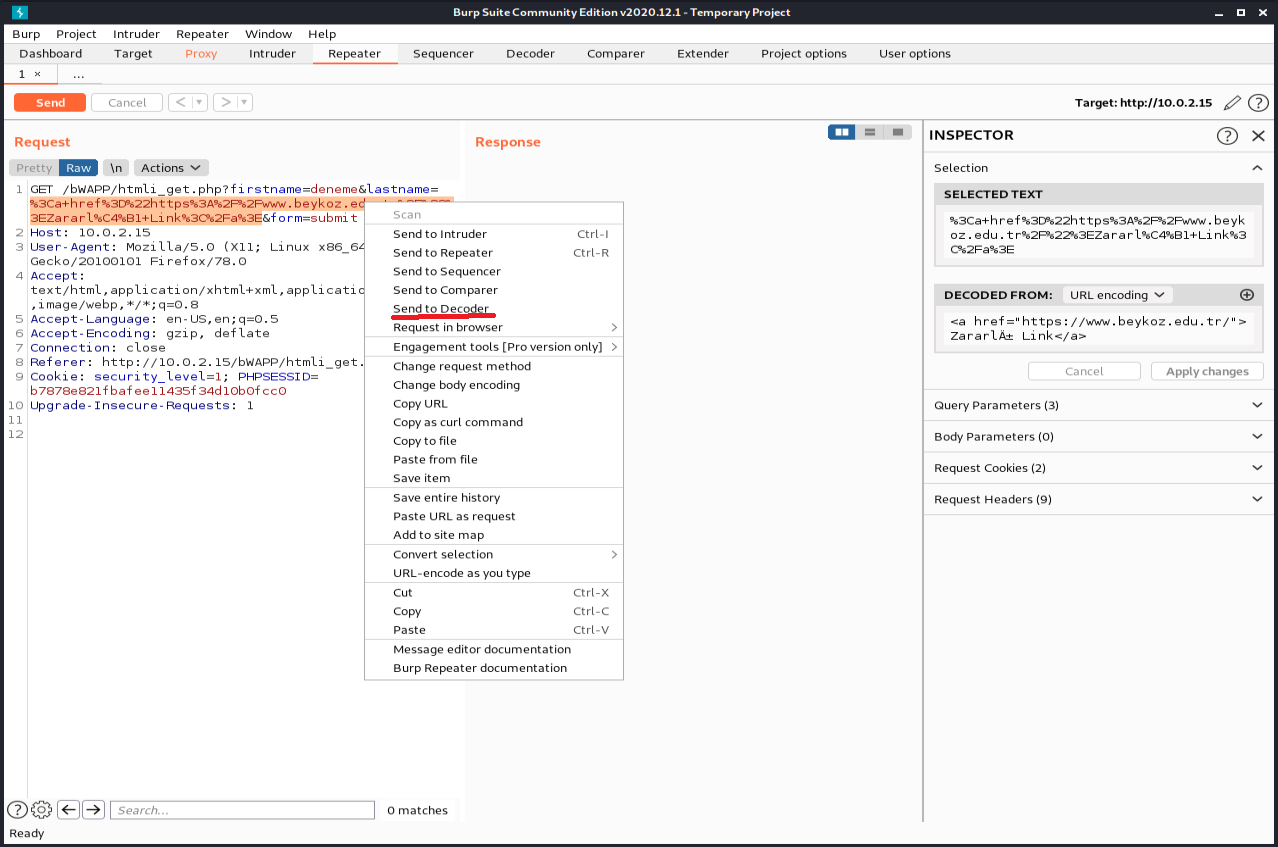
“<a href="https://www.beykoz.edu.tr/">Zararlı Link</a>” kodunu HTML encode şeklinde yolladı. Bu yüzden yolladığımız hiç bir HTML karakteri kabul edilmeyecektir. Burada yapmamız gereken isteğimizi farklı bir formatta yollamak. Bunun için “Actions” bölümüne veya sağ tıklayarak “Sen to Repeater” a tıklamamız gerekiyor. Bu sayese paketimizde değişiklik yapabileceğiz. Ayrıca en üste bakarak paketlerde “GET” methodu kullanıldığını görmüş olduk.



