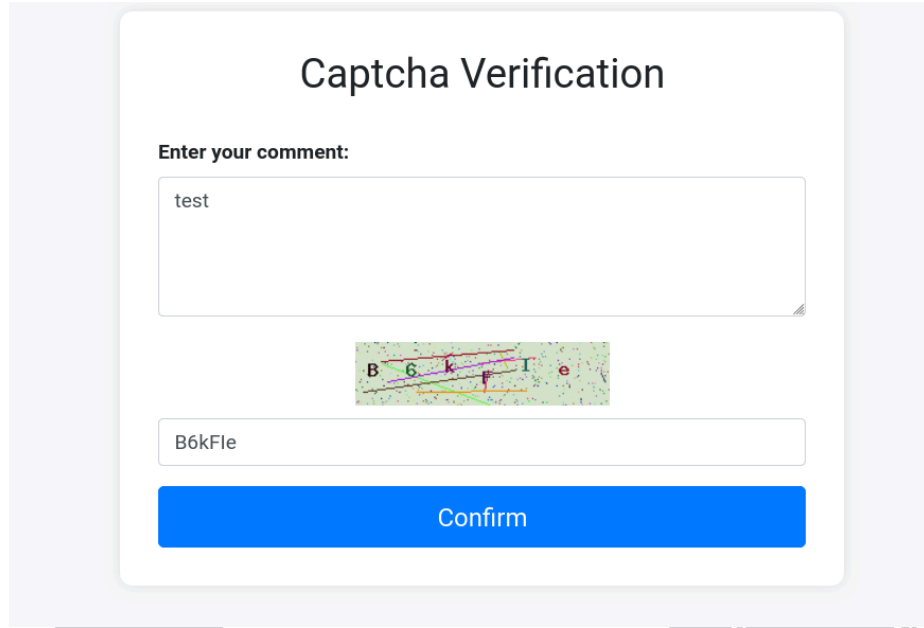


YAVUZLAR WEB GÜVENLİĞİ & YAZILIM TAKIMI

CAPTCHA BYPASS ÖDEV RAPORU

1. Captcha Bypass

“Captcha Bypass” laboratuvarına girildiğinde yorum kaydetmek için captcha doğrulaması gerektiği fark edildi.



Şekil 1.1. Laboratuvar Ekran Görüntüsü


Captcha'nın doğrulamamızı beklediği değeri girip onayladığımızda yorumumuz arkaplanda doğrulanarak kaydedildi.

Captcha Verification

Captcha verification successful, note added.

Enter your comment:

sevgilerden bir demet!



B6kFle

Confirm

Reset Table

#	Comments
1	Basit Düşün.)
2	Kaynak Kodlardan yararlan!
3	test

Şekil 1.2. Doğrulanmış Captcha Ekran Görüntüsü


Doğrulan yorumumuz kaydedildikten sonra yeni bir yorum kaydetmek için tekrar captcha doğrulamasından geçmemiz gerekti. Captcha doğrulaması için bir önceki captcha doğrulamasında kullanılan değer girildiğinde girilen yorum onaylandı ve kaydedildi.

Captcha Verification

Captcha verification successful, note added.

Enter your comment:

