项目方式讲解WAF建设

宜人贷 / 刘铁铮

自我介绍

- 十年以上的安全从业经历
- 在安全厂商、百度做过安全研究
- 在支付公司负责过企业安全建设
- 当前在宜人贷负责安全架构，完成办公网安全产品、WAF产品建设

目录

- 需求
- 方案设计
- 技术实现
  - WAF基础功能
  - WAF扩展功能
  - WAF运营后台
  - 功能和性能测试
- 经验总结

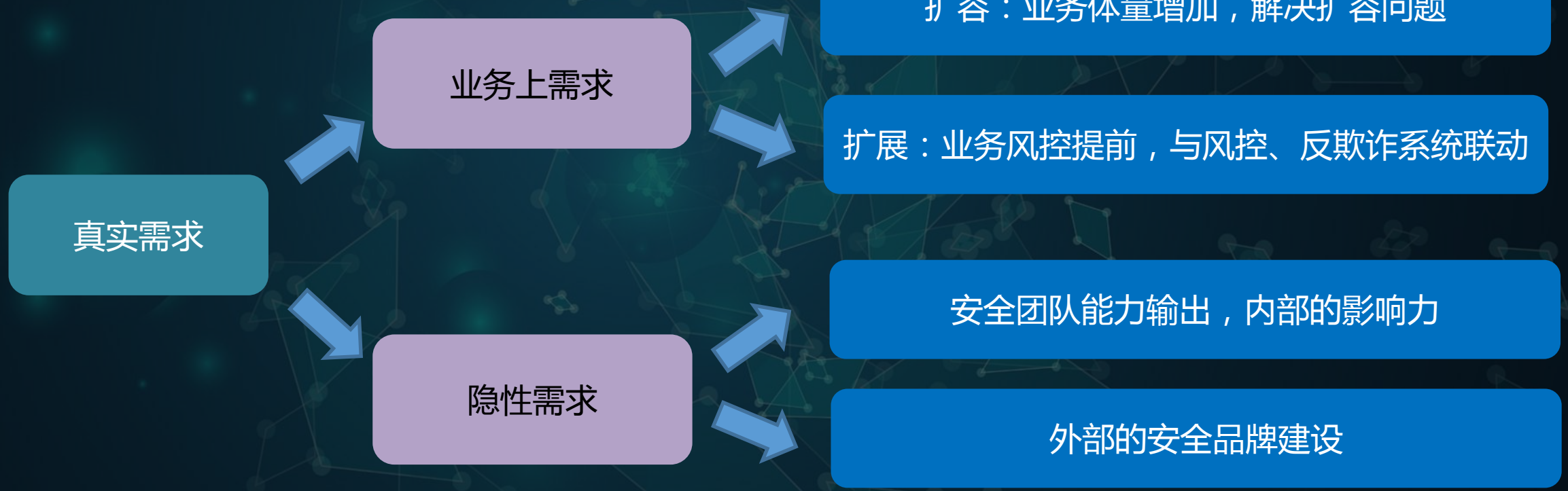
需求

我们遇到了哪些问题？

商业产品的扩容、扩展问题

- 扩容，商业盒子产品必然会面临的问题；
- 扩展，商业WAF产品通常只具备传统web安全防御能力，很难与其他安全产品有效联动、形成合力

真实的需求是什么？


真实需求
业务上需求
隐性需求
