ICS 35.240.50
L 67

DB33

浙　　江　　省　　地　　方　　标　　准

DB 33/ XXXXX—XXXX

基于智能动态污点跟踪的应用安全检测及漏洞管理技术规范

Technical specification of application security detection and vulnerability management based on intelligent dynamic stain Tracking

点击此处添加与国际标准一致性程度的标识

（征求意见稿）

－XX－XX 发布　　XXXX－XX－XX 实施

浙江省市场监督管理局　　发布

目　　次

前　言

本文件按照GB/T　1.1—2020《标准化工作导则　第1　部分：标准化文件的结构和起草规则》的规定起草。

本文件由浙江省公安厅提出并归口。

本文件主要起草单位：XXX。

本文件主要起草人：XXX、XXX、XXX。

本文件为首次发布。

本文件由XXX负责解释。

基于智能动态污点跟踪的应用安全检测及漏洞管理技术规范

1 范围

本文件规定了基于智能动态污点跟踪的Web应用安全检测能力要求、应用漏洞管理能力要求、安全管理中心安全功能要求、安全管理接口通用要求、安全建设验证要求等内容。

本文件适用于Web应用系统在开发上线前的应用安全测试阶段、Web应用系统安全评级和Web应用在线运行时自动化测试等过程。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适合于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 7408—2005 数据元和交换格式 信息交换 日期和时间表示法

GB/T 22239—2019 信息安全技术 网络安全等级保护基本要求

GB/T 25069—2010 信息安全技术 术语

GB/T 31506—2015 信息安全技术 政府门户网站系统安全技术指南

GB/T 32917—2016 信息安全技术 web应用防火墙安全技术要求与测试评价方法

GB/T 37931—2019 信息安全技术 web应用安全检测系统安全技术要求和测试评价方法

ISO/IEC 9075-1:2011 信息技术.数据库语言.结构化查询语言(SQL).第1部分:框架(Information technology. Database languages. SQL. Part 1:Framework (SQL/Framework))

ECMA-262 ECMAScript语言规范（ECMAScript Language Specification）

RFC 2616 超文本传输协议 HTTP 1.1（Hypertext Transfer Protocol HTTP 1.1）

RFC 3548 Base16、Base32和Base64数据编码（The Base16, Base32, and Base64 Data Encodings）

RFC 6229 流密码RC4的测试向量（Test Vectors for the Stream Cipher RC4）

RFC 8259 JavaScript对象符号(JSON)数据交换格式（The JavaScript Object Notation (JSON) Data Interchange Format）

3 术语和定义

GB/T 25069—2010、GB/T 32917—2016和GB/T 31506—2015界定的以及下列术语和定义适用于本文件。

3.1

Web应用系统 web application

一种通过Web访问的应用程序，由完成特定任务的各种Web组件构成，并通过Web将服务提供给用户。

注：常见的Web组件包括但不限于HTML文件和图像等静态文件，以及可通过执行获得执行结果的动态脚本文件，所

有这些组件相互协同为用户提供一组完整的服务。

3.2

交互式应用安全检测　interactive application security testing(IAST)

一种应用程序安全测试技术，通过在服务端部署Agent程序，能收集、监控Web应用程序运行时函数执行、数据传输，能高效、准确地识别安全缺陷及漏洞，同时能准确定位漏洞所在的代码文件、行数、函数及其参数。

3.3

安全管理中心　security management center

能为一个或多个具有交互式应用安全检测能力的Web应用系统远程提供Web安全检测策略制定、下发以及Web安全漏洞收集、智能综合分析的Web安全管理平台。

3.4

服务端请求伪造　server-side request forgery (SSRF)

由于Web应用系统的设计缺陷，使得攻击者可以通过篡改HTTP请求中的资源访问请求，诱骗Web服务器以自己的合法身份来访问攻击者以前无法访问的其它网络资源（如内网资源），从而达到攻击者的攻击目的。

3.5

安全连接密钥　secure connect key

一种用来实现安全管理中心安全地连接具有交互式应用安全检测功能的Web应用系统的随机字符串，该字符串由安全管理中心分配，通过安全的方式分发到所管理的Web应用系统，并由Web应用系统保存。Web应用系统在接收到来自安全管理中心的管理命令后，通过核对安全管理中心出示的安全连接密钥来验证安全管理中心是否具有远程管理权限。

