hgame week2

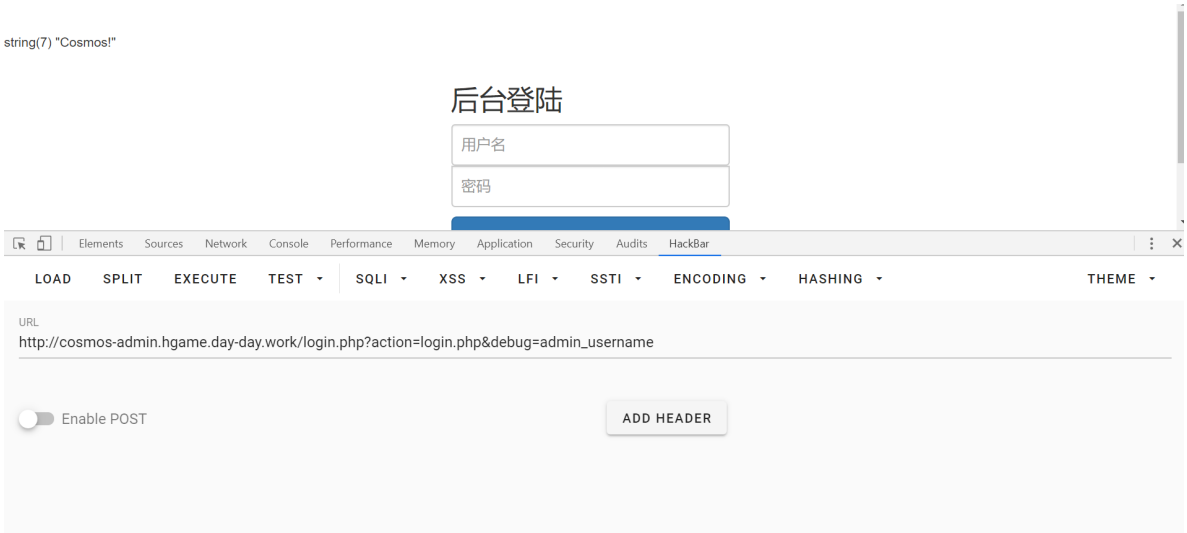
web

cosmos的博客后台

> 查看网页源代码没有什么收获，尝试伪协议读源码，构造 http://cosmos-admin.hgame.day-day.work/index.php?action=php://filter/read=convert.base64-encode/resource=login.php ，发现base64编码后的源代码，同理获得index.php和admin.php的源代码。

> 本来想直接读取config.php但是在index.php里被filter了，无法直接读取用户名和密码的具体值，于是继续进行代码审计。
>
> 发现debug参数传入变量名会显示变量值，在login.php里得知用户名和密码的变量为admin_username和admin_password，借助debug获取变量值。

```
//login.php
<?php
include "config.php";
session_start();

//Only for debug
if (DEBUG_MODE){
    if(isset($_GET['debug'])) {
        $debug = $_GET['debug'];
        if (!preg_match("/^[a-zA-Z_\x7f-\xff][a-zA-Z0-9_\x7f-\xff]*$/", $debug))
{
            die("args error!");
        }
        eval("var_dump($$debug);");
    }
}

if(isset($_SESSION['username'])) {
    header("Location: admin.php");
    exit();
}
else {
    if (isset($_POST['username']) && isset($_POST['password'])) {
        if ($admin_password == md5($_POST['password']) && $_POST['username'] ===
$admin_username){
            $_SESSION['username'] = $_POST['username'];
            header("Location: admin.php");
            exit();
        }
        else {
            echo "用户名或密码错误";
        }
    }
}
?>
```



