

入侵检测防护高效发现安全风险

莫展铵
平安人寿高级安全技术工程师

目 录

- 一、甲方视角下漏洞挖掘的痛点
- 二、工具原理与落地实践
- 三、漏洞治理的流程与扩展延伸
- 四、实施效果与总结

一、甲方视角下漏洞挖掘的痛点

甲方视角下漏洞挖掘的痛点

安全测试覆盖的功能不全：

- 受限于应用系统的架构（如系统间调用接口不可见）
- 功能调用的完整性（如前端页面功能缺陷）
- 接口的功能差异（如参数差异导致功能逻辑差异）

安全测试用例覆盖不全：

- 基于扫描工具的漏洞挖掘存在局限性
- 可投入的人力渗透资源有限
- 基于人员经验的漏洞挖掘方式存在差异

二、工具原理与落地实践

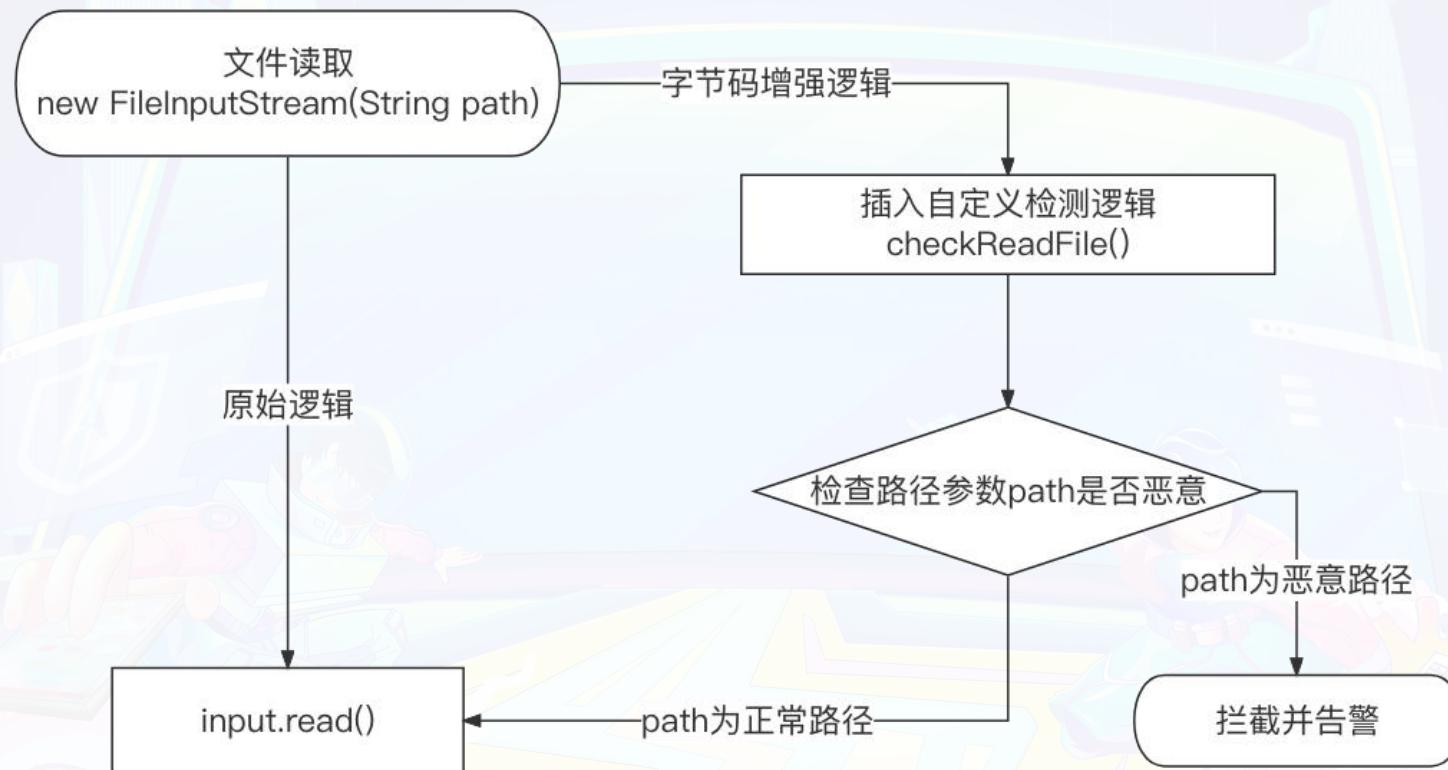
2.1 工具原理

为了解决上述的痛点，我们引入了RASP (Runtime Application self-Protection)

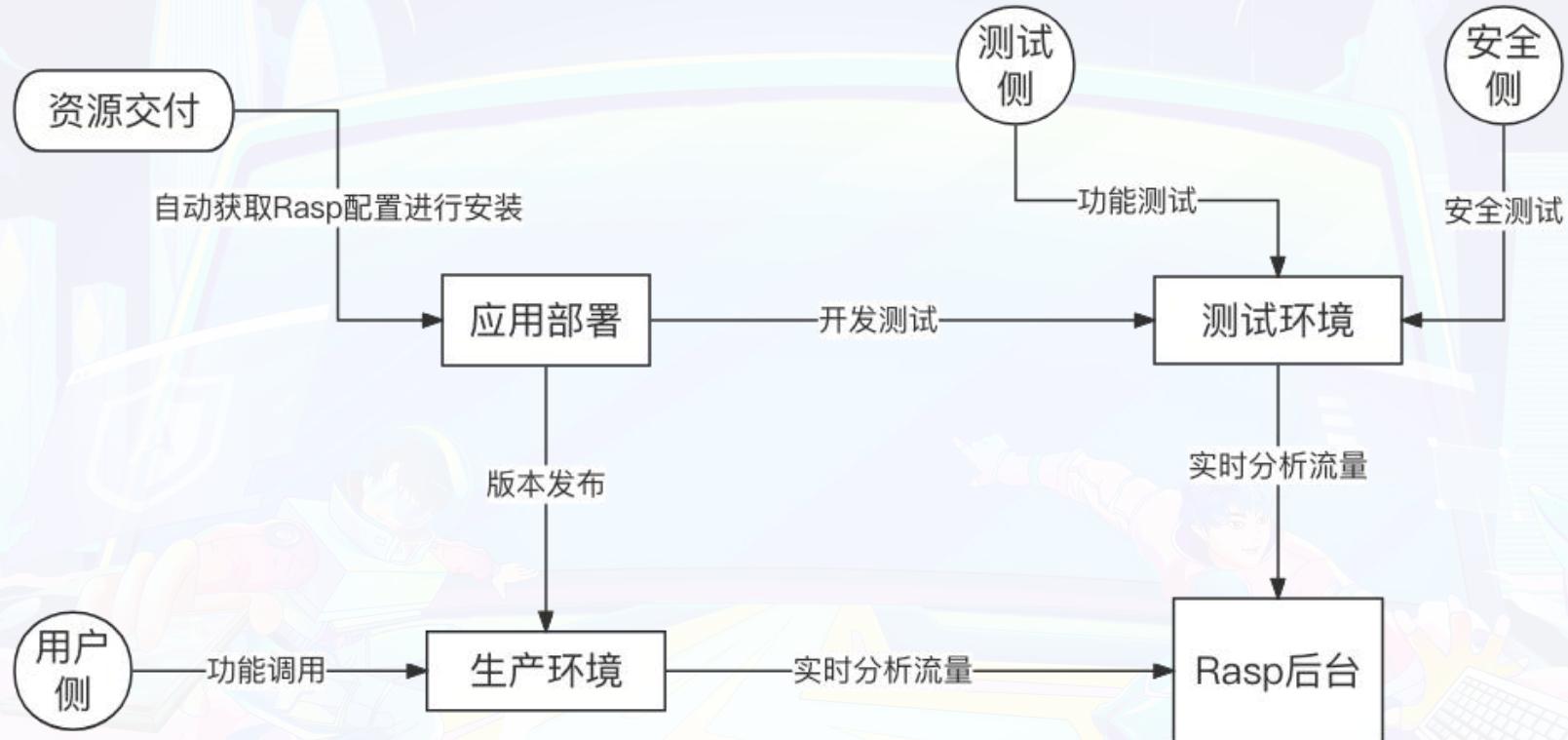
RASP通过JavaAgent & Instrumentation机制，对JVM (Java Virtual Machine) 进行注入，使用字节码增强技术对指定的类进行动态的操作和修改。

