安全玻璃盒

让安全更简单更智能

交互式应用安全检测系统

产品手册

杭州孝道科技有限公司

目录

1 系统登录

输入登录页面地址，打开登录页面，输入管理员分配的用户名及密码，即可登录成功，进入系统主页。如果忘记密码，可通过忘记密码功能设置新密码（忘记密码功能需要管理员在系统管理-系统配置-邮件设置中配置邮件服务器且需要用户账号绑定有效的邮箱）

2 添加 SecPoint

点击添加 SecPoint，进入 agent 安装引导界面，可根据提示进行 agent 的下载与安装。

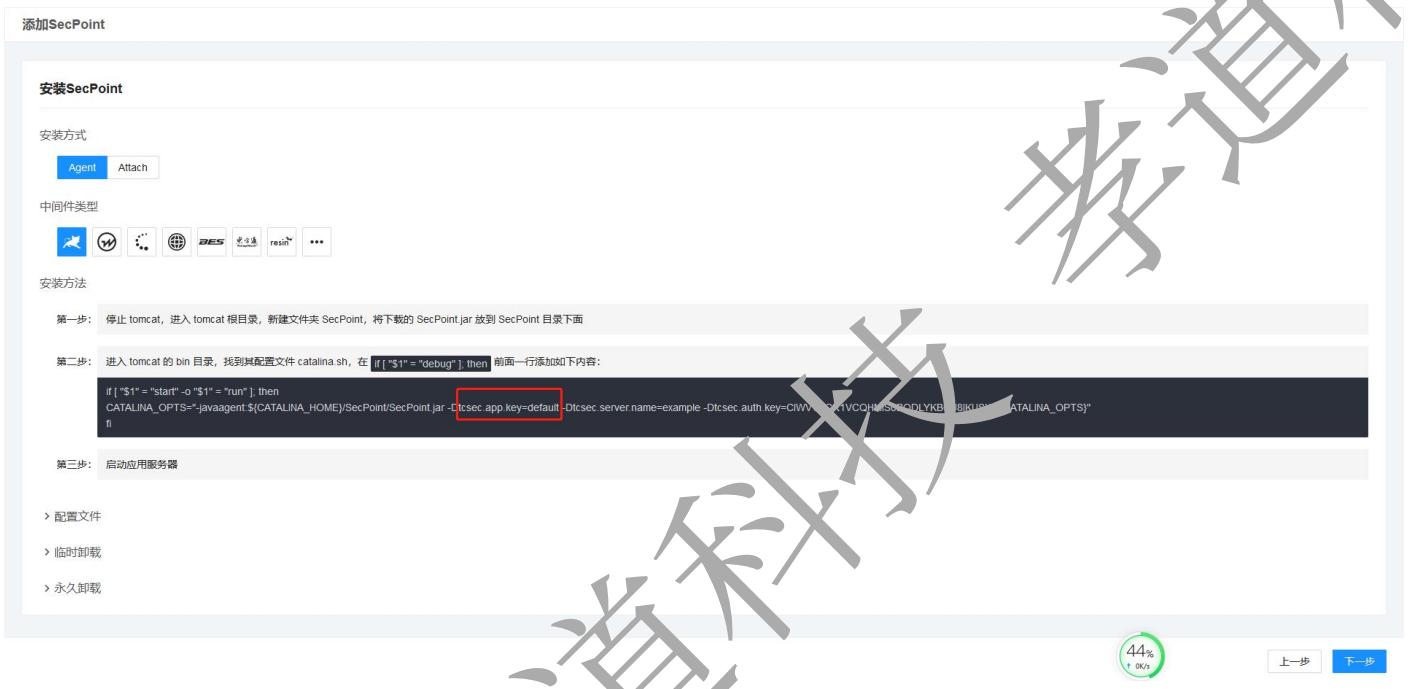
IAST 平台提供了 Java、PHP、.NET、PYTHON 四种 SecPoint 类型，您可以根据自己的应用选择相应的 SecPoint 及操作系统环境进行下载。

创建应用的时候需要输入应用的唯一标识 key，SecPoint 通过 key 与应用进行绑定。SecPoint 启动时需要在启动脚本中添加字段 -Dtcsec.app.key = 应用 key（在下载 SecPoint 页面选择需要绑定的应用，此处的应用 key 会相应变化）

参数说明如下：

参数名	是否必选	描述
-Dtcsec.cof.path	否	指定 agent 的配置文件位置
-Dtcsec.app.key	是	应用 key，用来指定绑定的应用
-Dtcsec.authorization	否	默认关闭，开启之后下载的

		agent 将不能跨组绑定，只能绑定下载时候选定小组的应用
-Dtcsec.data.version	否	指定 agent 版本
-Dtcsec.server.name	是	指定服务名称



3 主页

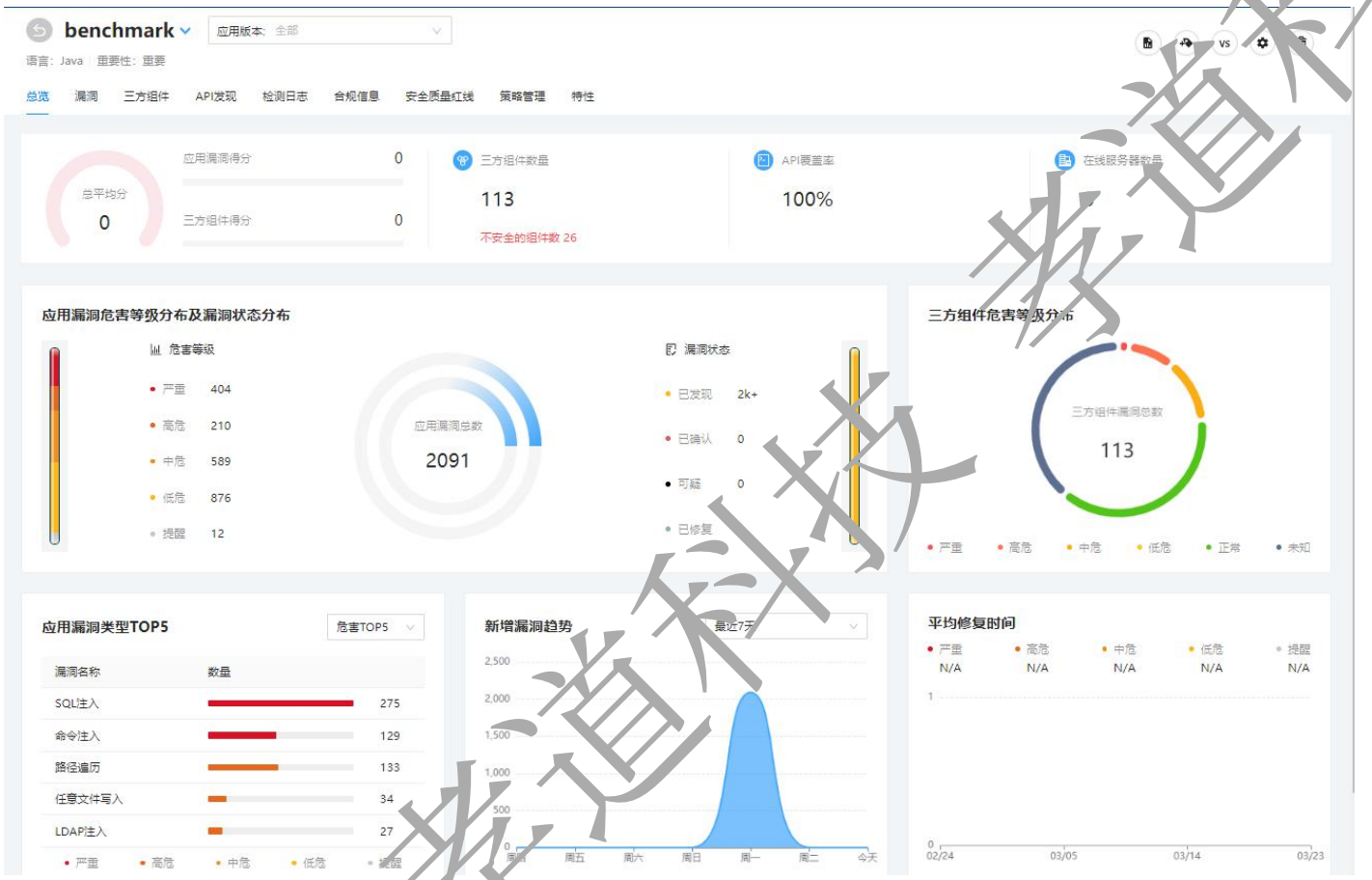
IAST 主页是对所有应用漏洞数据的统计，在主页我们可以查看到所有应用的漏洞情况、不安全的组件漏洞情况以及新增漏洞趋势、平均修复时间等，通过主页的数据展示，我们可以快速直观地了解到应用存在的风险及相关开发人员对漏洞的修复情况。

4 应用

4.1 总览

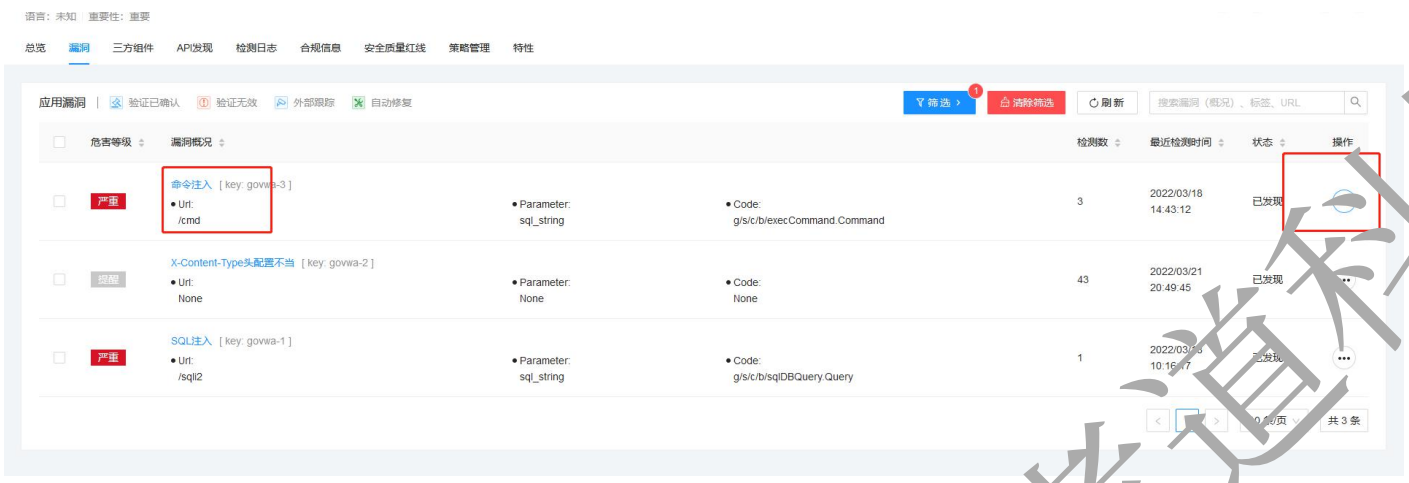
进入应用首先是应用列表页面，展示当前用户下的所有应用，以及应用的基本信息。点击具体的应用名称进入应用详情.

IAST 总览页面是对当前应用数据的统计，在这里您可以查看到当前应用基本信息、漏洞危害等级及状态分布情况、新增漏洞趋势和平均修复时间。通过总览页面，您可以快速直观地了解到当前应用所存在的风险及相关开发人员对漏洞的修复情况。



4.2 漏洞

漏洞列表展示当前应用下被检测出来的漏洞以及漏洞的基本信息，并能对漏洞做相应的操作，点击具体的漏洞名称，进入漏洞详情。



如下图所示，漏洞标签页展示了有关该漏洞的所有信息，包括属性、状态、检测记录、漏洞详情、漏洞风险、合规信息、修复建议等。

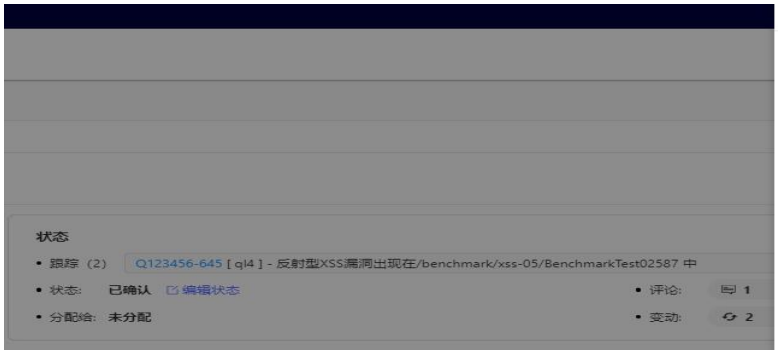
4.2.1 属性

属性栏是对该漏洞的一些基本信息的展示，包括漏洞的所属应用及其最近应用版本、漏洞 KEY、检测次数、暴露天数、code（该漏洞触发的具体代码位置），您可以从该页面对这条漏洞的基本属性

有大致了解。

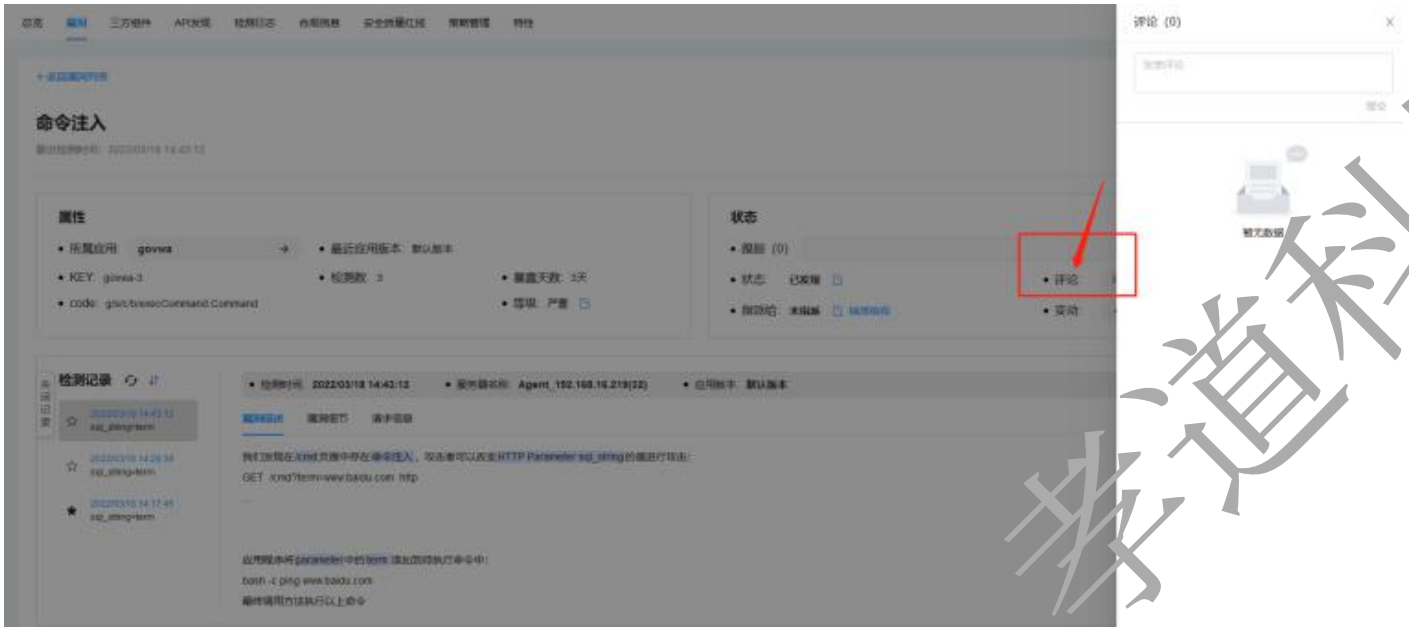
4.2.2 状态

在状态栏中您可以为该漏洞创建 JIRA 以及禅道问题，我们会对该漏洞创建的所有问题进行跟踪。

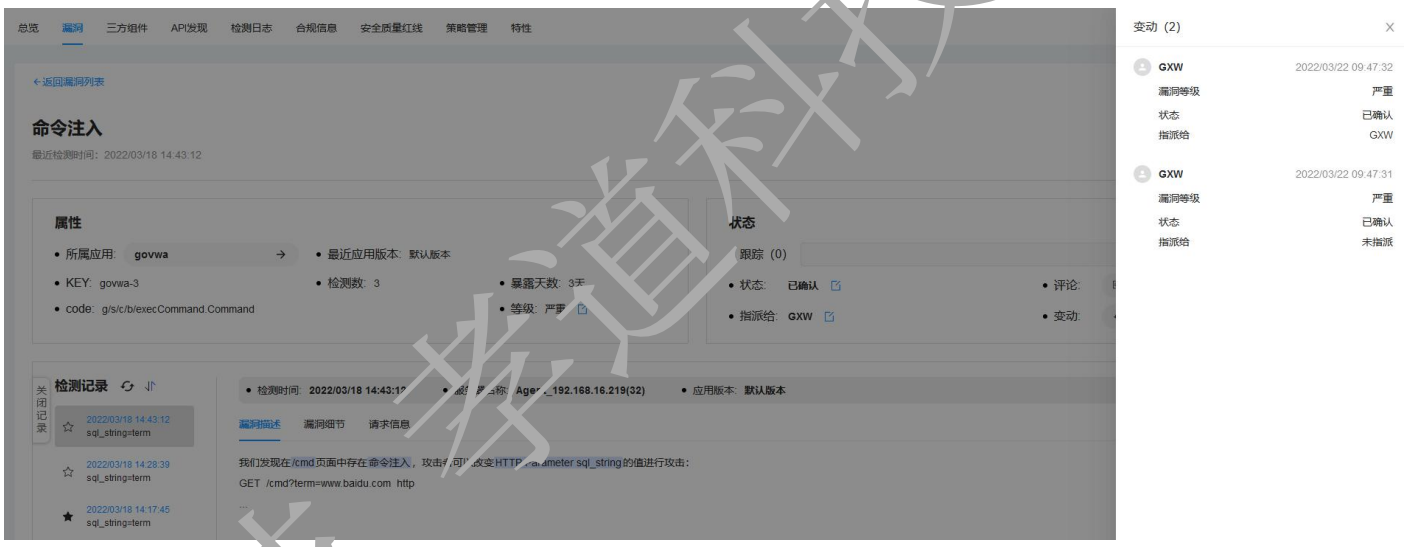


点击编辑状态，您可以对漏洞状态进行修改，且可以对当前状态进行备注。您可以将该漏洞分配给其他用户，方便对漏洞进行及时有效的处理。

点击评论栏，您可以看到所有用户对该漏洞留下的评论。



同时，在状态栏中也会展示目前该漏洞被分配的负责人，以及所有用户对该漏洞进行的状态变更操作。



4.2.3 检测记录

该栏中记录了该漏洞的所有检测记录，包括不同应用版本的漏洞记录。检测记录超过 50 条，将会自动删除多余的记录。对于第一次上报的记录和经过了主动验证的记录，会默认标星处理。标星的记录将不会被自动删除，且不计入自动删除的统计数量中。

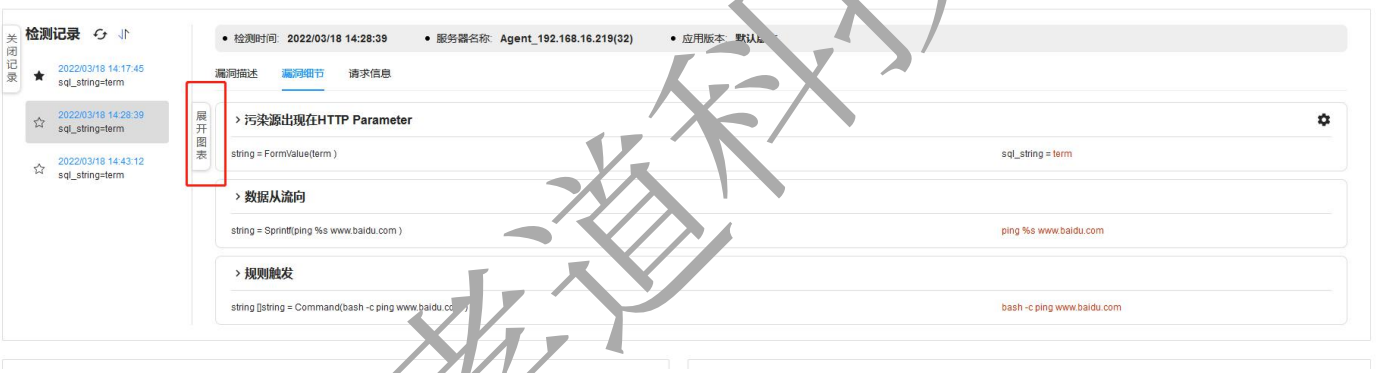
4.2.4 漏洞详情

漏洞详情中主要展示该条漏洞记录的详细信息，包括漏洞检测时间、所属服务器名称、应用版本

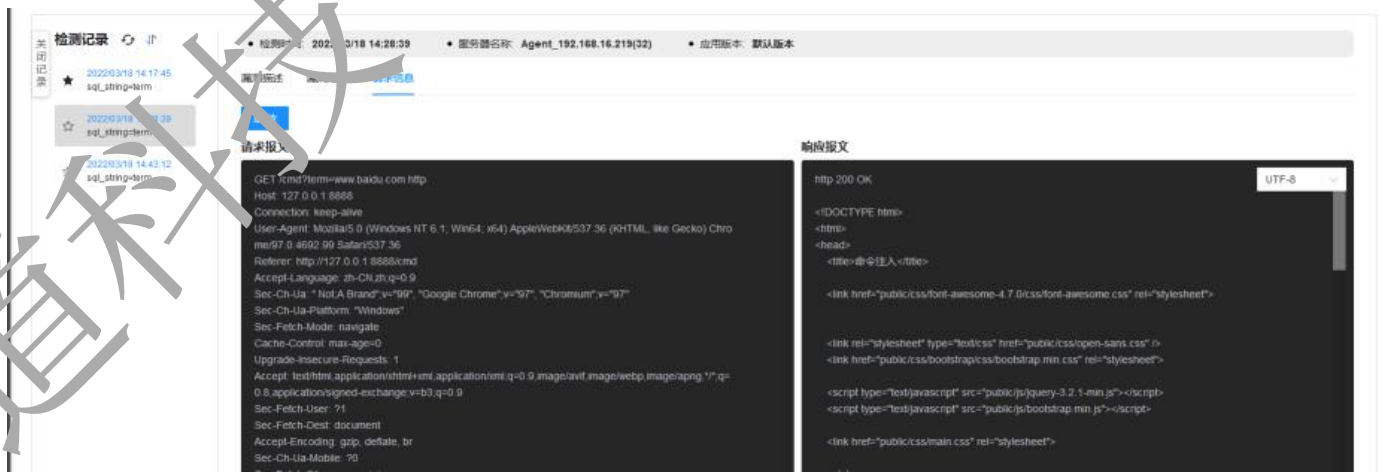
以及漏洞信息。漏洞信息主要包括漏洞描述、漏洞细节、请求信息，如果该条检测记录进行了主动验证，还将展示验证信息。