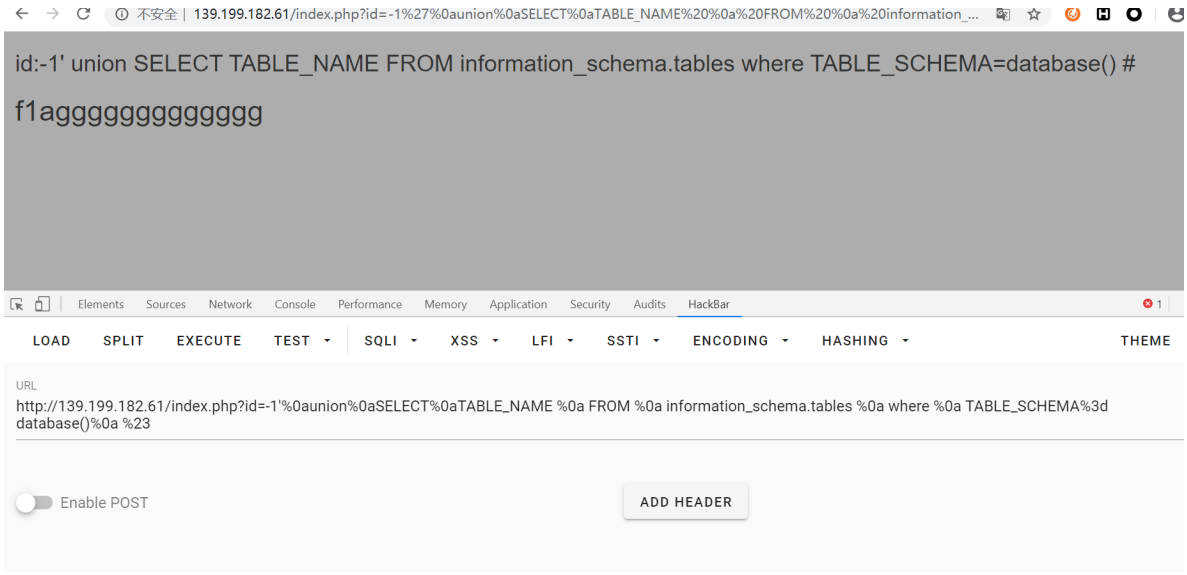
> 其中密码要求MD5后和原来密码相同，这里涉及到php里md5的绕过和==的弱比较，因为密码是0e开头，在和另一个字符串比较的时间会变成零，所以找一个MD5后也是0e开头的传入即可。

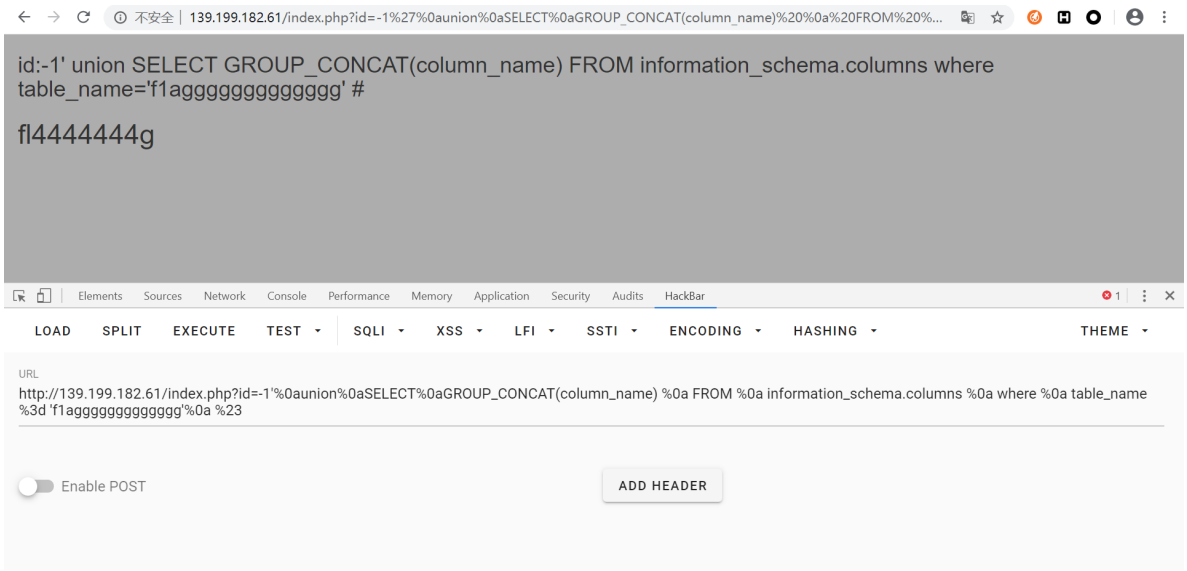
```
//admin.php
<?php
include "config.php";
session_start();
if(!isset($_SESSION['username'])) {
    header('Location: index.php');
    exit();
}

function insert_img() {
    if (isset($_POST['img_url'])) {
        $img_url = @$_POST['img_url'];
        $url_array = parse_url($img_url);
        if (@$url_array['host'] !== "localhost" && $url_array['host'] !==
"timgsa.baidu.com") {
            return false;
        }
        $c = curl_init();
        curl_setopt($c, CURLOPT_URL, $img_url);
        curl_setopt($c, CURLOPT_RETURNTRANSFER, 1);
        $res = curl_exec($c);
        curl_close($c);
        $avatar = base64_encode($res);

        if(filter_var($img_url, FILTER_VALIDATE_URL)) {
            return $avatar;
        }
    }
    else {
        return base64_encode(file_get_contents("static/logo.png"));
    }
}
?>
```