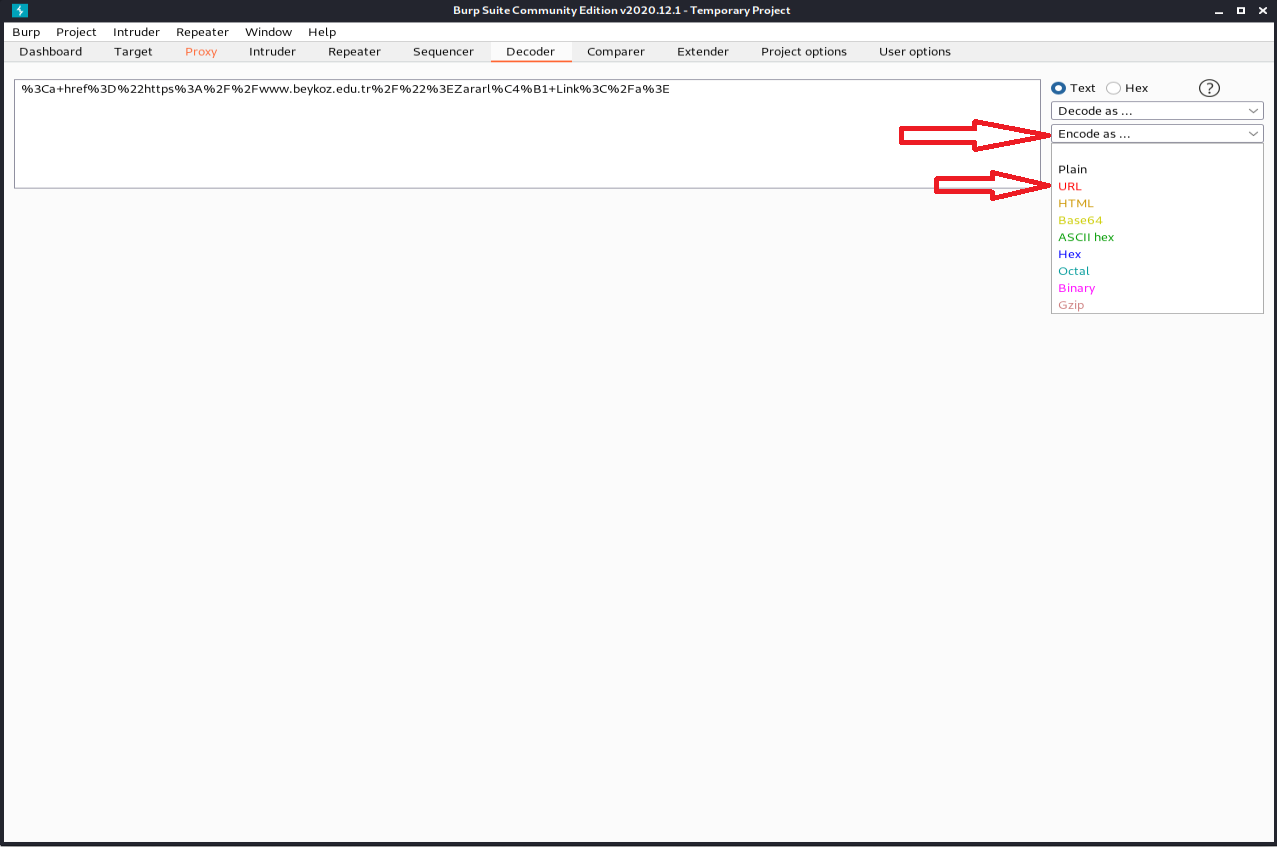
* Repeater kısmı Request ve Response olarak ikiye ayrılır.