在漏洞描述中描述了漏洞的形成原因，您可以快速看到漏洞存在的具体位置。

在漏洞细节中会按代码的执行顺序，将所有污染源、传播途径、规则触发展示在页面中。通过来源参数，我们可以追溯到漏洞出现的位置，之后分析传播途径，我们可以进一步了解到这些数据在程序内部是如何传递的，最终在规则触发阶段中，我们可以查看到污染数据被使用的具体操作。此外，您也可以展开图表，比较直观地查看漏洞的传播过程。



请求信息标签页展示了请求信息及响应信息。您可以通过编辑请求信息，之后点击重放按钮来重放该请求来验证漏洞是否存在。

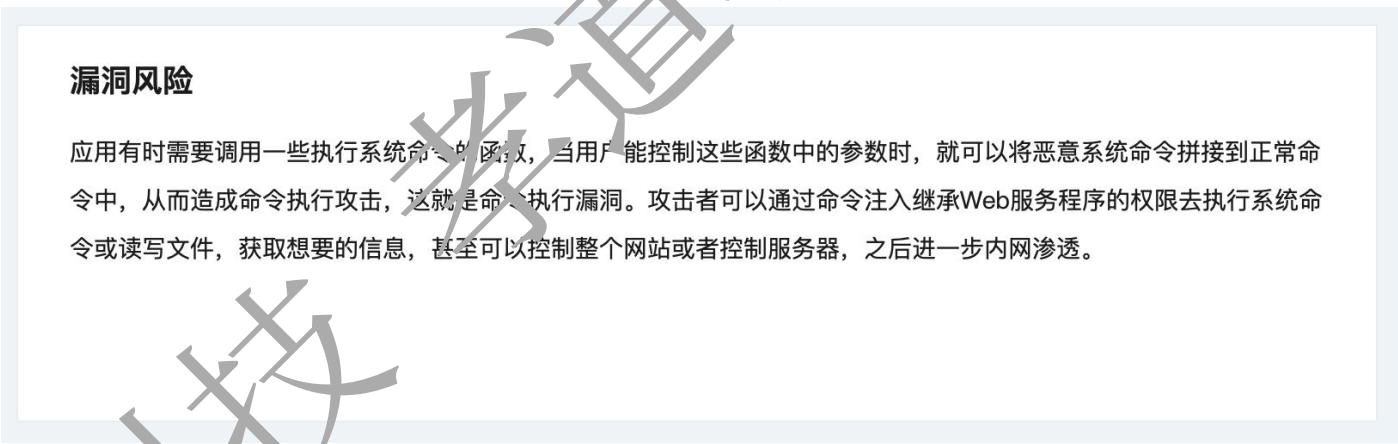


验证信息页面会将主动验证的结果展示出来，在验证信息中您可以查看到替换攻击载荷后重发的

请求信息，同时我们会给出该验证结果的验证依据。

4.2.5 漏洞风险

漏洞风险一栏主要描述了漏洞的形成原因及由该漏洞可能导致的风险。



4.2.6 合规信息

合规信息收录了 OWASP、CWE/SANS、PCI-DSS 等开放式 Web 应用安全项目的 10 项最严重的 Web 应用程序安全风险类别，这里会将该漏洞符合的各安全风险类别展示出来。

合规信息

• OWASP Top 10 2013	A1 – Injection
• PCI-DSS v3.2.1	6.5.1 – Injection Flaws
• GDPR	Injection
• OWASP Top 10 2017	A1 – Injection
• CWE/SANS 2011	CWE-77: Improper Neutralization of Special Elements used in a Command ('C...

4.2.7 修复建议

在修复建议中，针对各种不同类型的漏洞，我们给出了比较专业的修复建议，并且提供了安全代码以及不安全代码示例以供参考，这样即使开发人员对某些漏洞不太了解，也能够通过参考示例给出比较合理的修复方案。

4.3 三方组件

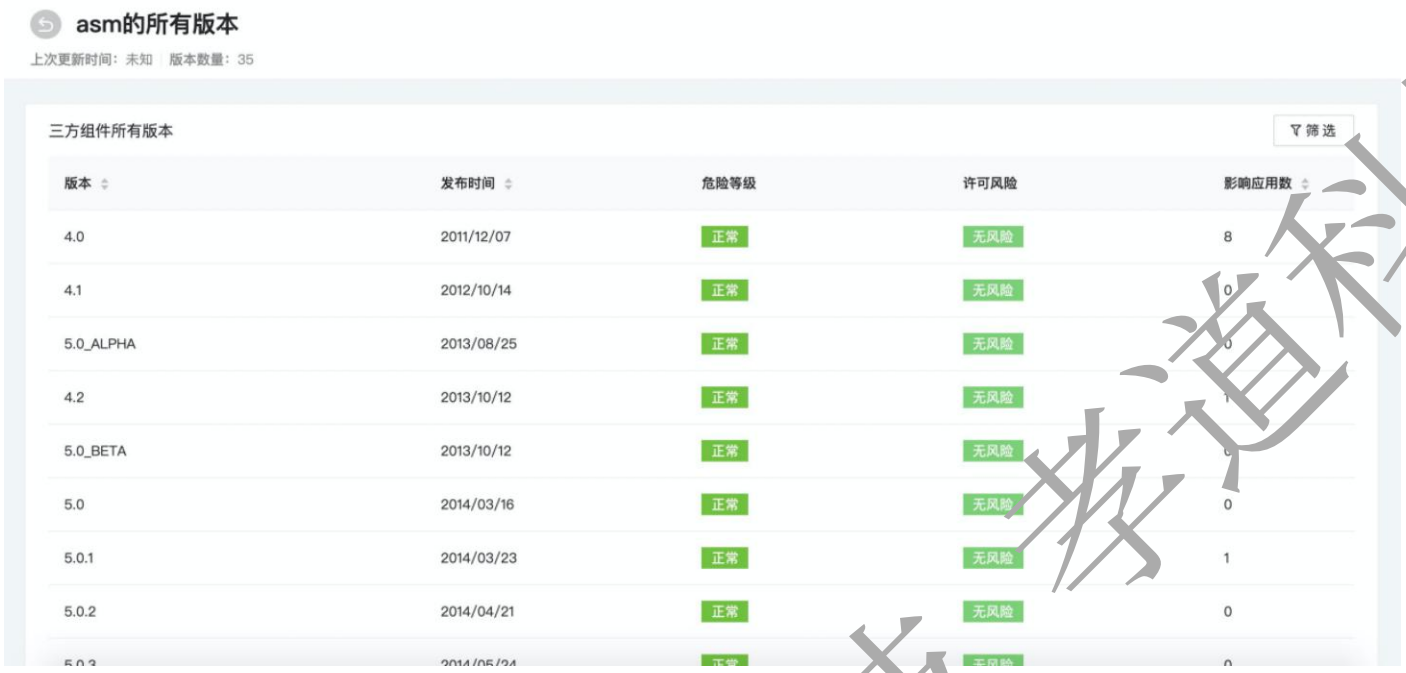
三方组件页面展示出了三方组件危害等级分布比例、组件使用版本分布以及组件许可风险等级分布比例等。

您可以按危害等级、所属服务器、许可风险筛选查看三方组件，对于风险等级较高的组件，可替换成安全版本使用。

危险等级中的未知状态是说明该 jar 包是用户自定义的 jar 包，或者三方组件库中匹配不到的 jar 包。

点击三方组件名称，可进入三方组件详情页面，查看该组件的详细漏洞信息。目前进行检测的漏洞标准为 CVE 标准、CNNVD 标准、自定义标准。

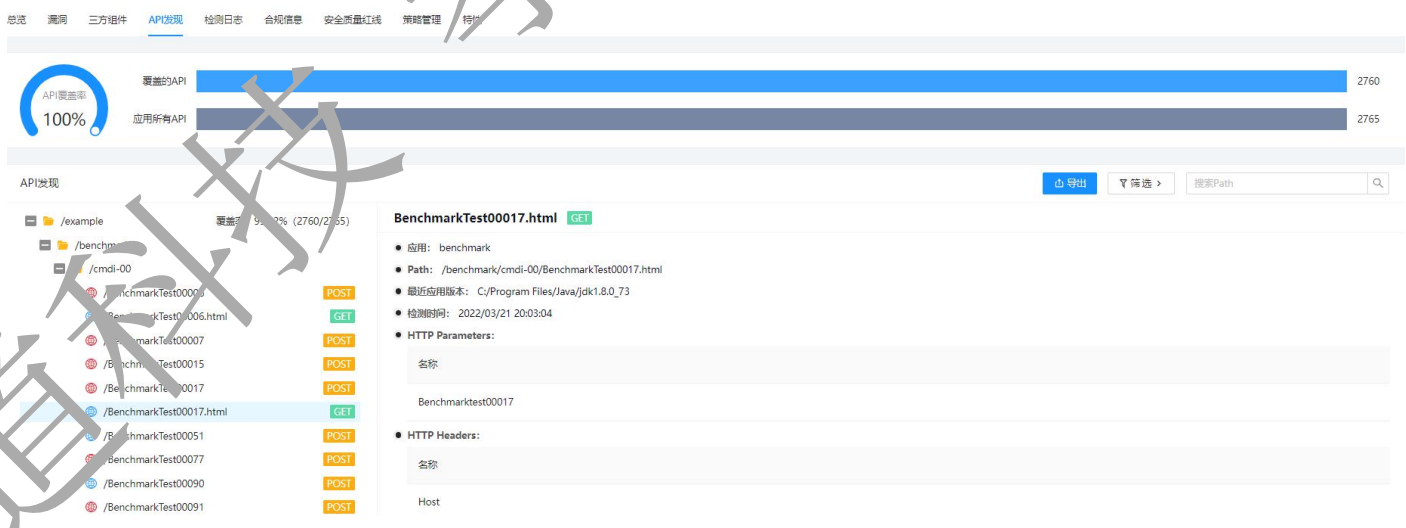
点击所有版本可进入所有版本详情页面，可以查看当前组件哪个版本是安全的，选择安全版本使用。



4.4API 发现

在 API 发现功能中，SecPoint 会根据加载到了 JVM 中的代码扫描出应用所有的 API，并且会实时监控并记录 HTTP/HTTPS 请求的访问日志，将 API 根据层级，以树形结构显示，树形结构点击图标可展开、收起。您也可以根据漏洞情况、检测情况、搜索框来筛选对应的 API。

如下图所示，API 发现展示出了 API 的具体信息当前 API 产生漏洞的数量，点击漏洞的数量，可以跳转到该应用的漏洞列表中并筛选展示具体的漏洞。也可以将发现的 API 导出至本地进行相关操作



鼠标悬浮到 path 后，会显示排除规则快捷配置图标，设置完成之后，系统将不再检测该 path 下

的相应漏洞。

4.5 检测日志

检测日志功能默认关闭，使用之前需要在应用特性中开启流量检测开关。

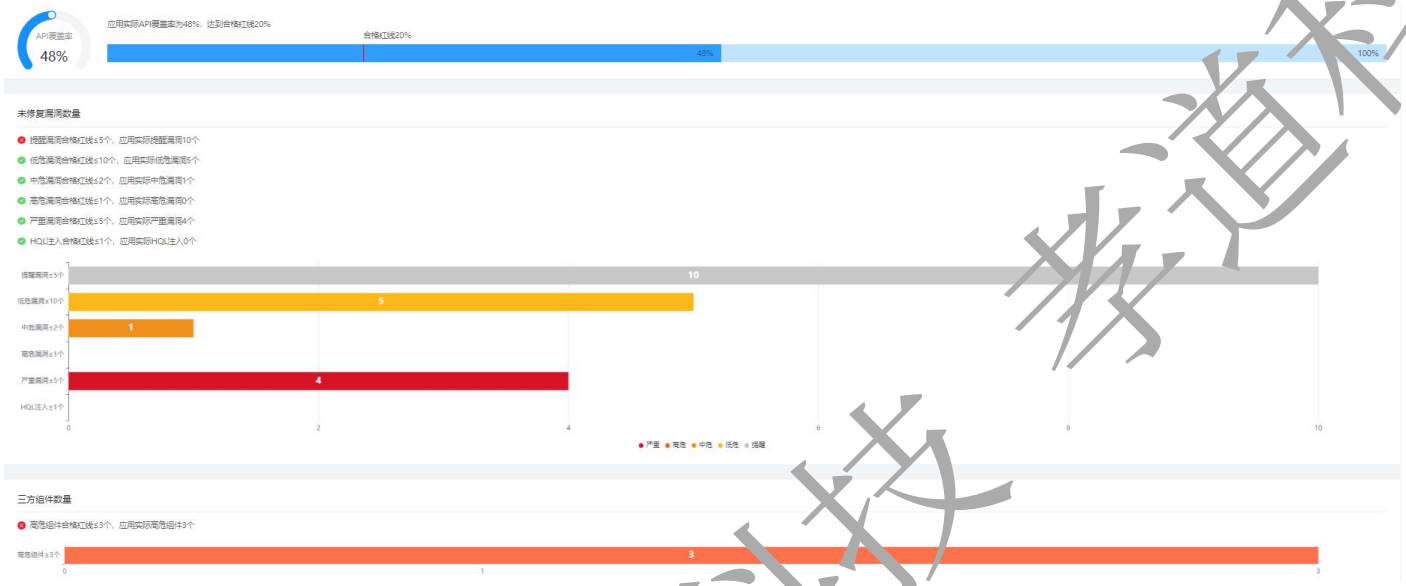
检测日志是访问被测应用时的 URL 记录，会清晰记录出该 URL 下出现的漏洞数量及漏洞类型。您可以根据漏洞类型、测试时间、搜索框来筛选对应的检测日志，也可以勾选日志导出统计表格。

4.6 合规信息

合规信息收录了 OWASP、CWE、SANS、PCI-DSS 等开放式 Web 应用安全项目的 10 项最严重的 Web 应用程序安全风险类别，astp 会将上报漏洞中符合各安全风险类别的统计并展示出来。

4.7 安全质量红线

安全质量红线页面展示应用质量达标状况，可以直观看到 API 覆盖率，未修复漏洞的数量情况以及三方组件的数量情况，所比较标准为引用的安全质量红线模板.



4.8 策略管理

4.8.1 检测规则

某一检测规则开启之后，系统就会对该种规则类型的漏洞进行检测，并将检测到的漏洞展示在漏洞列表中。

管理员设置的检测规则影响小组的默认检测规则，小组的默认检测规则影响应用的默认检测规则（受时间线影响）。

例如系统一共能检测 49 种漏洞，一开始管理员在策略管理处设置开启 43 种规则。之后管理员建立了 3 个小组，则这 3 个小组默认开启 43 种检测规则。小组 1 的小组管理员此时点击策略管理，看到的情况是开启 43 种检测规则。此时小组 1 创建了 3 个应用，则这 3 个应用默认开启 43 种检测规则。如果此时，小组 1 的小组管理员修改了检测规则为 46 种，并且又创建了 2 个应用。则原有的 3 个应用的默认检测规则不变，之后添加的 2 个应用的默认检测规则为 46 种。

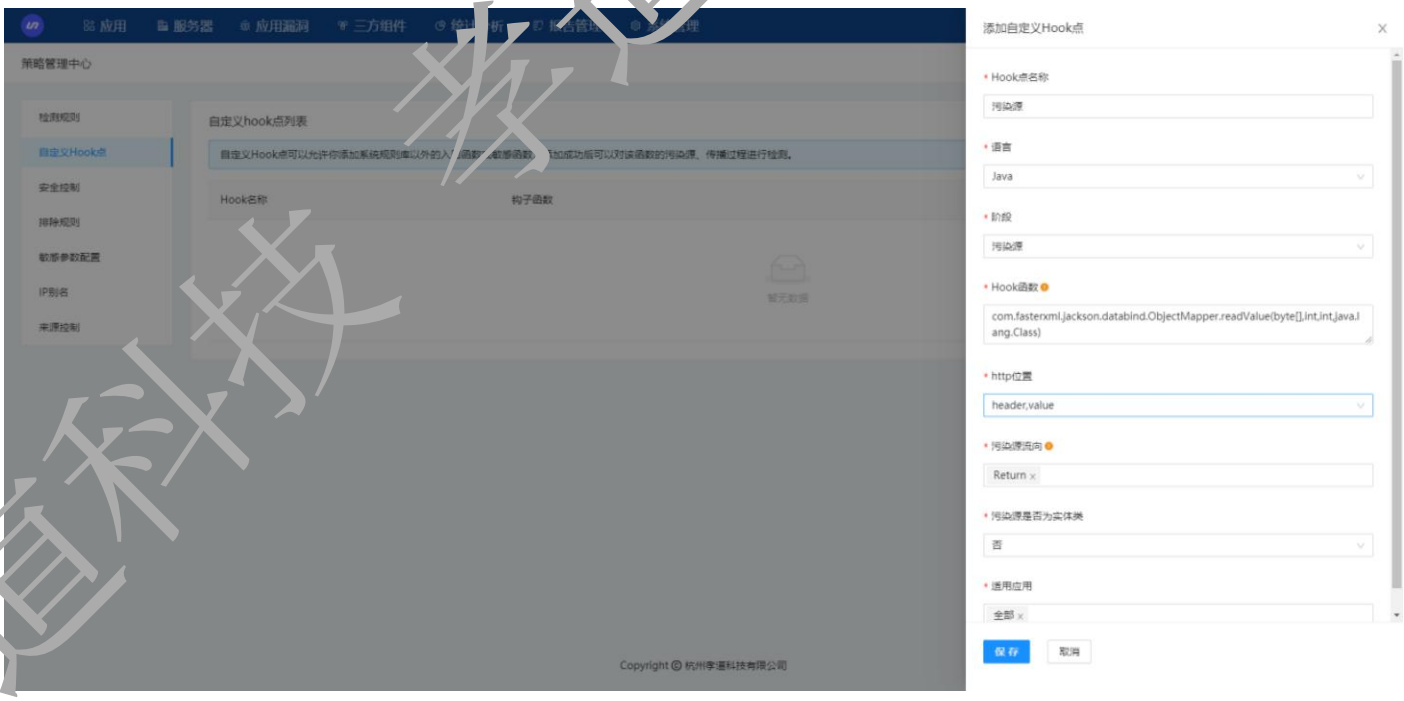
4.8.2 自定义 Hook 点

自定义 Hook 点主要为了增加对应用场景的适配性，在 IAST 自带的漏洞检测点缺失的情况下，可通过自定义 Hook 点进行不全，自定义 Hook 点可以在污染源、传播路径、规则触发三个阶段添加 hook 函数。