扩容：业务体量增加，解决扩容问题
扩展：业务风控提前，与风控、反欺诈系统联动
安全团队能力输出，内部的影响力
外部的安全品牌建设

方案设计

WAF产品功能组成

WAF的基础功能

传统WEB安全防御

WAF的扩展功能

CC防御 、反爬虫和会话分析

与风控和反欺诈系统联动

数据分析、情报数据

WAF的运营平台

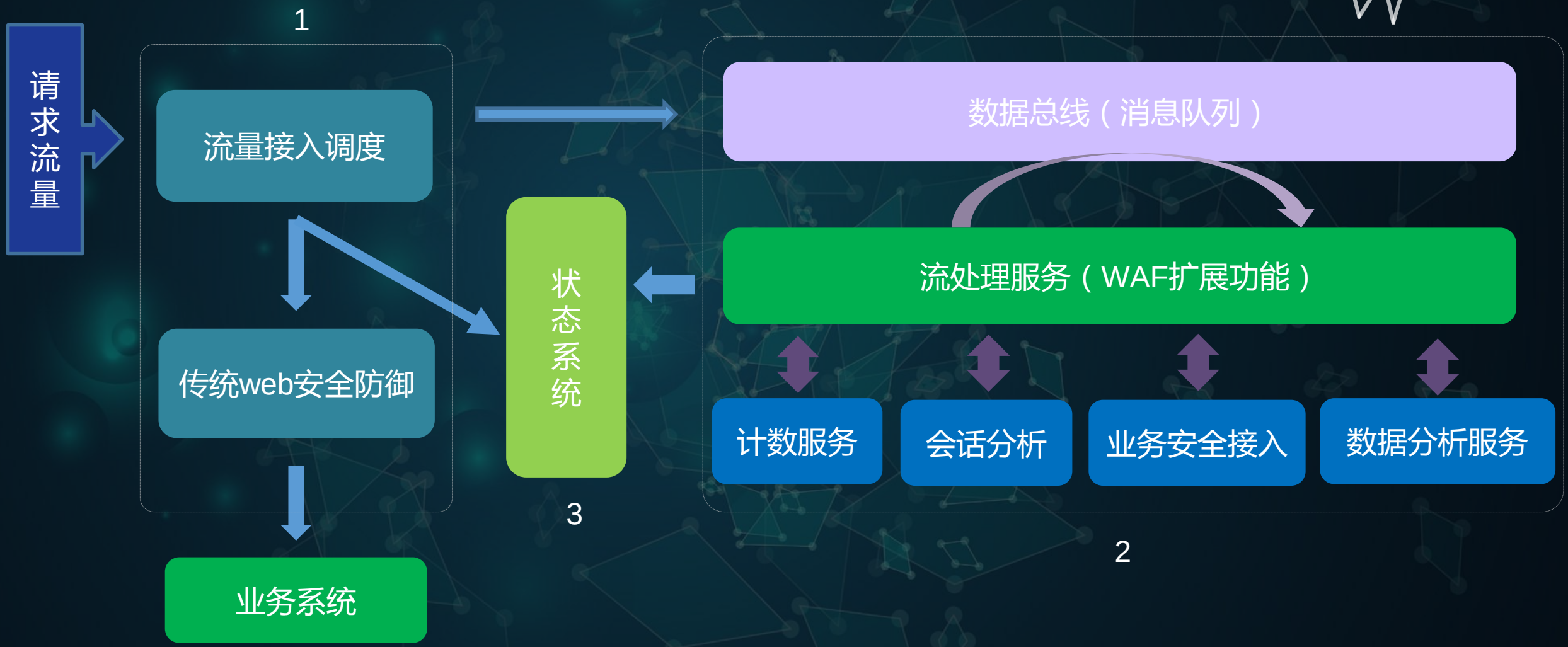
配置管理、规则管理

报表、日志、健康状况管理

数据分析平台管理、告警查询

设计 – 主流WAF产品分析

设计 – 框架和数据流图


1
请求流量
流量接入调度
传统web安全防御
业务系统
状态系统
3
数据总线（消息队列）
流处理服务（WAF扩展功能）
计数服务
会话分析
业务安全接入
数据分析服务
2

