GOPS 2020
Shanghai

GOPS

GOPS 全球运维大会

2020
-AIOps 风向标

上海站

指导单位：

主办单位：

大会时间：2020年11月27日-28日

大会地点：上海中庚聚龙酒店

GOPS 2020
Shanghai

SAST：腾讯代码安全建设实践

林桠泉　　腾讯安全平台部

林桠泉

腾讯安全专家

10年安全从业经验

福建中医药大学——中西医骨伤专业

《漏洞战争：软件漏洞分析精要》作者

微软全球最具价值安全研究员

技术领域聚焦：漏洞攻防研究、研发安全建设

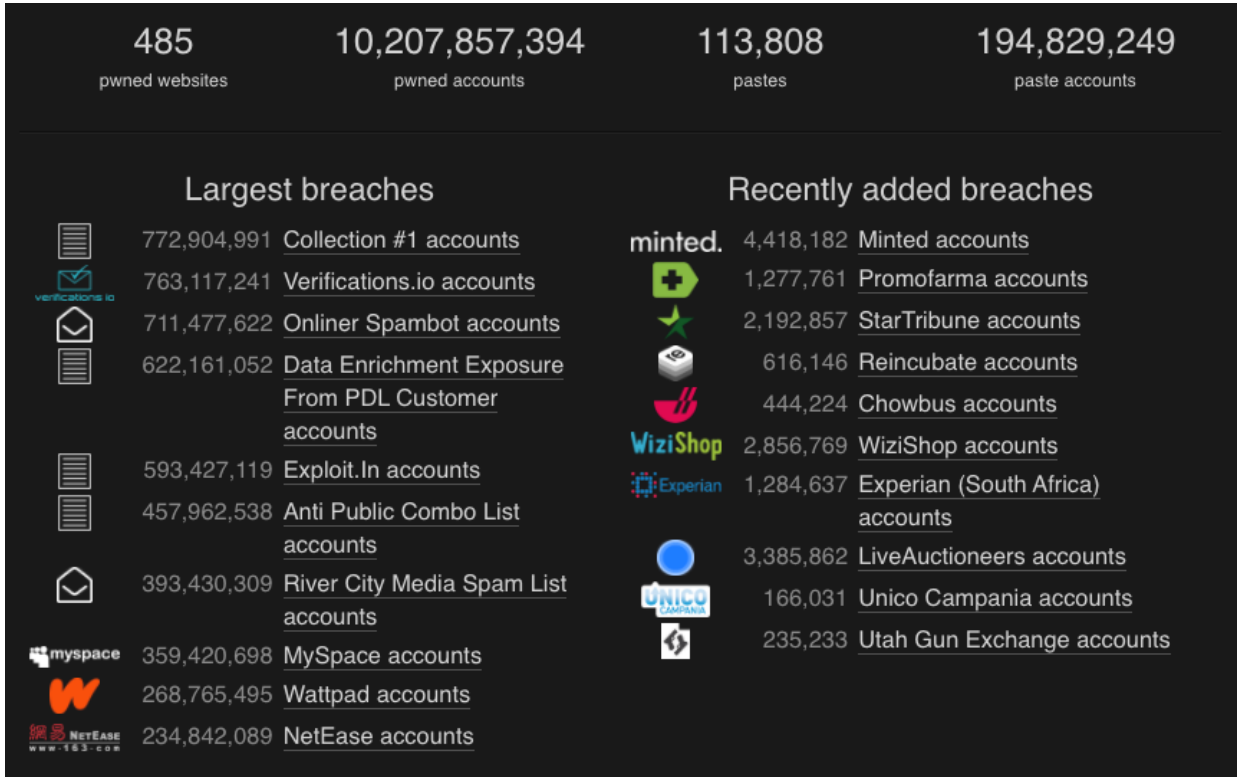
GOPS 2020
Shanghai

CONTENTS

目录