Write your comment here



Enter the captcha

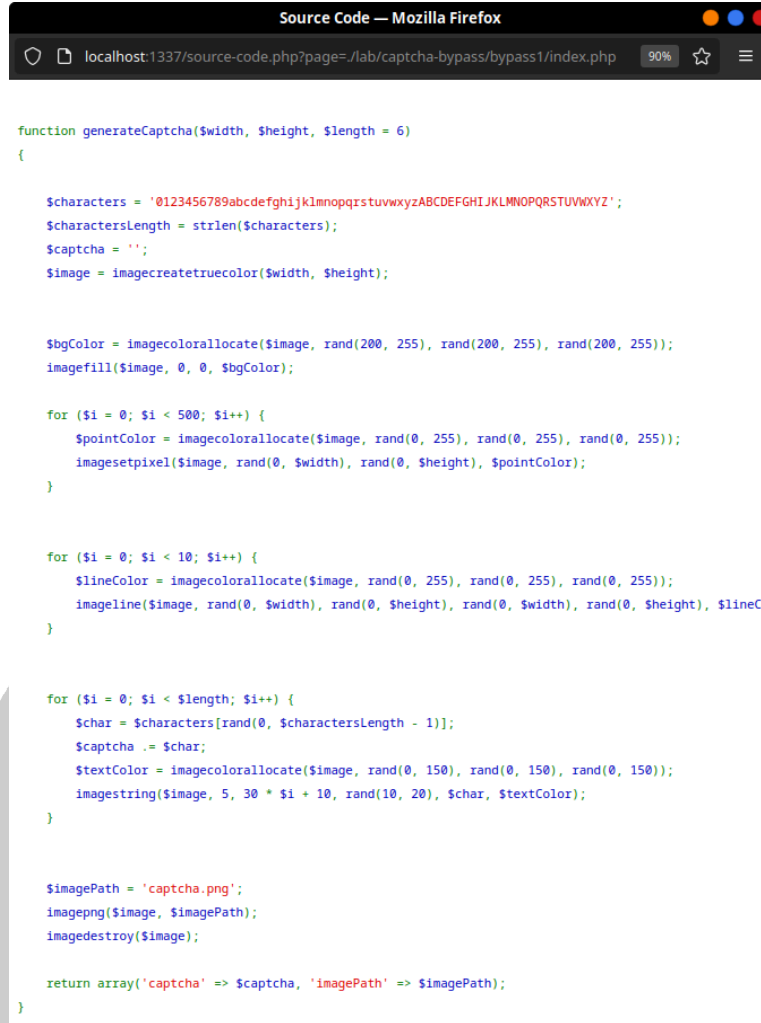
Confirm

Reset Table

#	Comments
1	Basit Düşün.)
2	Kaynak Kodlardan yararlan!
3	test
4	sevgilerden bir demet!

Şekil 1.3. Başarılı Captcha Bypass Ekran Görüntüsü

Böylelikle laboratuvar üzerinde Captcha Bypass zafiyeti uygulamış olduk.



```
function generateCaptcha($width, $height, $length = 6)
{
    $characters = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';
    $charactersLength = strlen($characters);
    $captcha = '';
    $image = imagecreatetruecolor($width, $height);

    $bgColor = imagecolorallocate($image, rand(200, 255), rand(200, 255), rand(200, 255));
    imagefill($image, 0, 0, $bgColor);

    for ($i = 0; $i < 500; $i++) {
        $pointColor = imagecolorallocate($image, rand(0, 255), rand(0, 255), rand(0, 255));
        imagepixel($image, rand(0, $width), rand(0, $height), $pointColor);
    }

    for ($i = 0; $i < 10; $i++) {
        $lineColor = imagecolorallocate($image, rand(0, 255), rand(0, 255), rand(0, 255));
        imageline($image, rand(0, $width), rand(0, $height), rand(0, $width), rand(0, $height), $lineColor);
    }

    for ($i = 0; $i < $length; $i++) {
        $char = $characters[rand(0, $charactersLength - 1)];
        $captcha .= $char;
        $textColor = imagecolorallocate($image, rand(0, 150), rand(0, 150), rand(0, 150));
        imagestring($image, 5, 30 * $i + 10, rand(10, 20), $char, $textColor);
    }

    $imagePath = 'captcha.png';
    imagepng($image, $imagePath);
    imagedestroy($image);

    return array('captcha' => $captcha, 'imagePath' => $imagePath);
}
```

Şekil 1.4. Laboratuvar Kaynak Kodu

Laboratuvarın kaynak kodlarından Captcha oluşturan fonksiyonunu (generateCaptcha) incelediğimizde Captcha oluşturulurken Captcha için herhangi bir geçerlilik tarihi veya ömür süresinin belirtilmediği görülüyor. Captchalar sadece session bitince siliniyor. Bu da daha önceden oluşturulmuş Captcha cevaplarının diğer Captchalar’da da kullanılabilmesine ve Captcha Bypass zafiyetine sebep oluyor. Captchalar oluşturulurken birkaç dakikayı geçmeyecek şekilde ömrünün belirlenmesi bu zafiyetin oluşmasını engelleyecektir.

2. Broken Captcha

“Broken Captcha” laboratuvarında önceki laboratuvara benzer şekilde kayıt işlemi gerçekleştirebilmemiz için Captcha doğrulamasını geçmemiz gerekiyor. Fakat bu laboratuvardaki Captcha doğrulaması matematiksel bir işlem gerektiriyor. Captcha’yı onlarca kez yenileyince matematiksel işlem olarak sadece toplama sorduğunu gözlemliyoruz.

