www.qconferences.com
www.qconbeijing.com

QCon北京2014大会 4月17—19日

伦敦 | 北京 | 东京 | 纽约 | 圣保罗 | 上海 | 旧金山

London · Beijing · Tokyo · New York · Sao Paulo · Shanghai · San Francisco

QCon全球软件开发大会

International Software Development Conference

www.infoq.com/cn

InfoQ

@InfoQ

infoqchina

软件
正在改变世界！

特别感谢
QCon上海合作伙伴

云WAF内功修炼

安全宝云WAF实践分享
@imiyoo

Who Am I ?

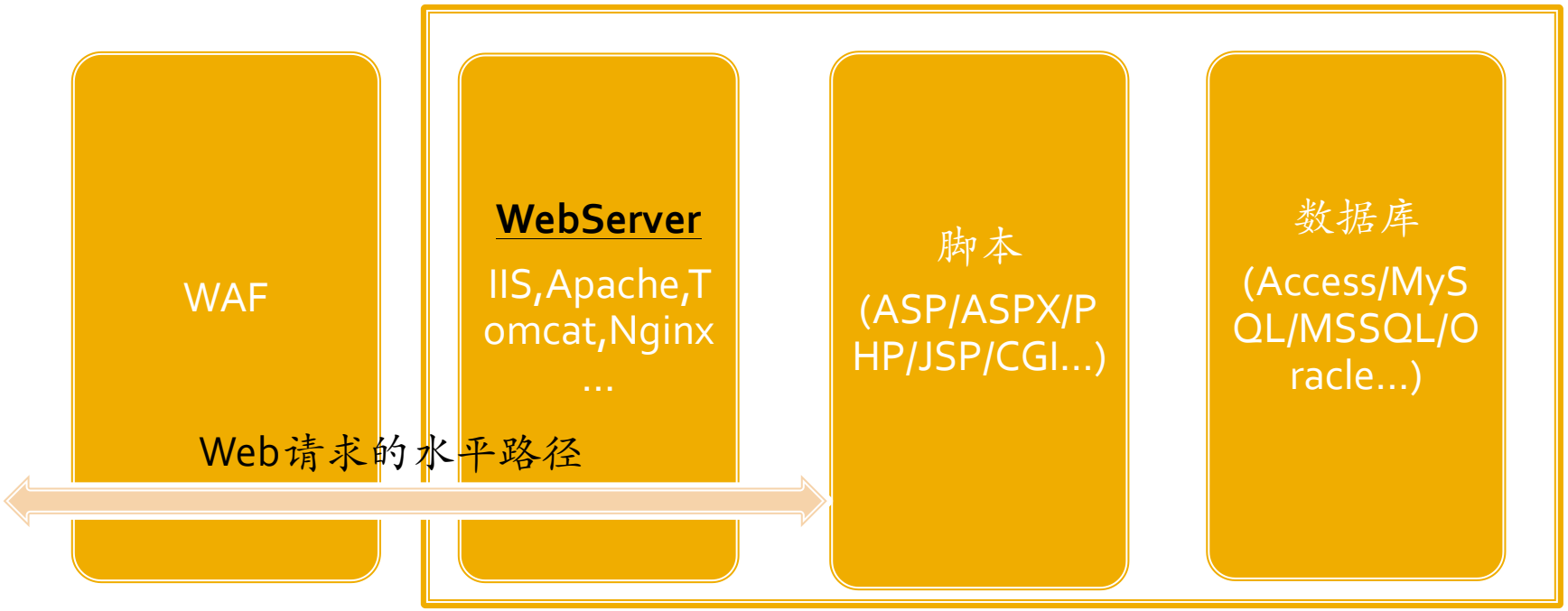
- 刘漩
  - 五年安全从业经验
  - 2010~ 安全宝 安全研究员
  - 国内安全团队安天365核心成员
  - 早期作品:
    - imiPhoneWall Android手机防火墙(已开源)
    - WatScan网站在线扫描器
  - 微博:@imiyoo
  - 博客:www.imiyoo.com

WAF的疑问

- WAF是什么，它有什么作用?
- 我们需要它吗?
- 为什么有了WAF还会被入侵?