注：安全管理中心在下发安全管理命令时，包含在安全管理命令中的安全连接密钥必须加密传输，以防止攻击者假冒安全管理中心身份非授权连接和管理Web应用系统。

3.6

会话签名密钥　session signature key

一种用来实现安全管理中心和具有交互式应用安全检测功能的Web应用系统之间通信过程中信息完整性的随机字符串，该会话签名密钥由安全管理中心生成，并通过一种安全方式下发到受管理的Web应用系统。安全管理中心在通过HTTP协议发送JSON格式的安全管理命令请求消息时，需要采用会话签名密钥对安全管理命令请求消息进行签名，Web应用系统则通过该会话签名密钥对接收到的安全管理命令请求消息中附加的签名进行验证。Web应用系统在向安全管理中心回复安全管理命令响应消息时，也同样需要采用该会话签名密钥进行签名以实现通信过程中的信息完整性。

注：为确保安全性，要求安全管理中心为各受管理的Web应用系统生成的会话签名密钥满足密码要求的随机性。

3.7

会话签名密钥　session signature key

一种用来实现安全管理中心和具有交互式应用安全检测功能的Web应用系统之间通信过程中信息完整性的随机字符串，该会话签名密钥由安全管理中心生成，并通过一种安全方式下发到受管理的Web应用系统。安全管理中心在通过HTTP协议发送JSON格式的安全管理命令请求消息时，需要采用会话签名密

钥对安全管理命令请求消息进行签名，Web应用系统则通过该会话签名密钥对接收到的安全管理命令请求消息中附加的签名进行验证。Web应用系统在向安全管理中心回复安全管理命令响应消息时，也同样需要采用该会话签名密钥进行签名以实现通信过程中的信息完整性。

注：为确保安全性，要求安全管理中心为各受管理的Web应用系统生成的会话签名密钥满足密码要求的随机性。

3.8

检测白名单　detect whitelist

用来定义某个需要Web应用系统交互式安全检测模块特殊处理的Web页面，要求安全检测模块不要对该页面或该页面的某个输入参数进行安全检测，以减少计算开销。

注：检测白名单通常由用户自定义，用户可以选择某些内容不进行安全检测，从而减少计算开销。

3.9

程序插桩　program instrumentation

在保证被测程序原有逻辑完整性的基础上在程序中插入一些探针（又称为“探测仪”，本质上就是进行信息采集的代码段，可以是赋值语句或采集覆盖信息的函数调用），通过探针的执行并抛出程序运行的特征数据，通过对这些数据的分析，可以获得程序的控制流和数据流信息，进而得到逻辑覆盖等动态信息。

3.10

智能动态污点跟踪　smart dynamic stain tracking

是一种智能跟踪并动态分析污点信息在程序中流动的技术。在漏洞检测中，通过程序插桩获得程序的控制流盒数据流信息，并使用智能动态污点跟踪技术将危险数据标记为污点数据，然后通过智能跟踪与污点数据相关的信息的流向，动态分析程序漏洞。

4 缩略语

下列缩略语适用于本文件。

XSS：跨站脚本攻击(Cross Site Script)

HTTP：超文本传输协议(Hypertext Transfer Protocol)

CSRF：跨站请求伪造（Cross-Site Request Forgery）

SSRF：服务端请求伪造（Server-Side Request Forgery）

XML：可扩展标记语言（eXtend Mark Language）

XXE：XML外部实体（eXtent XML Entity）

JSON：JavaScript对象表示法(JavaScript Object Notation)

LDAP：轻量级目录访问协议(Light Directory Access Protocol)

XPATH：XML路径操作语言（XML Path Language）

OS：操作系统（Operating System）

MVC：模式视图控制器编程模式（Model View Controler）

API：应用编程接口（Application Programming Interface）

URI：统一资源指示器（Universal Resource Indicator）

SQL：结构查询语言（Structure Query Language）

WVP：Web虚拟补丁（Web Virtual Patch）

5 概述

5.1 基于智能动态污点跟踪的应用安全检测技术概述

Web应用层一直是网络犯罪盒网路安全事件的重灾区，网络攻击者擅于利用创新和技术进步发现新漏洞并躲避安全检测。传统应用安全检测技术对于复杂应用场景（如微服务架构场景、应用通信加密场景、请求防重放等场景）无法正常进行漏洞检测，且传统应用安全检测技术无法解决满足日益增长的产品迭代需求与应用安全之间的矛盾。

