hgame LoRexxar的渗透之战wp

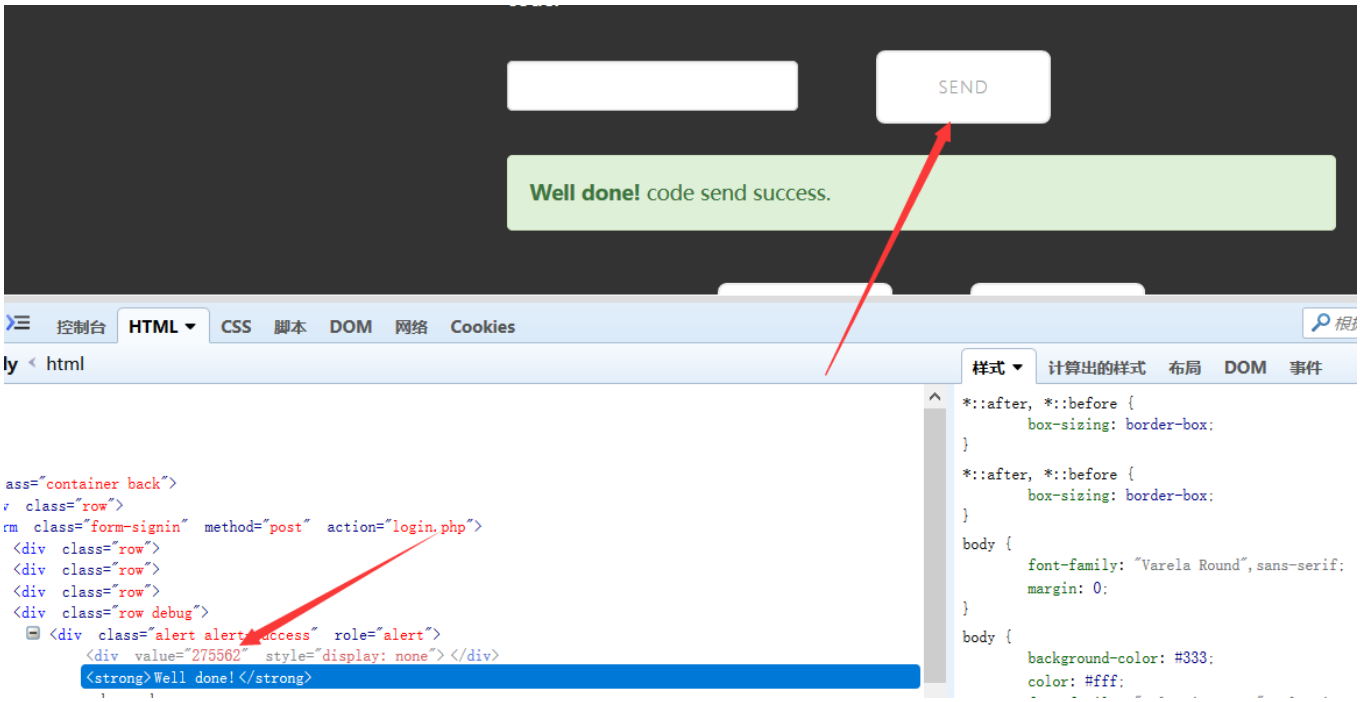
hgame

由于hgame是面向新生的ctf竞赛，所以pentest类的题目尤为难出，稍难的题目不但没人做的出来，甚至基础知识都没学过，所以在基础的web题目出完之后，我写了一个比较大的站来接触下之前以web题目形式出现的漏洞，也展示了一些日站才能接触到的漏洞

LoRexxar的渗透之战之一 POINT: 50
题目ID： 70
题目描述： 从这个题目开始将会是一个大型的渗透系列题目，前面遇到的web漏洞将会以不同的方式出现在题目中，你能抓住那些漏洞吗?
1、这个站才刚刚开始写，好像还什么都没有啊>x<
http://115.28.78.16:13333/d23fd789868fa2c8b3942a811f63adb7/0x01/
Hint: 1、这个站还没写完，也就意味着很多接口还是不能用的，那程序员怎么写代码呢？