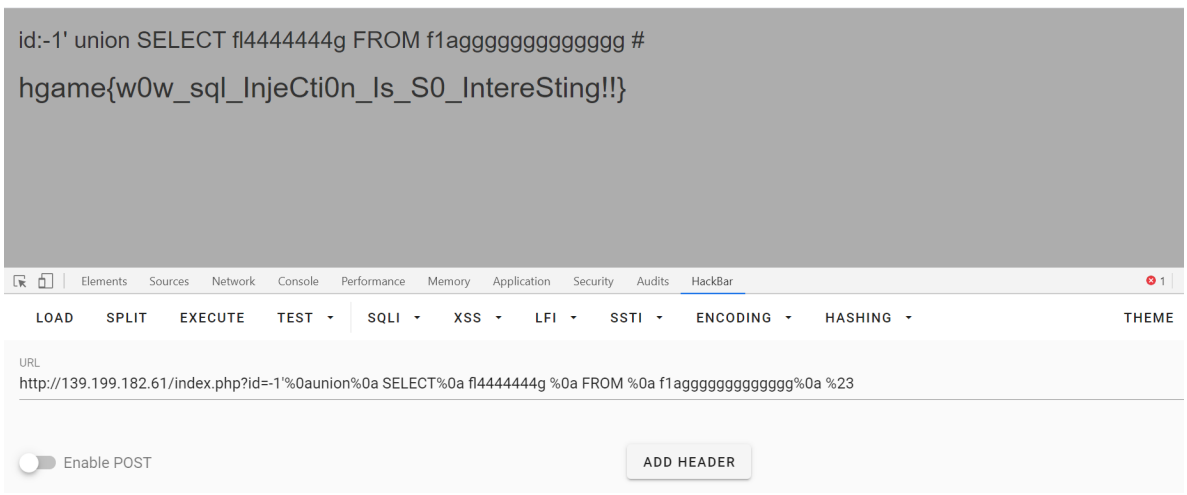
> 登录成功之后，代码审计admin.php，没有对imagl_url进行任何限制和处理，属于ssrf攻击，尝试了不同的协议dict，file，gopher；后来发现自己想复杂了，payload构造img_url=file://localhost/flag就可以了...
>
> 网页源代码里把图片的base64值解密得出flag

cosmos的留言板

> 首先根据题目描述，感觉应该是sql注入。
>
> 尝试id=1' union select database() #发现空格和小写select还有#被过滤，采用大写SELECT和%0a，%23，绕过
>
> 先用order by确认返回字段的数量，发现只有1，然后确认返回位置；接下来就一步一步注入获取表名和字段名。







