网络空间威胁对抗防御技术研讨会
暨 第十届安天网络安全冬训营

沧海横流

执行体全量识别与精细管控

# 防微杜渐
# 从代码维度审视执行体安全

安天 | 代码安全中心

# 目 录

# 01 代码安全趋势

# 过去十年供应链安全现状

**2014.04**

程序员Sean Cassidy在自己的博客上放出了OpenSSL一个 严重漏洞的细节，随后这一 “灾难级”的严重安全漏洞 席卷了整个互联网。国内互 联网大厂以及许多安全厂商 都被爆出漏洞

开源软件漏洞

**2015.09**

开发者使用非苹果公司官方渠道的XCODE工具开发苹果应用程序时，会向正常的苹果APP中植入恶意代码。被植入恶意程序的苹果APP可以在APP Store正常下载并安装使用。该恶意代码具有信息窃取行为，并具有进行恶意远程控制的功能

开发工具被污染

**2017.05**

来自瑞士安全公司Modzero的研究人员在检查Windows Active Domain的基础设施时发现惠普音 频驱动中存在一个内置键盘记录 器监控用户的所有按键输入

厂商预留后门

**2018.12**

驱动人生旗下多款软件携带后门病毒DTStealer， 仅半天时间感染了数万台电脑。该病毒进入电脑后，继续通过“永恒之蓝”高危漏洞进行全网传播，并回传被感染电脑的IP地址、 CPU型号等信息

升级劫持

**2020.04**

安全公司ReversingLabs发现Ruby语言官方模块包代码库RubyGems， 被植入了超725个恶意软件包。恶意软件包的下载量接近100000次。如， "atlas-client"是一个诱骗的诱饵程序包，被2100次下载，用来模仿的"atlas_client"

恶意软件

**2020.05**

GitHub 安全博客警告了针对 Apache NetBeans IDE 项目的 开源供应链攻击 Octopus Scanner，最终统计显示，有26个开源项目被植入了Octopus Scanner 后门

植入后门

**2020.12**

全球最著名的网络安全管理软件供应商SolarWinds遭遇国家级APT团伙高度复杂的供应链攻击并植入木马后门。该攻击直接导致包括美国关键基础设施、军队、政府在内的18000+企业客户全部受到影响：可任由攻击者完全操控

源代码被污染

**2021.01**

安全研究人员RyotaK，披露了他是如何渗透进微软Visual Studio Code的官方GitHub存储库的，微软Visual Studio Code的持续集成脚本中存在一个漏洞，该漏洞导致他能够渗透进微软Visual Studio Code的官方GitHub存储库并向其提交文件。 这种代码库中的安全问题将有可能被应用到软件供应链攻击中

开发环境漏洞

**2021.03**

安全研究人员通过一种新颖的软件供应链攻击方式（依赖关系混乱），成功的侵入了微软、苹果、 PayPal、特斯拉、优步等35家国际大型科技公司的内网， 在向这些公司提交安全报告后，他获得超过 13 万美元的奖金

依赖混乱

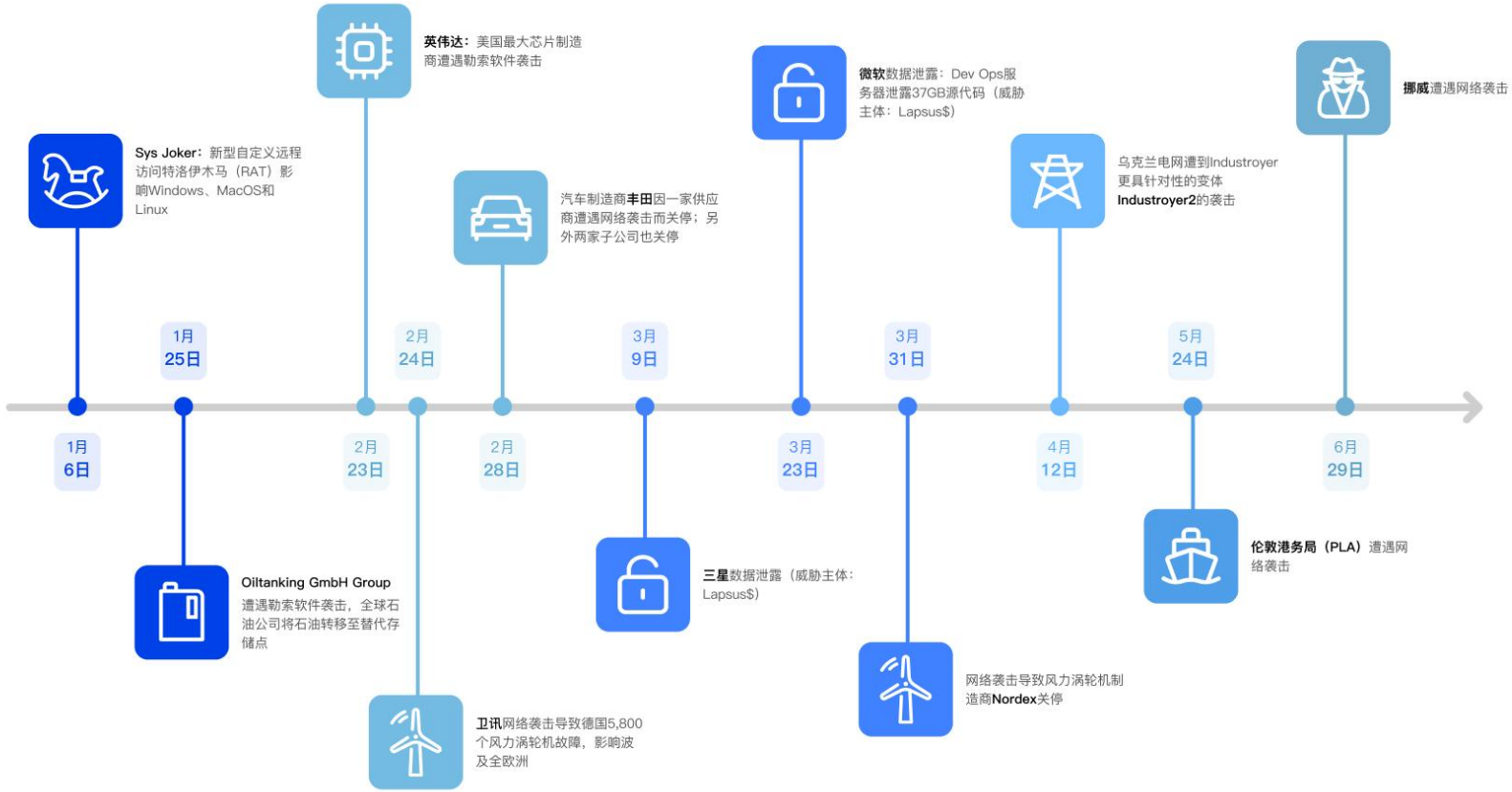
**2022.04**

2022年04月05日，境外黑客组织 “AgainTheWest” （以下简称“ATW组织”）在境外社交平台的群组中发布了针对我国卫生部的源代码泄露相关信息，并提供对应的数据下载。