一是个比较有趣的洞，也有不少人在实战中遇到过，稍微逛逛整个站，可以发现，注册的时候需要账号密码还有手机，而登陆的时候，会向手机发送验证码，而事实上，这个功能并没有写。

实战中也能遇到这样的问题，在一个站点还处于开发中，关于邮件和手机验证码的接口往往还没有完成，一般来说，部分开发人员会把验证码无效化，但也有部分开发人员会把验证码返回到前台来，所以f12看源码就能发现发送回来的验证码。



登陆成功就能看到flag了。

这里有个有意思的洞，让我们来看看源码

```
api/code.php

<?php
session_start();

if(!isset($_POST)){
    echo "hello, hacker, you can't go here....";
}

$code = (string)rand(100000,999999);

$_SESSION['code'] = $code;

echo "<div class=\"alert alert-success\" role=\"alert\">
        <div value=\"$code\" style=\"display: none\"></div>
        <strong>Well done!</strong> code send success.
      </div>"

?>
```

登陆时候的校验是

```
    if($code-0 != $_SESSION['code']){
        echo "<script>alert('code is wrong!')</script>";
        echo "<script>window.location.href='./login.php'</script>";
        exit;
    }
```

结果就是，如果code传入字符串或者不传入，然后不点击check来获取验证码直接登陆也是可以绕过的...

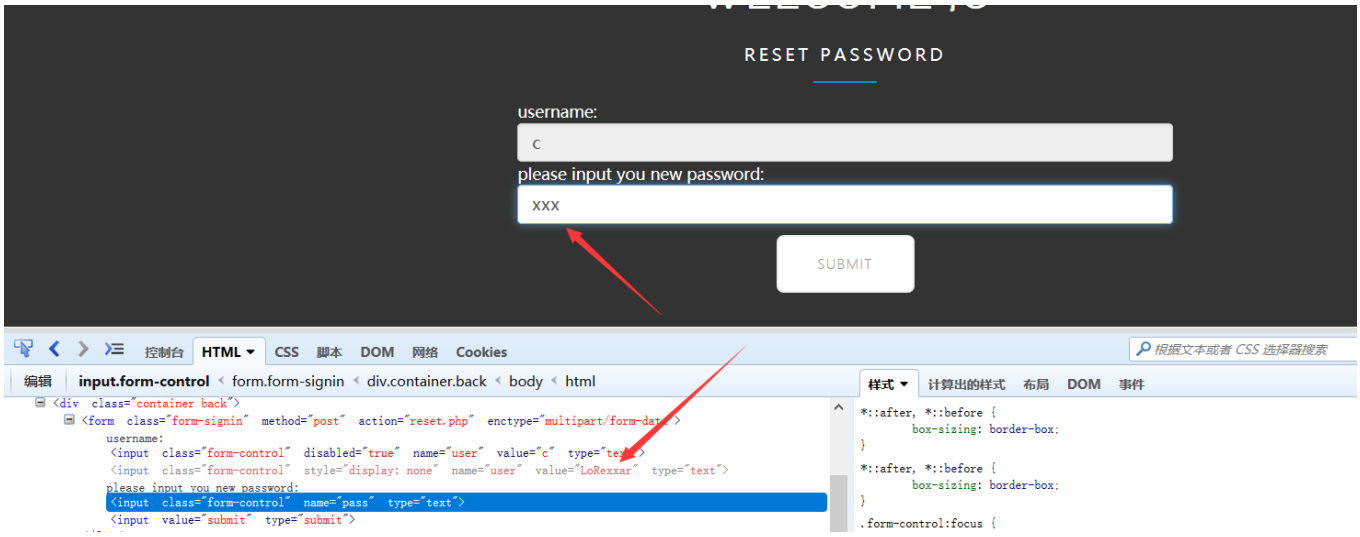
二

LoRexxar的渗透之战之二 POINT: 150
题目ID： 73
题目描述： 看上去站好像完整多了啊，那么哪里有问题呢？
http://115.28.78.16:13333/d23fd789868fa2c8b3942a811f63adb7/0x002/
Hint: 1、LoRexxar才握着真正的力量
2、渗透题各个题目单独存在，基本不存在漏洞通用
3、业务逻辑漏洞