基于智能动态污点跟踪的应用安全检测技术是一种新型的技术，它通过插桩技术，把代理程序融入到应用系统中，利用智能动态污点跟踪在应用系统运行过程中完整监控数据在应用系统内部的传递与变化，绘制出数据流图，通过对这些数据流图的逻辑进行分析，进而检测应用的脆弱性。该技术基于污点跟踪的漏洞识别方式使它能够完全适应带验证标签、加密、防重放等应用安全环境，在检测的过程中不会产生脏数据，对应用系统本身和应用系统运行结果不产生影响。同时，基于智能动态污点跟踪的应用安全检测技术能在应用系统运行时监控内部数据流向，记录漏洞产生的整个过程，漏洞信息能定位到污点传播过程中涉及的函数及其参数，为漏洞的管理修复提供指导。

5.2 基于智能动态污点跟踪的应用安全检测整体框架

如图1所示，基于智能动态污点跟踪的应用安全检测整体框架包括两部分：第一部分为嵌入了安全检测功能模块的Web应用系统，它一般部署在企业Web应用域中；第二部分为安全管理中心，它一般部署在Web安全管理域或云计算数据中心。由内嵌在Web应用系统中的安全检测模块和安全管理中心共同组成应用安全测试整体框架，形成一个包括多个Web安全检测客户端和一个安全管理中心的客户机/服务器模式框架，实现全路径的漏洞检测功能。

内嵌于web应用服务器的安全检测功能模块基于智能动态污点跟踪实现漏洞检测，它通过插桩技术获取到用户提交的数据，并对这些数据进行标记和跟踪，识别数据在应用系统内部传递和变化的过程，直至数据被递交至可引发安全漏洞的函数，根据整个过程绘制出数据流图。通过对整个数据传播过程进行分析，判断出这一数据流是否能引发安全漏洞。在应用系统启动加载的过程中，对应用系统所组成的文件特征进行识别，进而与内建第三方组件库进行匹配，判断是否存在具有安全风险的第三方组件。

部署在安全管理域的安全管理中心是基于智能动态污点跟踪的应用安全检测框架的管理控制平台。它负责对各个应用的漏洞检测策略管理和漏洞生命周期的管理。在检测策略管理中，需要能够针对每一个应用能够定义检测策略的开启或关闭、自定义漏洞的危害等级等。在安全管理中心模块，需能够展示该应用已检测的URL列表、漏洞生命周期管理、应用系统安全评级等。实现应用系统漏洞状态多维度全方面的评价与展示。

基于智能动态污点跟踪的应用安全检测框架除了需要确定内嵌在各Web应用系统中的安全检测模块的安全功能要求外，还需要明确定义安全检测模块与安全管理中心之间的安全管理接口，实现安全管理中心的Web安全检测策略和规则的下发以及安全检测模块的漏洞上报等操作。