4.8.2.1 污染源阶段（propagate）

污染源阶段可以对指定的值进行标记，值可指定为函数的参数、返回值 以及 this 对象。用户可以根据自己开发的需要对值进行污染源标记的函数可通过此规则进行覆盖。

污染源阶段填写示例：



Hook 点函数：函数完全限定名

Http 位置：指污染源输入的位置

污染源流向：指定值的位置，支持 R、this、P1-P36、三种类型，可多选

Return - 返回值

this - this 对象

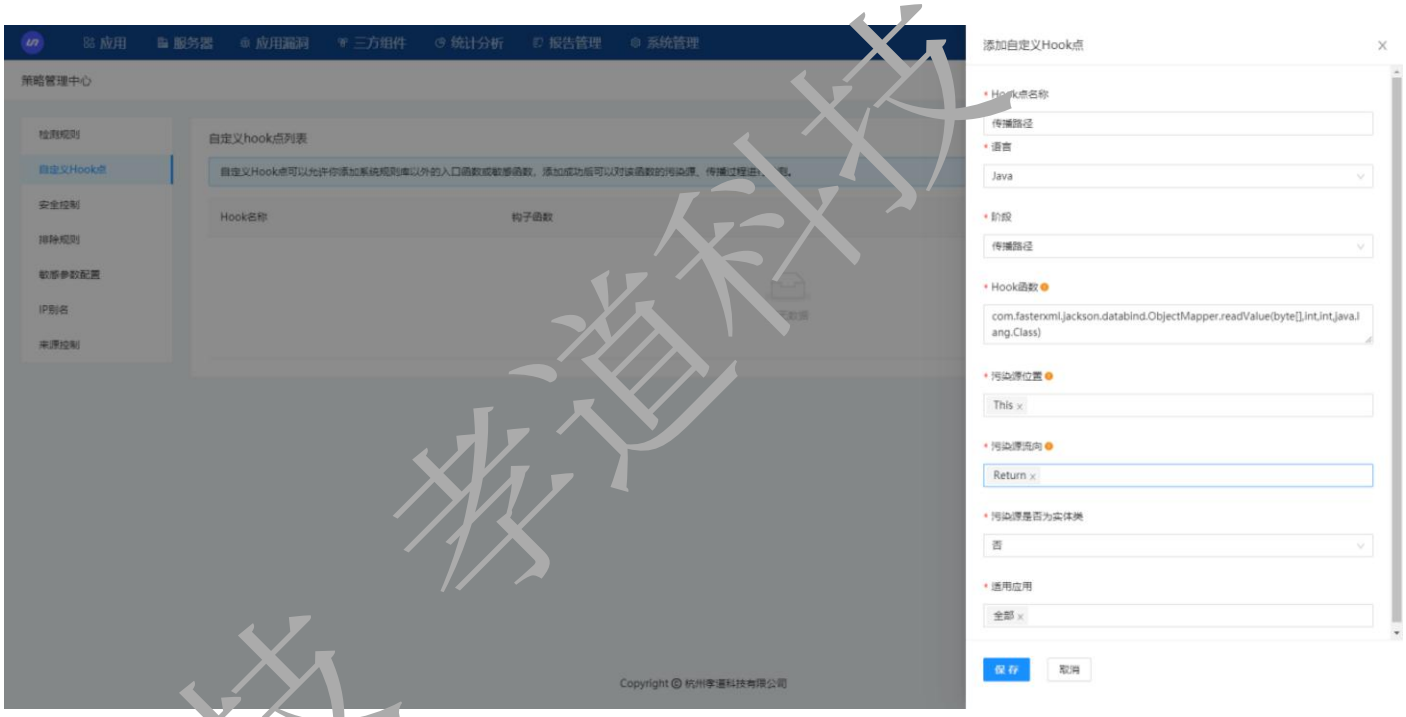
P1-P36 - 参数，P1 表示第一个参数

污染源是否为实体类：代表污染源是否能够被解析

4.8.2.2 传播路径阶段（propagate）

传播路径阶段（propagate）会完成值与值之间的污点标记传递，例如 String 对象 A 有污点标记，对 A 执行 append 方法产生的 B 对象也会有污点标记，此时 append 方法则是一个污点传播函数。

传播路径阶段填写示例：



Hook 点函数：函数完全限定名

污染源位置：指污染源输入的位置

污染源流向：指定值的位置，支持 R、this、P1-P36 三种类型，可多选

Return - 返回值

this - this 对象

P1-P36 - 参数，P1 表示第一个参数

污染源是否为实体类：代表污染源是否能够被解析

4.8.2.3 规则触发阶段（sink）

规则触发函数（sink）也叫敏感函数，如果含有污染源标记的参数在敏感函数被执行，则会上报漏洞，此时完成了一次完整的污点跟踪流程。以 sql 注入为例，getParameter 获得的值（http 参数）会进行污点标记，在经过污点传播，如果最后 sql executeQuery 执行数据库查询时参数也有污染源标记，则证明执行数据库查询的 sql 语句可被用户控制，此时会上报 sql 注入漏洞，该功能可以为自定义添加的漏洞规则添加规则触发点。

规则触发阶段填写示例：

Hook 点函数：函数完全限定名

触发位置：指定触发敏感函数的参数，支持 this、P1-P36、三种类型，可多选

this - this 对象

P1-P9 - 参数，P1 表示第一个参数

污染数据是否由用户控制：代表污染源是否为外部输入

污染数据不能造成危害的条件：填写正则表达式，满足改表达的污染源则表示经过了过滤，将不会触发漏洞

4.8.3 安全控制

安全控制是允许您添加某些被认为能够保证数据安全的方法。比如您使用 java.net.URLDecoder.decode 进行 SQL 注入的数据过滤，你可以添加 java.net.URLDecoder.decode(java.lang.String*, java.lang.String)，*表示参数位置，再次检测到 SQL 注入漏洞，会将其标记为“没问题，内部安全

控制”状态。

同时，您可以在漏洞细节中快捷增加安全控制，如下图所示：

4.8.4 排除规则

排除规则主要是用于对应用进行配置，配置哪些规则是不进行检测的。可以通过以下三种方式来添加排除规则 ：

1）排除类型为输入

当排除类型为 输入 的时候，输入类型的选项有 Parameter，Header，Query String，Body，Cookie。

当输入类型为 Parameter， Header 和 Cookie 时，需要填写输入参数，输入参数可以使用通配符*，如 aaa* 则表示会匹配 aaa*（如匹配 aaa234,aaahh 这样的参数）。

2）排除类型为 path

可以输入单个 path 或者多个 path，多个 path 需要用换行分开，如

/News/user

/News/add

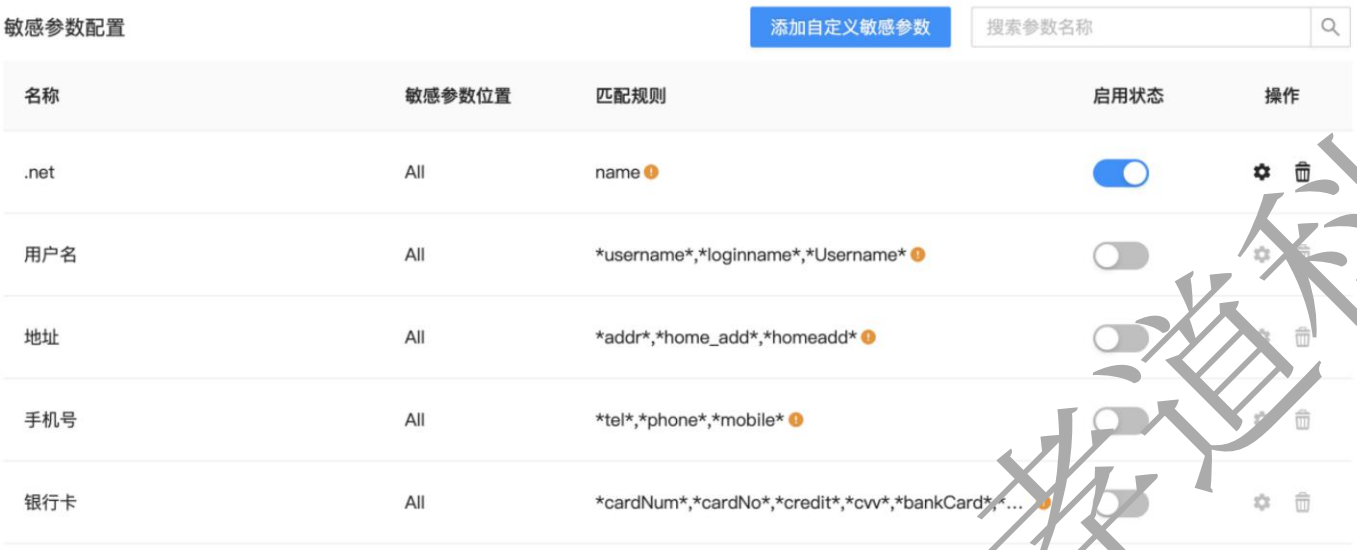
也可以使用*，如/News/*表示以下路径的漏洞都不进行检测。

3）排除类型为 package

排除类型为 package 时，你可以填入 com.tcsec.test，此时 IAST 和 RASP 将不会检测 com.tcsec.test 包下所有代码，此配置需重启应用才能生效。

4.8.5 敏感参数配置

通过自定义配置添加系统内置以外的敏感参数，以确定请求是否包含敏感数据并应检查是否存在相关漏洞，提升对敏感数据的保护。



敏感参数配置会影响以下两种漏洞类型的漏洞检测

4.8.6 熔断配置

熔断分为两种设置类型：

1、线程级熔断设置（针对某个监测点）

2、系统级熔断设置（整个 agent 检测服务）（单线程级和系统级的两个条件均为或的关系，两者达到其一就触发熔断）

4.8.7 自定义代码

自定义用户代码用于配置应用中代码的类型，方便系统区分用户代码和三方组件代码。

4.8.8.1 包含模式

使用包含模式匹配的代码将被视为用户代码。包含模式比排除模式具有更高的优先级，这意味着，如果在两个列表中都找到相同的模式，则匹配的代码将被视为您自己的代码。您可以使用 * 作为通配符。您应该指定包名，如：com.mycompany.myapp.*，org.thirdpartyorg.*。

4.8.8.2 排除模式

使用排除模式匹配的代码将被视为第三方代码。默认情况下系统会根据流行的第三方组件包来帮助识别第三方代码，您在此处添加的模式将与默认排除模式一起使用。您可以使用 * 作为通配符。您应该指定包名，如：com.mycompany.myapp.*，org.thirdpartyorg.*。

4.8.8IP 别名

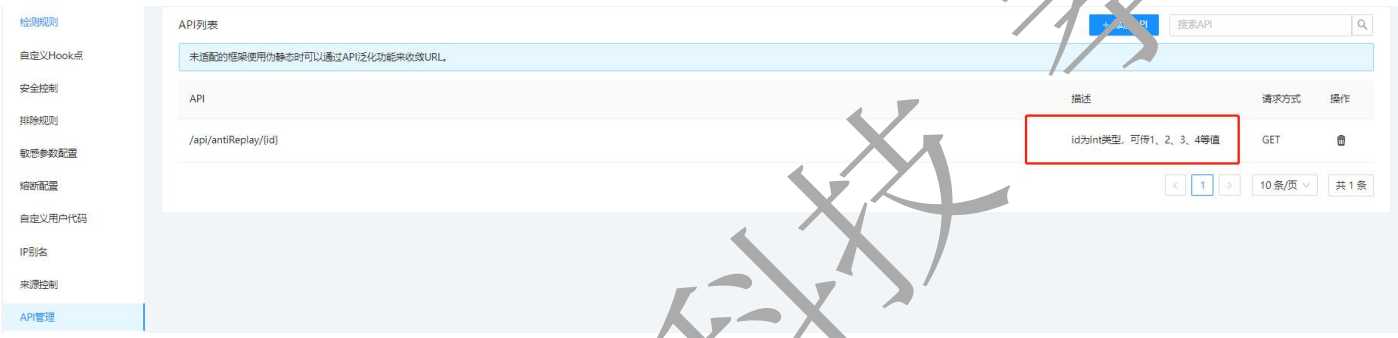
IP 别名可用于来源控制及 IP 管理中，一个 IP 别名中可以添加一个 IP 或多个 IP，具体的 IP 类型可以选择主机、范围及网段。

4.8.9 来源控制

当开启检测来源控制时：选择仅检测以下 IP，则仅对所列出 IP 的请求进行检测，对其他 IP 请求不进行检测；选择不检测以下 IP，则不检测所列出 IP 的请求，对其他 IP 进行检测。

4.8.10 API 管理

未适配的框架使用伪静态时可以通过 API 泛化功能来收敛 URL。可通过配置相应的请求方式，过滤多余 API.

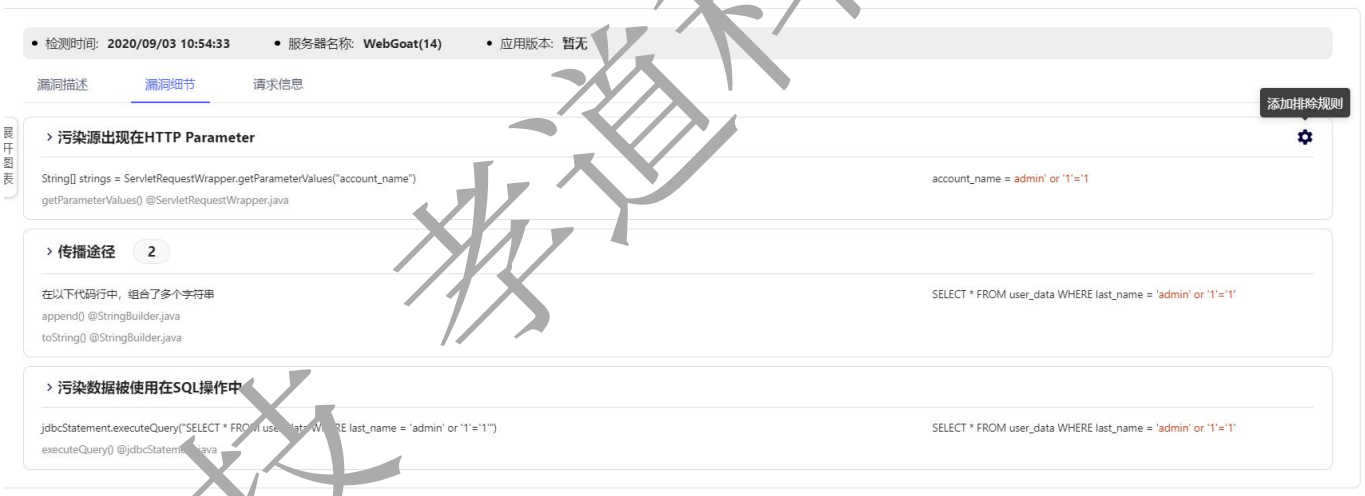


4.8.11 快捷配置

4.8.11.1 规则

1）在漏洞页面中，点击添加排除规则按钮，可以快捷添加排除规则，该页面的规则排除类型是 path，确认无误之后点击保存，该规则即可生效。

2）在漏洞细节中，鼠标悬浮到污染源一栏，显示添加排除规则快捷配置图标，点击按钮，会出现添加排除规则页面，该页面的规则排除类型是输入，确认无误之后点击保存，该规则即可生效。

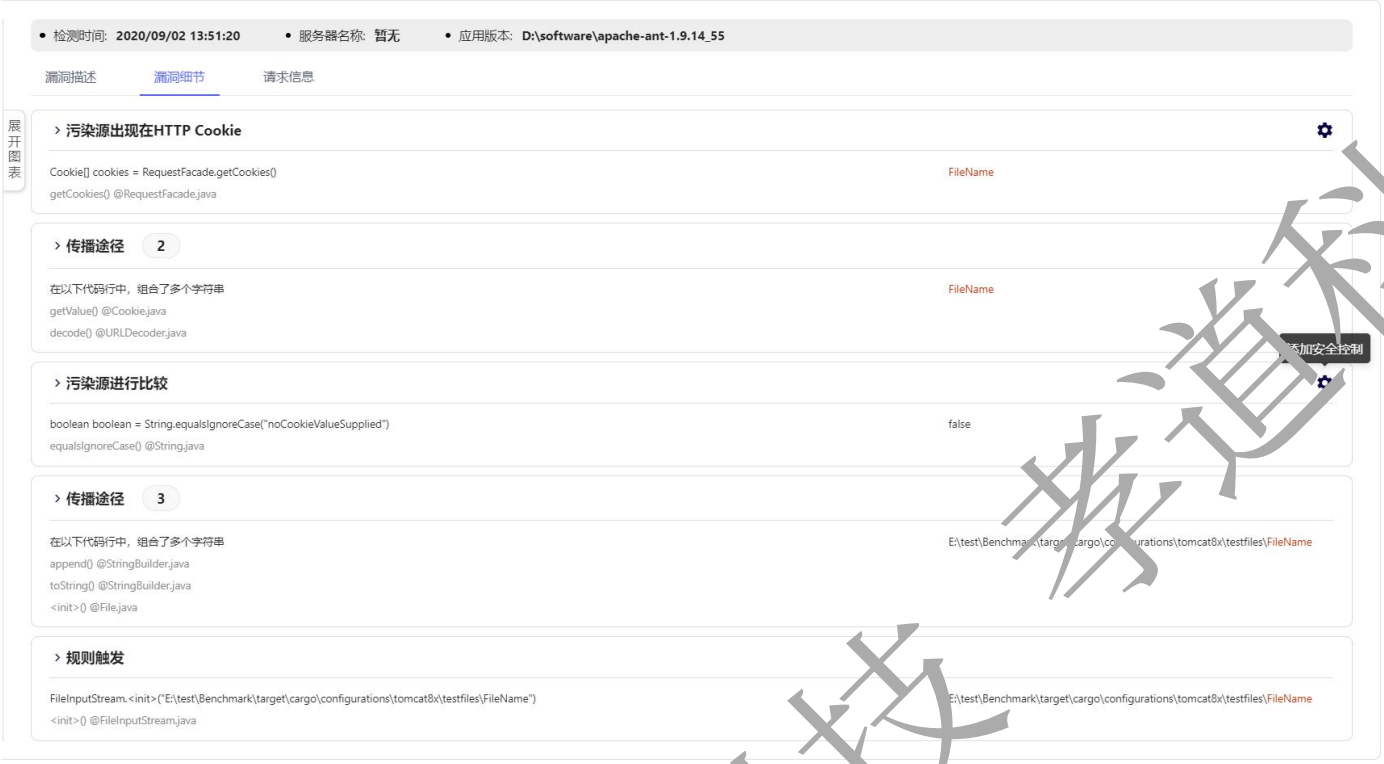


3）在 API 发现页面，鼠标悬浮到 path 后，会显示添加排除规则快捷配置图标，设置完成之后，系统将不再检测该 path 下的相应漏洞。

4）检测日志中，在操作里点击添加排除规则，会出现添加排除规则页面，该页面的规则排除类型是 path，确认无误之后点击保存，该规则即可生效。

4.8.11. 安全控制

在漏洞细节中，对于以下阶段可以快捷添加安全控制，点击按钮，会出现添加安全控制页面，该确认无误之后点击保存，该规则即可生效。



4.9 特性

4.9.1 跟踪集成

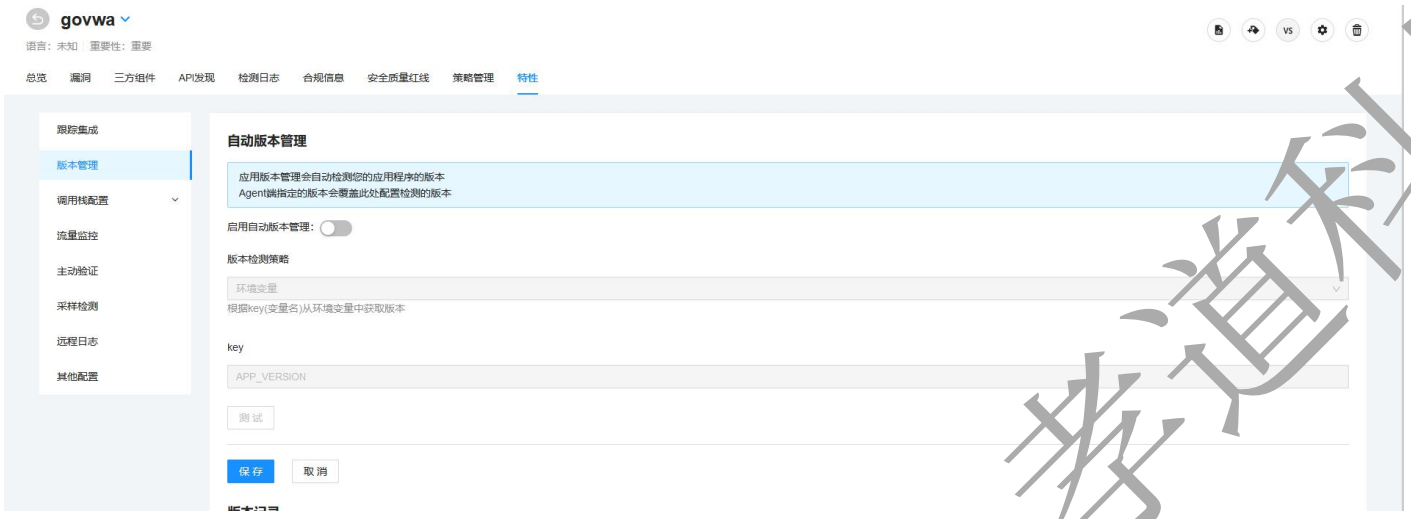
启用 JIRA/禅道跟踪集成，可对 IAST 平台的漏洞及有漏洞的三方组件进行跟踪管理。您可以在 JIRA/禅道平台创建项目，在 IAST 的应用-特性-跟踪集成中 配置 JIRA/禅道的账户信息并连接到 JIRA/禅道平台，连接成功后，可选择创建问题所属的 JIRA/禅道项目（正常情况下，一个应用对应一个 JIRA/禅道项目），并配置 IAST 危害等级与 JIRA/禅道优先级的映射，之后选择问题类型，点击保存完成 JIRA/禅道配置，然后就可以在漏洞列表或三方组件列表创建 JIRA/禅道问题了。