可能你还不知道什么是业务逻辑漏洞，我稍微解释一下就是在代码逻辑层产生的漏洞，而不是代码本身的漏洞。

这样的漏洞盲测应该会比看源码简单很多，这里是一个修改密码的业务逻辑漏洞。

我们看到修改密码这里是2步:
1、先校验旧密码，校验成功后，进入第二步
2、输入新的密码，修改成功



通过修改隐藏标签中的value，来修改LoRexxar账号的密码

登陆LoRexxar的账号就能看到flag了

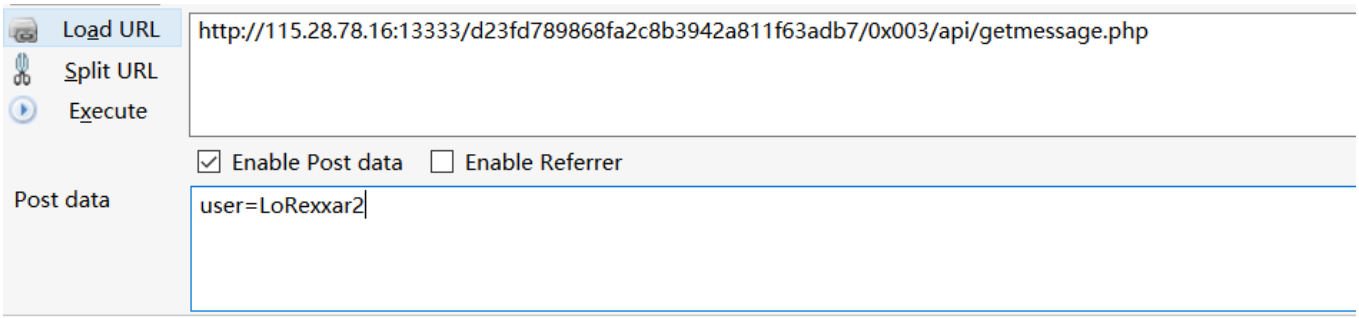
三

LoRexxar的渗透之战之三 POINT: 100
题目ID： 74
题目描述： LoRexxar最近觉得这个站不太优雅，所以决定试试看ajax解决方案
http://115.28.78.16:13333/d23fd789868fa2c8b3942a811f63adb7/0x003/
Hint: 1、LoRexxar2才握有真正的力量
2、这题可能有很多做法，那么最简单的是什么？
3、水平越权

这里漏洞点我觉得说的挺明白，正常随便注册个账号，然后给自己发送一些信息，然后在user页面，我们发现通过ajax的方式，获取了messages数据

最重要的是，我们发现请求的时候戴上了用户名

于是试试看请求别人的信息



- hctf{rugu0_y0u_p1ngx1ngyu3qu4n_du0l1h4i}

成功了

四

LoRexxar的渗透之战之四 POINT: 400

题目ID： 75
题目描述： 看上去好像没什么问题了啊，可惜，你们对力量一无所知o(^▽^)┛
http://115.28.78.16:13333/d23fd789868fa2c8b3942a811f63adb7/0x004/
Hint: 1、你可能对力量一无所知
2、sql注入