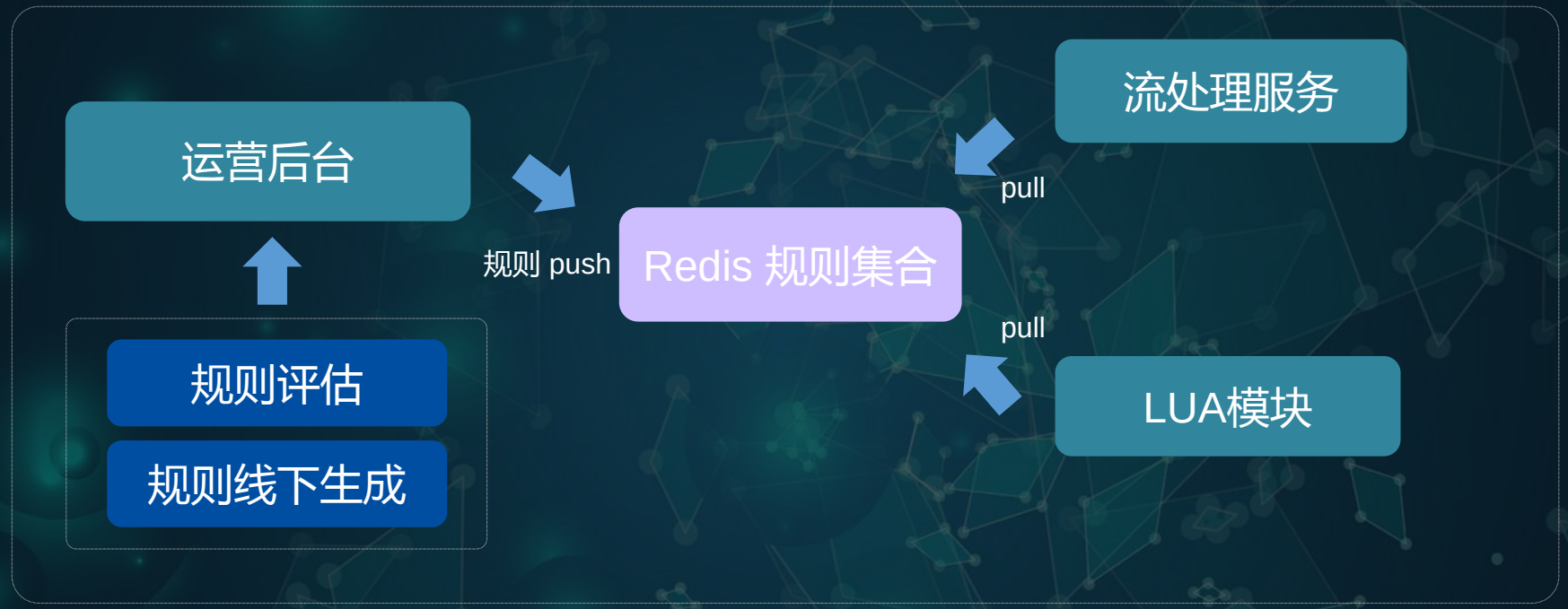
技术实现

WAF基础功能

执行流程

1. 利用nginx(openresty)，解析流量
2. 配置信息、规则、黑白名单由运营后台同步到redis
3. 利用LUA在各个执行阶段分段处理
   - ① 在init-worker阶段定时同步redis数据
   - ② 在access阶段执行规则判定和动作执行
   - ③ 在header阶段将sessionid写入cookie，以便后续流程的多维分析
   - ④ 在body阶段执行敏感数据过滤
   - ⑤ 在log阶段完成日志的输出

WAF基础功能 - 规则推送



- 规则来源
  - ① Modsecurity规则集提取
  - ② 商业WAF规则
  - ③ 宜人贷自积累的行业内规则
- 规则推送
  - ① 规则评估
  - ② Timer执行
  - ③ Redis写入

WAF基础功能 - 规则优化

- 规则执行效率
  - ① systemtap-toolkit 工具调优正则
  - ② 先匹配字符串，匹配后再执行正则匹配

```
$ ./ngx-pcre-stats -p 24528 --total-time-top --luajit20
 Tracing 24528 (/path/to/nginx/sbin/nginx)...
 Hit Ctrl-C to end.
 ^C
 Top N regexes with longest total running time:
 1. pattern /WEB_ATTACK/: 15103us (total data size: 82184)
 2. pattern / __cf__\d+/: 11143us (total data size: 25916)
 3. pattern /[^\x01-\xff]/: 10233us (total data size:
102825)
```

