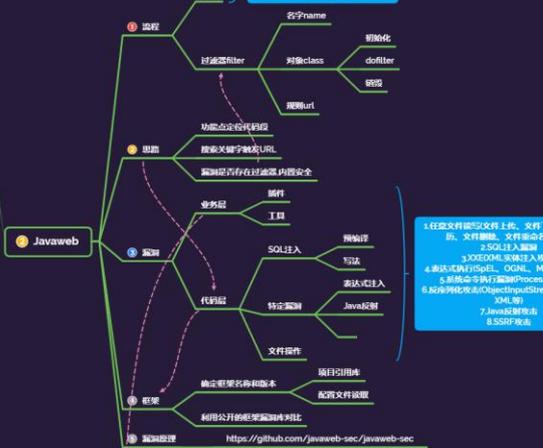
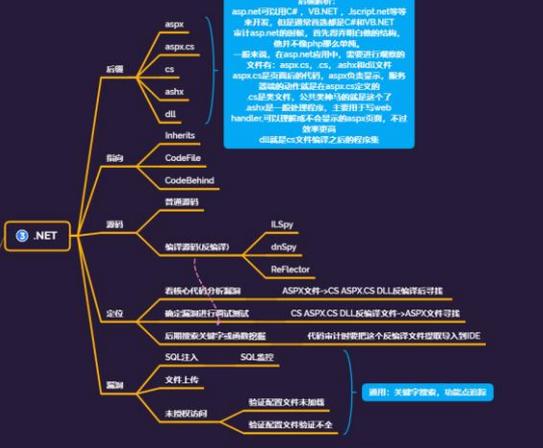
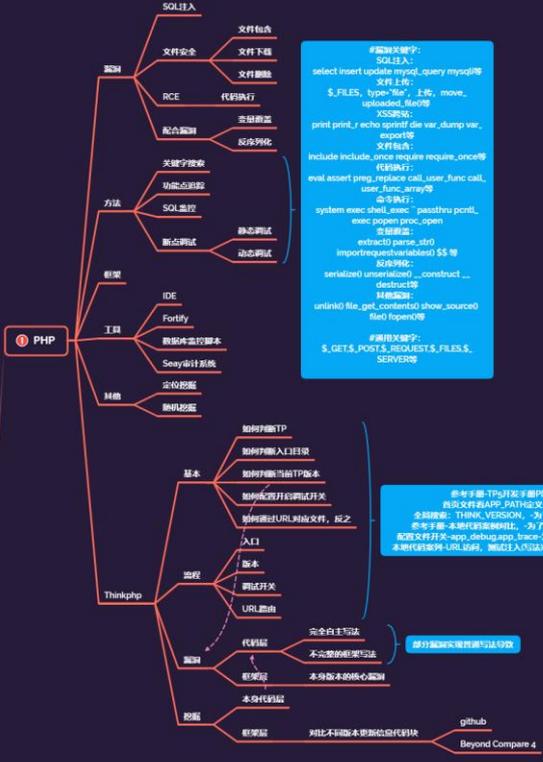


代码审计-SAST&IAST 项目&火线洞态&Agent 部署&插件安装&产品测

评

代码审计-小迪安全



名称	DAST	SAST	MAST
AWVS	是	否	否
XRAY	是	否	否
APPSCAN	是	否	否
RIPS	否	是	否
Snyac	否	是	否
VCG	否	是	否
Fortify	否	是	否
Findbugs	否	是	否
AWVS	是	否	否
XRAY	是	否	否
APPSCAN	是	否	否
RIPS	否	是	否
Snyac	否	是	否
VCG	否	是	否
Fortify	否	是	否
Findbugs	否	是	否

#知识点:

- 1、代码审计-开源版&商业版
- 2、代码审计-单语言&多语言
- 3、代码审计-DAST&SAST&IAST

#Java 审计知识点:

<https://xz.aliyun.com/t/7945> java 代码审计常规思路和方法.pdf

SQL 注入, XSS 跨站, RCE 执行, 反序列化, 身份验证, SPEL, SSTI, 三方组件安全等

#章节点:

- 1、语言审计-PHP&.Net&Java&Python
- 2、漏洞审计-注入&上传&RCE&未授权等
- 3、框架审计-ThinkPHP&Spring&Flask 等
- 4、工具审计-RIPS&VCG&Fortify&Bandit 等
- 5、技术审计-动静态调试&DAST&SAST&IAST 等

#简要点:

- 1、代码审计必备知识点:

环境搭建使用, 工具插件安装使用, 掌握各种漏洞原理及利用, 代码开发类知识点。

- 2、代码审计开始前准备:

审计目标的程序名, 版本, 当前环境(系统, 中间件, 脚本语言等信息), 各种插件等。

- 3、代码审计挖掘漏洞根本:

可控变量及特定函数, 不存在过滤或过滤不严谨存在绕过导致的安全漏洞。

- 4、代码审计教学计划:

审计项目漏洞原理->审计思路->完整源码->应用框架->验证并利用漏洞。

5、代码审计教学内容

对比项	DAST	SAST	IAST
测试对象	Web应用程序	Web应用程序 APP的漏洞	Web应用程序 APP的漏洞
部署成本	低	低	高
使用成本	较低, 基本无需人工验证	高, 人工排除误报	低, 基本没有误报
漏洞检出率	中	高	较高
脏数据	非常多	较少	几乎没有
研发流程集成	测试/线上运营阶段	研发阶段	测试阶段
误报率	低	高	极低 (几乎为0)
测试覆盖度	低	高	高
检查速度	随测试用例数量稳定增加	随代码量呈指数增长	实时检测
逻辑漏洞检测	支持部分	不支持	支持部分
影响漏洞检出率因素	与测试payload覆盖度相关 企业可优化和扩展	与检测策略相关 企业可在定制策略	与检测策略相关 企业可定制测量
第三方组件漏洞检测	支持	不支持	支持
支持语言	不区分语言	区分语言	区分语言
支持框架	不区分框架	区分框架	区分框架
侵入性	较高, 脏数据	低	低
风险程度	较高, 扫挂/脏数据	低	低
漏洞详情	中, 请求	较高, 数据流+代码行数	高, 请求+数据流+代码行数
CI/CD集成	不支持	支持	支持
持续安全测试	不支持	支持	支持
工具集成	无	开发环境集成 构建工具、问题跟踪工具	构建工具、自动化
其他	无法定位漏洞的具体代码行数和产生漏洞的原因		不支持C, C++和Golang等语言

对比项	DAST	SAST	IAST
商业产品	AppScan、AWVS、 webinspect burpsuite	Fortify、Checkmarx	默安-雳鉴IAST 新思Seeker软件 开源网安SecZone VulHunter 、墨云VackBot等, 国外: Contrast Security等
开源产品	Owasp ZAP、Xray	Raptor、RIPS、Seay源代码审计系统、 VCG等	百度RASP等
部署成本	低	低	高
使用成本	较低, 基本无需人工验证	高, 人工排除误报	低, 基本没有误报
漏洞检出率	中	高	较高

演示案例：

➤ 代码审计利器-IAST-插件

➤ 代码审计利器-IAST-Agent

DAST&SAST&IAST

DAST: 动态应用程序安全测试 (Dynamic Application Security Testing) 技术在测试或运行阶段分析应用程序的动态运行状态。它模拟黑客行为对应用程序进行动态攻击, 分析应用程序的反应, 从而确定该 web 应用是否易受攻击。

SAST: 静态应用程序安全测试 (Static Application Security Testing) 技术通常在编码阶段分析应用程序的源代码或二进制文件的语法、结构、过程、接口等来发现程序代码存在的安全漏洞。

IAST: 交互式应用程序安全测试 (Interactive Application Security Testing) 是 2012 年 Gartner 公司提出的一种新的应用程序安全测试方案, 通过代理、VPN 或者在服务端部署 Agent 程序, 收集、监控 Web 应用程序运行时函数执行、数据传输, 并与扫描器端进行实时交互, 高效、准确的识别安全缺陷及漏洞, 同时可准确确定漏洞所在的代码文件、行数、函数及参数。IAST 相当于是 DAST 和 SAST 结合的一种互相关联运行时安全检测技术。

目前还有些商业版平台未介绍如下:

静态: CheckMarx 奇安信代码卫士等

IAST: 悬镜灵脉 IAST 默安雳鉴 IAST 等

PHP -Seay RIPS CheckMarx Fortify VCG Kunlun-M

NET -VCG Fortify CheckMarx

Java-Fortify Fortify CheckMarx

Python-Bandit Fortify CheckMarx

JS-Kunlun-M NodeJsScan Fortify CheckMarx

Go-Gosec CheckMarx

Bandit

参考: <https://bandit.readthedocs.io/>

安装: `pip install bandit`

linux:

安装后会在当前 Python 目录下 bin

使用: `bandit -r 需要审计的源码目录`

windows:

安装后会在当前 Python 目录下 script

使用: `bandit -r 需要审计的源码目录`

`D:\Python3\Scripts>bandit.exe -r F:\python_webapp\www\`

Kunlun-M

1、安装依赖库: `pip install -r requirements.txt`

2、配置文件启用: `cp Kunlun_M/settings.py.bak Kunlun_M/settings.py`

3、初始化数据库: `python kunlun.py init initialize`

4、加载规则数据库: `python kunlun.py config load`

Web 使用: `D:\Python38\python.exe kunlun.py web -p 9999`

Cli 使用: `D:\Python38\python.exe kunlun.py scan -t`

#代码审计利器-IAST-火线洞态测评报告

文档: <https://doc.dongtai.io/docs/introduction>

安装: <https://github.com/HXSecurity/DongTai>

控制台: <https://iast.huoxian.cn/project/projectManage>

测试台: <https://labs.iast.huoxian.cn/#/images/index>

主要测评: 应用漏洞&组件安全

在线靶场测评: [openrasp&spring](#)

本地应用测评: IDEA_Plugin 插件

Agent 部署测评: [DongTai OpenApi](#)

1、IDEA_Plugin 插件

<https://github.com/HXSecurity/DongTai-Plugin-IDEA/releases>

<https://doc.dongtai.io/docs/getting-started/agent/plugin/java-agent-idea>

2、Agent 部署

<https://doc.dongtai.io/docs/category/agent-%E5%AE%89%E8%A3%85%E6%8C%87%E5%8D%97>

#代码审计利器-IAST-Semmler 测评报告

<https://lgtm.com/> 代码平台接入测评

测评: <https://lgtm.com/>

#补充说明:

目前免费可试用的四个 IAST 平台:

1、openrasp-iast/openrasp-iast 是一款灰盒扫描工具,目前开源的 IAST 扫描器,通过安装 Agent 和扫描器,能够结合应用内部 hook 点信息,针对获取到的 url 请

求参数进行... 从而检测到安全漏洞

涉及资源：

补充：[涉及录像课件资源软件包资料等下载地址](#)