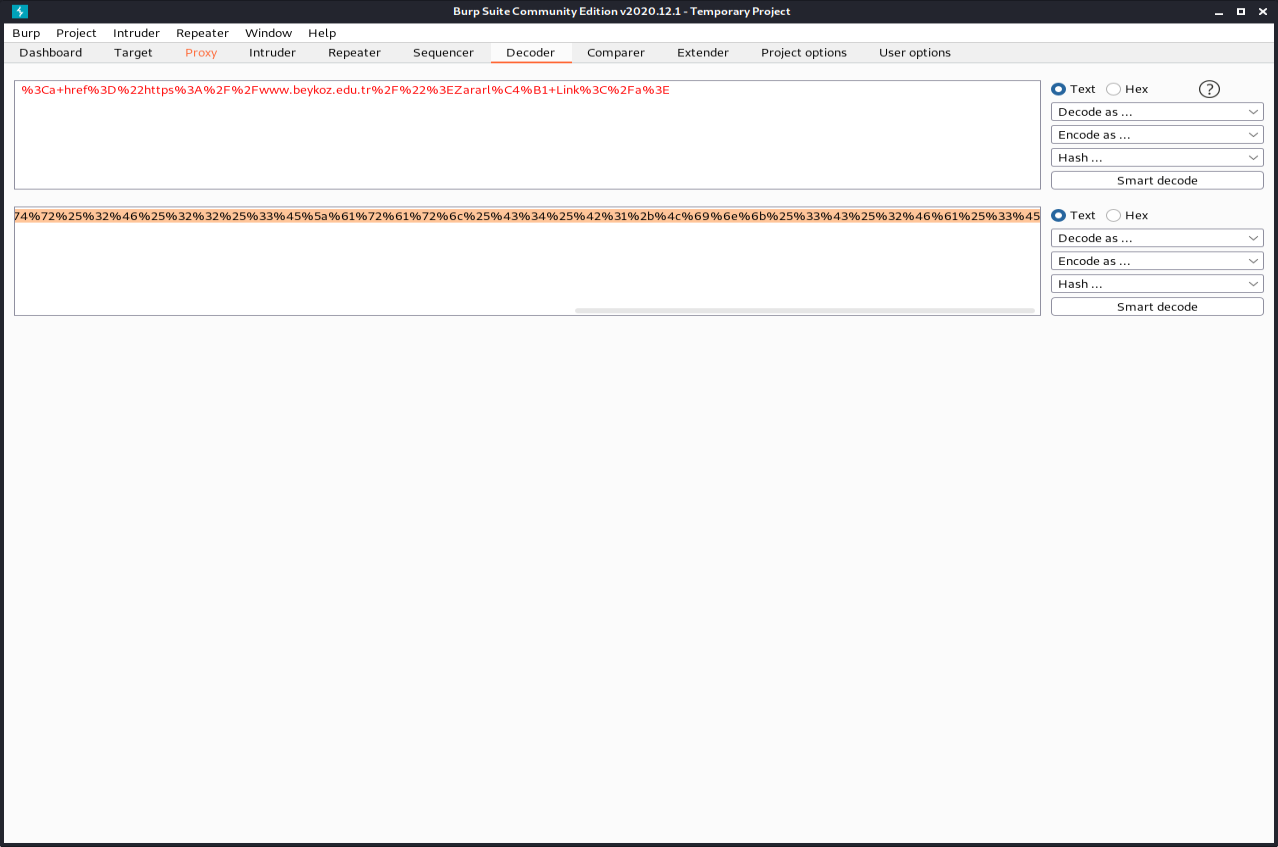
* Burada işaretli olan HTML encode lu koda sağ tıklayarak veya kopyalayarak Decode bölümüne yolluyoruz.