什么是WAF

- Web Application Firewall
  - Web应用的中间层，对Web请求进行攻击过滤



WAF的划分

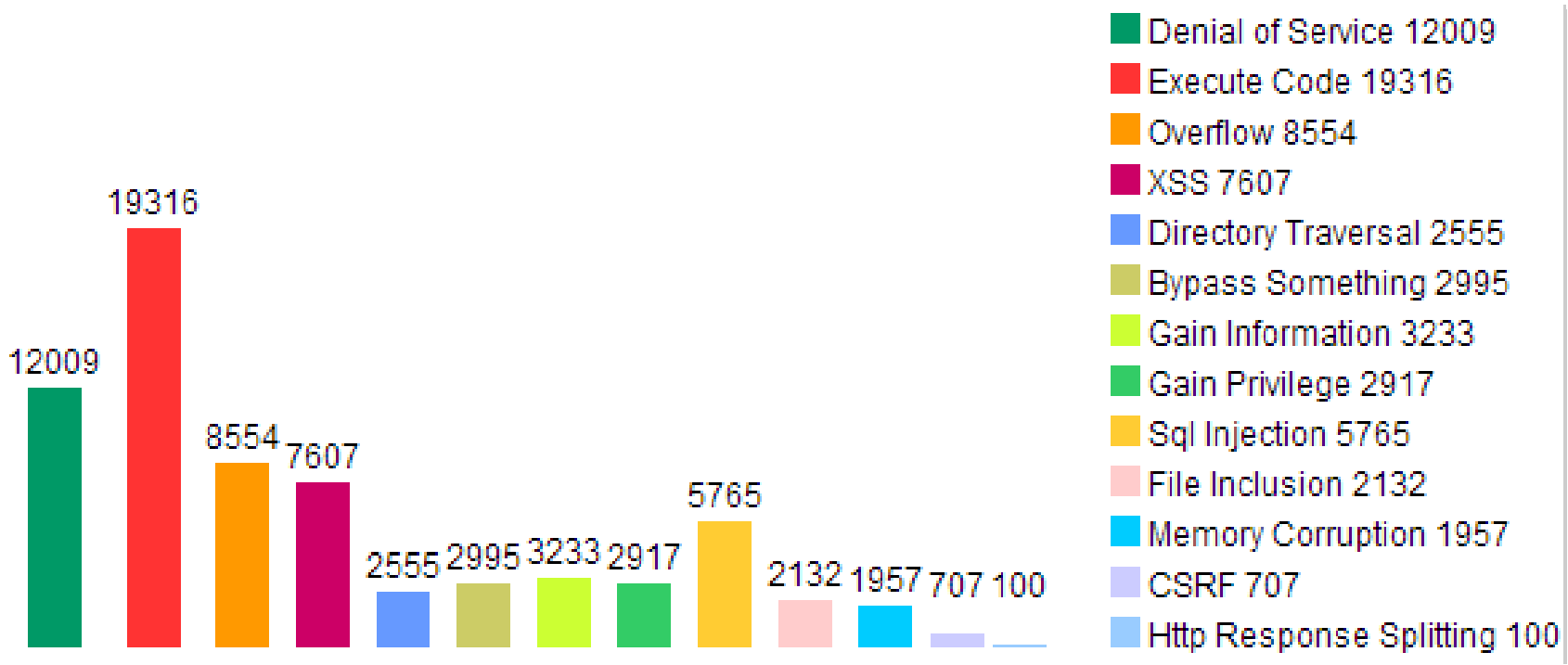
- 根据Web请求的水平路径划分:
  - 网络层，硬件WAF，云WAF
  - 主机层，软件WAF，安全狗
  - 代理层，基于WebServer进行实现，外部表现为WebServer的模块
  - 脚本层，基于WebServer传递过来的Web请求

WAF防御思想

- 攻击的特点:
  - 攻击二象性:已知攻击与未知攻击
  - 攻击的两个维度:形式与漏洞
- 完美防御思想:
  - 攻击防御思想:“黑”和“白”
- WAF的核心原理:
  - 运用‘黑’、‘白’思想
  - 特征匹配、漏洞签名
  - 对匹配结果进行响应(拦截、记录)

我们需要它吗?