4.9.2 版本管理

启用版本管理后，系统会自动检测您应用程序的版本。

不同的应用采用了不同的设计方式，我们提供了环境变量、Java 系统属性、Java Manifest、配置文件(properties/yaml)、自定义实现类五种版本检测策略（版本检测策略仅支持 Java 应用），您可以根据自己的需要选择应用的版本检测策略，参考辅助说明完成版本检测策略的添加。

如下图所示，自定义实现类的版本检测策略配置方式。



系统采集到应用的不同版本会记录到版本列表中，漏洞的最后归集也会对应到响应的版本上面，后续可以根据版型信息查询对应的相关内容。当然，系统设置了“默认版本”，在无法获取版本的情况下，漏洞信息会统一汇聚到“默认版本”。

4.9.3 调用栈配置

调用栈配置可配置生效于污染源阶段，传播阶段，tag 阶段，触发阶段。当产生大量的漏洞触发时，全量上报会对系统性能产生影响通过配置限定上报数量。

4.9.4 流量监控

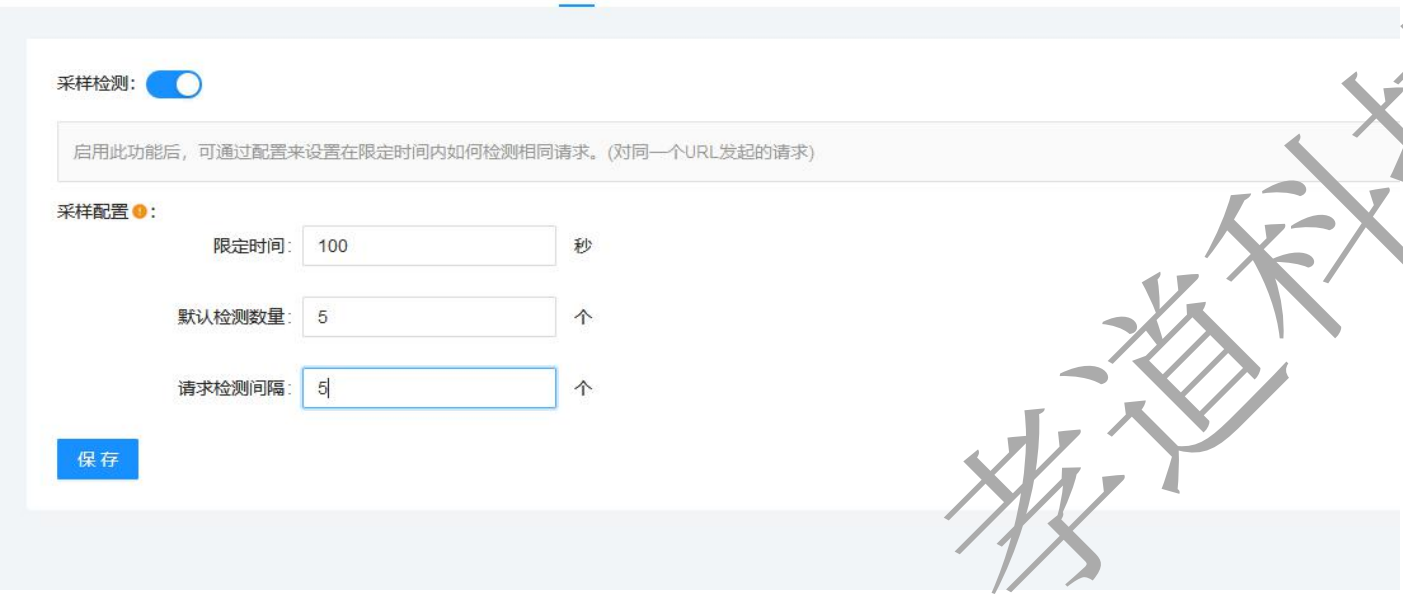
流量监控通过 SecPoint 实时监控 HTTP/HTTPS 请求对应用的访问，您可以在检测日志页面查看结果。如果发现性能下降，则可能要停用此功能。

4.9.5 主动验证

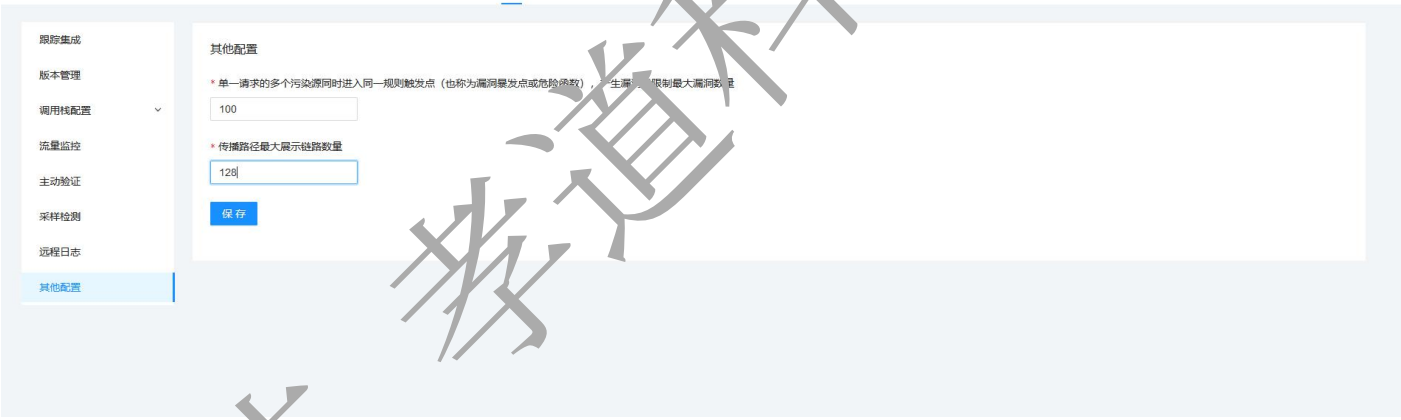
主动验证通过向 Web 服务器发送带有特定 payload 的 HTTP 请求可以很大程度上提高漏洞检测的准确性。

4.9.6 采样检测

启用此功能后，可通过配置来设置在限定时间内如何检测相同请求。(对同一个 URL 发起的请求)，因为检测会实时抓取请求进行分析，会造成一定的服务资源压力。配置开启采样检测，在一定时间段内以一定的时间间隔检测相应数量，可缓解服务压力。

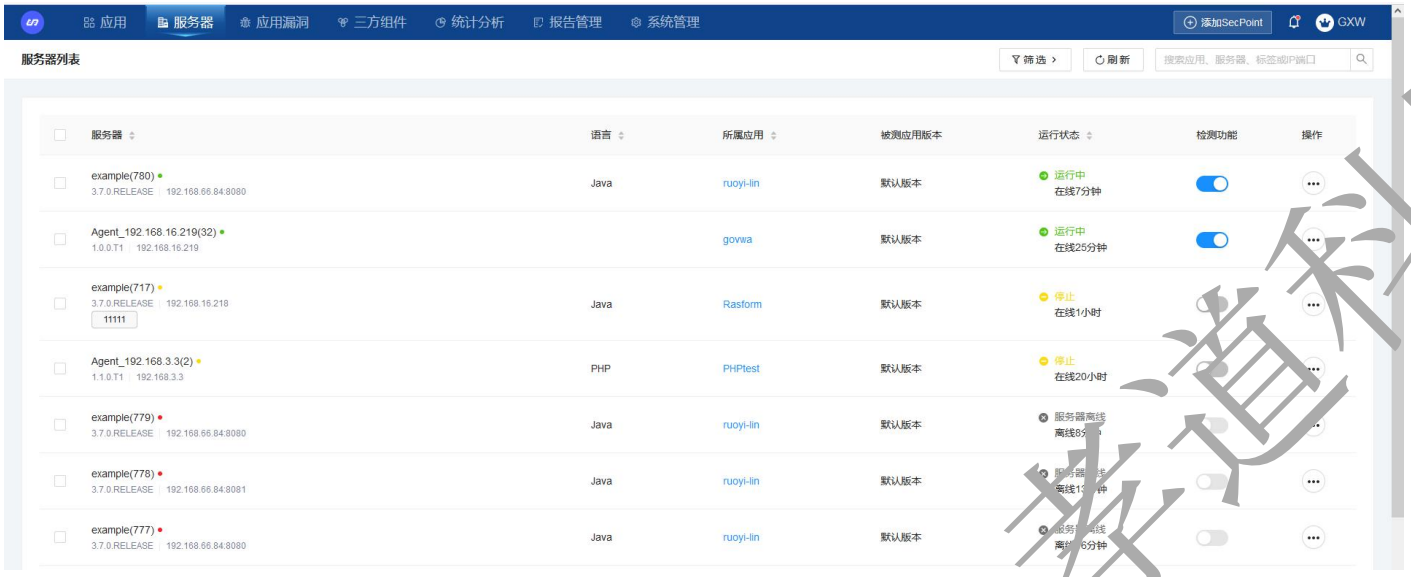


4.9.8 其他配置



5 服务器

服务器列表展示了部署了 SecPoint 的应用所对应信息，一个 SecPoint 对应一个服务器。列表中展示了服务器基本信息，包括 SecPoint 版本、开发语言、所属应用、检测状态等。您可以通过所属应用、检测状态、升级状态、在线状态、搜索框来筛选对应的服务器。勾选服务器可以对在线服务器检测开关批量开启关闭，开启时检测功能开启，关闭时检测功能关闭且占用许可会释放，勾选离线服务器记录批量删除服务器连接信息



5.1 服务器设置

5.1.1 添加标签

您可以为服务器添加标签来标记该服务器。

5.1.2 设置服务器

在操作中点击设置，进入设置服务器页面，您可以修改服务器名称，同时可以根据自己的需求进行 Secpoint 日志配置，包括日志空间配置、日志等级、日志清理及存储位置配置。

并发检测量功能适用于被检测系统并发量比较高的情况，如：对被检测系统进行压力测试时、WEB 扫描器对被检测系统进行安全扫描等场景。该功能用于控制 SecPoint 同时最大能跟踪检测的并发量。

6 应用漏洞

如下图所示，漏洞标签页展示了有关该漏洞的所有信息，包括属性、状态、检测记录、漏洞详情、漏洞风险、合规信息、修复建议等。

6.1 属性

属性栏是对该漏洞的一些基本信息的展示，包括漏洞的所属应用及其最近应用版本、漏洞 KEY、检测次数、暴露天数、code（该漏洞触发的具体代码位置），您可以从该页面对这条漏洞的基本属性

有大致了解。

6.2 状态

在状态栏中您可以为该漏洞创建 JIRA 和禅道问题，我们会对该漏洞创建的所有问题进行跟踪

点击编辑状态，您可以对漏洞状态进行修改，且可以对当前状态进行备注。我们也支持您将该漏洞分配给其他用户，方便您对漏洞进行处理。

点击评论栏，您可以看到所有用户对该漏洞留下的评论。

同时，在状态栏中也会展示目前该漏洞被分配的负责人，以及所有用户对该漏洞进行的状态变更操作。

6.3 检测记录

该栏中记录了该漏洞的所有检测记录，包括不同应用版本的漏洞记录。检测记录超过 50 条，将会自动删除多余的记录。对于第一次上报的记录和经过了主动验证的记录，会默认标星处理。标星的记录将不会被自动删除，且不计入自动删除的统计数量中。

6.4 漏洞详情

漏洞详情中主要展示该条漏洞记录的详细信息，包括漏洞检测时间、所属服务器名称、应用版本以及漏洞信息。漏洞信息主要包括漏洞描述、漏洞细节、请求信息，如果该条检测记录进行了主动验证，还将展示验证信息。

在漏洞描述中简单描述了漏洞的形成原因，您可以快速看到漏洞存在的具体位置。

在漏洞细节中会按代码的执行顺序，将所有污染源、传播途径、规则触发展示在页面中。通过来源参数，我们可以追溯到漏洞出现的位置，之后分析传播途径，我们可以进一步了解到这些数据在程序内部是如何传递的，最终在规则触发阶段中，我们可以查看到污染数据被使用的具体操作。此外，您也可以展开图表，比较直观地查看漏洞的传播过程。

请求信息标签页展示了请求信息及响应信息。您可以通过编辑请求信息，之后点击重放按钮来重放该请求。

验证信息页面会将主动验证的结果展示出来，在验证信息中您可以查看到替换攻击载荷后重发的请求信息，同时我们会给出该验证结果的验证依据。

6.5 漏洞风险

漏洞风险一栏主要描述了漏洞的形成原因及由该漏洞可能导致的风险。

漏洞风险

攻击者通过精心设计的输入，可以更改SQL语句的结构并执行SQL注入攻击。当开发人员通过拼接用户提供的数据来构建SQL语句而不进行验证或编码时，就可能造成SQL注入攻击。这种攻击的目的是迫使数据库检索和输出用户无法访问的数据。例如，攻击者可以在脆弱的应用程序上使用SQL注入来查询数据库中的客户信用卡号和其他数据，即使这些数据不是用户权限范围内可以查询的数据。SQL注入还可能造成提权、帐户劫持等安全问题，在某些情况下，攻击者还可能获得对数据库服务器的shell访问权。

6.6 合规信息

合规信息收录了 OWASP、CWE/SANS、PCI-DSS 等开放式 Web 应用安全项目的 10 项最严重的 Web 应用程序安全风险类别，这里会将该漏洞符合的各安全风险类别展示出来。

合规信息

- OWASP Top 10 2013　A1 – Injection
- CWE/SANS 2011　1 CWE-89 Improper Neutralization of Special Elements
- PCI-DSS v3.2.1　6.5.1 – Injection flaws
- GDPR　Injection
- OWASP Top 10 2017　A1 – Injection

6.7 修复建议

在修复建议中，针对各种不同类型的漏洞，我们给出了比较专业的修复建议，并且提供了安全代码以及不安全代码示例以供参考，这样即使开发人员对某些漏洞不太了解，也能够通过参考示例给出比较合理的修复方案。

修复建议

1. 对SQL语句进行预编译和绑定变量.
2. 对进入数据库的特殊字符进行转义处理，或编码转换.
3. 对输入的数据进行类型检查，比如数字型的数据就必须是数字，相应的数据库中的存储字段必须为int型.
4. 数据长度应该严格规定，能在一定程度上抵御较长的SQL注入语句的执行.
5. 通过对数据库强制执行最小权限原则，来减缓SQL注入漏洞的影响。籍此，应用程序的每一个软件组件都只能访问、并仅影响它所需要的资源.
6. 避免网站显示SQL错误信息，比如类型错误、字段不匹配等，防止攻击者利用这些错误信息进行一些判断.

不安全代码示例

```
String name = request.getParameter("name");
...
Statement statement = con.createStatement();
String sql = "select * from users where name = '" + name + "'"; //通过拼接的方式构建sql查询语句，存在sql注入的风险
ResultSet rs = statement.executeQuery(sql);
...
```

安全代码示例

```
...
String name = request.getParameter("name");
String sql = "select * from users where name = ?"; //通过预编译的sql查询语句实现了将代码与变量分离
PreparedStatement st = con.prepareStatement(sql);
st.setString(1, name);
ResultSet rs = st.executeQuery();
...
```

参考链接

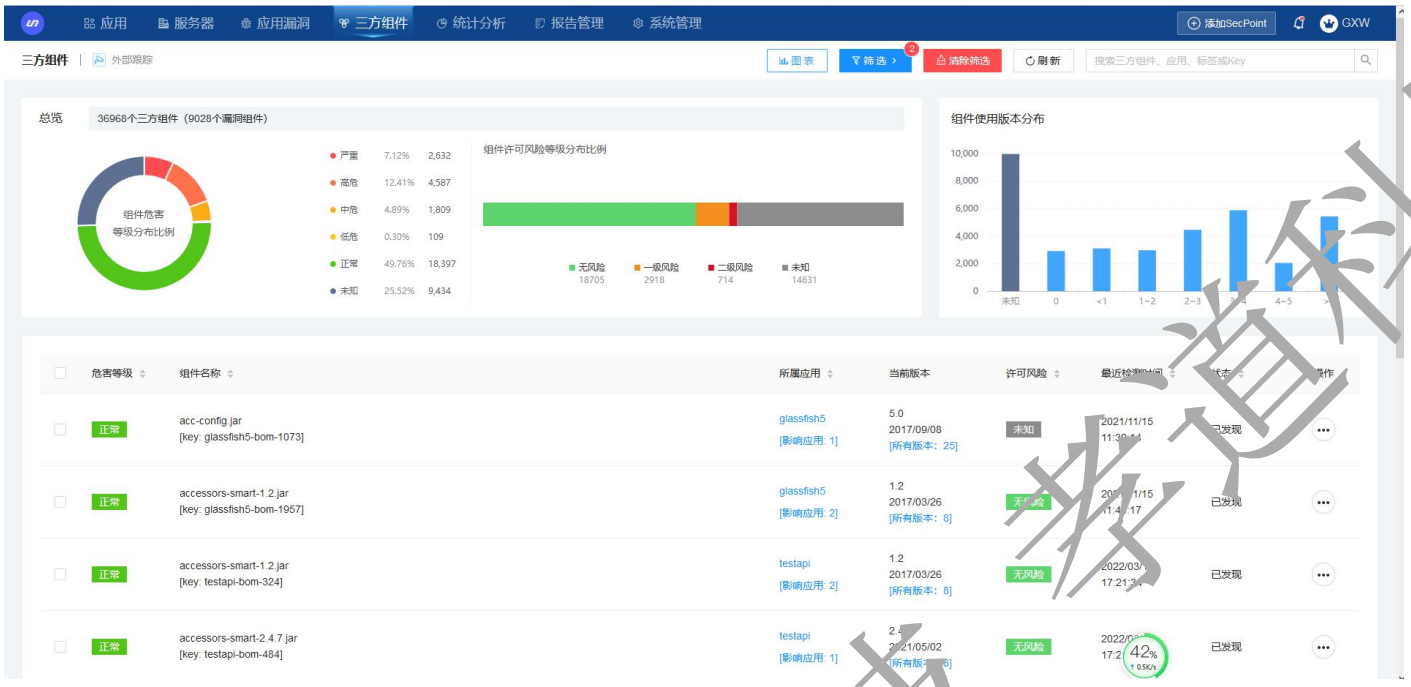
https://cwe.mitre.org/data/definitions/89.html
https://www.owasp.org/index.php/SQL_Injection
https://cwe.mitre.org/data/definitions/89.html
http://www.unixwiz.net/techtips/sql-injection.html

7 三方组件

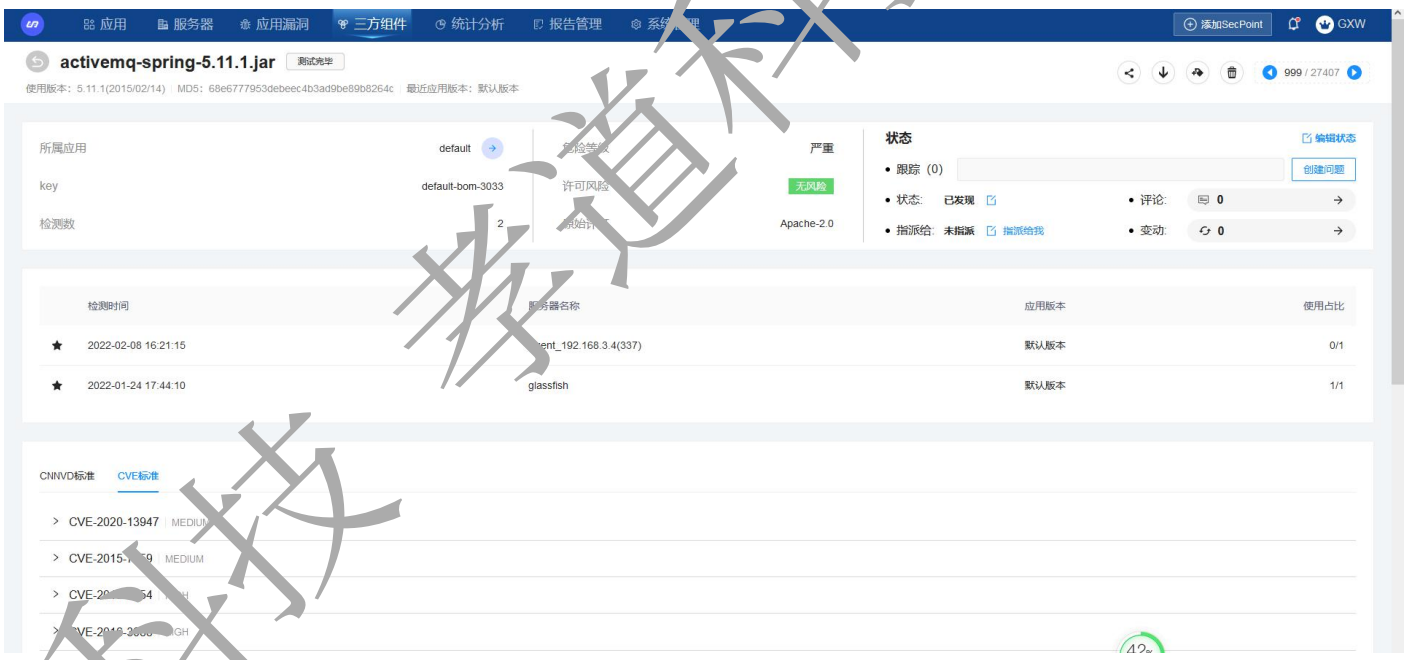
三方组件页面展示出了三方组件危害等级分布比例、组件使用版本分布以及组件许可风险等级分布比例等。