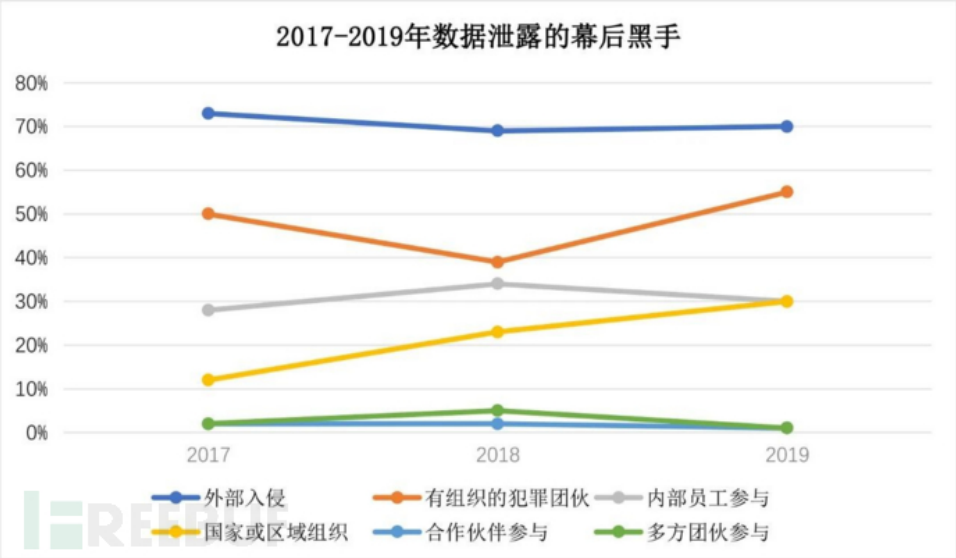
企业数据泄露现状

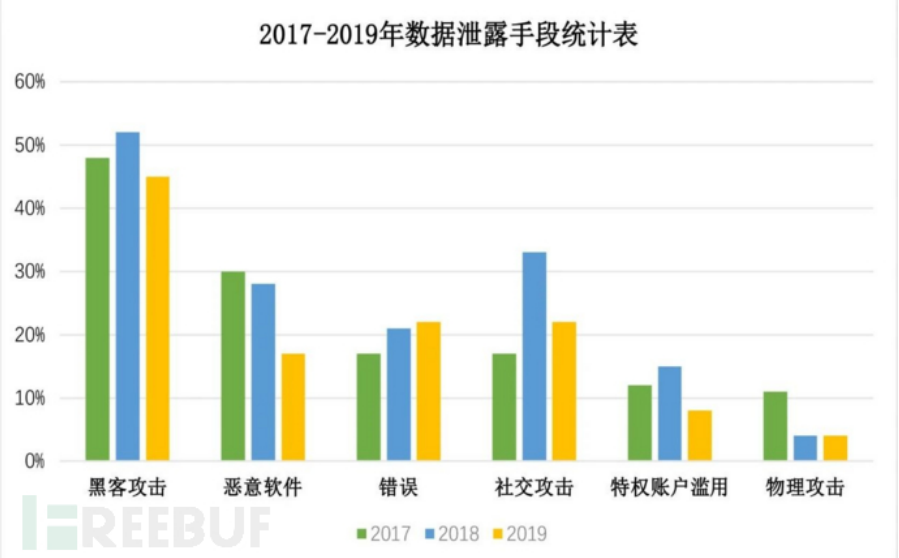
那些年发生过的数据泄露事件



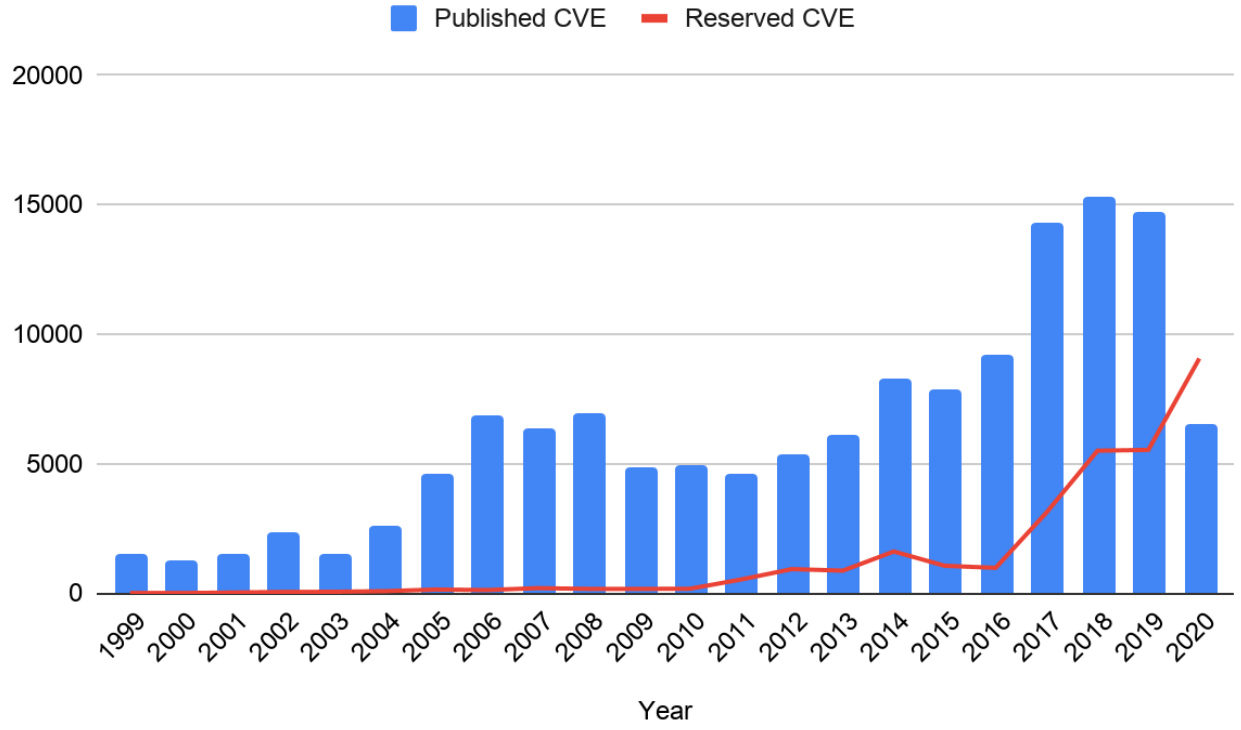
外部入侵是数据泄露的主要根源

漏洞攻击是外部入侵的主要攻击手段





全球软件漏洞数量持续增加



SAST概览

什么是SAST

静态应用安全测试 (SAST)，不运行程序的情况下，对代码或二进制文件进行代码安全检测。

SAST（静态应用安全测试）工作流程

为什么要做SAST

1. 安全左移（Shift Left）
2. 漏洞发现率高