misc

cosmos的午餐

> 打开发现pcapng文件和一个log文件，目测是流量审计的题目，通过查找资料后发现log文件是用来解密tsl，解密后，在导出对象http里看看有没有压缩包之类的，发现有x-www-form-urlencoded（主要和里面其他文件类型不同，所以点开看看），发现pk开头是压缩包！

```
Server: AmazonS3

PUT /534d46e63462e9032827eb34f8ad713920200118125937/e418689cfbcc624a45f53e32ffe31f8b15d211c5?
partNumber=1&uploadId=Z7WOUeW0sU0lTxoRgWIGhgldnB8OJBMtJIqvLG4DKRAgfi8i46CU6480BKvO33URN9SP1E9EXzOz3
SArL3A43lGqzk.Tq2ditYOsnHsxTVGGtUk9Is9vi4xItCEx3vas&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Credential=AKIAJMSYV525U4GRIMRQ%2F20200118%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-
Date=20200118T125939Z&X-Amz-Expires=3600&X-Amz-SignedHeaders=host&X-Amz-
Signature=bdce3fc539028c51dbeb84990b890ec1e093915395bfc01684153506aa4a8a51 HTTP/1.1
Host: wetransfer-us-prod-outgoing.s3.amazonaws.com
Connection: keep-alive
Content-Length: 1220624
Accept: application/json, text/plain, */*
Origin: https://wetransfer.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/
79.0.3945.117 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Sec-Fetch-Site: cross-site
Sec-Fetch-Mode: cors
Referer: https://wetransfer.com/
Accept-Encoding: gzip, deflate, br
Accept-Language: zh-CN,zh;q=0.9

PK.........z2P....`...z.......Outguess with key.jpg..WX.Q..."...E.RD@...E..H.Az...P.H..
EP.^".....          ..o..>.......w3sg..9...{.9w.......*.*."""".K...B..(C....--....
..G8.........?.....-h..@J8GE.....l...o~.....7.............o~.....7...4.v.?......`.x...X.
6[..@.`.p$.....T.............................]HDPX. ...[.....Y.98....4!x.9.....k         k.+.
9:..{...?.....H.....)..2P.w.;o.{P.W.^2Py..<].p...B.
r...2.OU...P.......D..X...AHDZZZHXTHTT..B.....
*....!. '.w1........o.[...o!.....[...o.[....^.?\.;W...!8"Ws.%.......\..5.?..\.F|.:.....)..d.Bz.:.%.
9..!.QQRP......n..]..$...I).m...n...$   '&....&..M..N.;!.CB........5....5(
jh           .....&!hL..D...v...}....u.H9<...I.B.....Nw.....3.............|b...R.2JO.UT..5..
...MLml........}|!P?...a..."...>$..~LK....._PXT\YU]S[W...........otl|brjzf........kk.}xt|rz.
9...]D.b...O..I....~......!.np...}....uH.<ns..!..$.KE.9.c].....=.....g..b..fX..d..
```

> 压缩包下载后解压发现是一张图片，名字outguess with key，提示要用outguess，key根据题目描述在图片备注里，打开虚拟机ubuntu
>
> outguess -kk "gUNrbbdR9XhRBDGpzz" -r "kkk.jpg" iiit.txt
>
> 得到文件 打开后显示https://dwz.cn/69rOREdu
>
> 进入后下载了一个压缩包，打开一个二维码，扫描得到flag

所见即为假

> 根据题目描述还有压缩包名字fake，怎么看都感觉像伪加密.... 尝试直接打开发现要密码，用7zip直接打开伪加密文件；另外压缩包给了提示F5 key: NlID7CQon6dBsFLr
>
> 继续用f5对图片进行解密，得出一大串：
> 526172211A0701003392B5E50A01050600050101808000B9527AEA2402030BA70004A70020CB5BDC2D80000008666C61672E7478740A03029A9D6C65DFCED5016867616D657B343038375E7A236D737733344552746E46557971704B556B32646D4C505736307D1D77565103050400
>
> 放入hex解码后得出flag

地球上最后的夜晚

> 解压后一个pdf一个压缩包，由于最近一直在做各种隐写，首先先去查查pdf隐写，发现wbs43open这个工具，联系到pdf文件名no password解密无需密码，直接用这个工具解密，得到Zip Password: OmR#O12#b3b%s*IW，打开发现是word文档，在压缩包里右键查看文件直接进入隐藏的zip，发现是一些xml文件；查询后得知flag一般放在word文件下的document里，进入word文件夹后发现secret.xml，在里面获得flag