RASP——应用安全测试技术

孙政豪

二十一世纪什么最重要？

- 人才
- 信息

- 两名乌克兰黑客窃取华尔街企业内幕信息

  两年获利超过一亿美元

- 提前获知Verisign 2013 财报将公布 15%年增长率

  黑客随即买入 $2.4 百万股票

- 美年大健康的销售人员向其宣称已经完全掌握了爱康国宾的客户资料、销售价格以及销售策略等信息。
- 2014年开始，爱康国宾陆续出现团体客户大面积异常流失的情况，在爱康国宾向新客户进行投标时，竞标成功率也远低于往年以及同期全国平均水平。爱康国宾因为美年大健康低价竞标等方式造成团体客户流失，损失巨大。

信息安全刻不容缓

“没有网络安全就没有国家安全”

常见防护方法

分层次防御

- 提高攻击者被检测到的概率
- 降低攻击者成功得手的几率

超过80%攻击都发生在应用层

——Gartner

安全软件开发生命周期
Security Software Development LifeCycle

- **产品研发初期**
- 安全顾问规划
- 规划安全里程碑
- 安全元素集成入产品

- **设计**
- 1.设计安全架构和指导方案。
- 2.形成安全且有效的规范完档

- **漏洞模型**

- **标准、检验、工具**

- 1.遵从安全编码规范
- 2.使用安全扫面工具(fuzzing tools, static-analysis tools, etc)

- **安全性推动**
- **代码**reviews
- 安全性测试
- 主流漏洞监测
- 达到预期准则

- **安全性把控**
- 安全团队的检测
- 完整的测试验证
- 遵守安全准则

- **RTM 和 部署**

- **安全响应**
- 规范反馈响应机制
- 实时处理反馈
- 反思

黑箱动态弱点扫描
Dynamic Application Security Testing / DAST

- 常被称为动态应用系统安全测试 / 网站黑箱扫描

  动态测试 / 弱点扫描

- 何为动态应用系统安全测试

  针对网页应用系统

  输入不同值来检查应用系统是否有暴露的漏洞

  依照一份事先定义的弱点清单尝试破解目标

黑箱动态弱点扫描
Dynamic Application Security Testing / DAST

- 主要用于发现应用程序及代码在开发过程中，由于开发者缺乏安全意识，疏忽大意极为容易导致应用系统存在可利用的安全漏洞：

SQL 注入	XSS	上传漏洞	CSRF
敏感信息泄露	目录遍历	目录列出	弱口令
链接地址重定向	任意文件读取	远程代码执行	恶意代码

黑箱动态弱点扫描
Dynamic Application Security Testing / DAST

DAST常用的工具包括：

工具名称	主要用途
nmap	获取主机开放的 服务、端口信息
nessus	对主机进行漏洞 扫描
本地/远程溢出工具	通过漏洞远程进入系统/本地提权
SQLMap	扫描应用程序SQL注入漏洞
Appscan	扫描应用程序漏洞
WebInspect	扫描应用程序漏洞
Retina	对主机进行漏洞 扫描