“ATW组织”通过已泄露的存储库访问权限或对已泄露的存储库账号进行暴力破解后获取到访问权限，进而完成此次源代码窃取。

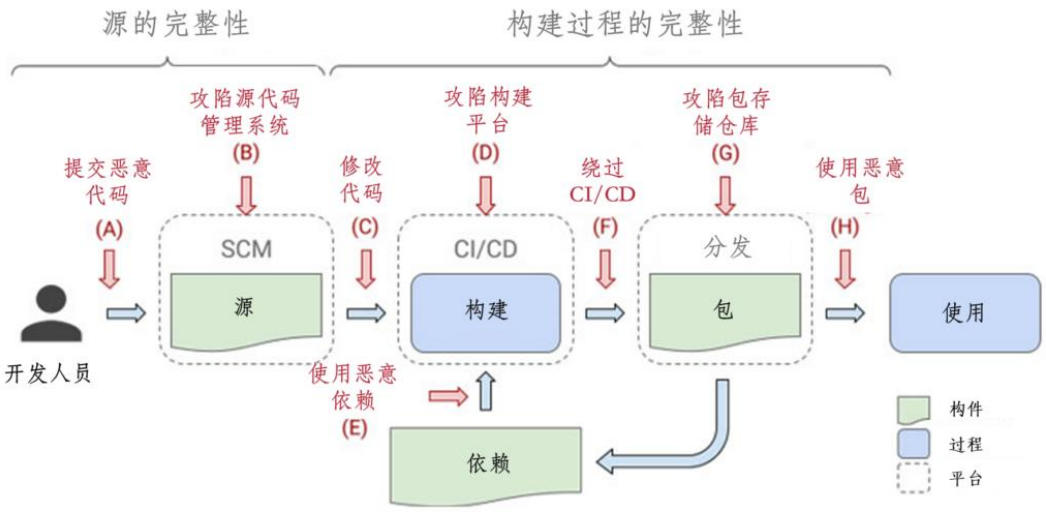
源代码泄露

# 2022年供应链重大事件



# 代码安全趋势

2021年6月，谷歌发布了名为Supply chain Levels for Software Artifacts（软件构建的供应链级别，简称为 SLSA ）的解决方案。




- **SLSA解决的问题：**
  - ✓ 软件生产商想要保护他们的供应链，但不知道具体如何保护
  - ✓ 软件消费者希望了解并限制他们遭受供应链攻击的风险，但没有方法
  - ✓ 单独的工件签名只能防止我们关心的攻击的一个子集

- **SLSA制定的标准是软件生产者和消费者的指导原则：**
  - ➢ 软件生产者可以遵循这些准则来使他们的软件更加安全
  - ➢ 软件消费者可以根据软件包的安全状况做出决定

# 代码安全所面对的场景

A

作为安全开发系统：

- 在编码阶段出现的漏洞未及时发现修复或无修复措施无法修复，再在后期修复成本较高，周期长
- 由于研发人员主要专注于固定业务，可能缺乏安全相关的能力，导致漏洞难以被修复

通信

金融

能源

涉密

B

作为代码安全监管系统：

- 在上线之前，难以针对第三方开发人员是否留下后门等隐患，作出有效的排查；并且面对如log4j重大安全事件时，难以对现有上线服务做自动化排查
- 缺乏与第三方开发人员周期性沟通安全问题的流程，无法及时把控项目的安全问题，最终导致项目携带难以发现的安全漏洞却验收

# 从代码维度审视执行体安全

软件组件分析帮助开发人员识别与许可证有关的风险和开源库的漏洞。它有助于为开发人员和安全团队提供升级组件的建议。

1. 对组件进行把控——面向源代码工程和二进制文件，SCA能根据企业安全策略限制不安全组件的使用，防止开发人员使用高风险的组件，并推荐替代品。

2. 软件物料清单（SBOM）——对软件交付物的静态快照，能够显著提高软件组件的透明度，从而能够对软件做信息验证。

3. 预防使用恶意组件——攻击者常常会发布一些与知名组件名字类似的恶意组件，而SCA能够识别这些恶意组件，以防止不知情的开发者下载使用这些组件。

4. 防止依赖混淆攻击——通常构建系统可能会从上游拉取一些最新版本的组件包，攻击者将私有模块的“更高版本”上传到公开仓库中时会造成攻击机会，从而导致客户端自动下载恶意“最新版本”。而SCA能够提供包来源信息和校验信息，以确保依赖项都是在预期内的。

# 从代码维度审视执行体安全

静态分析安全测试能够发现代码中的潜在漏洞。有助于让研发人员发现代码中潜在的风险。

1

降低漏洞修复成本——静态分析安全测试能够在不执行代码的情况对代码安全做分析，使得它适合SDLC的早期阶段。从最初的编码阶段开始测试应用程序有助于减少修复所发现的漏洞所需的时间和成本。

2

提高研发人员的安全意识——静态分析安全测试会为漏洞提供修复措施，和安全的实现方式，帮助开发人员在代码编写的过程中提升安全意识。

3

减小意外引入后门的可能——研发人员为了方便代码调试，可能会在代码中硬编码一些条件或者预留一些调试接口，而如果这些代码被引入到正式打包流程，都可能会导致程序被预留后门。

4

API安全治理——研发人员不应该去调用一些已经被弃用或者带安全风险的方法或者接口，静态分析安全检测系统能够提高这些不推荐API的可见性，从而避免研发人员在开发过程中使用。

# 02 组件安全治理

# 组件治理难点

直接间接依赖

SCA是否只需要提供每个依赖的升级信息?

A

B

如何给出最优的升级版本?