您可以按危害等级、所属服务器、许可风险筛选查看三方组件，对于风险等级较高的组件，可替换成安全版本使用。

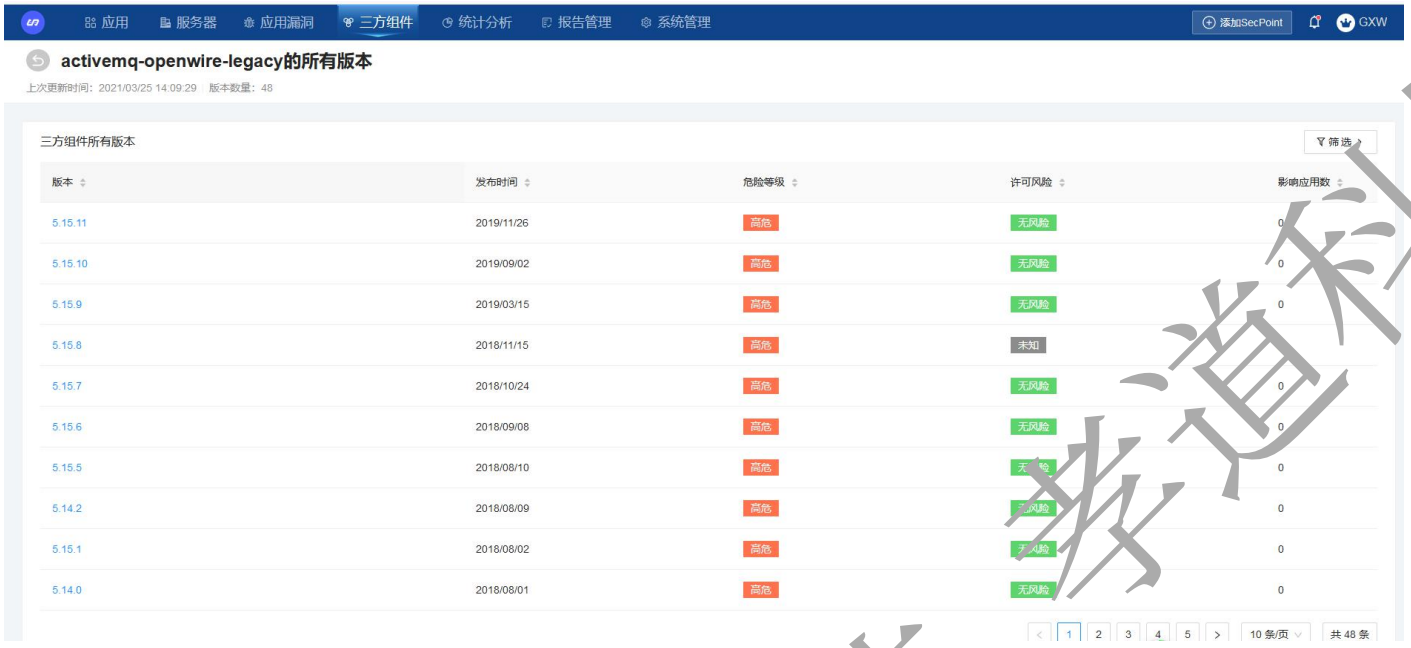
危险等级中的未知状态是说明该 jar 包是用户自定义的 jar 包，或者三方组件库中匹配不到的 jar 包。



点击三方组件名称，可进入三方组件详情页面，查看该组件的详细漏洞信息。目前进行检测的漏洞标准为 CVE 标准、CNNVD 标准、自定义标准。



点击所有版本可进入所有版本详情页面，可以查看当前组件哪个版本是安全的，选择安全版本使用。



8 统计分析

IAST 提供丰富的数据呈现方案，将应用数据转换成可视化图表动态展示。通过对各个维度数据的对比，可以让管理者对研发团队的安全能力状况有一个大致的了解，客观全面地认识整个公司研发团队的安全能力现状。通过应用的总漏洞数、高风险漏洞数、漏洞平均修复时间和修复率可对项目团队的安全开发能力进行评估，方便后续研发团队的安全开发水平的提升。

8.1 漏洞分析

漏洞分析模块针对小组和应用两个维度下的所有漏洞，以危害等级对漏洞数量、新增漏洞数量进行统计分析，并可以根据漏洞发现时间、小组名、应用名进行筛选统计，目前漏洞分析统计支持两种视图之间的切换，组件视图、应用视图。

8.1.1 漏洞总览

可选择半年或者一年的时间长度对整个系统中所有新增的总漏洞数以及高风险漏洞（严重、高危）数分别进行统计，以便对所有项目的漏洞数量变化有全局了解。

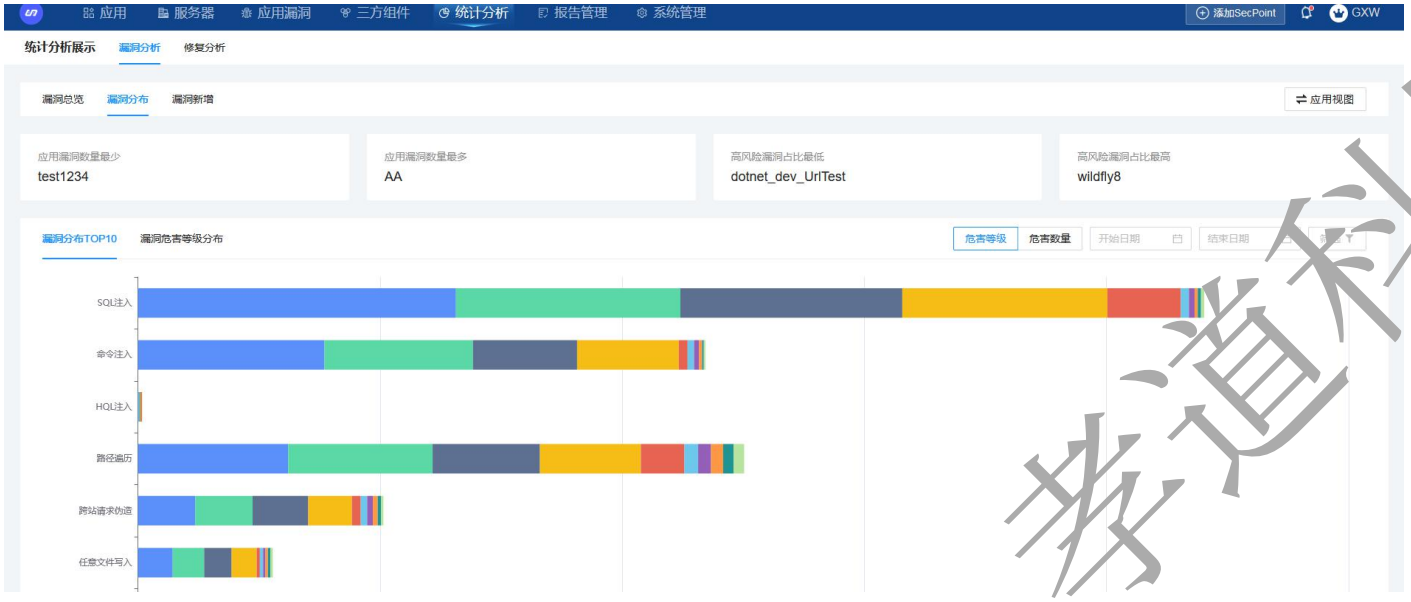
8.1.2 漏洞分布

8.1.2.1 小组视图

该页面以堆叠图和表格的形式展示所有小组 Top10 的漏洞等级、数量及其分布信息。表格中可对不同等级漏洞进行排序，顶部筛选栏可对具体小组和漏洞发生的时间进行筛选。

8.1.2.2 应用视图

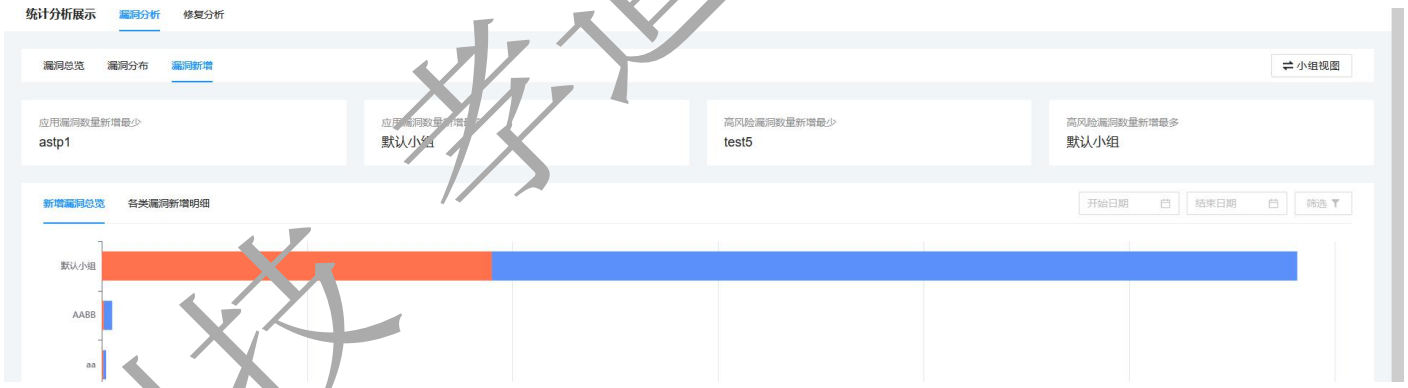
该页面以堆叠图和表格的形式展示所有应用 Top10 的漏洞等级、数量及其分布信息。表格中可对不同等级漏洞进行排序，顶部筛选栏可对具体应用和漏洞发生的时间进行筛选。



8.1.3 漏洞新增

8.1.3.1 小组视图

该页面将每个小组新增漏洞总数、高风险漏洞总数及各类型漏洞新增明细分别以条形图和表格的形式进行展示。表格中可根据漏洞等级进行排序，默认以当前选中项扩展排序规则。顶部筛选栏可根据应用名称和时间进行筛选，默认展示最近一年内所有项目的数据。



8.1.3.2 应用视图

该页面将每个应用新增漏洞总数、高风险漏洞总数及各类型漏洞新增明细分别以条形图和表格的形式进行展示。表格中可根据漏洞等级进行排序，默认以当前选中项扩展排序规则。顶部筛选栏可根据应用名称和时间进行筛选，默认展示最近一年内所有项目的数据。

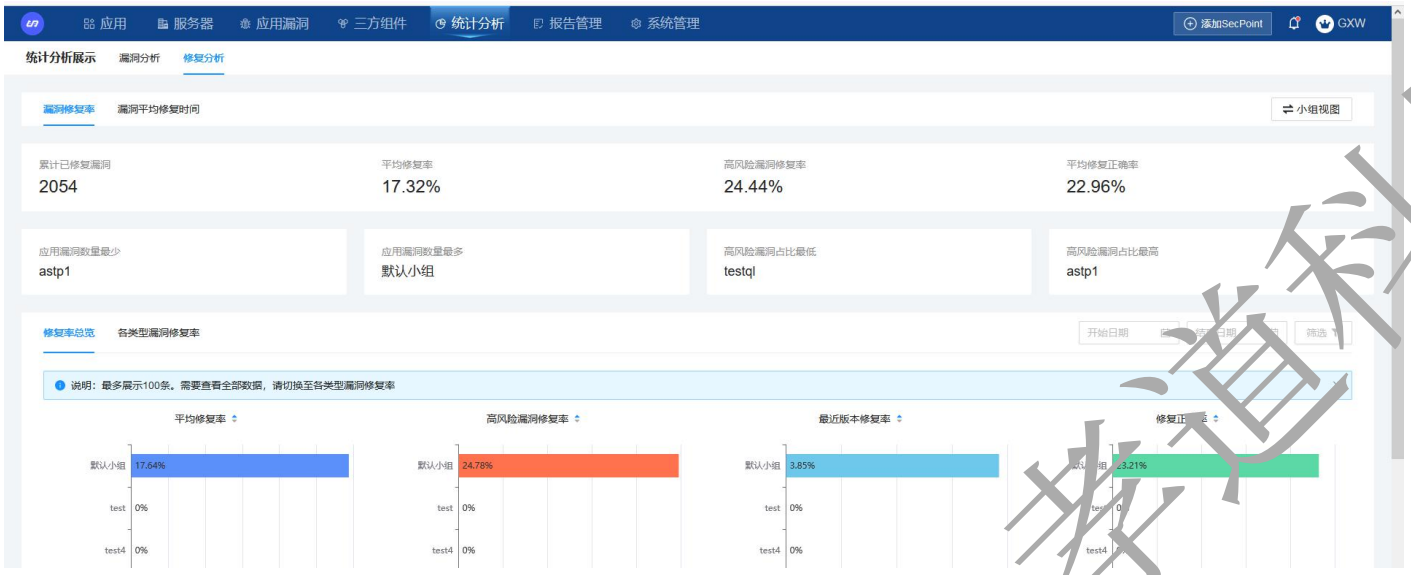
8.2 修复分析

修复分析模块针对每个小组和应用的所有漏洞，对不同漏洞等级的修复时间及其修复率进行统计，并可以根据时间、小组名、应用名进行筛选，目前修复分析统计统计支持两种视图之间的切换：组件视图、应用视图。

8.2.1 漏洞修复率分析

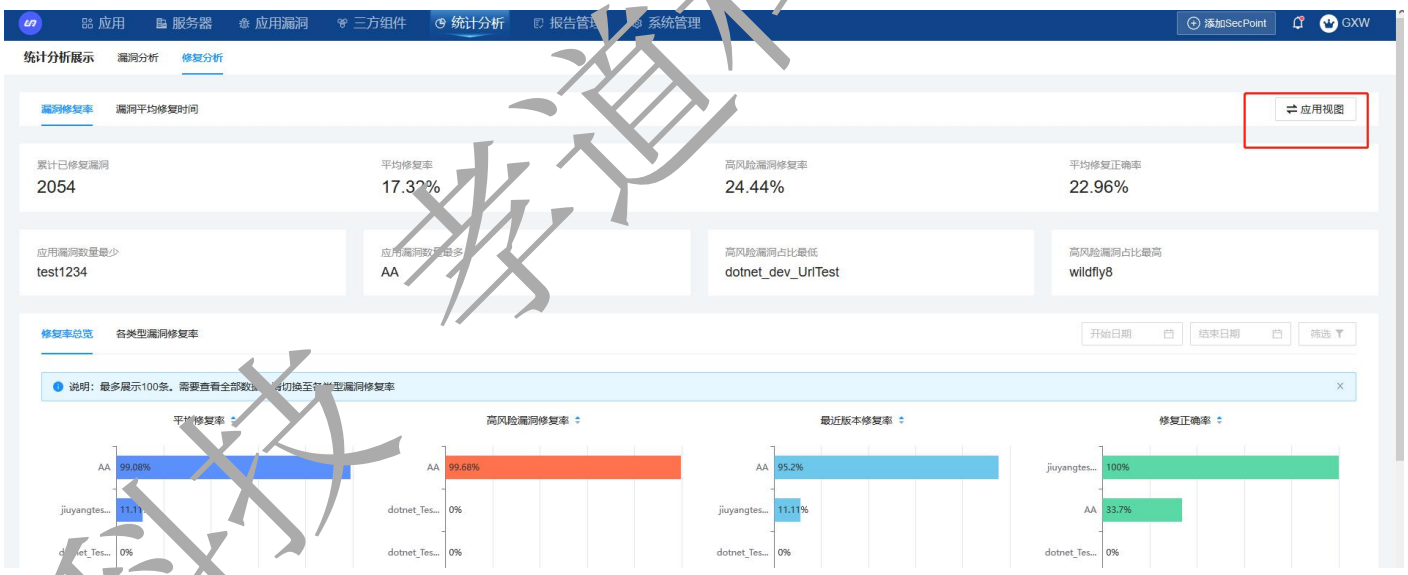
8.2.1.1 小组视图

该页面将每个小组所有漏洞及高风险漏洞的修复率、最近应用版本修复率、修复正确率以及各漏洞类型修复率分别以条形图和表格的方式进行展示，当修复率低于某个设定数值可标红提醒。表格中可根据不同选项进行排序，默认以当前选中项扩展排序规则。顶部筛选栏可根据小组名称和时间进行筛选，默认展示最近一年内所有项目的数据。



8.2.1.2 应用视图

该页面将每个应用所有漏洞及高风险漏洞的修复率、最近应用版本修复率、修复正确率以及各漏洞类型修复率分别以条形图和表格的形式进行展示，当修复率低于某个设定数值可标红提醒。表格中可根据不同选项进行排序，默认以当前选中项扩展排序规则。顶部筛选栏可根据应用名称和时间进行筛选，默认展示最近一年内所有应用的数据。

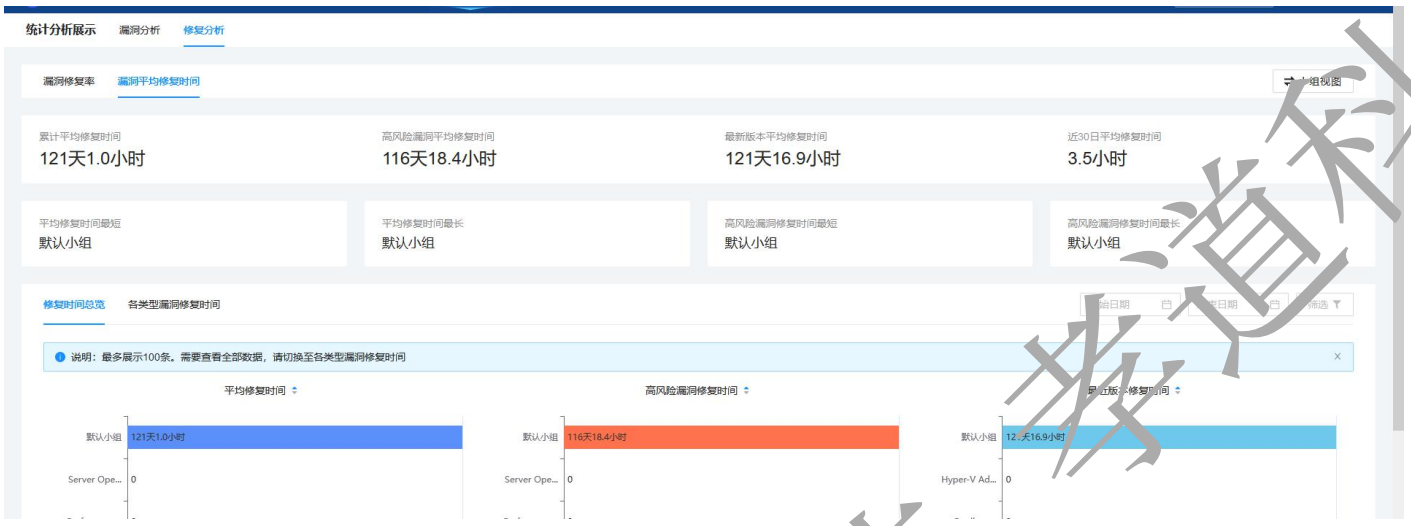


8.2.2 漏洞平均修复时间

8.2.2.1 小组视图

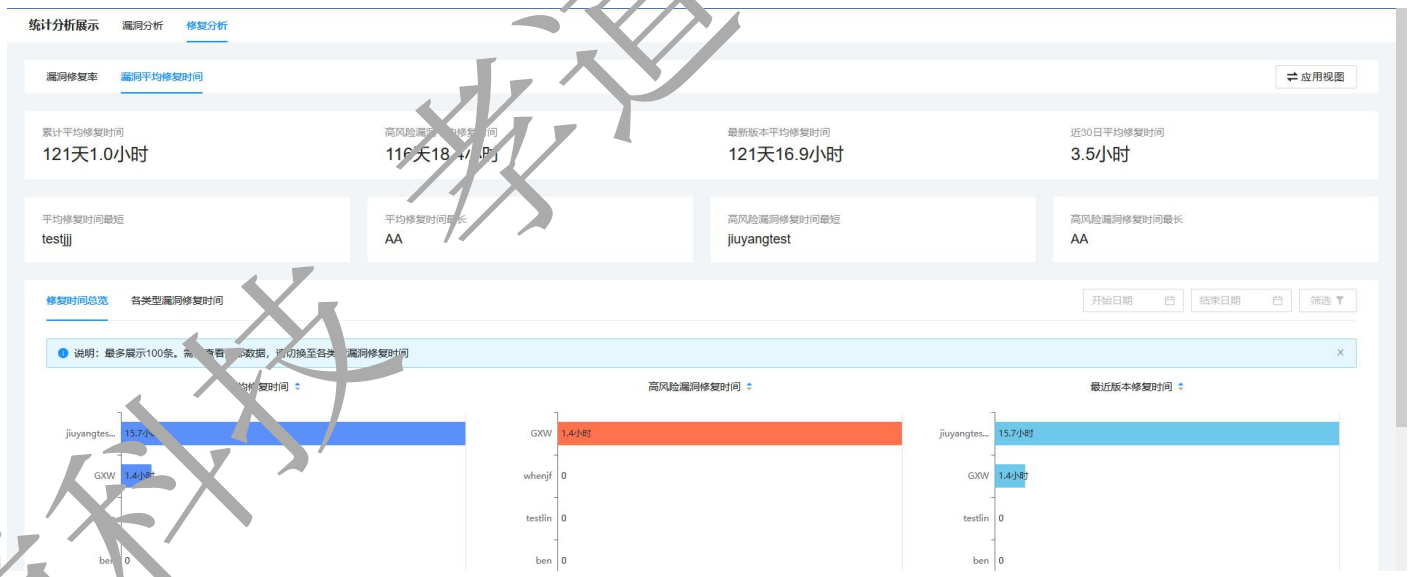
该页面将每个小组所有漏洞及高风险漏洞的修复时间、最近应用版本修复时间以及各漏洞类型修复时间分别以条形图和表格的形式进行展示，当修复时间高于某个设定数值可标红提醒。表格中可根