黑箱动态弱点扫描
Dynamic Application Security Testing / DAST

- **优势：**

  揭露一些只有在运行时才会暴露出的漏洞

- **缺点：**

  能找到的漏洞有限

  无法完整包含整个应用系统

  任何改动都需要重新扫描

白箱静态源码扫描
Static Application Security Testing / SAST

- 属于应用系统网站弱点扫描的一种，亦常称为

  静态应用系统安全测试、白箱测试、源码检测

- 何为静态源码扫描

  检查应用系统的原始码（source code）来找出安全漏洞

  能以更多元的角度去检查程式流中可能被攻击者发现的漏洞

白箱静态源码扫描
Static Application Security Testing / SAST

常用静态分析技术

- ◆词法分析
- ◆ 语法分析
- ◆ 抽象语法树分析
- ◆语义分析
- ◆控制流分析
- ◆数据流分析
- ◆污点分析

白箱静态源码扫描
Static Application Security Testing / SAST

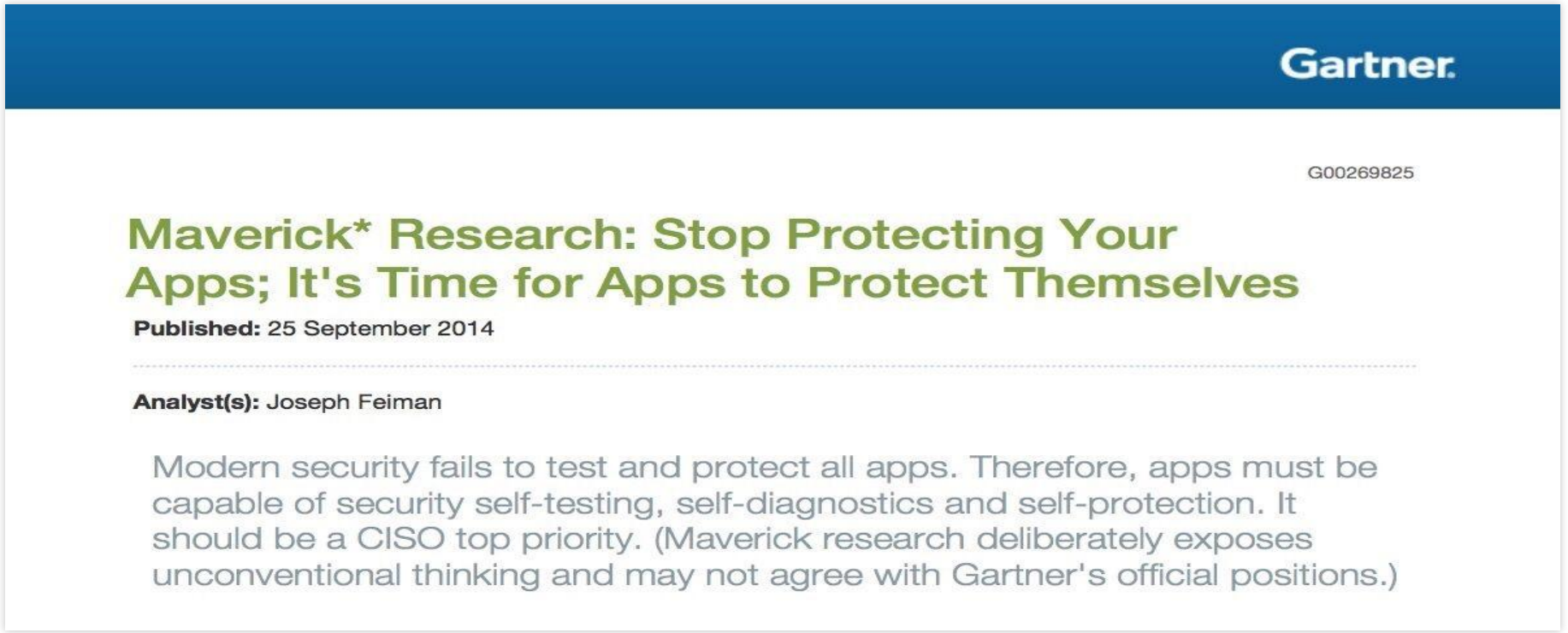
常用静态分析工具

公司名称	软件名称	支持语言
HP	Fortify SCA	各种主流语言
CheckMax	CxSuite	各种主流语言
University of Maryland	Findbugs	Java
Coveriter	Prevent	JAVA .NET C/C++
Veracode	SecurityReview	JAVA .NET

白箱静态源码扫描
Static Application Security Testing / SAST

- **优势：**
  - 能在开发阶段及早修复问题，比事后修补容易
  - 弱点可从源码中直接修正
  - 测试范围包含了各种程式语言
- **缺点：**
  - 误判
  - 可能暴出一些不会被外部（攻击者）发现的问题
  - 结果有风险等级，但不知道哪个会被攻击者锁定

RASP 运行时自我保护技术



Gartner

G00269825

Maverick* Research: Stop Protecting Your Apps; It's Time for Apps to Protect Themselves

Published: 25 September 2014

Analyst(s): Joseph Feiman

Modern security fails to test and protect all apps. Therefore, apps must be capable of security self-testing, self-diagnostics and self-protection. It should be a CISO top priority. (Maverick research deliberately exposes unconventional thinking and may not agree with Gartner's official positions.)