通过在函数方法的前后注入漏洞检测逻辑，进而实现漏洞发现以及漏洞拦截的效果。

2.1 工具原理



2.2 落地实践



2.2 落地实践

未采用RASP

安全测试覆盖面

漏洞发现效率

漏洞防护效果

采用RASP

安全测试覆盖面

漏洞发现效率

漏洞防护效果

VS

三、漏洞治理的流程与延伸扩展

3.1 漏洞治理的流程

告警面板集成至安全运营平台，可根据应用系统、漏洞类型、接口URL等层次进行筛选，提供批量处理、漏洞建项、状态变更等数据流转功能。

The screenshot displays a web-based application for managing security vulnerabilities. At the top, there is a search bar with dropdown menus for '全部' (All), '导出' (Export), and '批量处理' (Batch Processing). Below the search bar are several filter fields: '时间' (Time) with '开始日期' (Start Date) and '结束日期' (End Date) inputs; '拦截状态' (Interception Status) with '拦截请求' (Interception Request) and '记录日志' (Log Record); '漏洞类型' (Vulnerability Type) containing 'Spring监控未授权访问' (Spring Monitoring Unauthorized Access), '命令执行' (Command Execution), and '+ 27 ...'; '处理状态' (Handling Status) set to '全部' (All); '目标URL' (Target URL) and 'Hostname/IP' (Hostname/IP) fields; and a '搜索' (Search) button.

The main area shows a table of detected vulnerabilities:

<input type="checkbox"/>	最后发现	URL	系统名称	攻击来源	最后状态	漏洞类型	报警消息	处理状态	操作
<input type="checkbox"/>	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	拦截请求	目录遍历	[REDACTED]	待处理	处理 详情
<input type="checkbox"/>	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	记录日志	SQL 注入	[REDACTED]	待处理	处理 详情
<input type="checkbox"/>	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	记录日志	SQL 注入	[REDACTED]	待处理	处理 详情
<input type="checkbox"/>	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	记录日志	SQL 注入	[REDACTED]	待处理	处理 详情

3.1 漏洞治理的流程

- 告警时间
- 漏洞类型
- 漏洞参数
- 应用堆栈
- 资产信息

报警详情

漏洞详情	请求信息	资产信息	修复建议
报警时间 2023-09-11 16:15:26			
报警消息 [SQL注入]SQLi - SQL query structure altered by user input, request parameter name: [REDACTED]			
数据库类型 postgresql			
执行的SQL语句 select count(*) [REDACTED]			
应用堆栈 MD5 a61622c2d858d9221ff3c07e7adc04f4			
应用堆栈 org.postgresql.jdbc.PgConnection.prepareStatement(PgConnection.java) org.postgresql.jdbc.PgConnection.prepareStatement(PgConnection.java:1745) org.postgresql.jdbc.PgConnection.prepareStatement(PgConnection.java:430) [REDACTED]			

3.1 漏洞治理的流程

- 接口URL
 - 请求来源
 - HTTP 请求方式
 - HTTP 请求头
 - HTTP 请求参数

报警详情

漏洞详情 **请求信息** 资产信息 修复建议

请求编号
1b6d38555e4343b8b5a1a30e01aadb48

请求 URL
POST http://[REDACTED]/query

请求来源
30.188.136.121

完整 Header 信息

```
accept:/*  
ark:  
cmsl:  
conr:  
cont:  
cont:  
host:  
secr:  
user:  
x-ac:  
x-b:  
x-b:  
x-b:  
x-sp:
```