据不同选项进行排序，默认以当前选中项扩展排序规则。顶部筛选栏可根据小组名称和时间进行筛选，默认展示最近一年内所有小组的数据。



8.2.2.2 应用视图

该页面将每个应用所有漏洞及高风险漏洞的修复时间、最近应用版本修复时间以及各漏洞类型修复时间分别以条形图和表格的形式进行展示，当修复时间高于某个设定数值可标红提醒。表格中可根据不同选项进行排序，默认以当前选中项扩展排序规则。顶部筛选栏可根据应用名称和时间进行筛选，默认展示最近一年内所有应用的数据。



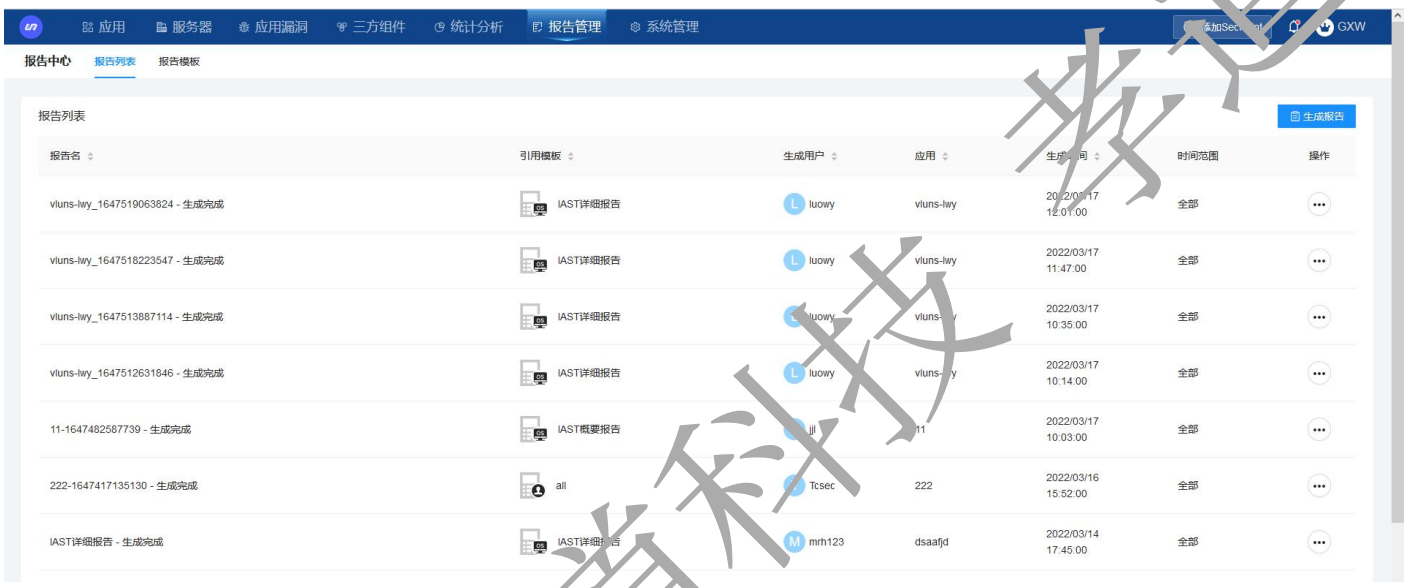
9 报告管理

报告管理主要提供了用户使用系统模板或者自定义模板生成报告的功能，您可以在报告模板页面

中创建自定义的模板，在报告列表中您可以通过系统默认模板或者自定义添加模板进行报告生成、下载以及管理。

9.1 报告列表

报告列表会展示系统生成的报告记录，将会记录生成的报告名、引用模板、生成用户、生成时间、生成报告的应用、时间等。在列表中也可以进行报告的下载及删除操作。



您也可以在该页面进行报告生成，选择报告类型、模板、应用及报告名后，即可生成相应的报告。

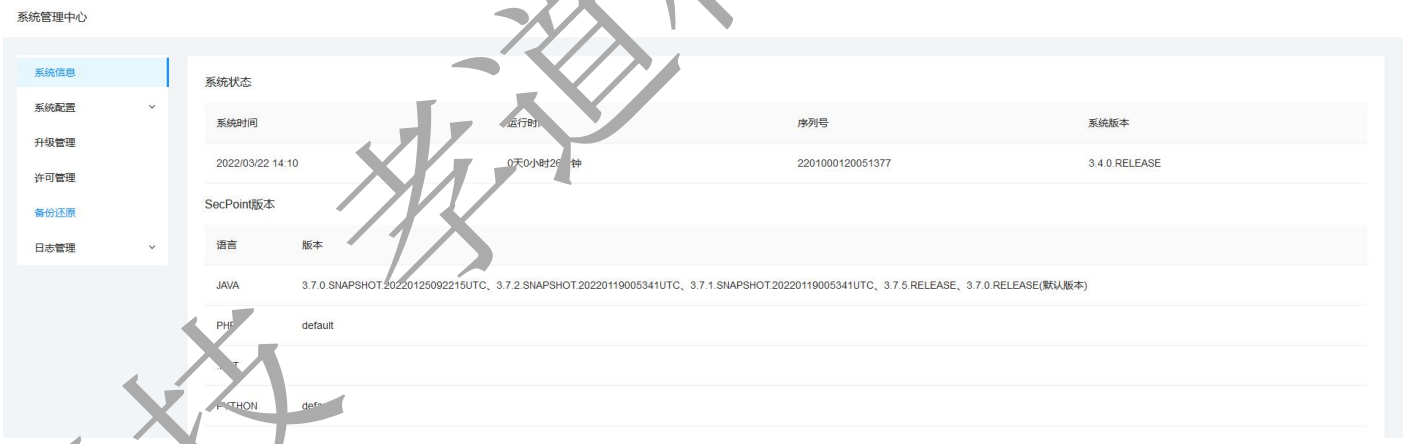
9.2 报告模板

报告模板中提供了两种系统默认模板，包括 IAST 概要报告模板、IAST 详细报告模板。您也可以创建模板，选择您需要的模块进行模板自定义。

10 系统管理

10.1 系统信息

系统状态主要是展示了系统的持续运行时间及版本等信息



10.2 系统配置

10.2.1 分享管理

启用分享管理，允许应用漏洞和三方组件的分享，可以自由决定管理员、小组管理员、普通用户分享的权限，关闭时，不允许任何用户进行应用漏洞/第三方组件分享。

10.2.2 邮件设置

邮件设置功能用于配置发送邮件的服务器，配置成功后，忘记密码功能及事件管理中的邮件告警功能才能正常使用。建议 SMTP 服务器及邮箱地址选择企业邮箱，因为个人邮箱的收发会受到一定的限制，相似邮件收发过多可能被判定为垃圾邮件。具体邮件服务器配置方式可根据自己公司使用的企业邮箱自行查询。

邮件配置完成之后，可点击测试发送测试邮件。如果测试邮件发送不成功，请检查配置，如果测试邮件发送成功、邮箱也接收到，说明邮件配置成功。

10.2.3 邀请注册

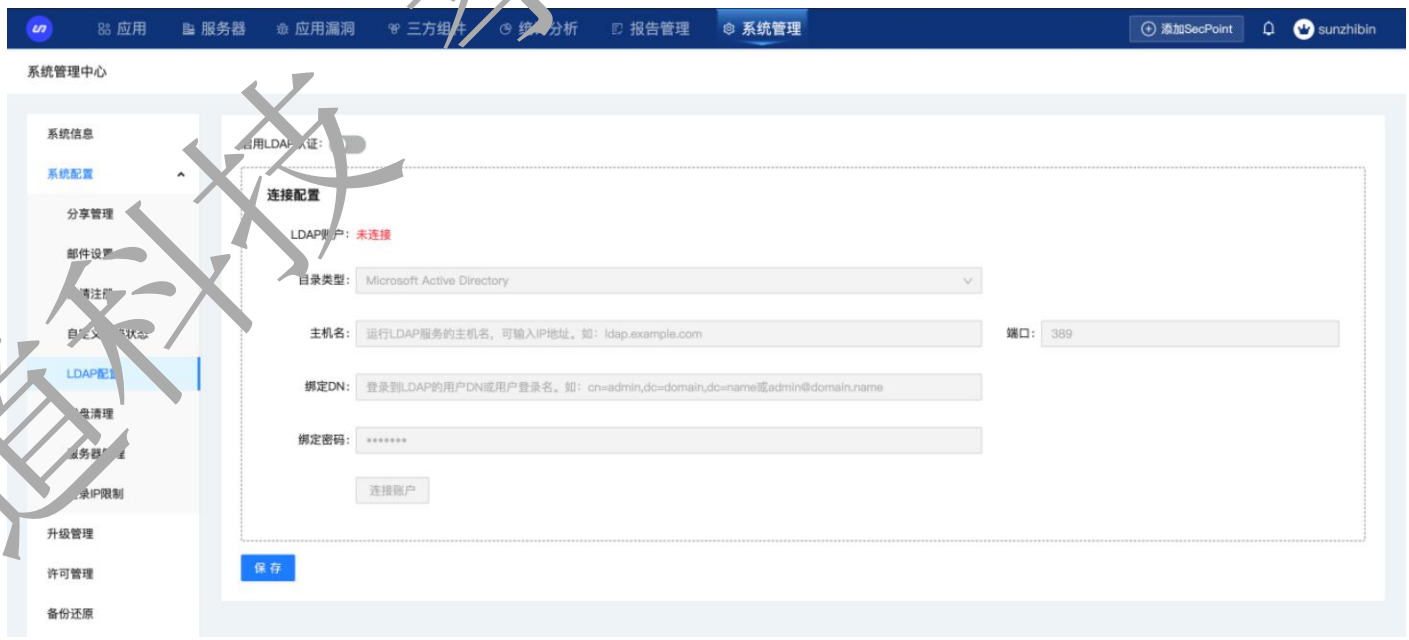
邀请注册用于邀请新用户注册 IAST 平台，开启时，勾选管理员即只有管理员可以发出邀请注册的链接，勾选管理员、小组管理员即管理员、小组管理员都可以邀请用户注册。关闭时，邀请用户注册功能被禁用。

10.2.4 自定义漏洞状态

通过自定义漏洞状态，您可以将漏洞的状态定义为适合自己公司及应用的状态，建议将状态修改为系统现有状态的近义词。如果您希望将漏洞状态还原为初始状态，可以点击恢复默认。

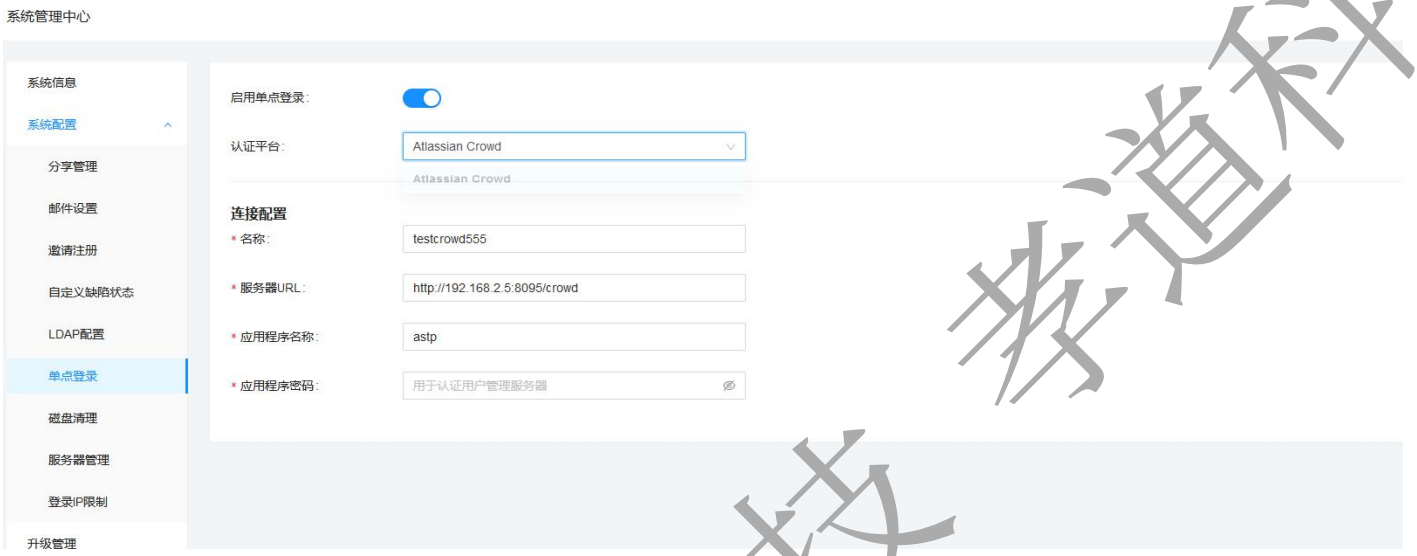
10.2.5 LDAP 配置

LDAP 配置完成并启用后，即可同步 LDAP 服务器上的用户信息到 IAST 平台，用户即可以使用 LDAP 账户进行登录。



10.2.6 单点登录

配置 crowd 连接后，可以直接使用 crowd 平台的账户密码登录，crowd 平台登录后的用户默认所属组为默认小组，管理员可在 IAST 平台分配所属组



10.2.7 磁盘清理配置

通过磁盘清理配置，您可以配置定期清理检测日志、攻击日志、系统日志、IAST 平台后台日志、备份数据及操作日志。

立即清理：数据量超过容量阈值时最早产生的超出容量阈值 80%部分的数据会被清理。

立即清空：清空该项所有数据。

10.2.8 服务器管理

服务器管理开启时，保留离线服务器连接记录，关闭时自动清理离线服务器的连接记录

10.2.9 登录 IP 限制

限制访问 IAST 平台的来源 IP。可输入多个 IP，多个 IP 之间以;分隔。同时允许模式匹配，仅支持*，如：192.168.1.*;192.168.2.*

10.3 升级管理

10.3.1 在线更新

配置在线升级平台服务器的地址，可立即更新三方组件漏洞库，也可以配置周期，定期更新三方组件漏洞库。

10.3.2 离线更新

点击下载按钮导出三方组件 MD5 值的列表文件到本地，然后联系相关工作人员获取新的的三方组件漏洞库，之后点击上传按钮，将新的文件上传至 IAST 平台，即可完成三方组件漏洞库的离线更新（如果如无导出文件，也可以直接联系相关工作人员）。

10.3.3 系统升级

系统升级包主要包含 IAST 平台升级文件、SecPoint 升级文件、三方组件库文件及系统相关补丁，获取升级包后，点击上传升级包，按照提示即可完成升级。

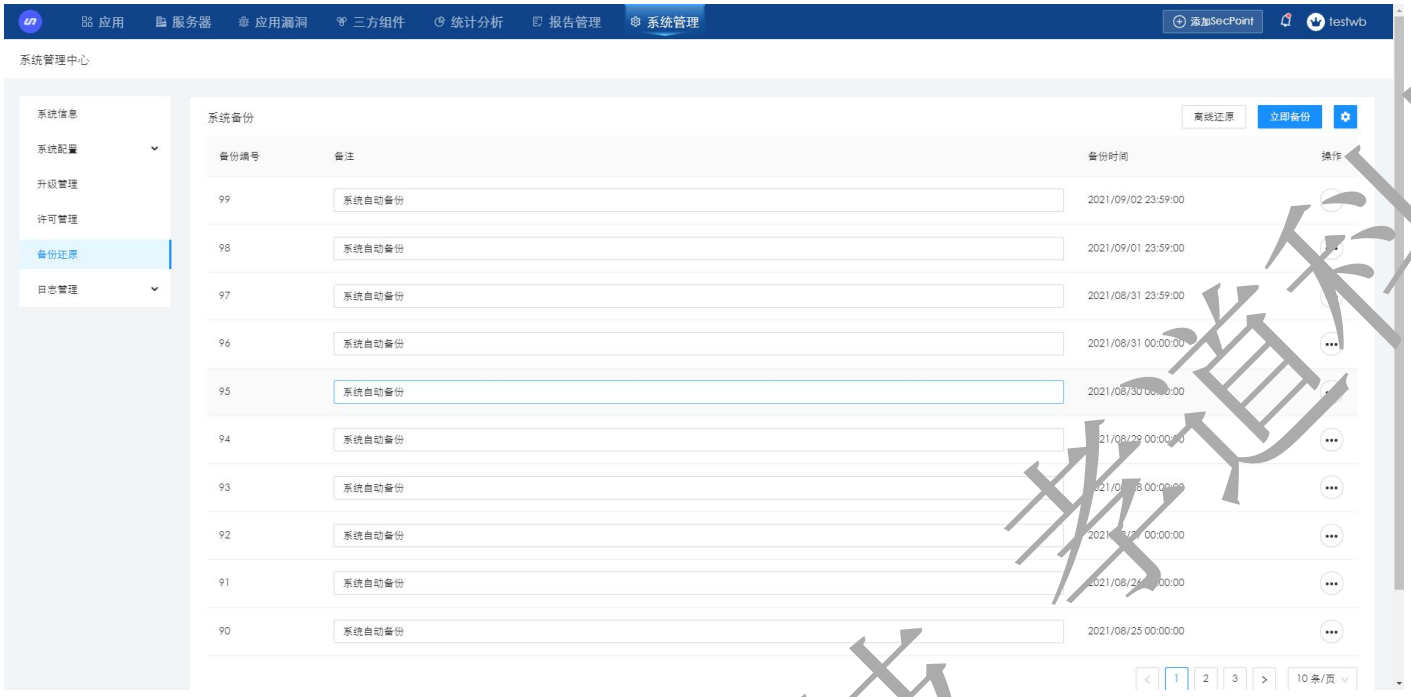
10.4 许可管理

当许可时间到期后或者许可数量不够使用时，可下载许可申请文件或者 base64 格式申请文件，然后联系安全玻璃盒工作人员，获取新的许可文件，导入系统，重新登录即可使用。

10.5 备份还原

备份恢复功能主要用于对系统数据库信息进行备份，当出现误删或者误操作影响系统的正常使用时，可通过还原操作，将系统还原至之前正常的一个节点。如果系统文件损坏，您也可以重新安装系统，然后将备份数据重新导入平台，之前备份的数据在新的系统中仍能正常使用。

您可以手动备份，也可以选择自动备份，设置备份的周期。



10.6 日志管理

10.6.1 系统日志

此处可以查看系统的告警日志，当服务器出现状况时，可以及时的得到通知。同时 Agent 升级日志、系统的升级日志及三方组件的升级日志都会在此展示。

10.6.2 操作日志

操作日志记录了登录、退出、新增、修改、删除、下载等操作的日志以及相应的搜索，当系统出现异常问题时，可以在此处追踪原因。

10.6.3 日志收集

您可以通过 astp 日志收集下载 astp 日志，可根据需要选择下载一天内、一周内、两周内、一月内的日志，如果需要收集 SecPoint 日志请前往服务器列表进行收集。