- "漏洞"一直存在着



攻击自动化、批量化、规模化

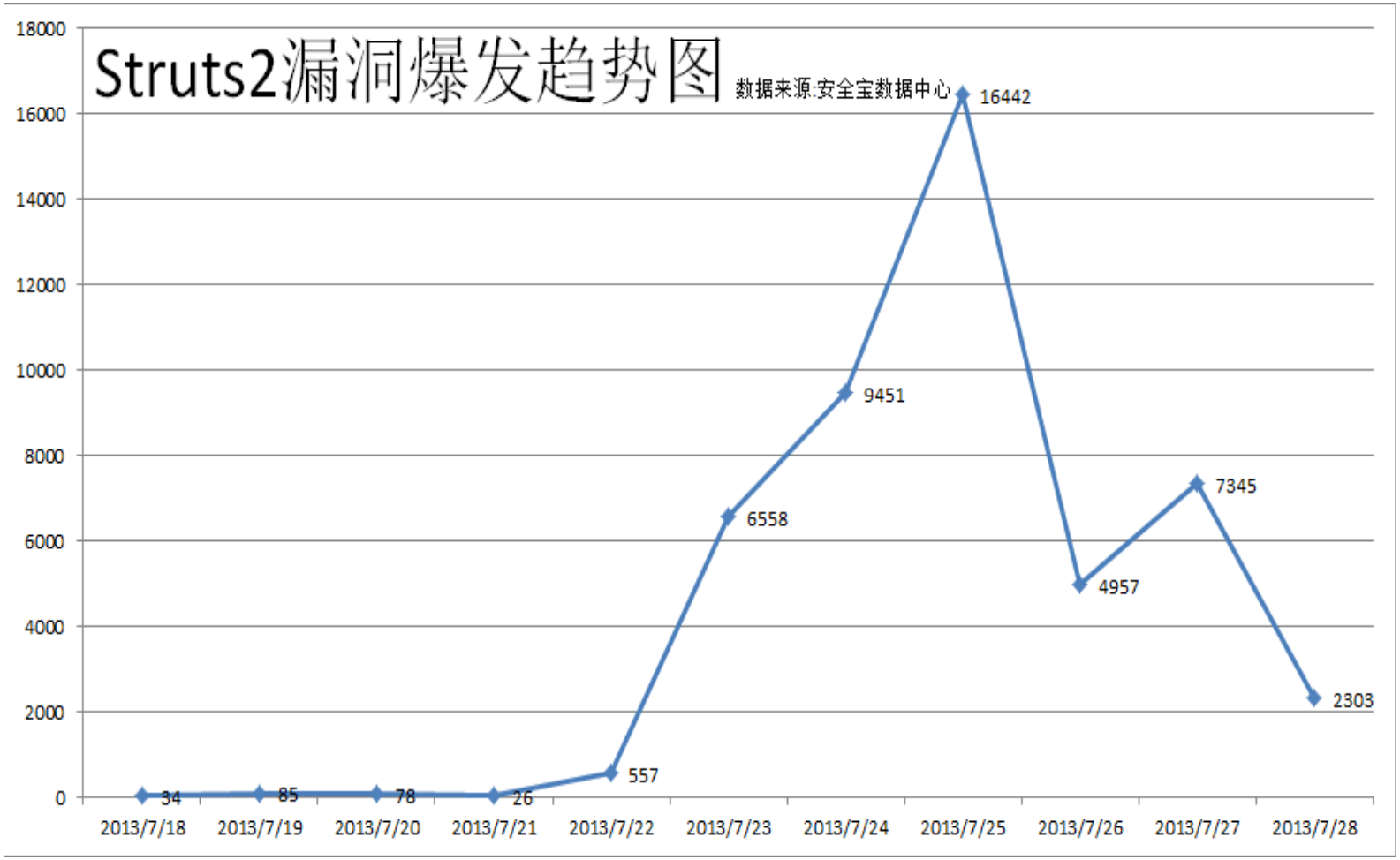
Struts2漏洞

2013-07-18爆发，Apache,苹果官网相继被黑


```
*Details: *
#show the webroot
http://vmbuild.apache.org/continuum/groupSummary.action?redirect:${%23a%3d(new%20java.lang.ProcessBuilder(new%20java.lang.String[]{'whoami'}))
/home/continuum/apache-continuum-1.4.1/apps/continuum
*Proofs of concept: *
#id
uid=1001(continuum) gid=1001(continuum) groups=1001(continuum)
#/sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:56:ae:00:0b
          inet addr:140.211.11.54  Bcast:140.211.11.255  Mask:255.255.255.0
          inet6 addr: fe80::250:56ff:feae:b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:22081926 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7627912 