3. 供应链安全检测（软件成分分析，SCA）
4. 可定位至漏洞代码，便于漏洞修复

3

行业SAST产品调研与应对

商业SAST产品调研

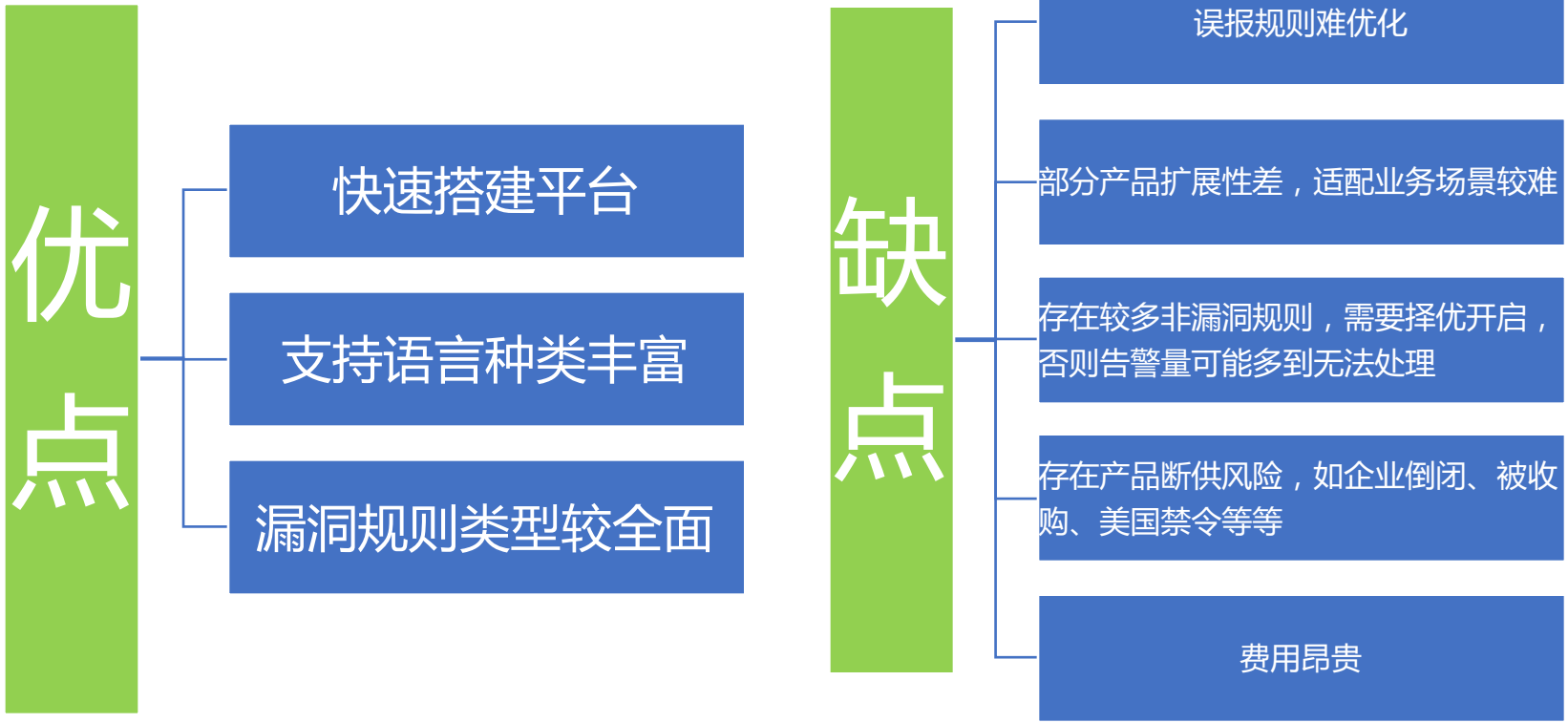
产品	支持语言	误报率	漏报率	扫描速度	输入方式	持续集成	依赖编译	修复建议	自定义规则	支持框架	报告输出	价格（元）
Coverity	20种	中	中	中	源码目录	支持	是	一般	支持	70+种	web	13万/年
CheckMarx	20种	极高	高	慢	zip	支持	否	优秀	支持	40种	web/pdf/rtf/csv/xml	85万/年
CodeQL	5种	中	中	慢	源码目录	不支持	是	差	支持	82种	json	67万/年