11 用户中心

鼠标移入右上角用户名，进入用户中心，即可进行小组新建、用户添加、策略配置等操作。

11.1 个人设置

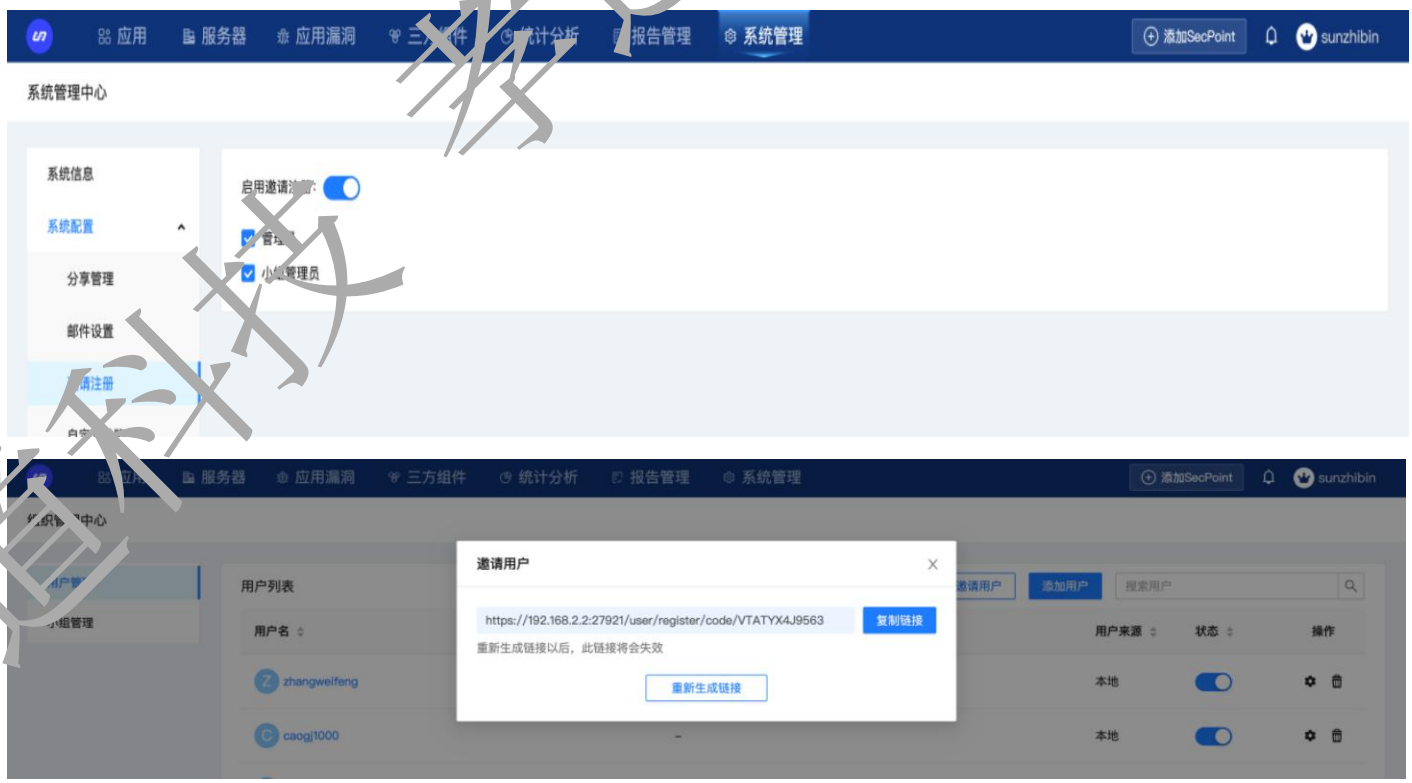
您可以在个人设置中修改当前账户的个人信息及密码，如果需要当前账户一直保持在线，您可以在个人信息中将会话超时时间设置为 0，如果不需要一直保持在线，您可以根据自己的需要设置会话超时时间。

11.2 组织设置

11.2.1 用户管理

IAST 平台可通过邀请注册、平台添加及 LDAP 同步三种方式添加用户

11.2.1.1 使用邀请注册功能需要在系统管理-系统配置-邀请注册页面启用此功能，开启之后点击邀请用户，复制邀请链接发给需要注册的用户，用户在浏览器打开链接，输入相应的信息点击确认注册即可完成账号的注册，注册完成之后可直接登录。



11.2.1.2 您也可以在 IAST 平台直接添加用户，添加用户的时候可以通过选组功能，直接为用户分配小组。如果添加用户时时未选择小组，用户默认者属于默认小组。

11.2.1.3 在系统管理-系统配置-LDAP 配置页面完成 LDAP 的配置并保存，即可从 LDAP 服务器同步用

户到 IAST 平台。

11.2.2 小组管理

可在小组管理中添加不同的小组，小组许可分配方式有共享和配额两种，默认为共享模式，共享即不需要给小组分配许可，所有共享模式的小组都使用系统可分配的许可。

当许可分配方式为配额时，添加小组时需要给小组分配检测许可数量。

示例：系统共有 50 个 IAST 许可，建了三个小组 A、B、C；A 和 B 小组是共享模式，C 是配额模式，分配了 10 个许可。此时 A 和 B 共享使用 40 个，C 独享 10 个同时且 C 不能超过 10 个许可；

给小组分配许可额度的时候，可分配数量（检测的数量）=许可总数量-已分配给所有小组的额度-共享额度的小组已授权的数量。

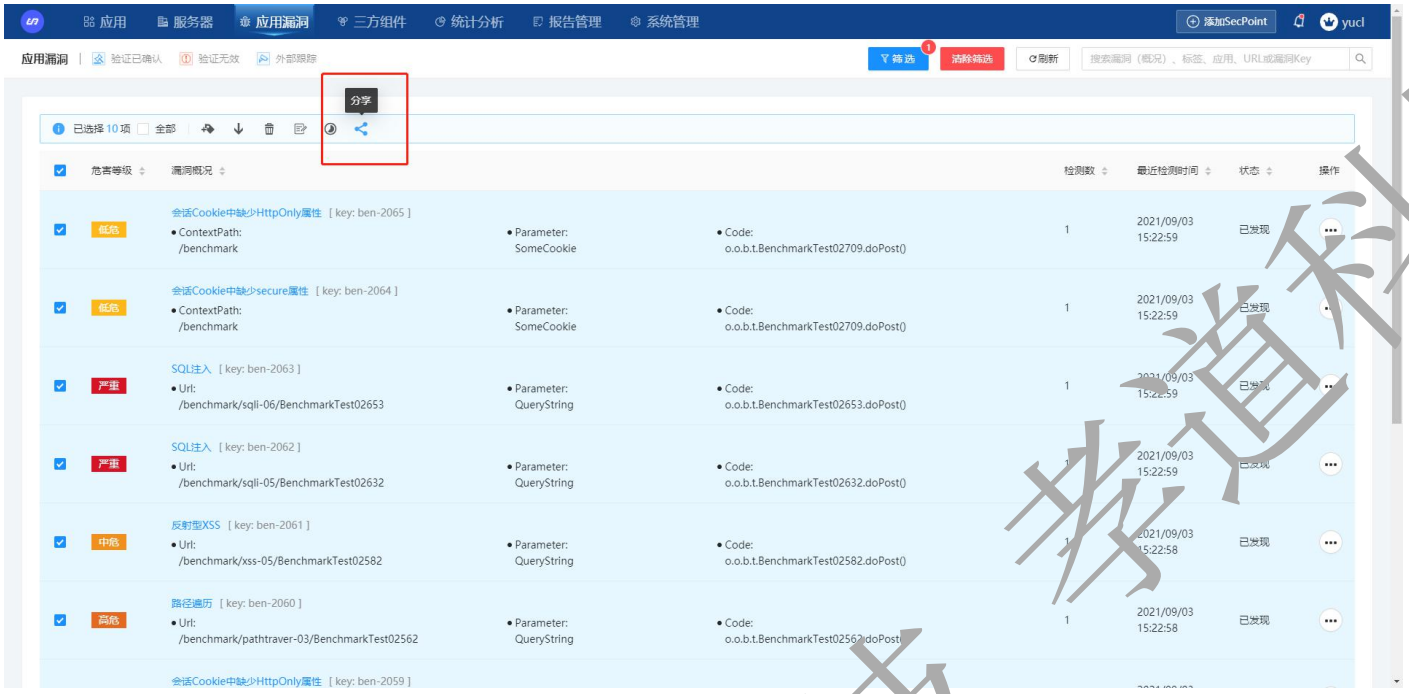
添加小组时，可以通过选择组内用户功能为该小组添加用户。

11.3 分享

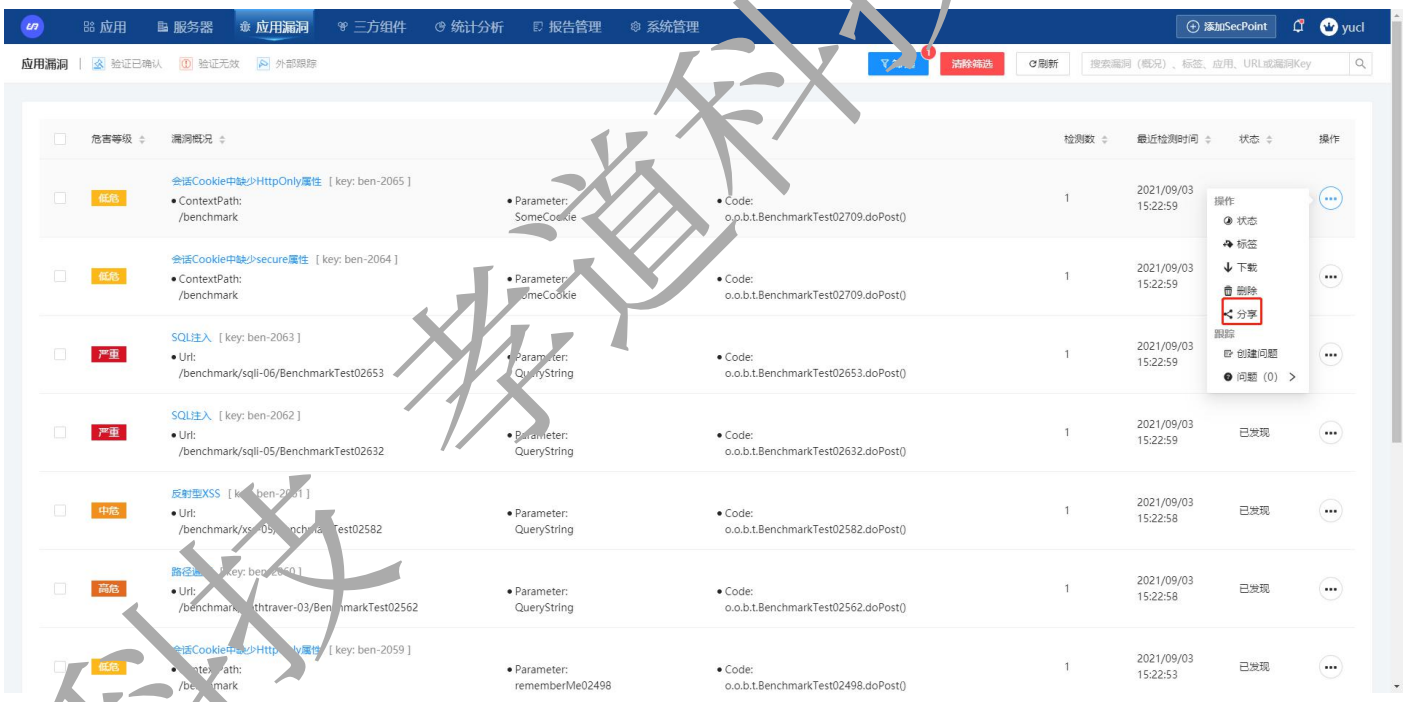
11.3.1 分享

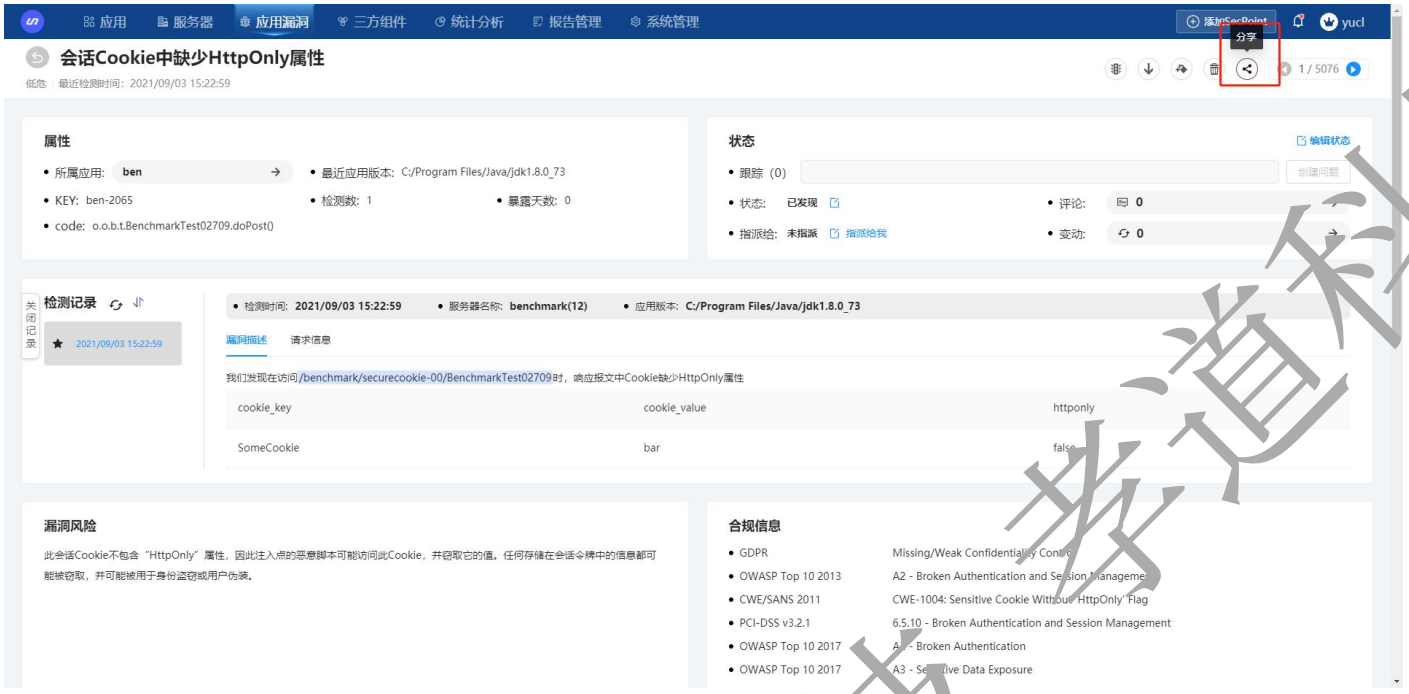
漏洞和问题组件均可以进行分享，生成的分享链接可以直接查看漏洞以及组件详情，并进行状态的修改。

批量分享：


应用
服务器
应用漏洞
三方组件
统计分析
报告管理
系统管理
添加SecPoint
yucl
应用漏洞
验证已确认
验证无效
外部跟踪
筛选
清除筛选
刷新
分享
已选择 10 项
全部
危害等级
漏洞概况
检测数
最近检测时间
状态
操作
会话Cookie中缺少HttpOnly属性 [key: ben-2065]
• ContextPath: /benchmark
• Parameter: SomeCookie
• Code: o.o.b.t.BenchmarkTest02709.doPost()
1
2021/09/03 15:22:59
已发现
会话Cookie中缺少secure属性 [key: ben-2064]
• ContextPath: /benchmark
• Parameter: SomeCookie
• Code: o.o.b.t.BenchmarkTest02709.doPost()
1
2021/09/03 15:22:59
已发现
SQL注入 [key: ben-2063]
• Url: /benchmark/sqli-06/BenchmarkTest02653
• Parameter: QueryString
• Code: o.o.b.t.BenchmarkTest02653.doPost()
1
2021/09/03 15:22:59
已发现
SQL注入 [key: ben-2062]
• Url: /benchmark/sqli-05/BenchmarkTest02632
• Parameter: QueryString
• Code: o.o.b.t.BenchmarkTest02632.doPost()
1
2021/09/03 15:22:59
已发现
反射型XSS [key: ben-2061]
• Url: /benchmark/xss-05/BenchmarkTest02582
• Parameter: QueryString
• Code: o.o.b.t.BenchmarkTest02582.doPost()
1
2021/09/03 15:22:58
已发现
路径遍历 [key: ben-2060]
• Url: /benchmark/pathtraver-03/BenchmarkTest02562
• Parameter: QueryString
• Code: o.o.b.t.BenchmarkTest02562.doPost()
1
2021/09/03 15:22:58
已发现
会话Cookie中缺少HttpOnly属性 [key: ben-2059]

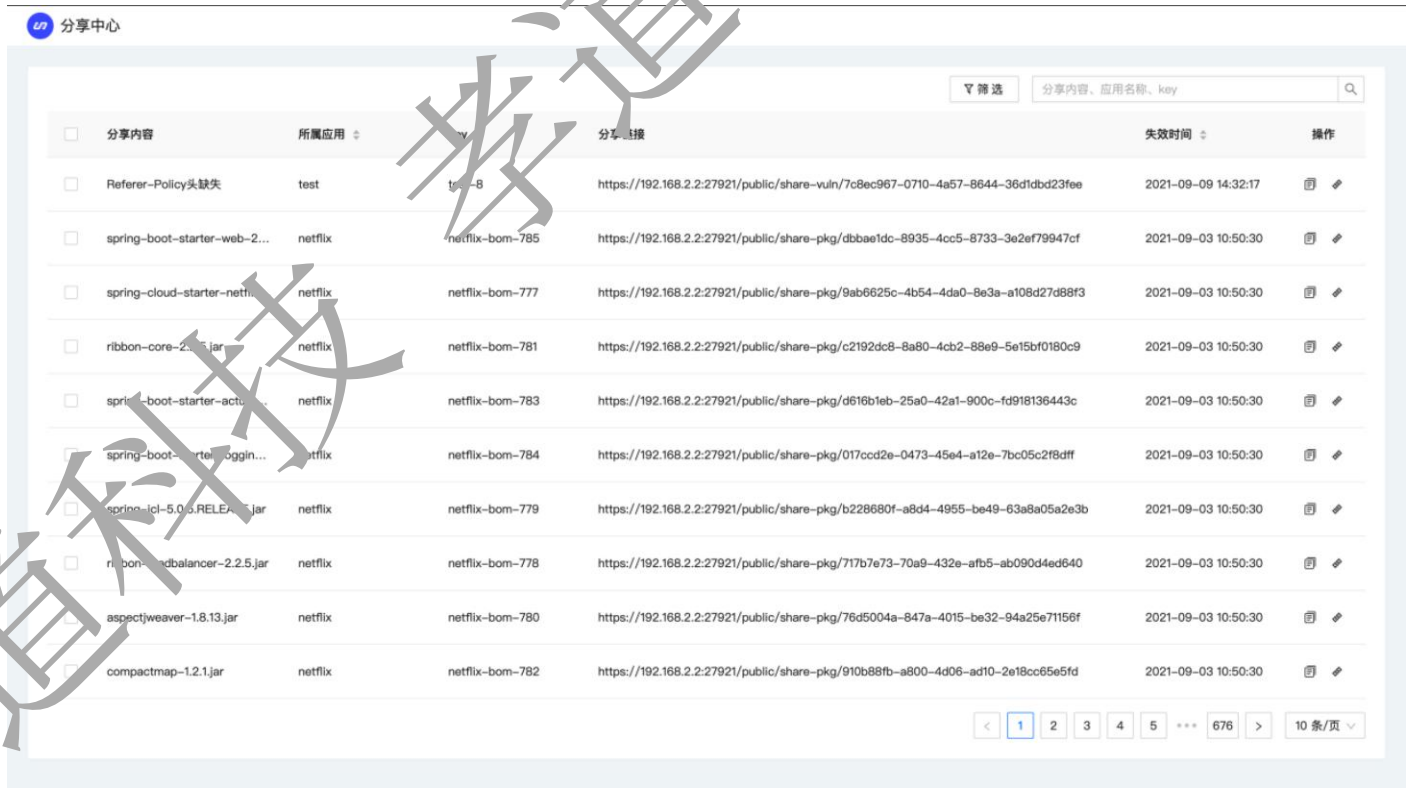
单个分享：


应用
服务器
应用漏洞
三方组件
统计分析
报告管理
系统管理
添加SecPoint
yucl
应用漏洞
验证已确认
验证无效
外部跟踪
筛选
清除筛选
刷新
危害等级
漏洞概况
检测数
最近检测时间
状态
操作
会话Cookie中缺少HttpOnly属性 [key: ben-2065]
• ContextPath: /benchmark
• Parameter: SomeCookie
• Code: o.o.b.t.BenchmarkTest02709.doPost()
1
2021/09/03 15:22:59
操作
状态
标签
下载
删除
分享
跟踪
创建问题
问题 (0)
会话Cookie中缺少secure属性 [key: ben-2064]
• ContextPath: /benchmark
• Parameter: SomeCookie
• Code: o.o.b.t.BenchmarkTest02709.doPost()
1
2021/09/03 15:22:59
SQL注入 [key: ben-2063]
• Url: /benchmark/sqli-06/BenchmarkTest02653
• Parameter: QueryString
• Code: o.o.b.t.BenchmarkTest02653.doPost()
1
2021/09/03 15:22:59
SQL注入 [key: ben-2062]
• Url: /benchmark/sqli-05/BenchmarkTest02632
• Parameter: QueryString
• Code: o.o.b.t.BenchmarkTest02632.doPost()
1
2021/09/03 15:22:59
已发现
反射型XSS [key: ben-2061]
• Url: /benchmark/xss-05/BenchmarkTest02582
• Parameter: QueryString
• Code: o.o.b.t.BenchmarkTest02582.doPost()
1
2021/09/03 15:22:58
已发现
路径遍历 [key: ben-2060]
• Url: /benchmark/pathtraver-03/BenchmarkTest02562
• Parameter: QueryString
• Code: o.o.b.t.BenchmarkTest02562.doPost()
1
2021/09/03 15:22:58
已发现
会话Cookie中缺少HttpOnly属性 [key: ben-2059]
• Parameter: rememberMe02498
• Code: o.o.b.t.BenchmarkTest02498.doPost()
1
2021/09/03 15:22:53
已发现