实时应用自我保护技术（Runtime Application Self - Protection）也称RASP技术

是2014年9月Gartner的调研员Feiman提出的一种全新概念。

实时应用自我检测
Runtime Application Security Test / RAST

- 在应用运行时结合上下文进行**自我测试与监控**
- 详细记录攻击行为（or合法性校验行为）
- 堆栈信息代码级定位漏洞
- 可根据漏洞的严重等级视时间和经历对其进行修补

为什么需要RASP技术

- 程序完成的太久远，找不到源代码
- 漏洞数量太多
- 开发团队缺乏安全经验
- 第三方供应商的漏洞修复周期长
- 系统中存在未知的漏洞

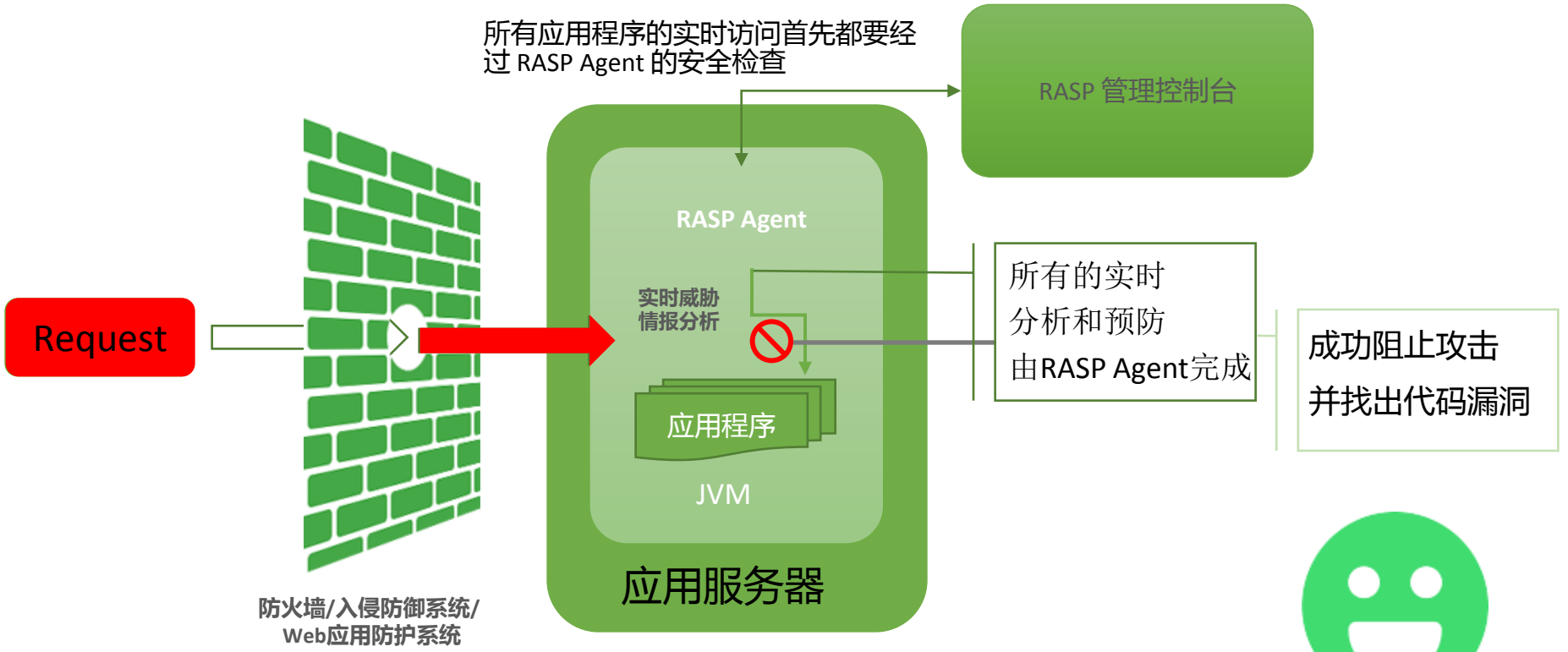
所以，你需要使用RASP技术打**虚拟补丁**，保护你带病上线的应用程序

它像一剂疫苗注入到应用中，与应用一起运行，对外提供服务

结合应用的逻辑和数据流，在运行时对访问应用的代码进行检测

对于已知漏洞，相当于为其打了虚拟补丁，起到补偿控制的作用

RASP技术请求示例图



该代理安装在应用程序服务器作为实时威胁智能筛选器

Q & A

Thank You

谢谢您的聆听