版本复杂性和知识库质量

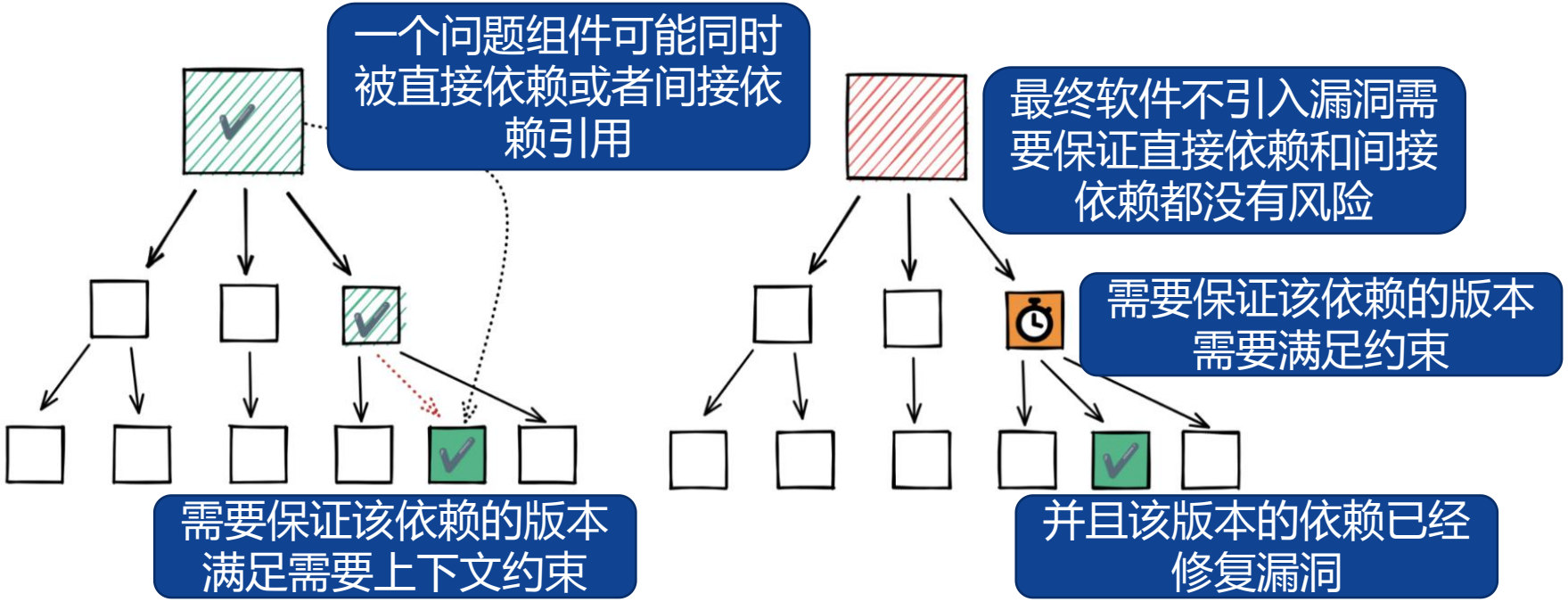
升级是否是最佳解决方案

版本之间的接口兼容性该如何保证?

C

# 间接依赖vs直接依赖

组件本身的安全性也取决于其本身的依赖项的安全性。



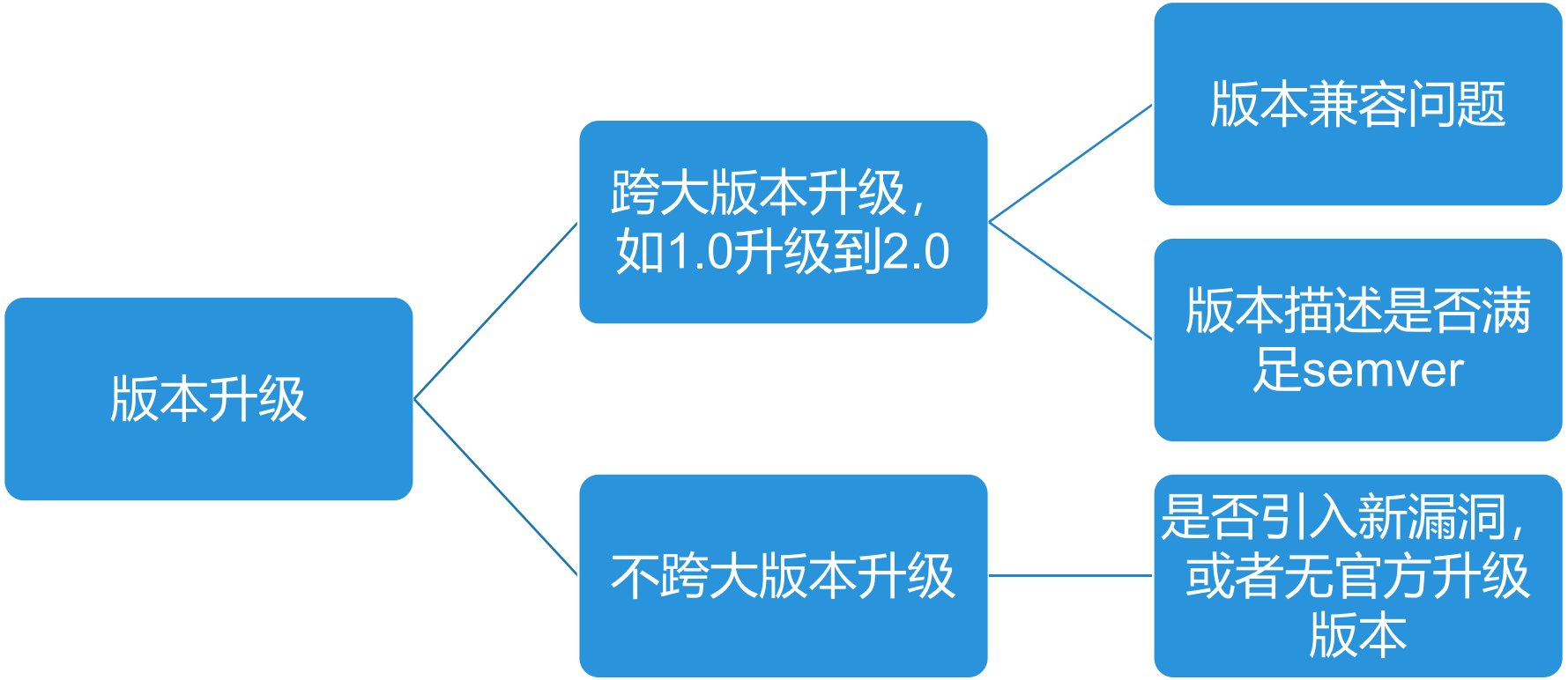
# 版本的复杂性和知识库质量

包管理生态和漏洞库的不统一。

| 包管理器 | 作用域类型 | 版本规定 |
| --- | --- | --- |
| Maven | GAV | 无约定 |
| npm | 命名空间（可选）和包名 | semver |
| pip | 包名 | 无约定 |
| golang | url和路径 | 无约定 |
| php composer | Vendor和包名 | semver |
| Rust cargo | 包名 | 无约定 |
| .Net NuGet | 包名 | Semver(可选) |
| GHSA数据库 | Ecosystem标签和语言特定格式 | 语言特定 |
| Nvd数据库 | CPE | 无约定 |