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:26173286052 (26.1 GB)  TX bytes:3491916802 (3.4 GB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
```

"病毒式"传播


Struts2漏洞爆发趋势图
数据来源:安全宝数据中心
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
34
85
78
26
557
6558
9451
16442
4957
7345
2303
2013/7/18
2013/7/19
2013/7/20
2013/7/21
2013/7/22
2013/7/23
2013/7/24
2013/7/25
2013/7/26
2013/7/27
2013/7/28

WAF的不足

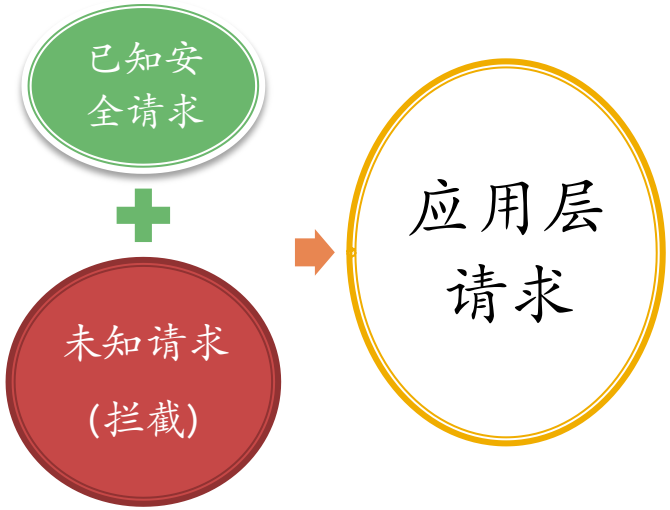
- WAF只是Web安全的基础防御措施，属于**相对的安全**，不是绝对的安全!
- 因为**漏洞的未知性**和**攻击形式的多样性**，以及**大数据的复杂性**，导致WAF不可避免存在误报和漏报.....
- 所以WAF需要修炼.....

基础修炼

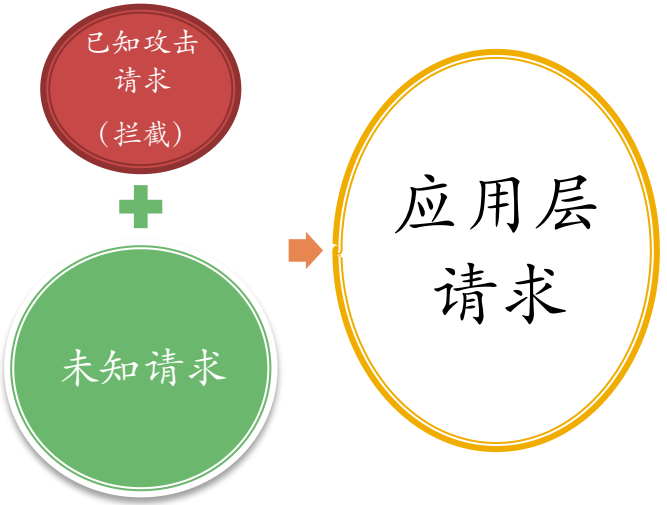
- 规则是针对攻击形式还是针对网站?
  - 白规则 OR 黑规则
- 规则如何适应大量网站?
  - 自动化 AND 自学习