SonarQube	22种	中	中	中	zip、代码仓库	支持	是	一般	支持	9种	web/pdf	781万/10亿行
RIPS (SonarQube收购)	2种	中	中	快	zip	支持	否	优秀	不支持	30+种	web/pdf/json	46万/年
Pinpoint (阿里收购)	2种	高	高	慢	zip、代码仓库	支持	是	优秀	不支持	-	web/pdf	20万/年
Fortify	27种	高	高	慢	源码目录	支持	是	一般	支持	-	pdf/html/doc/xls	100万/年

注：

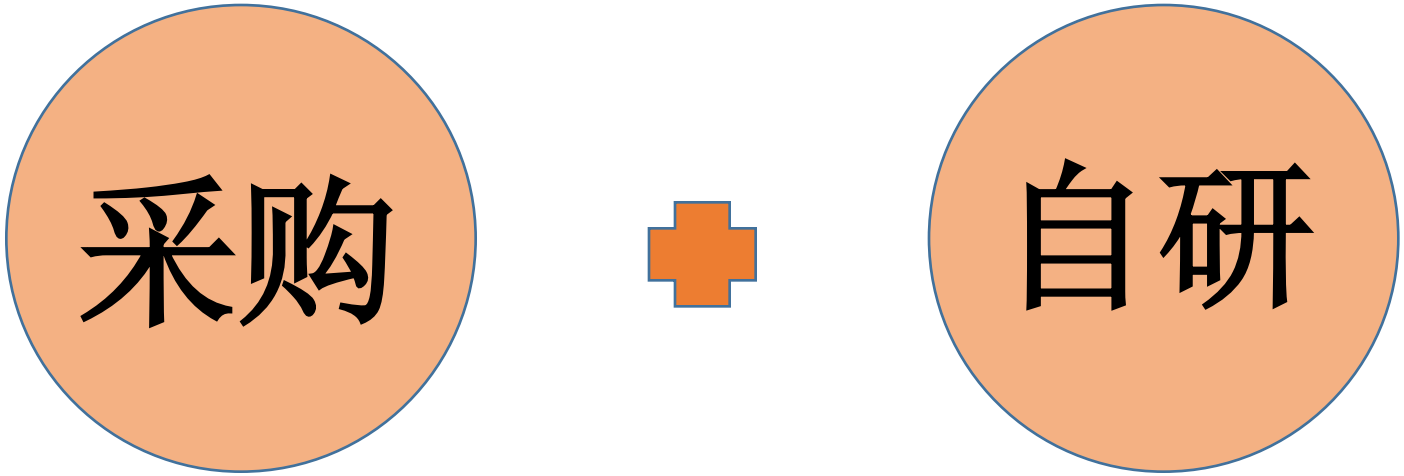
1、仅开启安全漏洞规则，不包括代码规范、提示类安全风险，测试以自建靶场样本为主。