11.3.2 分享中心

分享中心可以管理已经进行分享的三方组件和应用漏洞，支持批量取消分享或者导出分享表格，单条记录可以进行取消分享和复制连接



11.4 事件管理

11.4.1 小组规则

小组规则可针对不同的事件建立触发规则，IAST 平台主要有发现新漏洞、一段时间内未再发现漏洞、上报系统日志三种事件，每种事件都会触发 IAST 平台主动进行不同的操作。

以发现新漏洞为例，如下图配置，每当 dubbo 应用产生新的 sql 注入漏洞且未进行验证，如果 IAST 平台配置了邮件服务器，就会发邮件到收件人账户。

触发动作选择创建跟踪管理，如果 IAST 平台配置了 JIRA 连接，那么每当 dubbo 应用产生新的 sql 注入漏洞且未进行验证，就会在 JIRA 平台创建一个漏洞问题。

11.4.2 全局规则

全局规则是管理员添加的对多个组/多个应用起作用的规则（全局规则添加之后直接生效），只有管理员才能查看编辑全局规则。

11.4.3 模板规则

模板规则本身不触发动作，而是通过将模板规则自动复制到新建小组起作用，非管理员能编辑/删除复制到自己组内的模板规则，若您希望新建小组都有某些规则，则可以添加模板规则。

11.5 大屏展示

IAST 数据分析大屏

通过 IAST 数据分析大屏，您可以实时的查看到应用漏洞数、有漏洞的应用数等信息，也可以查看到应用新增漏洞的趋势。

此外，通过右上角的数据筛选，您可以分别查看不同应用的数据信息或者不同时间段的应用相关信息。

11.6 策略管理

11.6.1 检测规则

某一检测规则开启之后，系统就会对该种规则类型的漏洞进行检测，并将检测到的漏洞展示在漏洞列表中。

管理员设置的检测规则影响小组的默认检测规则，小组的默认检测规则影响应用的默认检测规则（受时间线影响）。

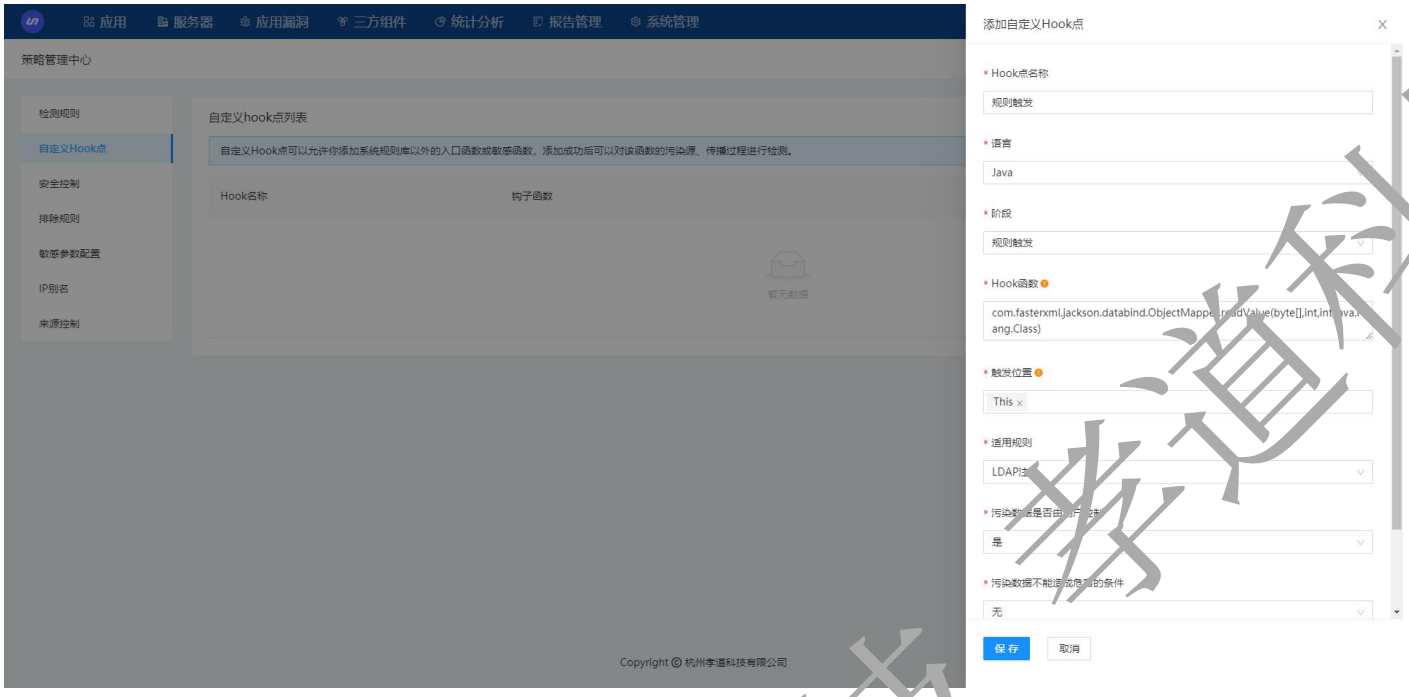
例如系统一共能检测 49 种漏洞，一开始管理员在策略管理处设置开启 43 种规则。之后管理员建立了 3 个小组，则这 3 个小组默认开启 43 种检测规则。小组 1 的小组管理员此时点击策略管理，看

到的情况是开启 43 种检测规则。此时小组 1 创建了 3 个应用，则这 3 个应用默认开启 43 种检测规则。如果此时，小组 1 的小组管理员修改了检测规则为 46 种，并且又创建了 2 个应用。则原有的 3 个应用的默认检测规则不变，之后添加的 2 个应用的默认检测规则为 46 种。

11.6.1.1 自定义检测规则

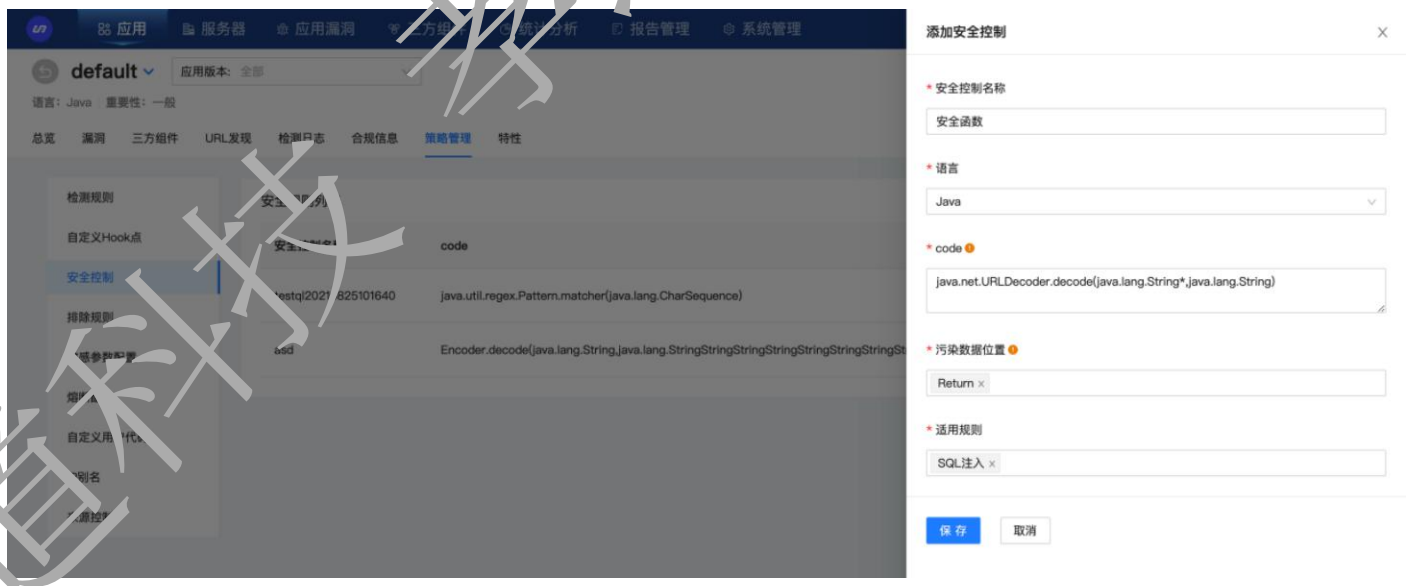
自定义添加检测规则主要为了增加对用户场景的适配性，在 IAST 自带的漏洞规则无法满足用户场景时，可通过自定义添加检测规则搭配自定义 Hook 点进行覆盖。

检测规则添加完之后，需要添加自定义 hook 点，在触发阶段关联上新增的漏洞类型

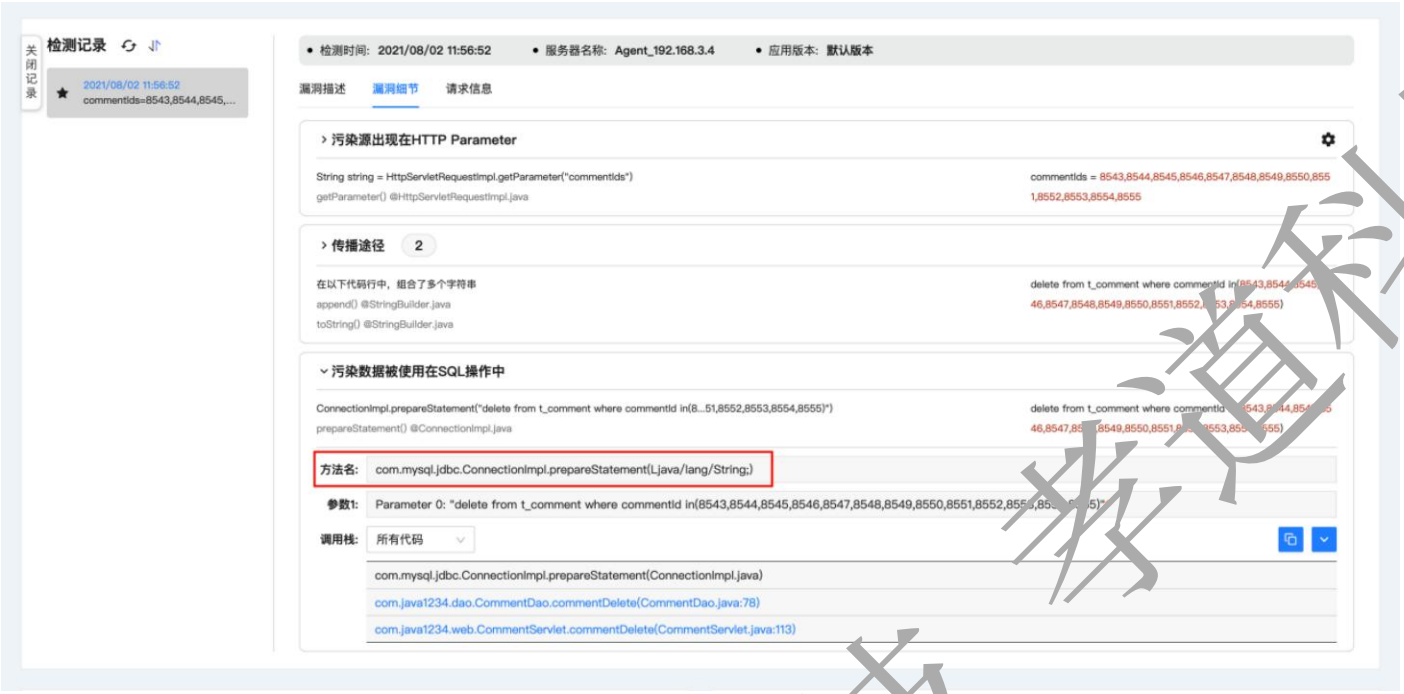


11.6.3 安全控制

安全控制是允许你添加某些被认为能够保证数据安全的函数。比如你使用 java.net.URLDecoder.decode 进行 sql 注入的数据消毒，你可以添加 java.net.URLDecoder.decode (java.lang.String*,java.lang.String)，*表示参数位置，再次检测到 SQL 注入漏洞，会将其标记为“没问题，内部安全控制”状态。



如下图所示，code 是漏洞细节中的方法名。



11.6.4 排除规则

排除规则主要是用于对应用进行配置，配置哪些规则是不进行检测的（比如可以配置某些 uri 白名单），排除 uri 下该种类型的检测规则。可以通过以下三种方式来添加排除规则 ：

11.6.4.1 排除类型为输入

当排除类型为 输入 的时候，输入类型的选项有 Parameter，Header，Query String，Body，Cookie。

当输入类型为 Parameter， Header 和 Cookie 时，需要填写输入参数，输入参数可以使用通配符*，如 aaa* 则表示会匹配 aaa* （如匹配 aaa234,aaahh 这样的参数）。

11.6.4.2 排除类型为 path

可以输入单个 path 或者多个 path，多个 path 需要用换行分开，如

/News/user

/News/add

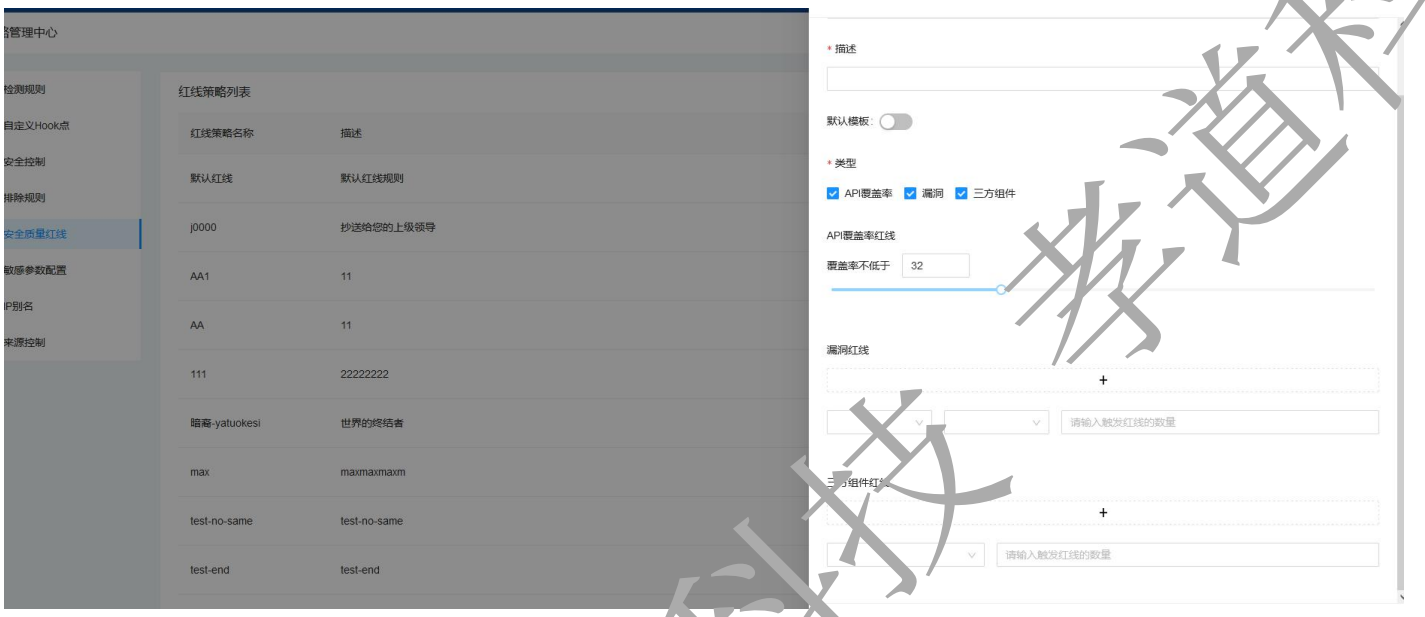
也可以使用*，如/News/*表示以下路径的漏洞都不进行检测

11.6.4.3 排除类型为 package

排除类型为 package 时，你可以填入 com.tcsec.test，此时 IAST 和 RASP 将不会检测 com.tcsec.test 包下所有代码，此配置需重启应用才能生效。

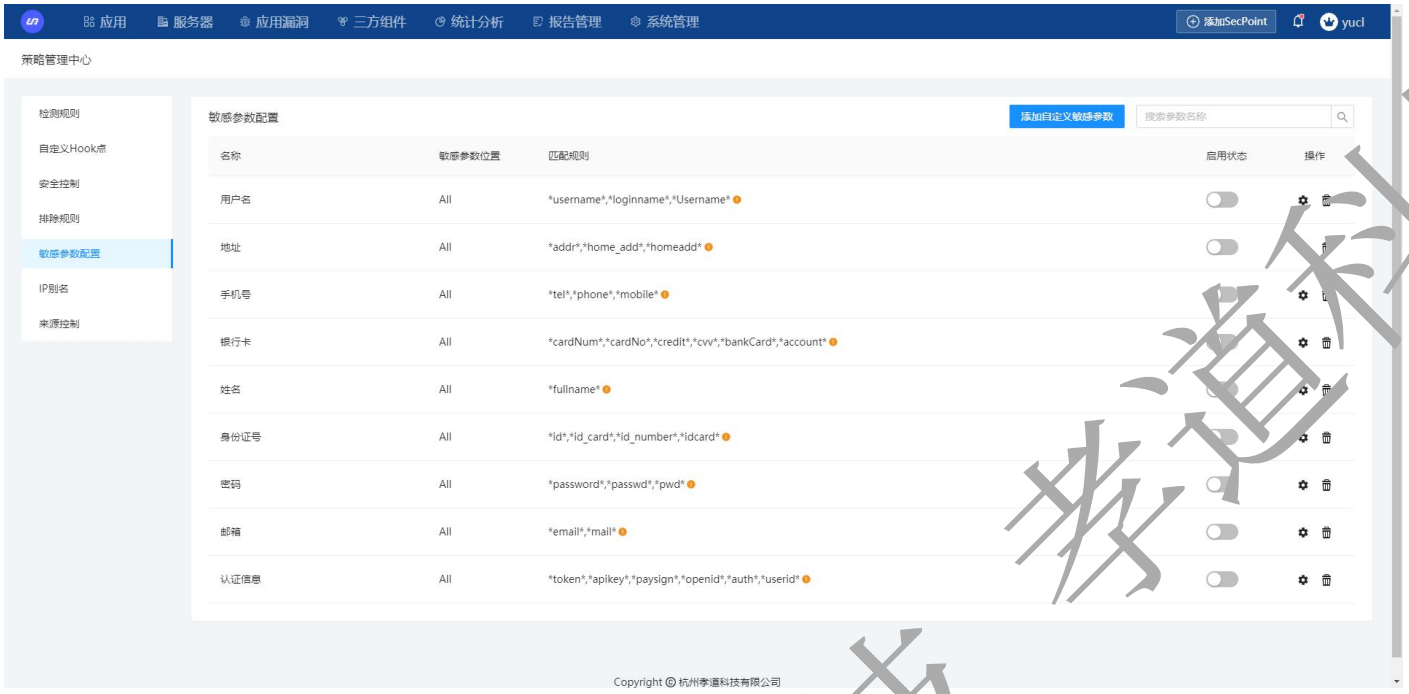
11.6.5 安全质量红线

红线策略可以应用到应用中用来做质量评判标准，可通过配置 API 覆盖率，漏洞红线以及三方组件红线做模板标准



11.6.6 敏感参数配置

系统级敏感参数配置，可以应用到所有应用的隐私合规的相关漏洞的检测范围，将一些敏感的参数（如：密码，手机号等）添加到列表，并且赋予对应的匹配表达式，启用该项内容时，SecPoint 将会对该参数进行合规检测。系统默认配置了常规的敏感参数，并处于关闭状态，用户需自行开启。



11.6.7IP 别名

IP 别名可用于来源控制及 IP 管理中，一个 IP 别名中可以添加一个 IP 或多个 IP，具体的 IP 类型可以选择主机、范围及网段。

11.6.8 源控制

当开启检测来源控制时：选择仅检测以下 IP，则仅检测列出的 IP 的请求，对其他 IP 的请求不进行检测；选择不检测以下 IP，则不检测列出的 IP 的请求，对其他 IP 进行检测。

11.7 WEB API

使用 IAST 平台提供的 API，需要 API 秘钥，API 秘钥在用户中心-API 设置中生成，是 APIKey 的值。

IAST 平台提供符合 HATEOS 标准的 RESTful API，可以通过重放 API 功能的响应信息提取到详细的漏洞信息或三放组件信息。具体字段代表的意义，可参考 API 文档字段说明。

如果需要在其它平台调用漏洞及三方组件 API，您可以查看具体接口文档规范，且可以通过重放 API 调试接口。

目前平台提供的 API 主要提供：基础业务、应用、服务器、三方组件几个大类。