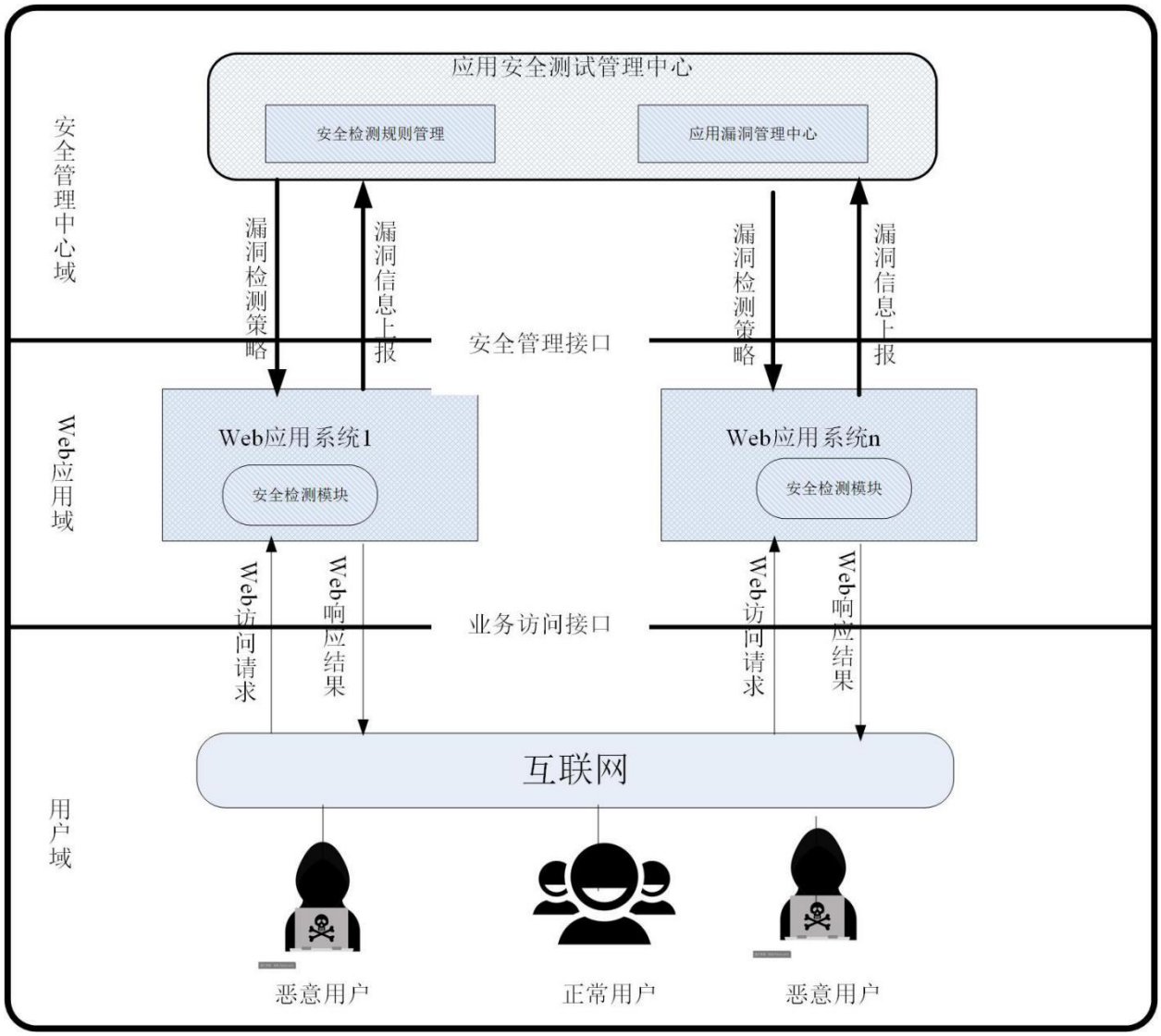

图 1　交互式应用程序安全检测框架

5.3　基于智能动态污点跟踪的应用漏洞管理技术概述

应用漏洞管理技术是指对基于智能动态污点跟踪的应用安全检测技术发现的漏洞进行管理，包括配置安全规则降低误报、针对漏洞创建分配任务、跟踪漏洞问题的各个周期等。基于智能动态污点跟踪的应用安全检测技术可以通过用户配置在智能动态污点跟踪的过程中新增桩点，配置相应的安全规则后，应用安全检测功能可以对用户定义的编码、过滤等类型的方法进行监控，进一步完善应用安全检测能力，使得应用安全检测策略与应用更加契合。

对于基于智能动态污点跟踪的应用安全检测技术发现的漏洞，安全管理中心可以对漏洞状态进行变更，包括已发现、已确认、可疑、没问题、已修复五个状态。针对每个漏洞，可创建对应的问题并分配给相关人员进行问题修复，同时，创建的问题可以同步到漏洞管理平台中进行管理。在这个过程中，安全管理中心会跟踪漏洞问题的全生命周期并进行实时同步。当应用进行迭代测试时，可创建不同的应用版本进行漏洞检测，通过对比不同应用版本间的漏洞差异区分出残余漏洞与新增漏洞。

6　安全检测能力要求

6.1 安全检测工作模式要求

应具有基于智能动态污点跟踪的应用安全检测功能的Web应用应：

a) 具有以下检测模式：
   1) 失效模式：当处于失效模式时，Web 应用系统的检测功能关闭，即 Web 应用系统不受安全检测系统监控；
   2) 检测模式：当处于检测模式时，Web 应用系统的安全检测功能模块开启，并依据安全管理中心下发的检测规则对漏洞进行识别与分析。
b) 接受来自安全管理中心的指令，可以在失效模式与检测模式之间切换。