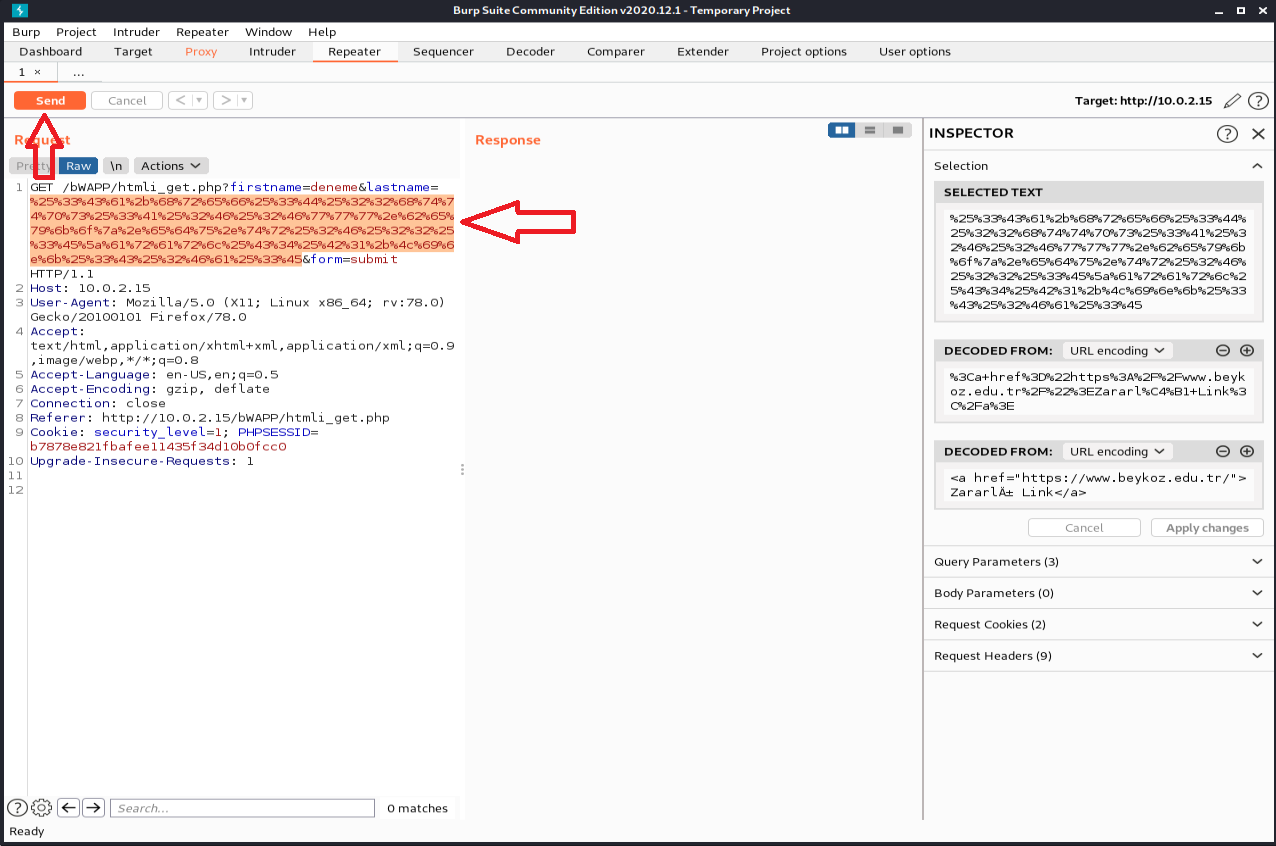
* Burada sağ kısımdaki ”Encode as ...” sekmesine tıkladıktan sınra “URL” yi seçiyoruz