2、依赖编译主要是面向C/C++/Java而言

商业SAST产品的优缺点



我们的做法

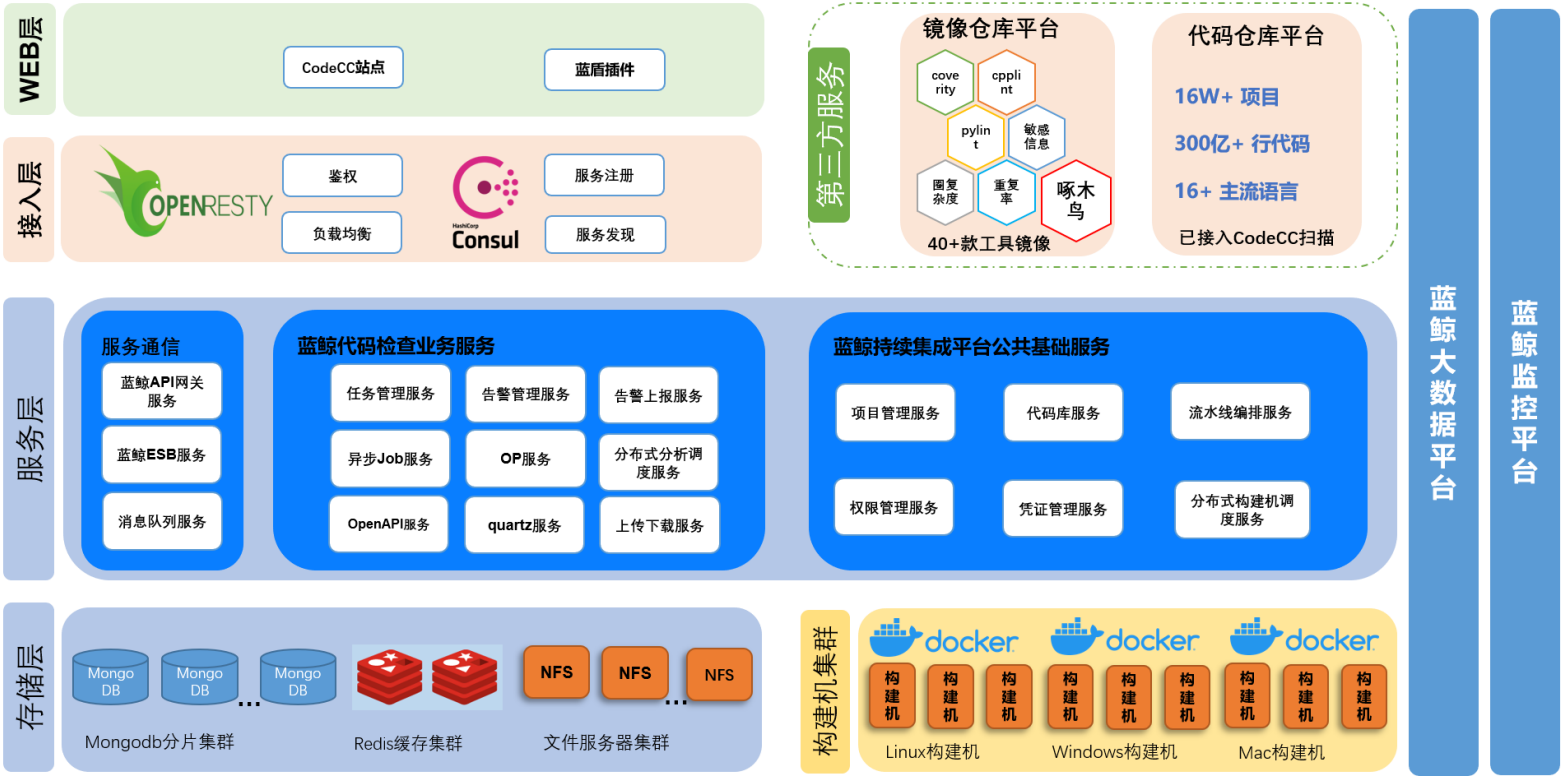


软件采购

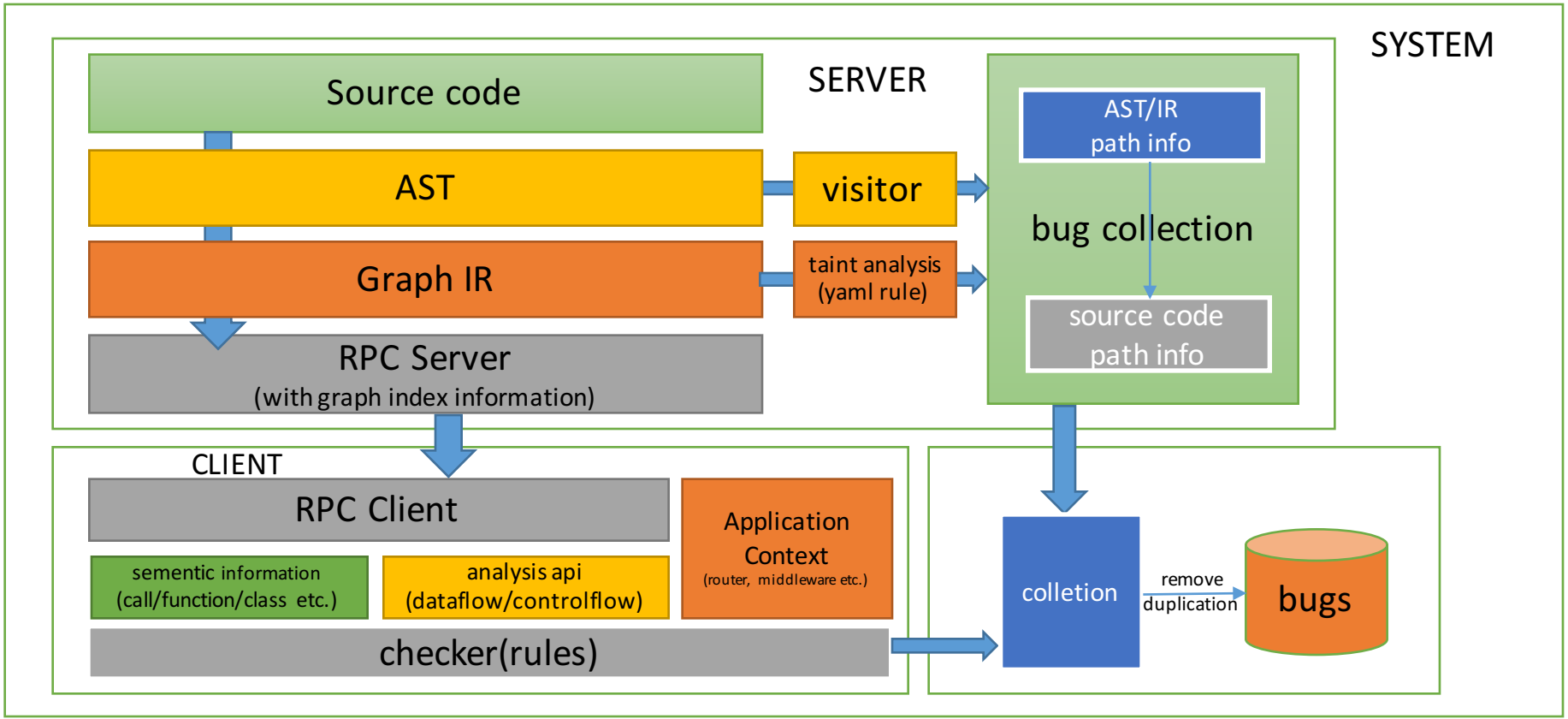
规则全开，误报告警多得无法运营，等于没发现问题

挑选低误报规则开启，取消非漏洞规则（如代码规范）

腾讯代码分析平台——BK-CodeCC



腾讯代码安全检测引擎——啄木鸟



腾讯SAST的挑战与展望

SAST面临的挑战

误报

- 整体占比不高，但基数大
- 如何平衡误报与漏报

人工

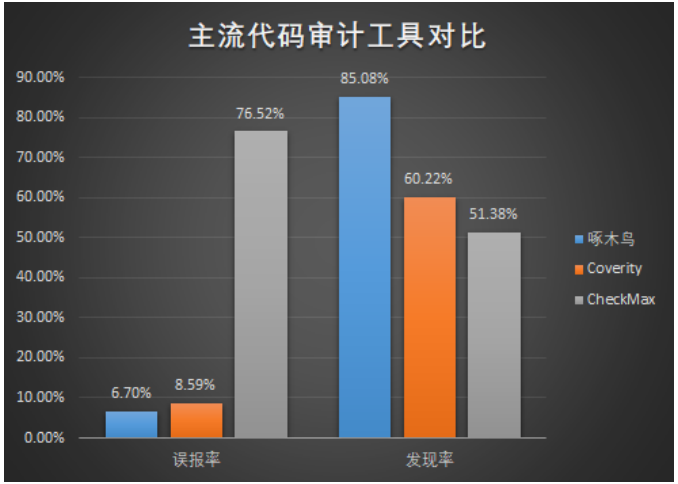
- 人工验证成本高