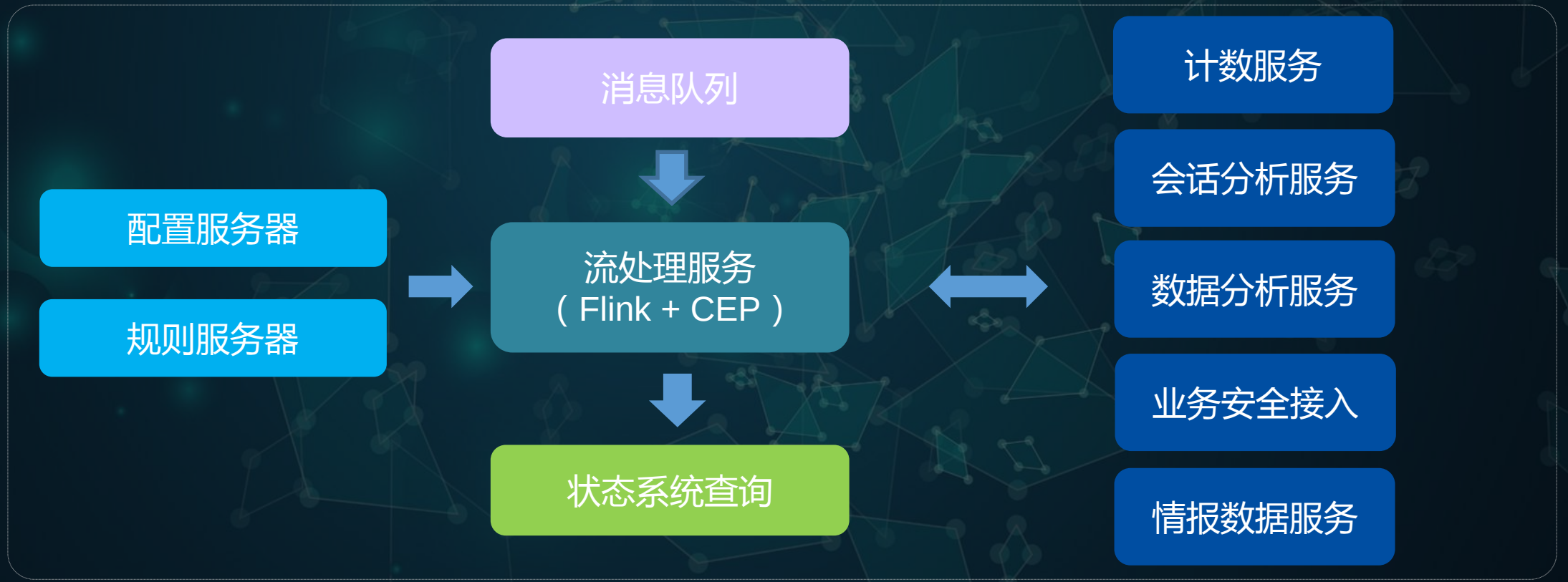
```
$ ./ngx-pcre-stats -p 24528 --worst-time-top --luajit20
 Tracing 24528 (/path/to/nginx/sbin/nginx)...
 Hit Ctrl-C to end.
 ^C
 Top N regexes with worst running time:
 1. pattern /\.cookie\b.*?\;\W*?domain\W*?\=/: 98us (data size: 36)
 2. pattern /(Content-Length|Transfer-Encoding)/: 89us (data size:
14)
 3. pattern / __cf__\d+/: 63us (data size: 8)
 4. pattern /[^\x01-\xff]/: 53us (data size: 13)
 5. pattern /\b(background|dynsrc|href|lowsrc|src)\b\W*?=/: 53us
(data size: 5147)
```

```
{
  "rule_id": "10100003",
  "match_type": "regex",
  "rule_name":"JAVA 获取参数",
  "key_word": "\\bget(?:runtime|parameter|inputstream|reader)|write)\\s*?\\(",
  "action": "deny"
},
```