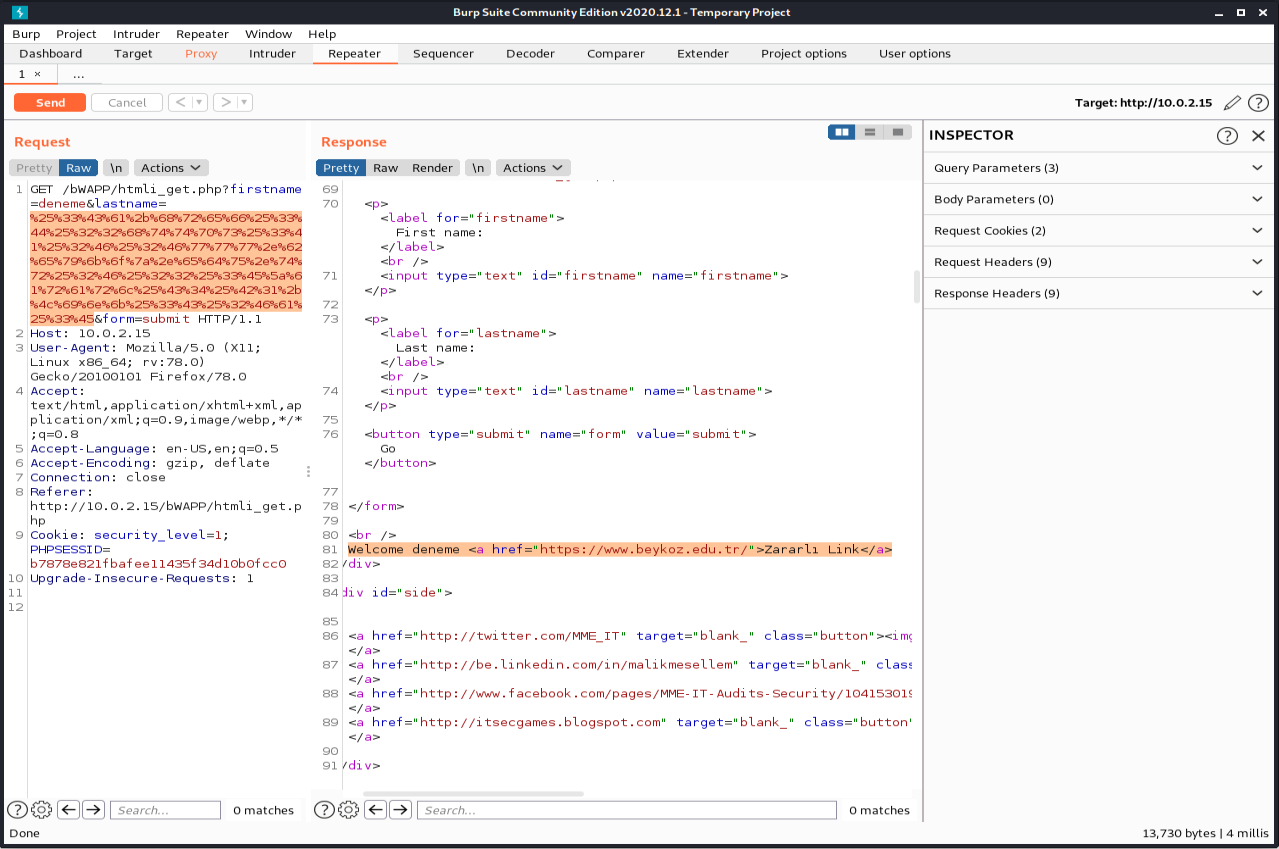
* Gördüğünüz gibi aşağıda URL encode formatında bir çıktı aldık.



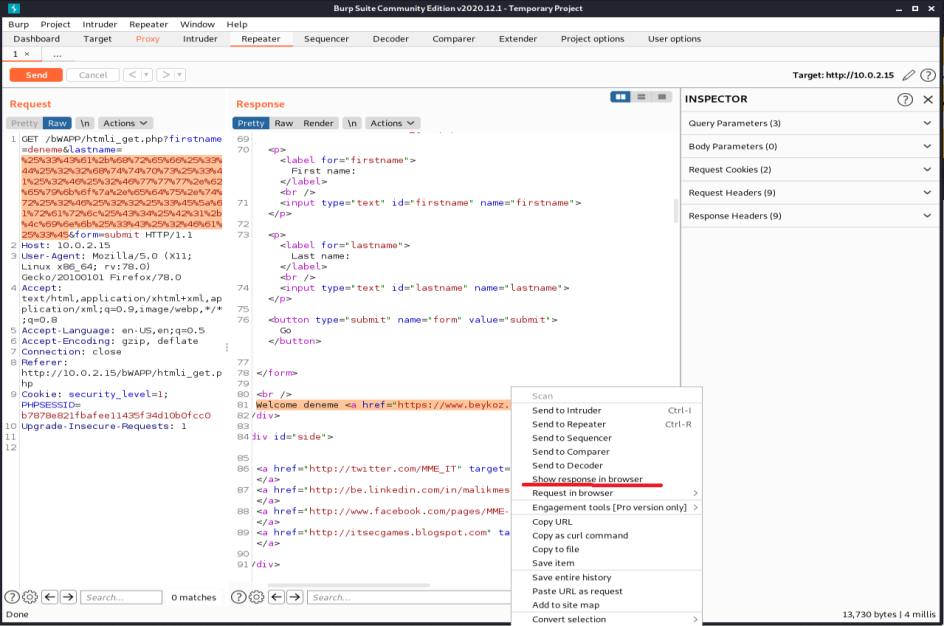
* Şimdi bu formatı kopyalayarak paketimize enjekte edip “Send” edelim.