# 版本之间的接口兼容性

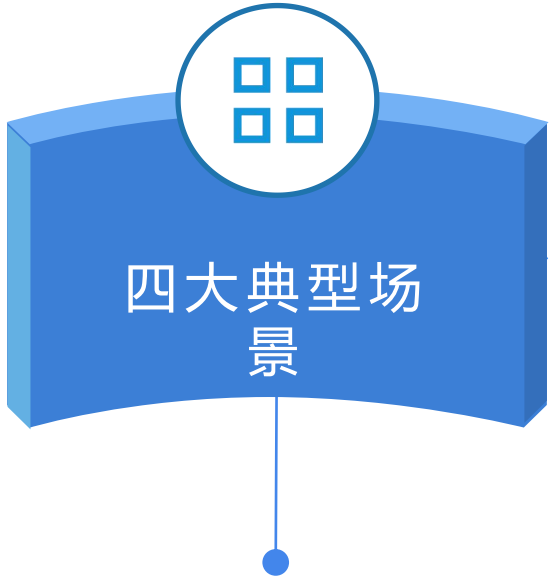
简单的版本升级同样也不一定能解决安全问题，并且还需要综合考虑研发成本。



# 组件安全治理

## 四大常见问题

- 第三方组件中的已知漏洞
- 第三方组件的软件许可问题
- 恶意第三方组件中的问题
- 引用版本过旧的第三方组件



## 四大典型场景

- 软件开发阶段
- 突发0Day高危漏洞
- IoT固件/二进制SDK分析
- 容器镜像安全

## 三大开源痛点

- 缺乏完善且高质量的漏洞数据库
- 资产透视深度不足
- 上游SDK/固件无源码

# 安天软件组件分析系统

## 丰富的安全组件库

支持主流编程语言生态，与包管理器深度结合。拥有识别900w+ 开源组件信息、6000w+ 开源组件版本信息，且拥有安天认证自有组件库。

## 全面的开源组件漏洞数据库

不仅覆盖NVD、CNNVD的漏洞数据，还具备大量的私有高质量漏洞数据库，使用PackageURL进行统一管理，能够明确组件来源。

## 支持二进制文件检测

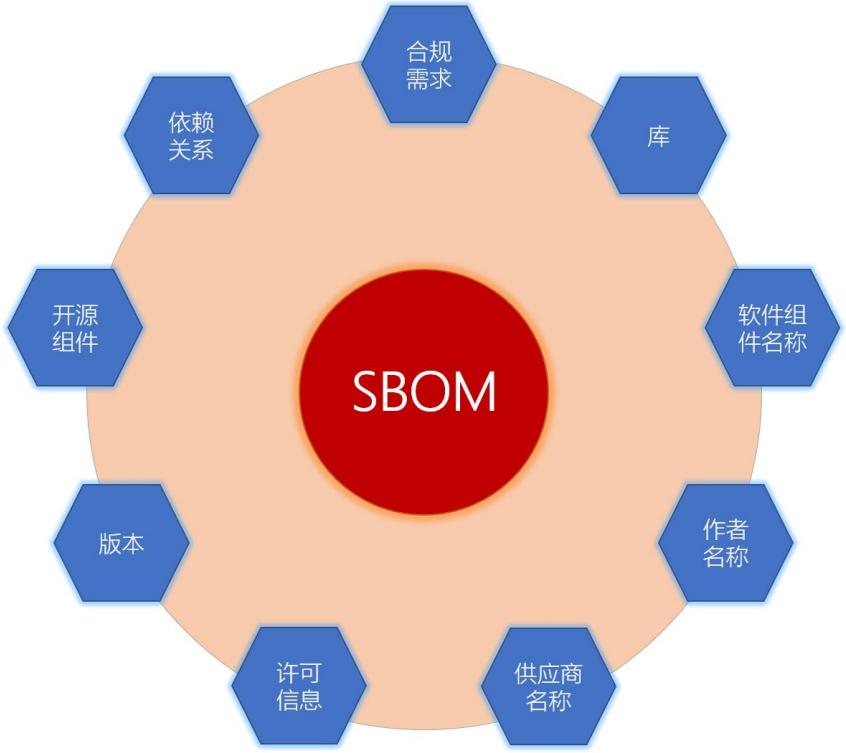
作为中下游厂商，拿到的SDK或固件往往都是无源码的。安天SCA能支持C/C++、C#、Golang、安卓应用编译后的文件的扫描。

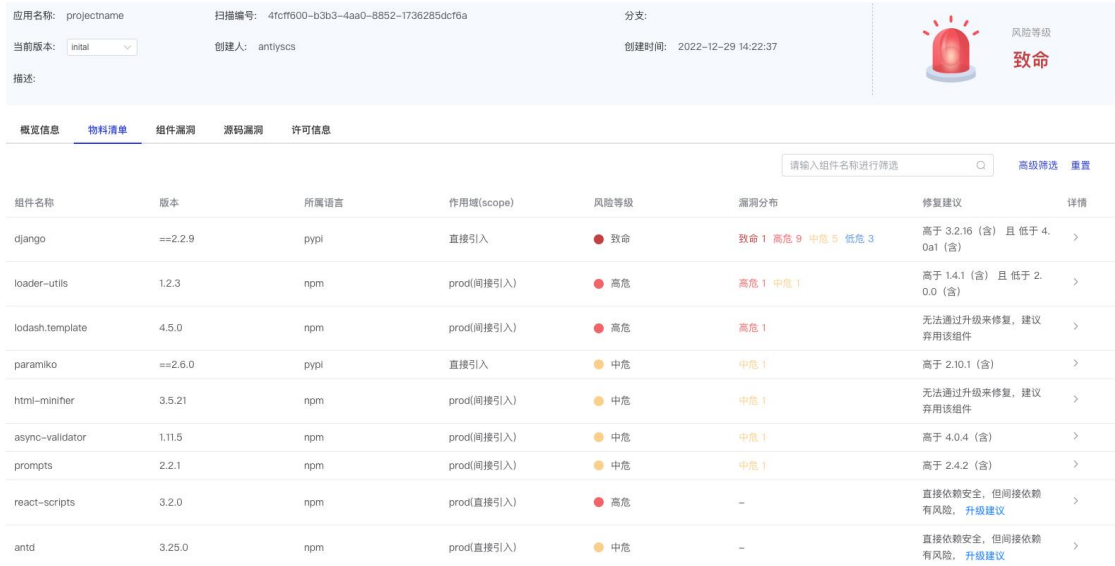
## 与开源生态深度结合

与Maven、npm等生态深度结合，提供私有的安全组件库，不仅能精准识别生命周期中各个阶段的风险组件，还能做到一键修复，极大降低开发人员的工作量。

# 安天软件组件分析系统

安天软件组件分析系统支持输出物料清单（SBOM），软件物流清单已经是国际公认的方法论，能够有效提高漏洞的可见性和许可证管理的能力。



应用名称: projectname　扫描编号: 4fcff600-b3b3-4aa0-8852-1736285dcf6a　分支:

当前版本: inital　创建人: antiyscs　创建时间: 2022-12-29 14:22:37

描述:

风险等级 致命

概览信息 | 物料清单 | 组件漏洞 | 源码漏洞 | 许可信息

请输入组件名称进行筛选　高级筛选　重置

| 组件名称 | 版本 | 所属语言 | 作用域(scope) | 风险等级 | 漏洞分布 | 修复建议 | 详情 |
|---|---|---|---|---|---|---|---|
| django | ==2.2.9 | pypi | 直接引入 | 致命 | 致命 1 高危 9 中危 5 低危 3 | 高于 3.2.16（含） 且 低于 4.0a1（含） | > |
| loader-utils | 1.2.3 | npm | prod(间接引入) | 高危 | 高危 1 中危 1 | 高于 1.4.1（含） 且 低于 2.0.0（含） | > |
| lodash.template | 4.5.0 | npm | prod(间接引入) | 高危 | 高危 1 | 无法通过升级来修复，建议弃用该组件 | > |
| paramiko | ==2.6.0 | pypi | 直接引入 | 中危 | 中危 1 | 高于 2.10.1（含） | > |
| html-minifier | 3.5.21 | npm | prod(间接引入) | 中危 | 中危 1 | 无法通过升级来修复，建议弃用该组件 | > |
| async-validator | 1.11.5 | npm | prod(间接引入) | 中危 | 中危 1 | 高于 4.0.4（含） | > |
| prompts | 2.2.1 | npm | prod(间接引入) | 中危 | 中危 1 | 高于 2.4.2（含） | > |
| react-scripts | 3.2.0 | npm | prod(直接引入) | 高危 | – | 直接依赖安全，但间接依赖有风险， 升级建议 | > |
| antd | 3.25.0 | npm | prod(直接引入) | 中危 | – | 直接依赖安全，但间接依赖有风险， 升级建议 | > |