6.2 安全漏洞检测能力要求

6.2.1 应用漏洞检测能力

6.2.1.1 OS 命令注入漏洞检测能力

应具备检测OS命令注入漏洞的能力。

6.2.1.2 HQL 注入漏洞检测能力

应具备检测HQL注入漏洞的能力。

6.2.1.3 SQL 注入漏洞检测能力

应具备支持ISO/IEC 9075-1中定义的ANSI-SQL的常见关系数据库管理 系统软件的SQL注入漏洞检测能力。

注：常见的支持ANSI-SQL的关系数据库管理软件包括MSSQL、Oracle、MySQL、db2、PostgreSQL等。

6.2.1.4 XPATH 注入漏洞检测能力

应具备检测XPATH注入漏洞的能力。

6.2.1.5 SMTP 注入漏洞检测能力

应具备检测SMTP注入漏洞的能力。

6.2.1.6 表达式注入漏洞检测能力

应具备检测JAVA语言中表达式注入漏洞的能力。

6.2.1.7 目录遍历漏洞检测能力

应具备检测目录遍历漏洞的能力。

6.2.1.8 XML 外部实体注入漏洞检测能力

应具备检测XML外部实体注入漏洞（XXE）的能力。

6.2.1.9 JAVA 反射注入漏洞检测能力

应具备检测JAVA语言中JAVA反射注入漏洞的能力。

6.2.1.10 LDAP 注入漏洞检测能力

应具备检测LDAP注入漏洞的能力。

6.2.1.11 NoSQL 注入漏洞检测能力

应具备检测NoSQL注入漏洞的能力。

6.2.1.12 不安全的反序列化漏洞检测能力

应具备检测JAVA语言中的不安全的反序列化漏洞的能力。

6.2.1.13 CSRF 漏洞检测能力

应具备检测CSRF漏洞的能力。

6.2.1.14 SSRF 漏洞检测能力

应具备检测SSRF漏洞的能力。

6.2.1.15 URL 跳转漏洞检测能力

应具备检测URL跳转漏洞的能力，包括：

a) Forward 类跳转；
b) Redirect 类跳转。

6.2.1.16 Cookie 敏感信息泄露漏洞检测能力

应具备检测Cookie敏感信息泄露漏洞的能力。

6.2.1.17 JSONP 劫持漏洞检测能力

应具备检测JSON劫持漏洞的能力。

6.2.1.18 XSS 漏洞检测能力

应具备检测XSS漏洞的能力。包括：

a) 反射型 XSS；
b) DOM 型 XSS。

6.2.1.19 使用弱加密算法漏洞检测能力

应具备检测弱加密算法漏洞的能力，包括：

a) MessageDigest 函数加密；
b) Cipher 函数加密。

6.2.1.20 非可信数据混入可信区域漏洞检测能力

应具备检测非可信数据混入可信区域漏洞的能力。

6.2.1.21 CRLF 注入漏洞检测能力

应具备检测CRLF注入漏洞的能力。

6.2.1.22 弱口令漏洞检测能力

应具备检测弱口令漏洞的能力，包括：

a) 登录弱口令；
b) 数据库连接弱口令。

6.2.1.23 不安全的 XMLDecoder 漏洞检测能力

应具备检测不安全的XMLDecoder漏洞的能力。

6.2.1.24 硬编码漏洞检测能力

应具备检测硬编码漏洞的能力，包括：
a) 硬编码密码；
b) 硬编码密钥。

6.2.1.25 readline 拒绝服务攻击漏洞检测能力

应具备检测readline拒绝服务攻击漏洞的能力。

6.2.1.26 正则表达式拒绝服务攻击漏洞检测能力

应具备检测正则表达式拒绝服务漏洞的能力。

6.2.1.27 Spring 自动绑定漏洞检测能力

应具备检测Spring自动绑定漏洞的能力。

6.2.1.28 不安全的 HTTP 方法漏洞检测能力

应具备检测不安全的HTTP方法漏洞的能力，包括以下不安全的HTTP方法：
a) OPTIONS；
b) HEAD。

6.2.1.29 不安全的 JSP 访问漏洞检测能力

应具备检测不安全的JSP访问漏洞的能力。

6.2.1.30 Session 重写漏洞检测能力

应具备检测Session重写漏洞的能力。

6.2.1.31 Session 超时时间配置不当漏洞检测能力

应具备检测Session超时时间配置不当漏洞的能力。

6.2.1.32 Web 服务器版本泄露漏洞检测能力

应具备检测Web服务器版本泄露漏洞的能力。

6.2.1.33 不充分的 SSL 连接漏洞检测能力

应具备检测不充分的SSL连接漏洞的能力。

6.2.1.34 使用弱随机数漏洞检测能力

应具备检测使用弱随机数漏洞的能力。

6.2.1.35 服务器请求不安全的资源漏洞检测能力

应具备检测服务器请求不安全的资源漏洞的能力。

6.2.1.36 脆弱的 SSL 加密算法漏洞检测能力