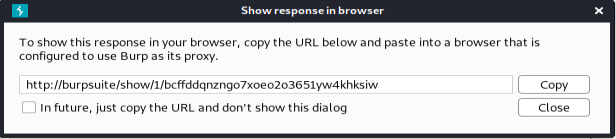
* Response kısmında yolladığımız Request in istediğimiz gibi çalışıp çalışmadığını kontrol edelim.



* Gördüğünüz gibi işaretli alanda “<a href="https://www.beykoz.edu.tr/">Zararlı Link</a>” kodunu başarılı bir şekilde enjekte ettik. Şimdi Response kısmına sağ tıklayarak “Show response in browser” i seçelim



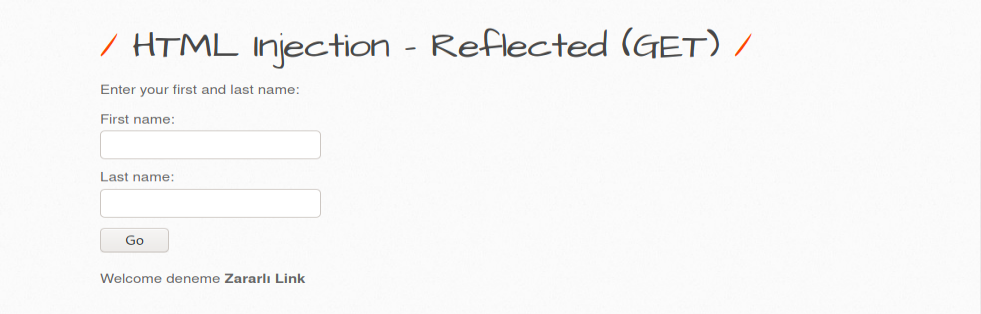
* Şimdi çıkan pencerede “copy” butonuna basarak linki kopyalayalım



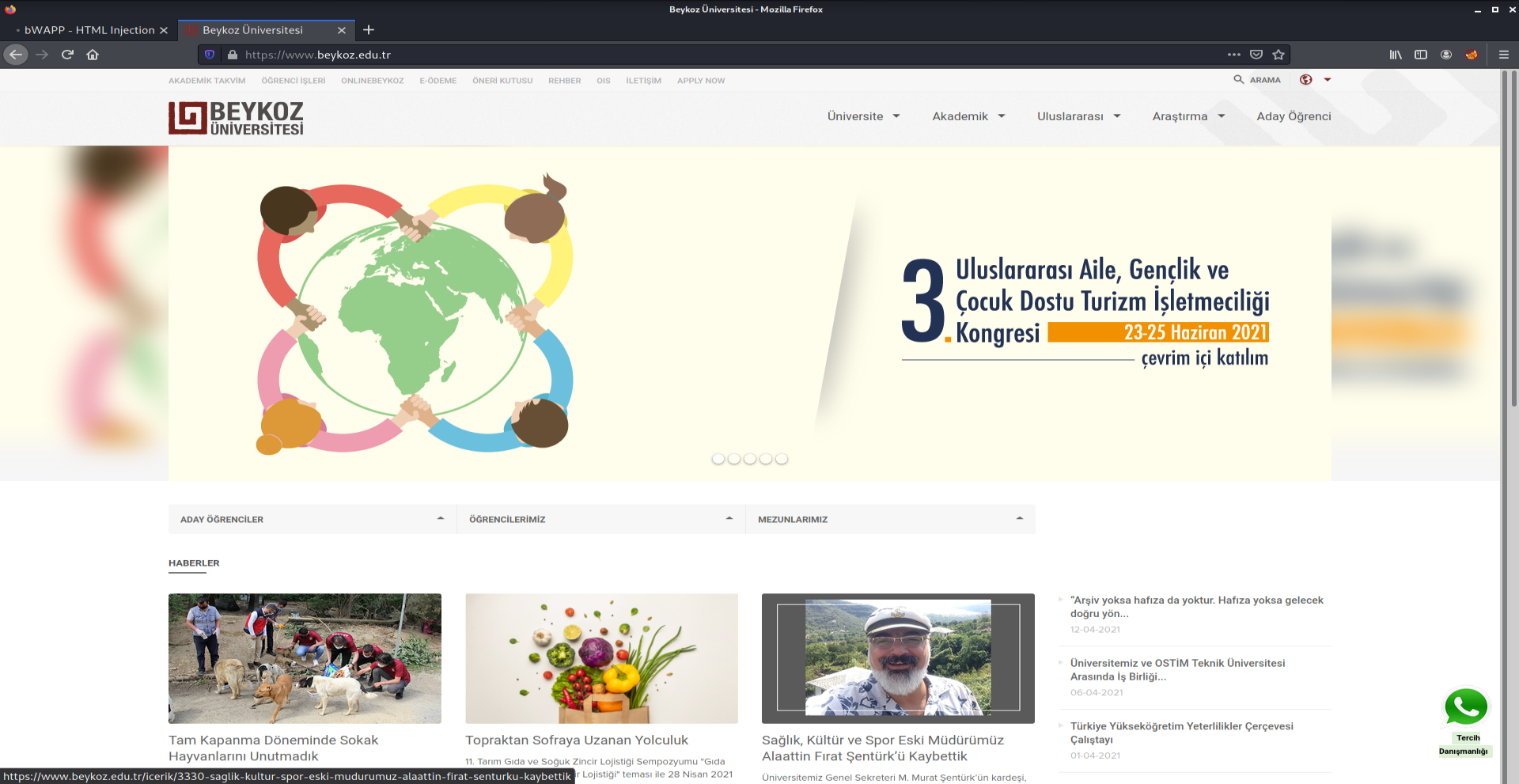
* Sırada ise link i tarayıcıya yapıştırark yolladığımız respone un görüntüsüne bakalım