# 安天软件组件分析系统

漏洞覆盖





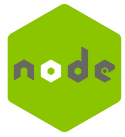

生态结合



开发平台



开发语言









# 03 代码安全治理

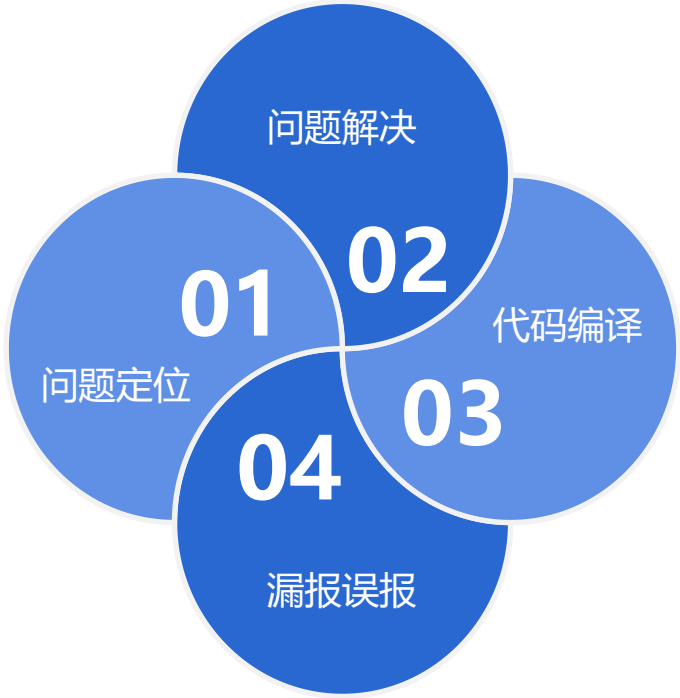
# 代码安全治理难点

## 难以追溯漏洞成因

在没有安全专家的情况下，难以发现问题本质

## 告警信息繁多

代码检测由于其特性，存在一定的误报且告警较多




## 源码问题无法解决

找到了问题代码的位置，但由于无法判断问题成因，不能有效的解决代码产生的问题，没有解决办法

## 代码无法编译

在无源码或源码无编译条件的情况下，代码审计困难重重

# 代码安全治理：静态分析安全测试系统

## 会做“阅读理解”的检测引擎

如果忽略了程序执行过程中的发生序关系，检测结果便会存在误报率高的问题。因此引入上下文敏感性分析来解决这一问题。上下文敏感分析是指在对程序进行过程间分析时，考虑函数调用的上下文信息。

## 支持二进制文件检测

支持多种语言类型和部分二进制文件进行检测，源码无编译条件下也可进行安全扫描

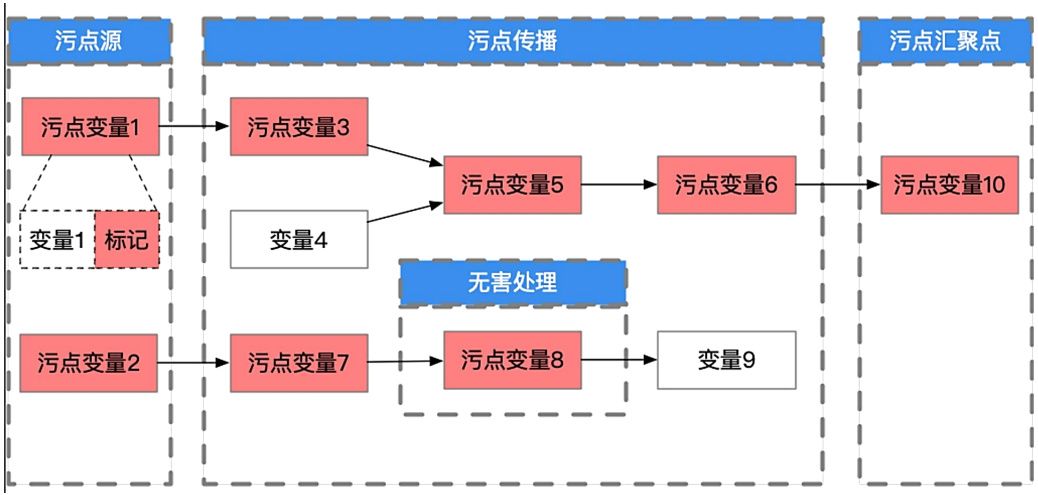
## 可视化的污点传播途径呈现

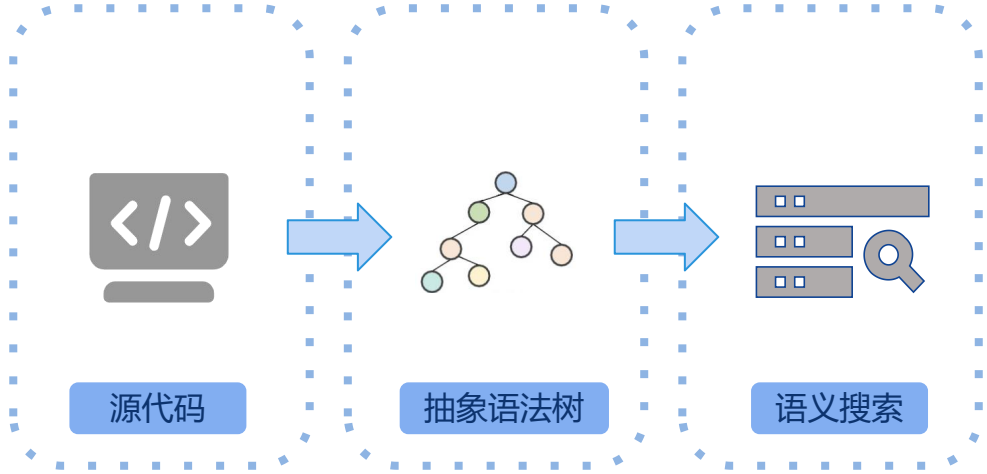
追踪分析源码的污点传播链路，通过对程序的污点传播路径进行分析，检测出由引入的用户数据流引起的深层风险漏洞。

## 输出详尽的检测报告

提供详细的检测报告，报告格式包括易于阅读的HTML格式之外，还可以输出SARFI格式，提供了详细的风险描述和修复建议，能够帮助开发人员提升了软件安全能力，便于日常安全运营。