WAF扩展功能 - 流处理服务

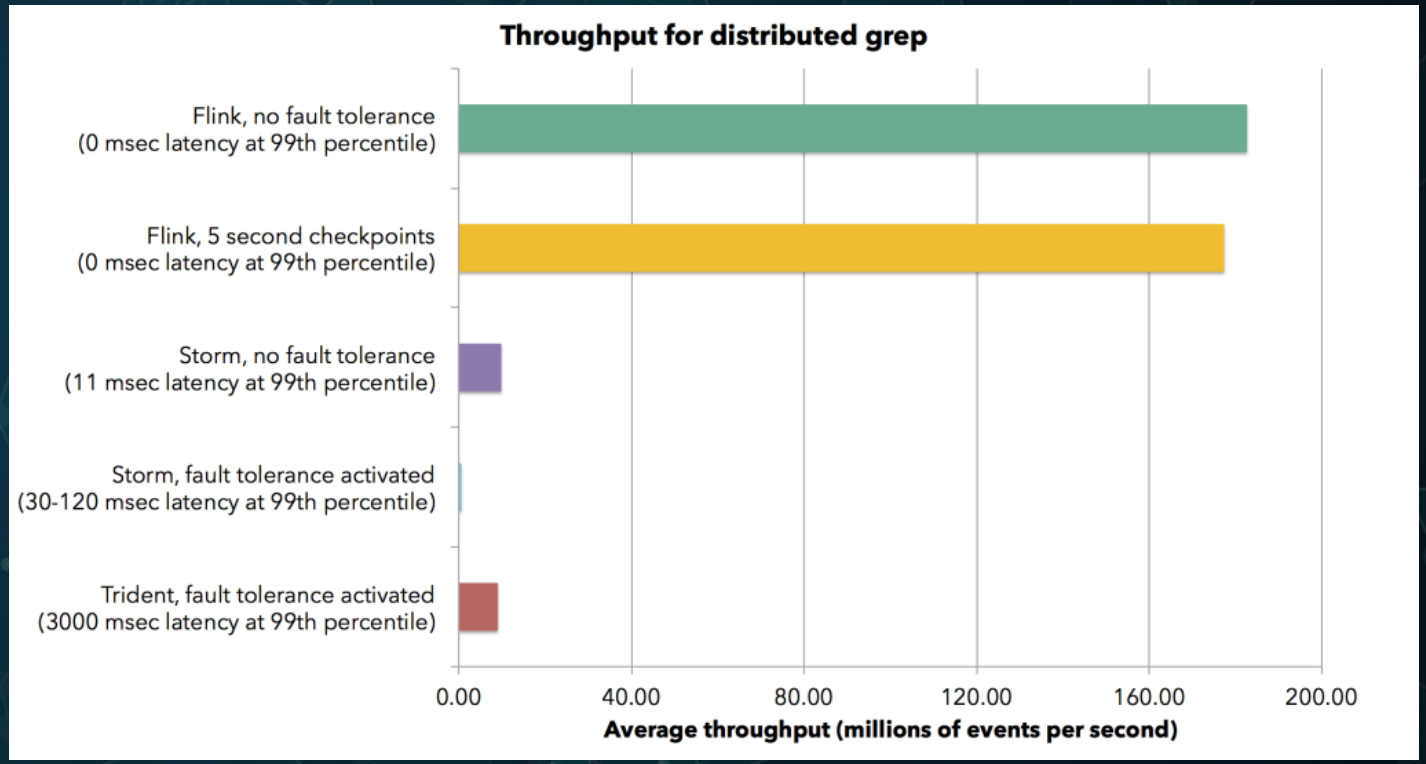
传统WAF　vs　WAF扩展

采用流计算方案 – 实时消费消息队列
对接各个微服务 – 扩展性



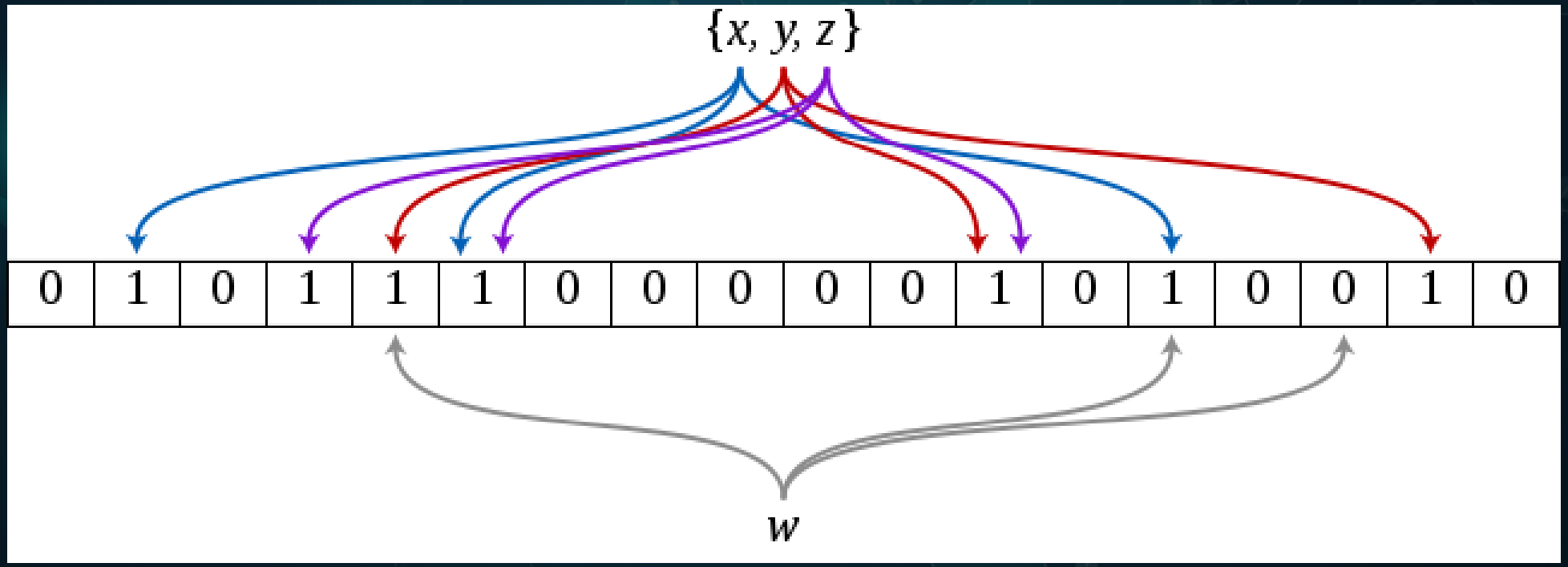
WAF扩展功能 - 为什么选择Flink？

1、纯流式系统
2、高吞吐
3、内置CEP复杂事件规则引擎



WAF扩展功能 - 计数服务

- 标准Bloom Filter ＆ Counting Bloom Filter
  - ① 布隆算法是一种高效利用空间的概率数据结构，用于检测一个元素是否属于一个集合；
  - ② 优点：实现简单，占用空间小，速度极快
  - ③ 缺点：有一定的误差



WAF扩展功能 – 会话分析

① SID合法性校验
② 指定域名的上下文分析
③ 相同SID下的基础安全规则触碰次数
④ Session封禁，不伤IP

WAF扩展功能 – 状态系统

- 业务安全提前做，触碰如下规则写入状态系统，为后续业务风控提前准备数据
  - ① 请求IP的情报信息
  - ② 请求IP or 设备指纹的请求频率计数
  - ③ 请求IP or 设备指纹的访问时段规则