基础修炼

- 白规则
  - 最大化拦截，误报最大化



- 黑规则
  - 最小化拦截，误报最小化



基础修炼

- 我们的选择:
  - 放弃白规则，走上“黑规则”之路
  - 放弃日志自学习，选择"规则自动生成"

误报

- 富文本误报
  - 搜索框、论坛帖子、后台模板修改等
  - 特点:
    - 内容多样化，随机性大，很容易触发规则
- 文件上传
  - 内容丰富，很容易触发注入、跨站规则

误报

- 静态正则规则的松散
  - 关键词与字符串傻傻分不清
    - 例子:aqbtest.com/vuln.php?data=**selectnamefromuser**
      - SQL注入拦截规则:select .* from .*
      - 区分单词与字符串:\bselect(\b).*(\b)from(\b).*

误报

- 正则缺少逻辑性，执行路径单一
  - 一次只能匹配一种条件
  - 如 Pading Oracle Attack的检测：
    - 攻击过程：
      - 使用WebResource.axd?d=xyz进行漏洞探测，错误密文产生500错误，而正确密文产生404
    - **正则显然无力有效匹配**

内功修炼-消除误报

- 文件、目录白名单
- POST协议区分对待
  - application/x-www-form-urlencoded
  - multipart/form-data(文件上传，后门拦截规则)
- 规则引擎逻辑化
  - 支持多条件组合判断
- SQL语法解析引擎
  - 解决关键词与字符串混淆

内功修炼-消除误报

- SQL语法分析引擎(Sqlite的lemon)
  - 语法分析引擎:http://www.hwaci.com/sw/lemon/
  - 工作原理
    - 词法分析
      - 先将原始数据解析为单词元符号(token)
    - 语法解析
      - 然后对分解出来的单词符号进行语法树解析
    - 攻击判定
      - 如果语法树属于攻击或可正常执行语法树形式，那么即存在SQL注入攻击

语法解析原理简介

- SQL语句
  - Select username,password from user where id=$id
  - 攻击语句:1 and 1=2 union select 1,1 from user

数字 | 逻辑运算符 | 数字 | 操作符 | 数字

U(操作关键字) | E(操作关键字) | (o)操作符 | k(SQL关键字) | (n)基础标志符

语法树解析:1&1 UEokn

存在攻击形式或可正常执行的SQL语法树

漏报

- 大数据包绕过
- 畸形数据包绕过
- 大小写绕过
- 编码绕过
  - URL编码
  - Unicode编码
- WebServer解码差异性绕过
- 复参数绕过
- URL二次编码绕过
- 数据库特性绕过
  - 注释符绕过
  - 冷门函数
  - 变量赋值
  - 。。。。。

攻击绕过

- 解码差异性绕过(WebServer)

测试环境	输入	输出	特点
IIS5.1+ASP	test.asp?test=1%1	11	百分号没有了
	test.asp?test=1%E	1E	百分号没有了
	test.asp?test=1%G	1G	百分号没有了
	test.asp?test=1%3F	1?	%3F被解码了
	test.asp?test=1%3G	13G	百分号去除了
	test.asp?test=1%GF	1GF	百分号去除了
	test.asp?test=1%GG	1GG	百分号去除了
IIS7.5+PHP	test.php?test=1%1	1%1	百分号保留
	test.php?test=1%G	1%G	百分号保留
Apache+PHP	test.php?test=1%E	1%E	百分号保留
	test.php?test=1%3F	1?	%3F被解码了
	test.php?test=1%3G	1%3G	百分号保留
	test.php?test=1%GG	1GG	百分号保留
Nginx+PHP	test.php?test=1%G	1%G	百分号保留
	test.php?test=1%E	1%E	百分号保留
	test.php?test=1%3F	1?	%3F被解码了
	test.php?test=1%3G	1%3G	百分号保留
	test.php?test=1%GG	1GG	百分号保留