# 代码安全治理核心能力




数据流分析技术

语义分析技术

# 能力场景适配

## 01 代码研发阶段

安天静态分析安全测试系统支持与用户的IDE深度结合，能够做一边编写代码，一边实时分析。

## 02 代码审计阶段

通过高精度数据流分析模块，能够逐步从攻击面最终到最终漏洞触发点，帮助审计人员理解漏洞成因。

## 03 代码测试阶段

结合配置文件，与CI/CD旁路接入，在代码测试阶段不仅能做到高精度的分析，并且还能与过往的漏洞修复情况做比对。

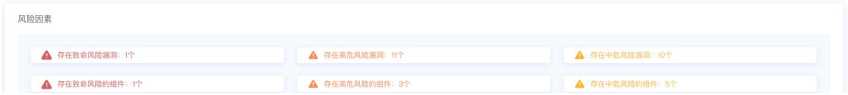
## 04 程序上架阶段

支持Java、Python等二进制程序包的分析，能够在上架之前发现程序潜在的安全风险。

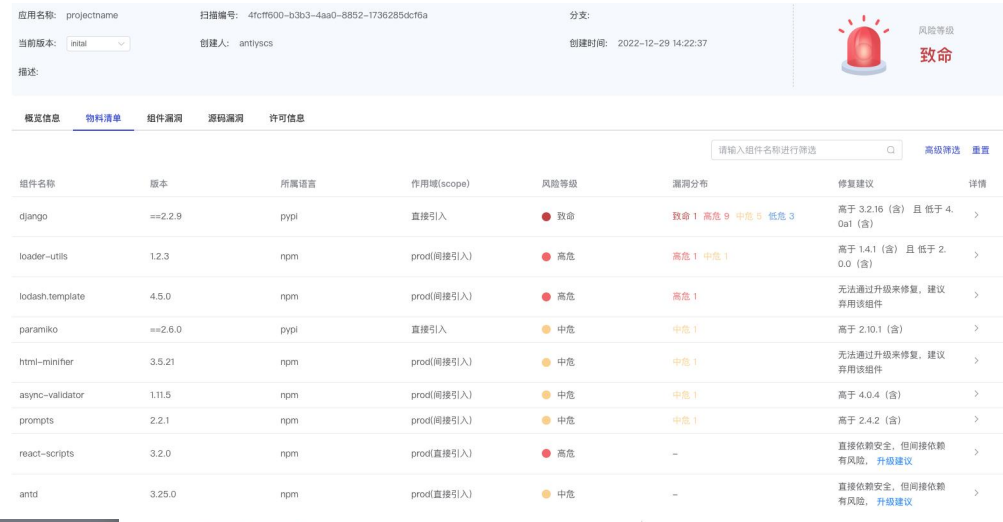
# 04 安天代码安全检测系统

# 安天代码安全检测系统（AntiySCS）











# 许可信息显示

安天代码安全检测系统

antiyscs

首页 > 详情

## 基本信息

应用名称: securibench-micro.jar　　扫描编号: 3c7b0b3b-d8a9-4823-a69c-929aff141f28　　分支:

当前版本: inital　　创建人: antiyscs　　创建时间: 2022-12-28 10:54:34

描述:

风险等级

高危

概览信息　物料清单　组件漏洞　源码漏洞　许可信息

| Full name | Identifier | FSF Free/Libre? | OSI Approved? |
|---|---|---|---|
| BSD Zero Clause License | 0BSD | | Y |
| Attribution Assurance License | AAL | | Y |
| Abstyles License | Abstyles | | |
| Adobe Systems Incorporated Source Code License Agreement | Adobe-2006 | | |
| Adobe Glyph List License | Adobe-Glyph | | |

# 源码漏洞分析

概览信息 | 物料清单 | 组件漏洞 | 源码漏洞 | 许可信息

请输入一级、二级名称

| 致命 | 高 | 中 | 低 | 所有 |
|---|---|---|---|---|
| 4 | 7 | 0 | 155 | 166 |

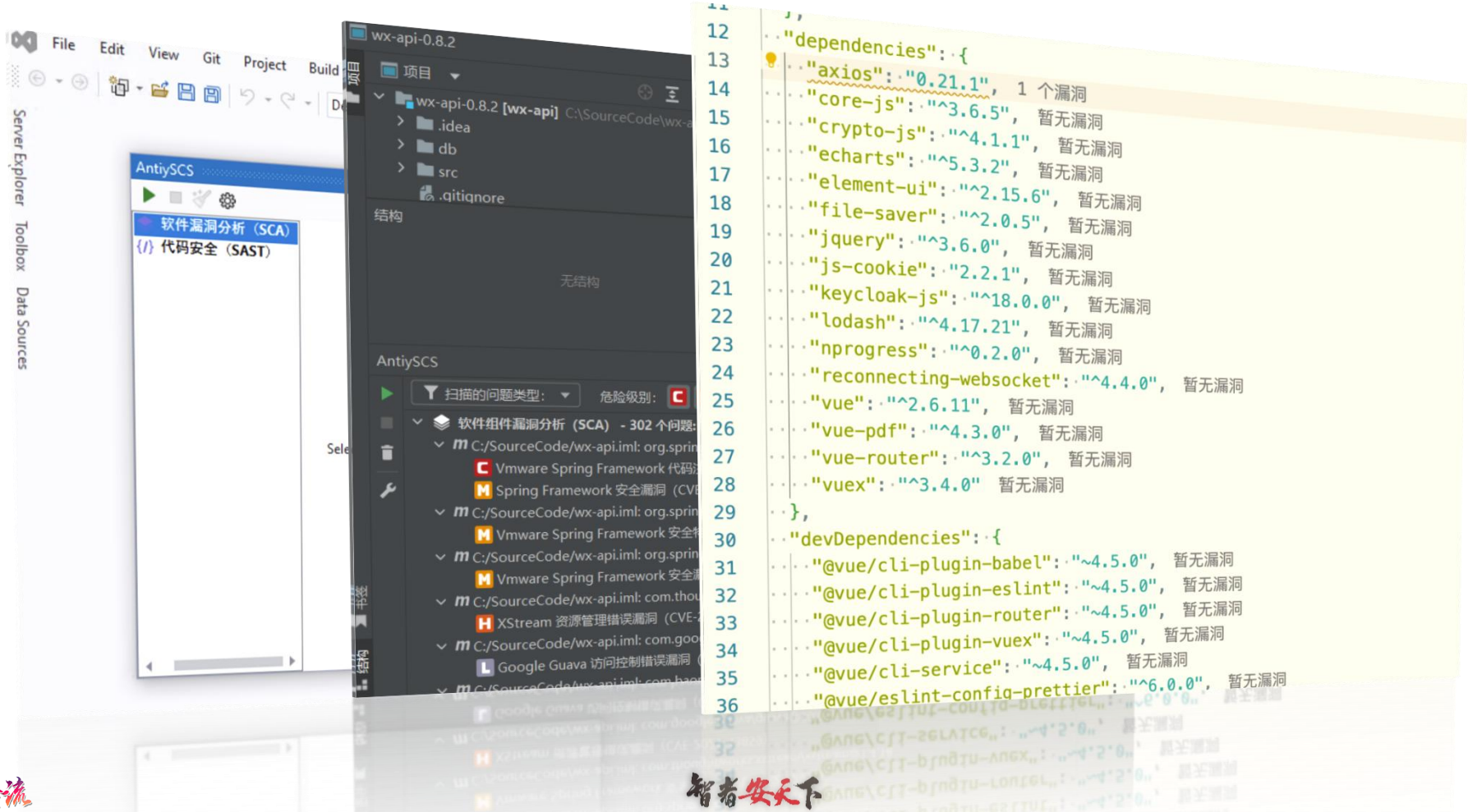
- 函数滥用（3）
  - 不安全的函数或模块（1）
    - 2.0.0/spug_api/apps/monitor/executors.py
  - 不安全的函数或模块（2）
- 代码执行（28）
  - XSS漏洞（2）
    - ug-2.0.0/spug_api/templates/web_ssh.html
    - ug-2.0.0/spug_api/templates/web_ssh.html
  - XSS漏洞（2）
  - XSS漏洞（1）
  - 命令注入（2）