Send Us Your Message

Name-Surname

Forward Message

Captcha

9 + 7 = ? Refresh

Send

View Messages

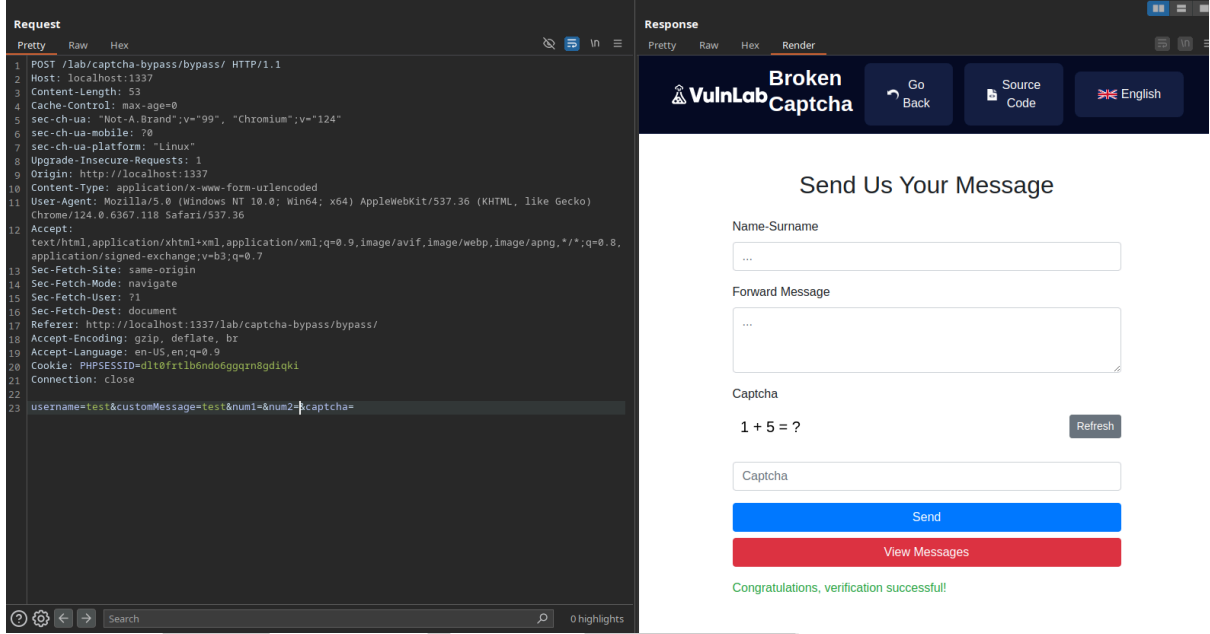
Şekil 2.1. Laboratuvar Ekran Görüntüsü

Formu doldurup gönderdikten sonra isteği, Burp Suite üzerinde dinlediğimizde aşağıdakine benzer bir istekle karşılaşıyoruz.

```
Request
Pretty Raw Hex
1 POST /lab/captcha-bypass/bypass/ HTTP/1.1
2 Host: localhost:1337
3 Content-Length: 48
4 Cache-Control: max-age=0
5 sec-ch-ua: "Not-A.Brand";v="99", "Chromium";v="124"
6 sec-ch-ua-mobile: ?0
7 sec-ch-ua-platform: "Linux"
8 Upgrade-Insecure-Requests: 1
9 Origin: http://localhost:1337
10 Content-Type: application/x-www-form-urlencoded
11 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
12 Chrome/124.0.6367.118 Safari/537.36
13 Accept:
14 text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,
15 application/signed-exchange;v=b3;q=0.7
16 Sec-Fetch-Site: same-origin
17 Sec-Fetch-Mode: navigate
18 Sec-Fetch-User: ?1
19 Sec-Fetch-Dest: document
20 Referer: http://localhost:1337/lab/captcha-bypass/bypass/
21 Accept-Encoding: gzip, deflate, br
22 Accept-Language: en-US,en;q=0.9
23 Cookie: PHPSESSID=dlt0frtlb6ndo6ggqrn8gdiqki
24 Connection: close
25
26 username=&customMessage=&num1=9&num2=7&captcha=
```

Şekil 2.2. Gönderilen İsteğin İçerik Görüntüsü

Burada isteğin body kısmında Captcha tarafından oluşturulan “num1” ve “num2” değerlerinin olduğunu görüyoruz. Captcha bu değerlerin sadece toplamını kontrol ettiği için bu değerleri “null” veya “0” olarak değiştiriyoruz ve isteği gönderiyoruz.



Şekil 2.3. Gönderilen İsteğin Sonucu

Gönderdiğimiz istekte Captcha tarafından toplanacak değerleri boş olarak değiştirdiğimiz için değerler toplandığında doğrulamayı sağlıyor ve Captcha'yı aşmış oluyoruz. Böylelikle web sitesi üzerinde Burp Suite ile Captcha'ya takılmadan istediğimiz gibi istek atabiliriz bu da sunucu tarafında şişmeye sebep olacaktır.

Captcha'nın düzgün çalışabilmesi için matematiksel işlem çeşitliliğin artırılması ve değerlerin kullanıcı tarafından erişilemez bir şekilde tutulması gerekmektedir.