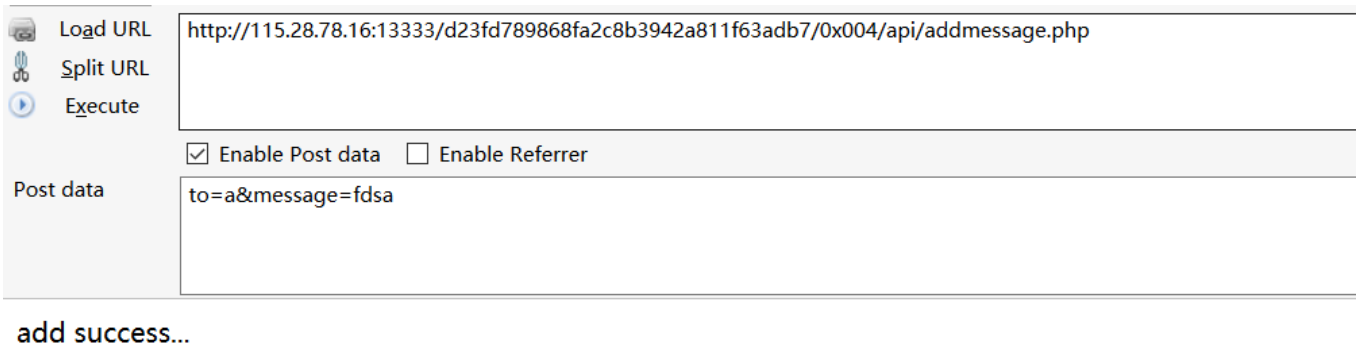
这可能是这个系列中最难的一题了，不过后来我也放了hint，这是一个sql注入题目，首先得找到漏洞点，稍微测试下，我们发现三的漏洞点改成了get，所以有输入而且会经过数据库的输入点只有发送信息了。

直接输入测试的话，会发现并没有发生什么，观察页面，我们发现了前端页面的转义

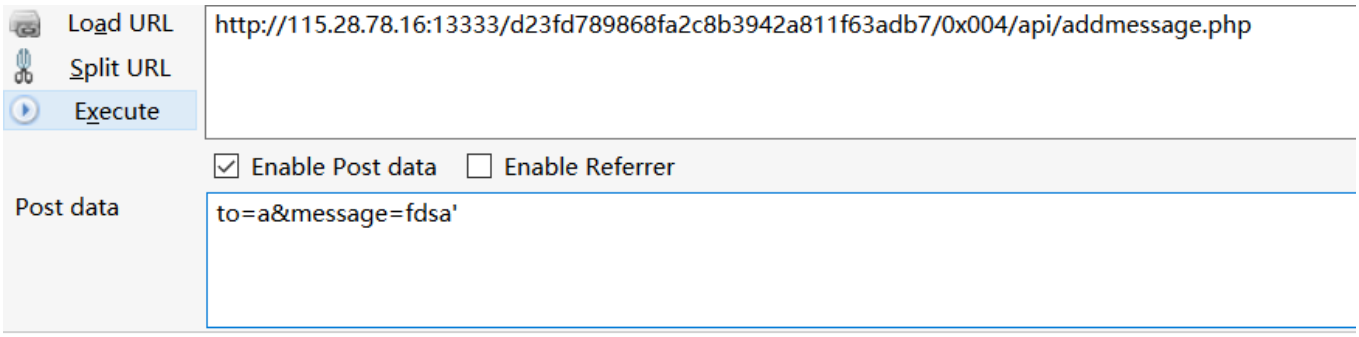
```
$(document).ready(function(){
 function trim(str){
        return str.replace(/(^\s*)|(\s*$)/g, "");
    }

    function addslashes(str) {
    str = str.replace(/\\/g, '\\\\');
    str = str.replace(/\'/g, '\\\'');
    str = str.replace(/\"/g, '\\"');
    str = str.replace(/\0/g, '\\0');
    return str;
    }

    function stripslashes(str) {
        str = str.replace(/\\'/g, '\'');
        str = str.replace(/\\"/g, '"');
        str = str.replace(/\\0/g, '\0');
        str = str.replace(/\\\\/g, '\\');
        return str;
    }

    var user = $("h1#user").text();
    user = trim(user.substring(9));

    $.get("api/getmessage.php", function(result){
  $("ul#message").html(result);
    });

    $("input#submit").click(function(){
  var to = addslashes($("form#subm input").val());
```

```
    var message = addslashes($("form#subm textarea").val());

          $.post("api/addmessage.php", {
              to: to,
              message: message,
          },function(result){
              if(result.indexOf("success") > 0){
                  console.log(result);
                  history.go(0);
              }else{
                  alert(result);
              }

          });
      });


});
```

也就是说我们直接对api操作的话，就可以绕过前段的过滤。



add success...



maybe gg...