executors.py × | web_ssh.html × | utils.py ×

```
193         self.rds.rpush(self.token, json.dumps({'key': key, 'status': 'info', 'data': message}))
194
195     def send_error(self, key, message):
196         message = '\r\n' + message
197         self.rds.rpush(self.token, json.dumps({'key': key, 'status': 'error', 'data': message}))
198         raise Exception(message)
199
200     def send_step(self, key, step, data):
201         self.rds.rpush(self.token, json.dumps({'key': key, 'step': step, 'data': data}))
202
203     def local(self, command, env=None):
204         command = 'set -e\n' + command
205             task = subprocess.Popen(command, env=env, shell=True, stdout=subprocess.PIPE, stderr=subprocess.STDOUT)
206
207         while True:
208             message = task.stdout.readline()
209             if not message:
210                 break
211             self.send_info('local', message.decode())
212         if task.wait() != 0:
213             self.send_error('local', f'exit code: {task.returncode}')
```

跟踪路径表 | 跟踪路径图 | 详细信息 | 修复建议

# 开发工具集成


File Edit View Git Project Build
Server Explorer Toolbox Data Sources
AntiySCS
软件漏洞分析（SCA）
代码安全（SAST）
wx-api-0.8.2
项目
wx-api-0.8.2 [wx-api] C:\SourceCode\wx-a
.idea
db
src
.gitignore
结构
无结构
AntiySCS
扫描的问题类型:
危险级别:
软件组件漏洞分析（SCA） - 302 个问题:
C:/SourceCode/wx-api.iml: org.sprin
Vmware Spring Framework 代码
Spring Framework 安全漏洞（CV
C:/SourceCode/wx-api.iml: org.sprin
Vmware Spring Framework 安全
C:/SourceCode/wx-api.iml: org.sprin
Vmware Spring Framework 安全
C:/SourceCode/wx-api.iml: com.thou
XStream 资源管理错误漏洞（CVE-
C:/SourceCode/wx-api.iml: com.goo
Google Guava 访问控制错误漏洞
12 "dependencies": {
13 "axios": "0.21.1", 1 个漏洞
14 "core-js": "^3.6.5", 暂无漏洞
15 "crypto-js": "^4.1.1", 暂无漏洞
16 "echarts": "^5.3.2", 暂无漏洞
17 "element-ui": "^2.15.6", 暂无漏洞
18 "file-saver": "^2.0.5", 暂无漏洞
19 "jquery": "^3.6.0", 暂无漏洞
20 "js-cookie": "2.2.1", 暂无漏洞
21 "keycloak-js": "^18.0.0", 暂无漏洞
22 "lodash": "^4.17.21", 暂无漏洞
23 "nprogress": "^0.2.0", 暂无漏洞
24 "reconnecting-websocket": "^4.4.0", 暂无漏洞
25 "vue": "^2.6.11", 暂无漏洞
26 "vue-pdf": "^4.3.0", 暂无漏洞
27 "vue-router": "^3.2.0", 暂无漏洞
28 "vuex": "^3.4.0" 暂无漏洞
29 },
30 "devDependencies": {
31 "@vue/cli-plugin-babel": "~4.5.0", 暂无漏洞
32 "@vue/cli-plugin-eslint": "~4.5.0", 暂无漏洞
33 "@vue/cli-plugin-router": "~4.5.0", 暂无漏洞
34 "@vue/cli-plugin-vuex": "~4.5.0", 暂无漏洞
35 "@vue/cli-service": "~4.5.0", 暂无漏洞
36 "@vue/eslint-config-prettier": "^6.0.0", 暂无漏洞

# 代码编写中实时检测

```
3   class AppApi:
4       def __init__(self, url):
5           self.url = url
6
7       def login(self, username, password):
8           data = {
9               "username": username,
10              "password": password
11          }
12
```