Form 参数

phoneNo= [REDACTED]

3.1 漏洞治理的流程

护白名单

添加白名单

* 选择应用:

* URL/URI ②:

URL 请输入

白名单备注:

请输入

检测点:

关闭所有检测点 Sping监控未授权访问 命令执行 任意文件删除 Transformer 反序列化 目录遍历 DNS请求 EVAL 代码执行 任意文件上传
 任意文件包含 JNDI请求 文件链接 OGNL 代码执行 任意文件下载 文件重命名 HTTP 响应采样检测 Sping监控未授权访问 SQL注入
 SQL 语句异常 SSRF 请求伪造 SSRF 请求伪造(重定向后) 任意文件上传(PUT) WebShell - 变形后门 WebShell - 命令执行 WebShell - 中国菜刀
 WebShell - 后门上传 WebShell - LD_PRELOAD 后门 任意文件写入 Echo XSS 跨站脚本攻击 BODY XSS 跨站脚本攻击 XXE 外部实体加载

取消 确认

MOZHANAN859-20230904

SQL 语句扫描

LCDP-LCDP 安全扫描与数字沙盒平台, 由白山云 STG

扩展延伸

次生安全风险的收敛:

- 基于应用框架特性、功能模块的差异，采用鉴权绕过、未授权访问、越权操作、攻击向量混淆等手法，结合常用的漏扫脚本工具、逻辑漏洞挖掘思维，批量对功能接口进行深入测试

联动安全合规:

- 联动合规，协同治理，对涉及客户、系统用户以及公司内部的敏感信息功能模块进行风险备案，审批管理，收敛敏感信息泄漏的风险点。

3.2 扩展延伸

调用脚本批量进行扩展检测：

POST /user/info HTTP/1.1

Host: test.com

Cookie: SESSION=用户凭证

Connection: close

Content-Type: application/json

Content-Length: 11

{"userNo": "1234"}

接口未授权：移除鉴权相关字段，最简HTTP请求访问

匹配用户参数，越权操作：/user/info?userNo=5678

版本迭代的其余接口：/user/v1/info

测试接口未删除：/user/infoTest

JSONP跨域检测、回调XSS：GET /user/info?callback=

鉴权、逻辑校验绕过：——

getRequestURI()绕过：/user;/info

UseTrailingSlashMatch绕过：/user/info/

UseSuffixPatternMatch绕过：/user/info.任意字符

Java正则匹配\r、\n绕过：/user/in%0afo

3.2 扩展延伸

合规审批面板，根据子系统、接口URL进行划分，提供涉敏接口批量处理、详情展示、审批处理功能

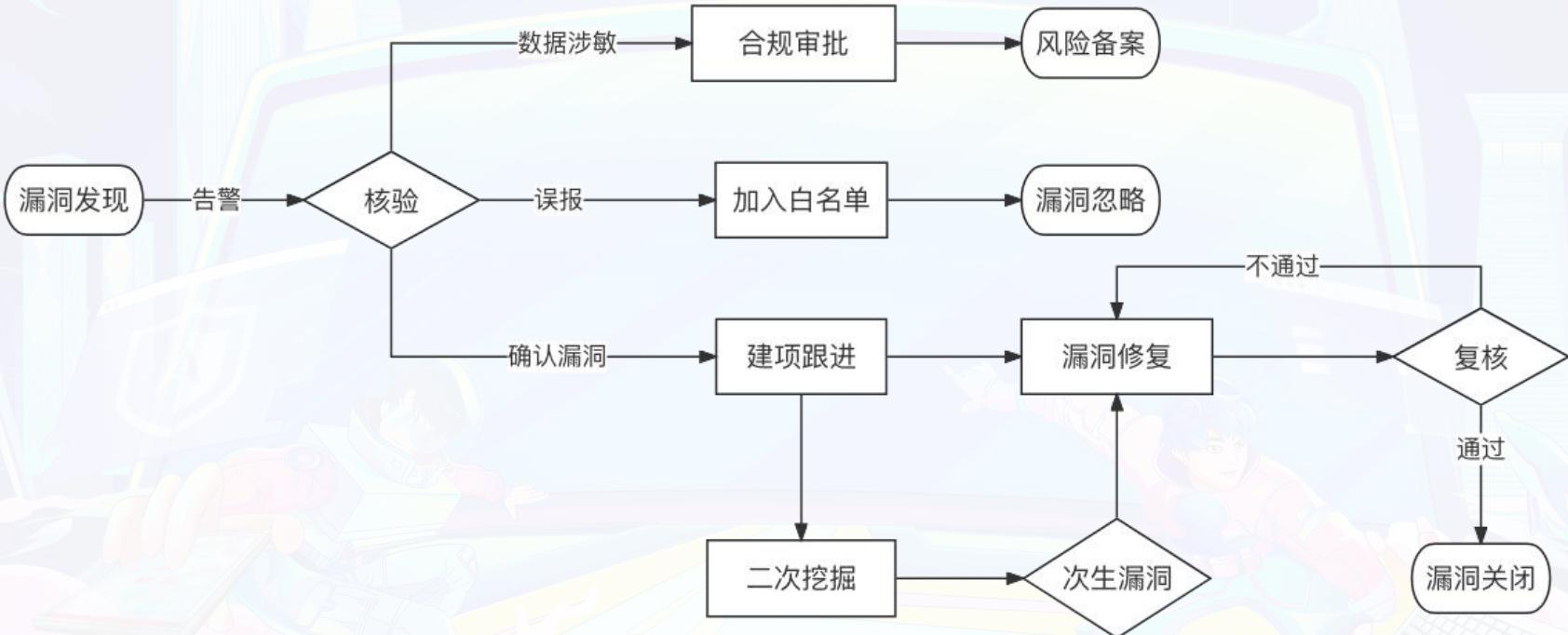
全部 导出 批量处理

时间: 2023-08-13 00:00:00 ~ 2023-09-13 23:59:59 处理状态: 审批中

目标URL: 搜索 重置

最后发现	URL	系统名称	攻击来源	最后状态	漏洞类型	报警消息	处理状态	操作
2023-09-12 23:54:00	[REDACTED]	[REDACTED]	[REDACTED]	记录日志	HTTP 响应采样检测	PII leak detected 7(Identity Card), e Number)	审批中	处理 提交审批 审批 详情
2023-09-12 23:16:52	[REDACTED]	[REDACTED]	[REDACTED]	记录日志	HTTP 响应采样检测	PII leak detected 0(Identity Card), e Number)	审批中	处理 提交审批 审批 详情
2023-09-12 20:56:00	[REDACTED]	[REDACTED]	[REDACTED]	记录日志	HTTP 响应采样检测	PII leak detected Number)	审批中	处理 提交审批 审批 详情
2023-09-12 18:58:05	[REDACTED]	[REDACTED]	[REDACTED]	记录日志	HTTP 响应采样检测	PII leak detected 8(Identity Card), e Number)、市(审批中	处理 提交审批 审批 详情

3.3 流程归纳



四、实施效果与总结

实施效果与总结

高效覆盖常规漏洞自动检测:

- 在多场景下，提高功能接口的扫描覆盖度，补足安全测试盲区
- 优化了系统持续迭代过程中，依赖人力测试导致安全测试用例覆盖不全的问题
- 安全人员可以更加专注在业务逻辑漏洞上的风险收敛以及扫描规则优化

精准告警，有效防护:

- 补足了依赖请求特征进行防护的传统安全工具短板，实现应用层级精细化的漏洞告警、攻击拦截以及资产定位功能
- 配置化、自动化批量处理误报，草木不再皆兵，监控告警不再头大



THANKS