攻击绕过

- 数据库特性绕过
  - Mysql的特殊符号
    - ‘.’、‘!’、‘~’、‘+’、‘-’

```
1  SELECT * FROM `qcon_test` where id=-1 union select~1,username,password from qcon_test;
```

信息 | 结果1 | 概况 | 状态

id	username	password
9223372036854775807	admin	admin

```
1  SELECT * FROM `qcon_test` where id=-1 union select 1.,username,password from qcon_test;
```

信息 | 结果1 | 概况 | 状态

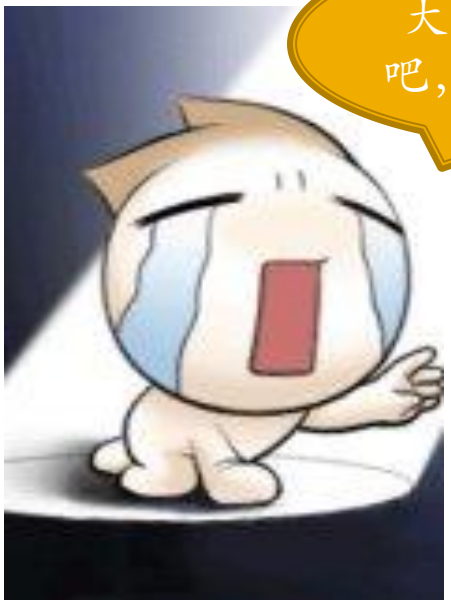
id	username	password
1	admin	admin

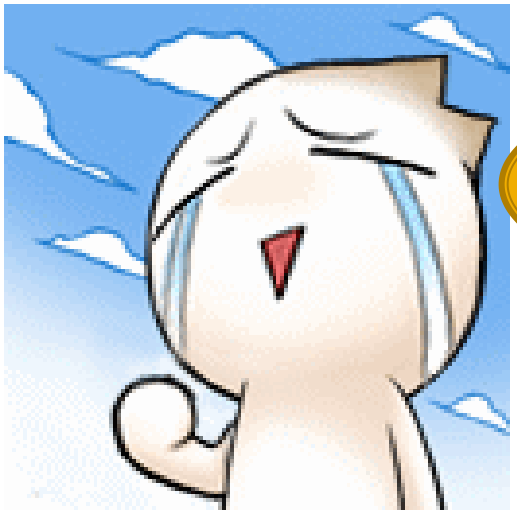
```
1  SELECT * FROM `qcon_test` where id=-1 union SELECT!1,username,password from qcon_test where id=1
```

信息 | 结果1 | 概况 | 状态

id	username	password
0	admin	admin

对于这些我们如何进行防御呢?