* Aşağıdaki “Welcome deneme **Zararlı Link”** yazısı bize doğru yolda olduğumuzu gösteriyor. Şimdi ise “**Zararlı Link**” e tıklayarak enjekte ettiğimiz linke gitmeyi deneyelim.



* Ve yaptığımız örnek başarılı bir şekilde gerçekleşmiş oldu.



HTML INJECTION NASIL ÖNLENİR ?

Her alanda olduğu gibi bu saldırı vektörünün en zayıf halkası insandır. Genel olarak bu saldırının ana nedeni Geliştiricilerin bilgisizliği veya dikatsizliğidir. Bu tarz saldırılar giriş ve çıkışta düzgün bir şekilde doğrulanmadığından dolayı gerçekleşir. Bu nedenle ilk kuralımız , ugun veri doğrulamadır.

Eğer büyük bir proje ise geliştiricileri ve tester ları daha bilgili ve deneyimli kişilerden seçmeli veyahut gerekli bilgi verilmelidir.

Geliştiricilerin herhangi bir html karakterini engelleyip engellemediği denetlenmeli eğer engellenmemiş karakter var ise fixlenmelidir.

Yönetici veya yönetici izni olanlar dışında kesinlikle klasör, dosya ve kodlar üzerinde kullanıcının herhangi bir rolü olmamalıdır.

Bu saldırı vektörünün camia da fazla görülmediği bilgisini eklemek lazım. Keza fazla karşılaşılmadığından genelde bu tarz denetimler yeteri kadar hatta hiç yapılmaz. Bu vektörün az görülmesi sonucu bu vektör hakkında pek fazla kaynak bulunmaz, haliyle geliştiren veya test eden kişilerin pek fazla bilgisi olmaz. Fakat her vektör gibi bu vektör de büyük zararlar verebilir.

Kaynakça

<https://tr.myservername.com/html-injection-tutorial>

[**https://www.dogukankaradag.com/html-injection-saldiri-turleri-ve-onlemleri/**](https://www.dogukankaradag.com/html-injection-saldiri-turleri-ve-onlemleri/)

[**https://www.softwaretestinghelp.com/html-injection-tutorial/**](https://www.softwaretestinghelp.com/html-injection-tutorial/)