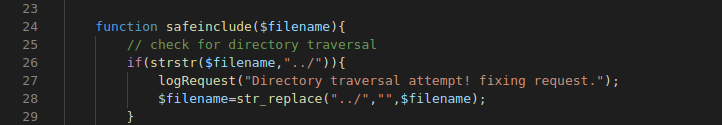
NATAS26

Natas 25 php kodlarını incelediğimiz zaman sayfanın sağ üst kısmında bulunan select elemanın değerine göre bize bir dosya getiriyor. Burada aklımıza ilk gelen açık dosya yollarını manüpüle edip *“etc/natas\_webpass/natas26”* dosyasının içeriği elde etmek (şifrelerin o dosya yolunda olduğunu önceki lablardan biliyoruz.). Path traversal işlemi için bizim linux komut satırına az da olsa aşina olmamız gerekiyor. Örneğin linux da üst klasöre dönmek için “cd ..” komutu kullanılır, veya üst klasörde bir dosya okumak için “cat ../{dosya\_ismi}” komutu ile üst klasöre geçmeden dosya okuma yapılabilir. Burada kullanacağımız payload genel olarak “../”dır. Peki bunu nasıl yedireceğiz?

Kodları incelediğimiz zaman setLanguage fonksiyonu eğer parametre olarak bir değer göndermiş isek o değeri parametre olarak safeinclude fonksiyonuna aktarıyor. Safeinclude fonksiyonu ise bazı basit validasyonlardan geçirerek göndermiş olduğumuz parametre ismindeki dosyayı ekrana basıyor.

Atlatmamız gereken ilk validasyon:



26. satırdaki if koşulu gönderdiğimiz parametrenin içinde “../” bulursa parametreden “../” karakterlerini kaldırıyor. Az önce demiştik ki payloadımız “../”dır. Bunu aşmanın yolu str\_replace fonksiyonuna takla attırmak.

<https://www.w3schools.com/php/phptryit.asp?filename=tryphp_func_string_str_replace>adresinden fonksiyonun tam olarak nasıl çalıştığını inceleyebiliriz.

Eğer parametre olarak “../” gönderirsek kodda oluşacak durum şu şekilde:



Ve bu kodun çıktısı boş olur çünkü “../” karakterini silmiş olur.

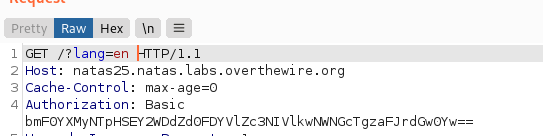
Bunu atlatmak aslında basit bir aldatmaya dayalı, madem fonksiyon gönderdiğimiz “../” karakterini siliyor, o zaman biz de “../” karakterini sildiği zaman ortaya gene “../” karakterini oluşturan bir payload gönderelim. “….//” payloadını gönderirsek ne olur?

-String ifadesi içinde “../” karakterini arar,

-İfadenin “3-5” karakterleri arasında “../” ifadesini bulur ve siler

-ifadenin içinde “../” karakteri kalır. Yani amacımıza ulaşmış oluruz.

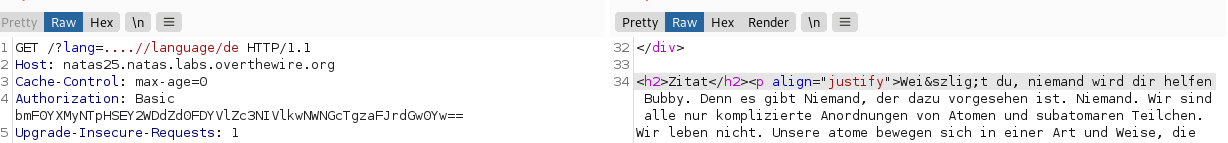
Burp suite aracında deneyelim:



Verisini gönderdiğimiz zaman bize standart olan ingilizce çıktıyı verecek. Kodları incelediğimiz zaman en dosyası language klasörünün altında bulunuyor. “….//language/de” parametresini gönderdiğimiz zaman bize farklı dildeki dosyayı getirse o zaman payloadımız işe yaramış demektir. Peki neden “….//language/du” çünkü kod otomatik olarak şöyle bir dosya yolu oluşturuyor:



Yani otomatik olarka language dosyasına giriyor. Gönderdiğimiz payload çalışırsa eğer ortaya çıkan dosya yolu şu şekilde oluyor. “language/../language/de” yani language klasörün üst dizinine çıkıyor ve tekrar language klasörüne giriyor ve de dosyasını okuyor.



Payload başarılı bir şekilde çalıştı. Peki ondan sonra ? (global parametreler)