由于这里是个insert注入，所以注法比较特别，猜测后台语句为

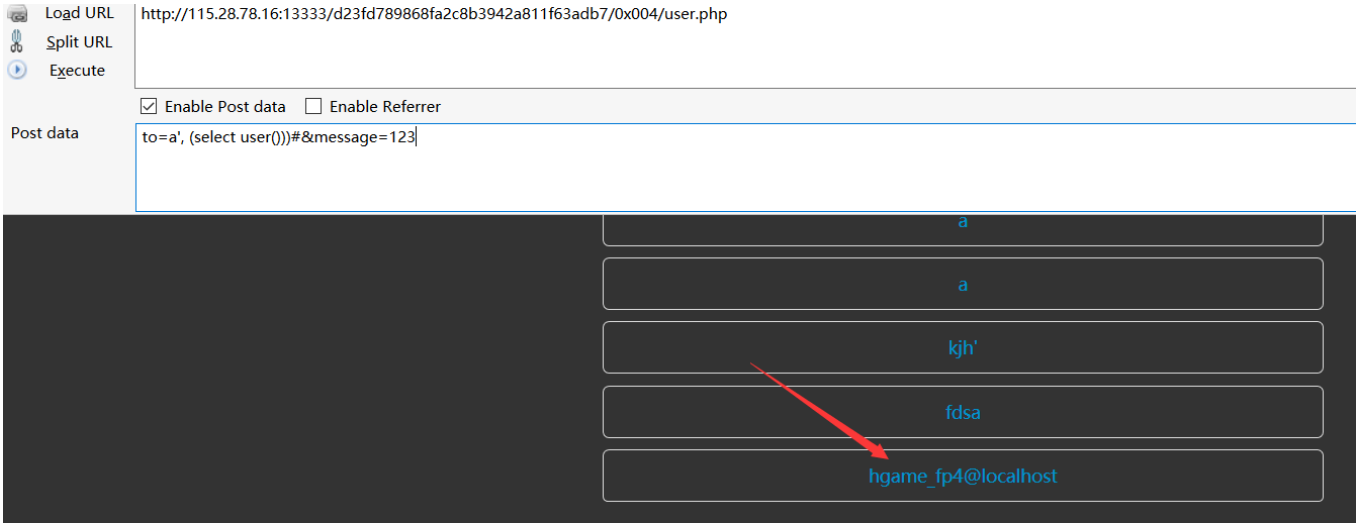
```
$query = "insert into m (fuser, user, message) values ('{$user}', '{$to}', '{$message}')";
```

我们的可控点为to和message

那我们可以构造语句为

```
$query = "insert into m (fuser, user, message) values ('{$user}', 'a', (select 'test'))#)', '{123}')";
```

构造显注



我这里就不演示了，flag在另一个表里

五

LoRexxar的渗透之战之五 POINT: 300
题目ID： 76
题目描述： 看上去好像没什么问题了，LoRexxar也会常常上线和大家聊天的
http://115.28.78.16:13333/d23fd789868fa2c8b3942a811f63adb7/0x005/
Hint: 1、你能从LoRexxar4获得真正的力量吗
2、xss，你可能需要注意CSP

提示已经很明显了，是xss，其实我看后台时候有很多同学都是对的payload，可能没注意我修改过的题意，很多同学都发给了LoRexxar账号

这题我就不说太多了，过滤测试很快就能得到，复写就可以绕过，由于CSP的原因，所以打到的cookie只有两种方式可以发送到自己手里，一种是跳转，一种是构造post发送站内信，都很简单，有兴趣可以看看我博客中的别的文章