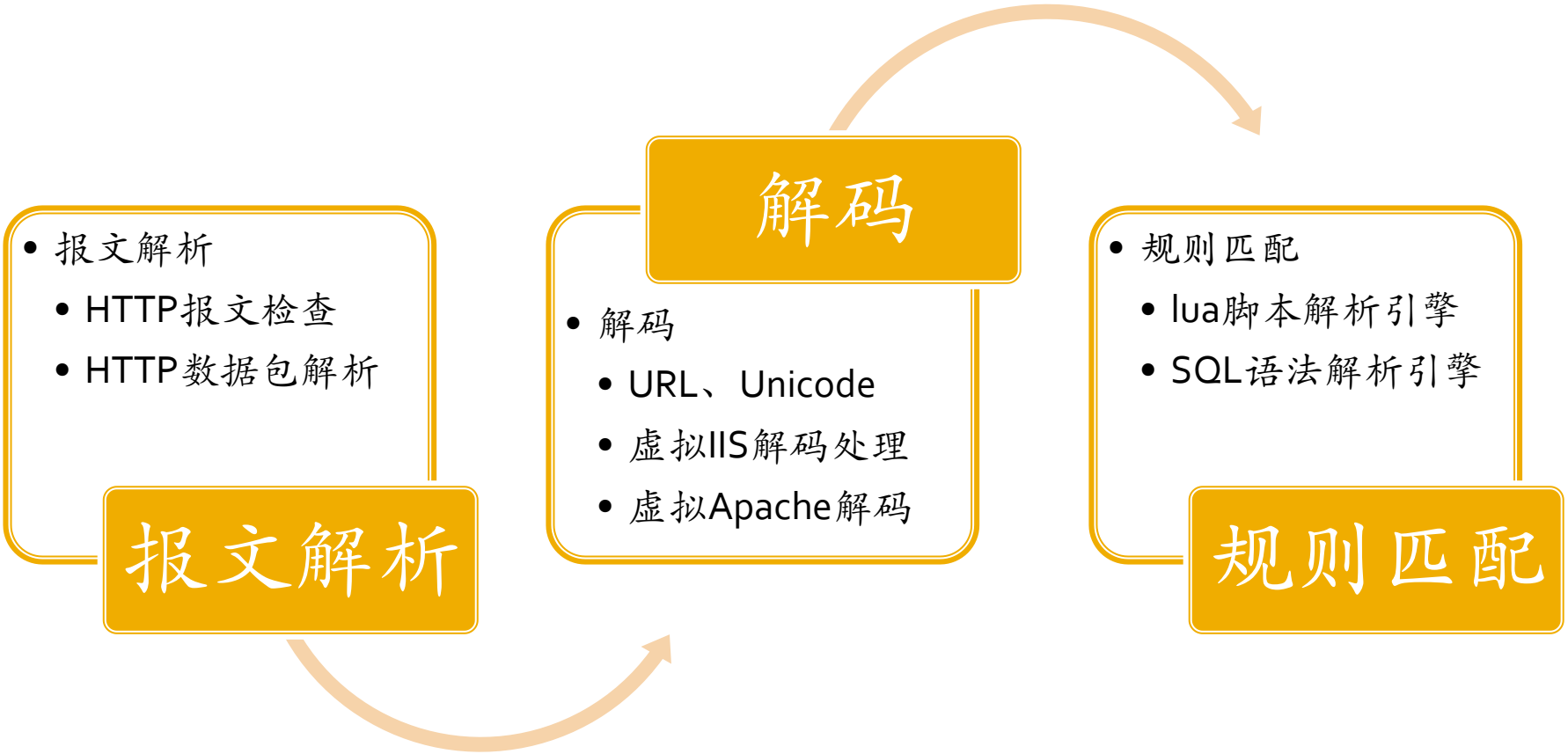




内功修炼-降低漏报

- 数据传输层面
  - 畸形、恶意的HTTP数据包丢弃
- WebServer层面
  - 大小写不敏感
  - 利用WAF底层进行解码，然后进行规则匹配
  - 兼容多种后端Web Server对预留字符的特殊处理,让规则进行多路径匹配拦截
- 脚本层面的绕过
  - 定制化规则(?)
- 数据库层面的绕过
  - SQL语句解析引擎
- 最新的oday攻击和攻击形式的多样性
  - 关注最新漏洞和新型攻击，添加特征规则

WAF数据处理流程


报文解析
• 报文解析
• HTTP报文检查
• HTTP数据包解析
解码
• 解码
• URL、Unicode
• 虚拟IIS解码处理
• 虚拟Apache解码
规则匹配
• 规则匹配
• lua脚本解析引擎
• SQL语法解析引擎

Q&A

Thanks