运营后台

- 多级配置，灵活降级
- 域名接入，规则管理
- 告警查询，报表输出
- 系统健康状况监控

日志和告警

告警日志
状态日志

功能和性能测试

无规则测试

30条规则测试

50条规则测试

CPU
负载

延时

规则数量	请求数量	并发数量	MQ超时	MQ超时数量	超时时间是否算入总时间	QPS	各阶段平均用时(单位：毫秒)						备注
							access阶段	send MQ	header_f	body_f	log阶段	总时间(毫秒)	
30	50w	30	200ms	9000	否	3000	0.29	0.13	0.01		0.011	0.55	
	50w	20	200ms	5000	否	3800	0.31	0.15	0.01		0.011	0.57	
	25w	10	200ms	500	否	4100	0.27	0.1	0.01		0.011	0.54	
	50w	30	2000ms	250	否	4800	1	0.89	0.01		0.011	1.32	
	50w	20	2000ms	85	否	5000	1	0.91	0.01		0.011	1.33	
	50w	20	20000ms	85	否	5000	1	0.91	0.01		0.011	1.33	
	50W	10	200ms	300	否	5159	0.27	0.14	0.01		0.11	0.53	
	50W	20	200ms	340	否	5859	0.58	0.45	0.01		0.11	0.84	
	50W	30	200ms	450	否	6021	0.38	0.26	0.01		0.11	0.65	
	50W	10	20000ms	0	否	4929	0.34	0.21	0.01		0.11	0.6	100% 7000ms 99%<5ms
	50W	20	20000ms	0	否	5400	1	0.96	0.01		0.11	1.38	100% 10639ms 99%<10ms
	50W	30	20000ms	0	否	4900	2.86	2.73	0.01		0.11	3.11	100% 10639ms 99%<12ms
	30w	10	20000ms	0	否	5381	1.48	1.21	0.01		0.11	2.86	100% 5238ms 99%<4ms
	30w	20	20000ms	0	否	5900	0.66	0.39	0.01		0.11	1.69	100% 7482ms 99%<6ms
	30w	30	20000ms	0	否	5900	2.7	2.45	0.01		0.09	4.46	100% 13199ms 99%<9ms
50	30w	10	20000ms	0	否	3700	1.17	0.12	0.01		0.16	2.72	100% 566ms 99%<6ms
	30w	20	20000ms	0	否	5100	1.3	0.4	0.01		0.14	3.94	100% 2825ms 99%<10ms
	30w	30	20000ms	0	否	4700	2.17	1.22	0.01		0.13	5.94	100% 9378ms 99%<15ms
100 有复杂规则	30w	10	20000ms	0	否	800	9.66	0.11	0.01		0.08	15.8	100% 1018ms 99%<31ms
	30w	20	20000ms	0	否	815	9.47	0.12	0.01		0.08	27.1	100% 662ms 99%<68ms
	30w	30	20000ms	0	否	793	9.83	0.21	0.01		0.08	38.6	100% 1074ms 99%<110ms
76 无复杂规则	30w	10	20000ms	0	否	775	9.58	0.1	0.01		0.08	15.2	100% 191ms 99%<32ms
	30w	20	20000ms	0	否	815	10	0.21	0.01		0.1	27.5	100% 1263ms 99%<78ms
	30w	30	20000ms	0	否	794	9.83	0.21	0.01		0.08	38.6	100% 367ms 99%<100ms

经验总结

团队

- 安全团队，特别是中小互联网公司的安全团队，在研发安全产品的过程中需要规避一些问题：
  ① 产品化，一个拿到生产环境使用的产品，健壮性、易用性、扩展性缺一不可
  ② 项目制，分工明确，有计划，这往往是大多安全团队比业务研发团队欠缺的
  ③ 好产品是迭代出来的，要抗的住压力，耐得住寂寞
  ④ 需要充分的需求分析、设计、评审
  ⑤ 需要足够多的测试
  ⑥ 需要具备产品、研发、数据分析、安全运营能力
  ⑦ 创造比破坏困难的多

规则策略和纵深防御

- 规则策略
  ① 宁漏报不误拦，影响业务的锅背不起
  ② 规则的数量和复杂程度影响到nginx的性能，规则和性能需要达到一种平衡
  ③ WAF属于典型的CPU密集型系统，95%的拦截发在在10%的规则上面，重点规则需要重点优化
  ④ 规则和业务匹配，就是说java后台就别上php的一些规则了

- 纵深防御和安全数据分析
  ① WAF产品做为流量入口，需要和其他安全系统联动，形成合力，重点解决真正危害业务的攻击行为
  ② 把机器学习算法模型作为规则的有效补充

参考资料

http://www.modsecurity.org
https://github.com/openresty/openresty-systemtap-toolkit
https://en.wikipedia.org/wiki/Bloom_filter
https://en.wikipedia.org/wiki/Hidden_Markov_model
https://data-artisans.com/blog/high-throughput-low-latency-and-exactly-once-stream-processing-with-apache-flink


THANKS