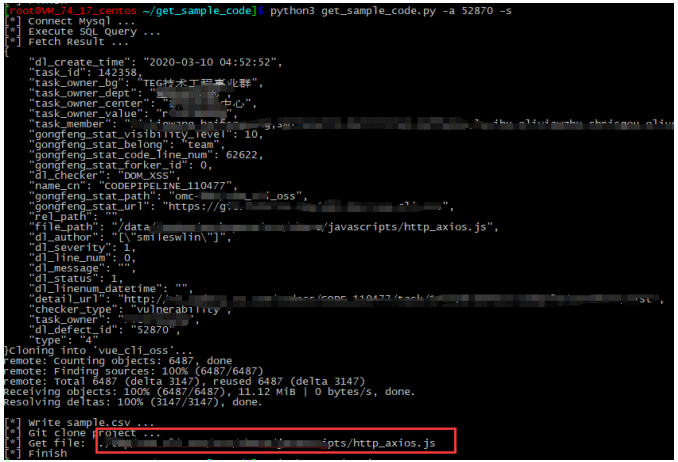
运营

- 代码规模量大，计算资源消耗大
- 风险数据运营与异常监控
- 修复推动与验证

寻找误报与漏报的平衡点

1. 优先保证准确率
   - 有争议的规则先不上
   - 前期允许漏报
   - 漏报的问题可通过添加规则解决，误报则需要更改原规则，不利于持续优化
2. 构建样本库
   - 漏洞代码样本
   - 安全代码样本





开发参与漏洞验证

安全工作人人有责

- 漏洞告警多，安全人员是无法全部人工验证一遍的
- 给开发培训漏洞相关知识
- 引导开发自主审查告警代码，标注告警（比如误报）
- 安全人员协助开发审查、修复和监督乱打标误报

安全运营

1、收集风险数据

- 告警类型、数量、代码提交者、时间、代码位置……

2、修复推动

- 安全分
- 奖惩机制

3、异常监控和告警

- 数据突升突降
- 异常忽略+屏蔽

4、宣传推广

- 视频、文章、邮件、安全比赛……
- 与代码仓库、研发平台合作，自动开启扫描与结果同步

展望

1. 自动修复技术
   - 代码仓库自动提交修复代码
   - IDE插件生成修复代码
   - 基于AI的自动修复技术
2. 代码分析与验证技术
   - 代码相似度分析识别漏洞代码
   - 自动生成PoC动态验证
   - 基于AI的漏洞检测技术

5

总结

总结

1. 自研引擎才能更好地适配业务场景，持续优化以提升检测能力
2. 构建自己的样本库，是提升和检验漏洞发现能力的关键
3. 探索前沿的代码分析、验证和自动修复技术

Thanks

高效运维社区

开放运维联盟

荣誉出品