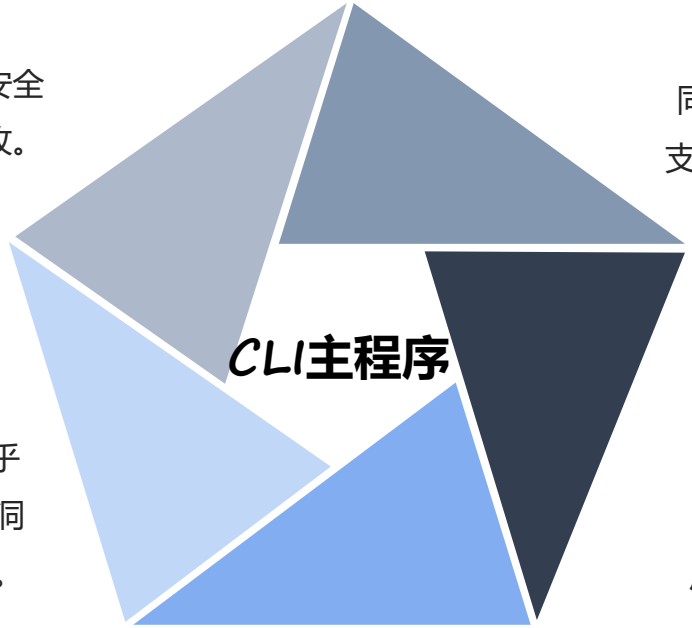
# 安天代码安全检测系统（AntiySCS）

## 一键组件升级

通过命令可对源码中可升级的不安全组件一键升级，省去用户手动修改。目前支持maven和npm。

## 多平台支持

同时支持Windows、Linux和macOS，支持x86-64和aarch64两种处理器架构。




## CUI界面输出

内置的CUI支持输出与网页报告几乎同等粒度的组件分析报告和代码漏洞报告，适用于命令行模式下的开发。

## 配置项管理

通过CLI，用户可以创建扫描规则模板，添加漏洞忽略等规则，允许用户根据实际情况灵活定制扫描配置。

## LSP语言插件

提供LSP接口插件，方便与各种支持LSP的IDE或者文本编辑器接入。

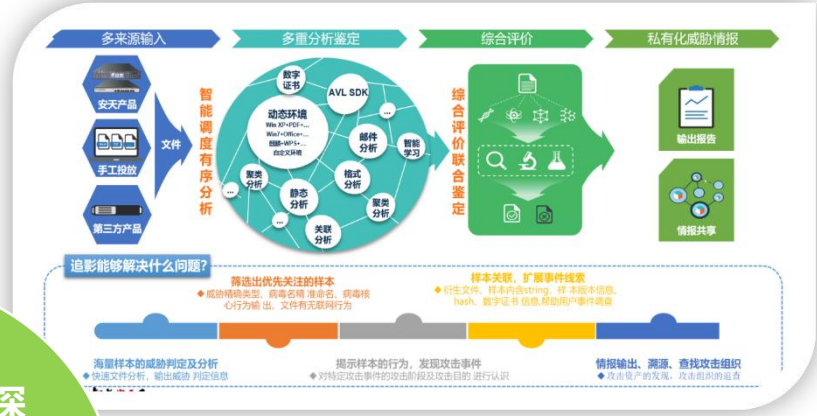
注：该功能于2023年春季正式推出

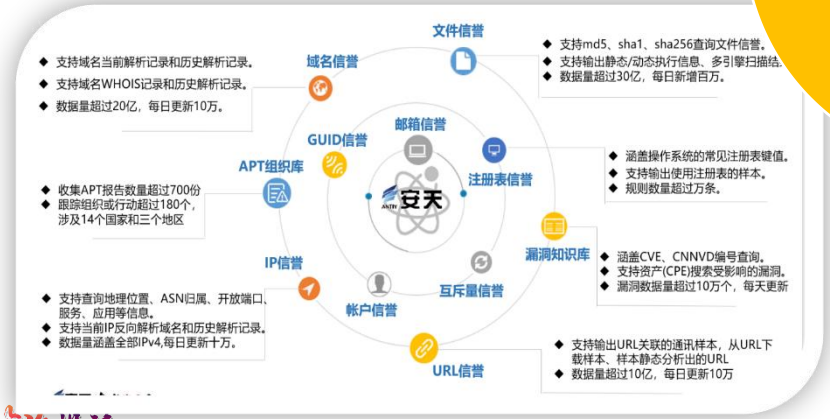
# AntiySCS优势能力金字塔

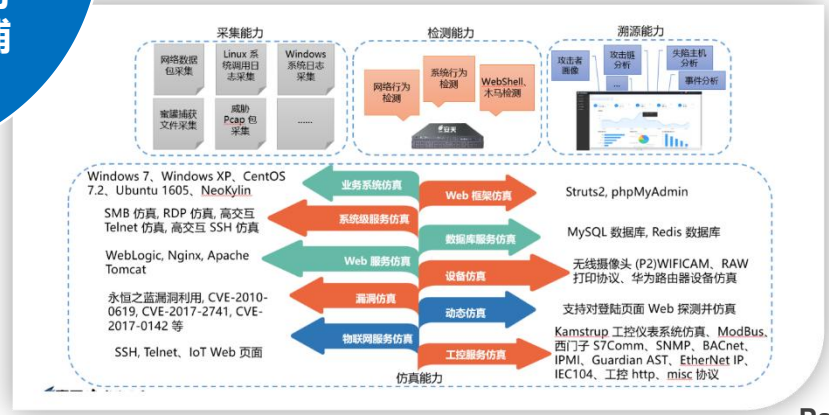

- **贴合业务** 一站式解决前期开发流程的代码安全问题，与开发流程深度结合
- **高可用性** 内置多套扫描引擎，多种分析方法，API、IDE、CI/CD丰富的业务接入部署方式
- **广覆盖面** 内置精准高效的扫描算法以及十余种分析鉴定器，支持扫描C/C++、Java、Golang、JavaScript、Node、Python、PHP等20余种主流语言的源代码工程

# 安天产品生态联动


全要素采集
300+种常见格式检测
数据按需采集，提升威胁追溯能力
支持ATT&CK / NSACSS两大威胁框架，呈现完整攻击链
检测结果标签化
检测规则开放式拓展
基于资产、事件进行关联
采集能力
识别能力
处理能力
检测能力
业务流量监测
零拷贝
流式协议解析
探海自带检测规则
文件检测规则
C2检测规则
网络行为检测规则
流量检测规则
导入/新建流量检测规则
新建自定义检测规则
安天 智者安天下


多来源输入
多重分析鉴定
综合评价
私有化威胁情报
安天产品
手工投放
第三方产品
文件
智能调度有序分析
动态环境
AVL SDK
静态分析
关联分析
综合评价联合鉴定
输出报告
情报共享
追影能够解决什么问题?
筛选出优先关注的样本
样本关联，扩展事件线索
海量样本的威胁判定及分析
揭示样本的行为，发现攻击事件
情报输出、溯源，查找攻击组织


文件信誉
域名信誉
GUID信誉
邮箱信誉
APT组织库
注册表信誉
漏洞知识库
IP信誉
互斥量信誉
帐户信誉
URL信誉
支持域名当前解析记录和历史解析记录。
支持域名WHOIS记录和历史解析记录。
数据量超过20亿，每日更新10万。
支持md5、sha1、sha256查询文件信誉。
支持输出静态/动态执行信息、多引擎扫描结...
数据量超过30亿，每日新增百万。
收集APT报告数量超过700份
跟踪组织或行动超过180个，涉及14个国家和三个地区
涵盖操作系统的常见注册表键值。
支持输出使用注册表的样本。
规则数量超过万条。
涵盖CVE、CNNVD编号查询。
支持资产(CPE)搜索受影响的漏洞。
漏洞数据量超过10万个，每天更新
支持查询地理位置、ASN归属、开放端口、服务、应用等信息。
支持当前IP反向解析域名和历史解析记录。
数据量涵盖全部IPv4,每日更新十万。
支持输出URL关联的通讯样本，从URL下载样本、样本静态分析出的URL
数据量超过10亿，每日更新10万


采集能力
检测能力
溯源能力
网络数据包采集
Linux 系统调用日志采集
Windows 系统日志采集
蜜罐捕获文件采集
威胁Pcap 包采集
网络行为检测
系统行为检测
WebShell、木马检测
攻击者画像
攻击链分析
失陷主机分析
事件分析
Windows 7、Windows XP、CentOS 7.2、Ubuntu 1605、NeoKylin
SMB 仿真, RDP 仿真, 高交互 Telnet 仿真, 高交互 SSH 仿真
WebLogic, Nginx, Apache Tomcat
永恒之蓝漏洞利用, CVE-2010-0619, CVE-2017-2741, CVE-2017-0142 等
SSH, Telnet、IoT Web 页面
业务系统仿真
系统级服务仿真
Web 服务仿真
漏洞仿真
物联网服务仿真
Web 框架仿真
数据库服务仿真
设备仿真
动态仿真
工控服务仿真
Struts2, phpMyAdmin
MySQL 数据库, Redis 数据库
无线摄像头 (P2)WIFICAM、RAW 打印协议、华为路由器设备仿真
支持对登陆页面 Web 探测并仿真
Kamstrup 工控仪表系统仿真、ModBus、西门子 S7Comm、SNMP、BACnet、IPMI、Guardian AST、EtherNet IP、IEC104、工控 http、misc 协议
仿真能力

# 感谢大家的关注

安天冬训营 wtc.antiy.cn