应具备检测脆弱的SSL加密算法漏洞的能力。

6.2.1.37 不安全的认证漏洞检测能力

应具备检测不安全的认证漏洞的能力。

6.2.1.38 日志注入漏洞检测能力

应具备检测日志注入漏洞的能力。

6.2.1.39 Autocomplete 配置不当漏洞检测能力

应具备检测Autocomplete配置不当漏洞的能力。

6.2.1.40 不安全登出漏洞检测能力

应具备检测不安全登出漏洞的能力。

6.2.1.41 HTTP 报文安全头缺失漏洞检测能力

应具备检测HTTP报文安全头缺失漏洞的能力，包括：

a) Cookie 的 HttpOnly 属性；
b) Cookie 的 Secure 属性；
c) Content-Type 头；
d) Content-Security-Policy 头；
e) Public-Key-Pins-Report-Only 头；
f) X-Xss-Protection 头；
g) Referer-Policy 头；
h) X-Frame-Options 头。

6.2.1.42 HTTP 报文安全头配置不当漏洞检测能力

应具备检测HTTP报文安全头配置不当漏洞的能力，包括：

a) X-Content-Type-Options 头；
b) X-Xss-Protection 头；
c) Access-Control-Allow-Origin 头；
d) Content-Security-Policy 头。

6.2.1.43 数据库密码以文本文件保存漏洞检测能力

应具备检测数据库密码以文本文件保存漏洞的能力。

6.2.1.44 Crossdomain.xml 配置不当漏洞检测能力

应具备检测Crossdomainn.xml配置不当漏洞的能力。

6.2.2 第三方组件安全检测能力

能基于程序片段特征分析应用系统构成，能识别已被确认存在安全风险（包括应用漏洞和许可风险）的组件。第三方组件安全漏洞数据库需包含CNNVD 标准/CVE 标准，并能定期保持同步。

6.3 安全管理接口技术要求

应提供安全管理接口，供安全管理中心调用，并满足以下要求：

a) 实现的安全管理接口可以对调用者进行身份认证，只有通过身份认证的安全管理中心才允许通过该安全管理接口实现对 Web 应用系统中内嵌安全防护模块的安全管理；

b) 安全管理中心和Web应用系统中内嵌安全防护模块之间通过安全管理接口传输Web安全防护策略或回传 Web 安全日志时，需要确保通信安全和数据安全，防止重放、窃听和篡改攻击。

6.4 开发语言环境技术要求

其内嵌的Web安全检测功能模块应支持以下Web开发语言环境：

a) JAVA；

b) ASP.net；

c) PHP。

7 安全管理中心安全功能要求

7.1 安全检测功能要求

7.1.1 应用管理功能要求

应用是安全管理中心最小管理单位，一个应用可以由一个或多个服务器组成。可针对于应用配置漏洞检测策略，也可根据应用对系统的安全状态进行评级或评分。

7.1.2 服务器管理功能要求

安全管理中心能需够识别每个安装客户端的应用系统服务器运行状态，并能够将服务器绑定至应用。能够对服务器设置日志等级、日志路径及并发检测等。

7.1.3 应用安全评分功能要求

应用安全评分应能够根据被测试应用系统的应用安全漏洞的严重等级、数量以及应用中所引用的存在漏洞的第三方开源组件的情况进行综合评分，使得安全管理员可以直观地掌握应用的状态。

应能够根据安全评分来实行应用安全安全评级，对应用安全进行统一监管与评价，促进整体应用安全的水平提升。

通过总览页面，应可以从安全评分、风险统计、应用漏洞趋势、应用漏洞状态统计全览应用系统的漏洞概况。

7.1.4 漏洞主动验证功能要求

安全管理中心应具备漏洞主动验证的管理功能，以提高漏洞检测结果的精确性。漏洞主动验证功能开启后，漏洞状态应注明漏洞验证状态，状态包括以下几类：

a) 通过验证：通过发送 payload，确认漏洞真实存在，且能被利用；

b) 验证失败：无法通过发送 payload 确定漏洞存在，需人工验证；

7.1.5 策略管理功能要求

7.1.5.1 检测规则

可配置检测规则的开关，能开启或者关闭对应漏洞类型的安全检测。

7.1.5.2 安全控制

可通过配置安全控制策略来提升漏洞检测结果的精确性。安全控制策略是允许添加某些被认为能够保证数据安全的函数或者方法，这些增加的函数或者方法将补充到智能动态污点跟踪过程中，不断完善安全检测能力，使得检测规则更加契合被检测应用。

7.1.5.3 来源控制

可配置来源控制策略来限制安全检测功能检测访问应用系统的请求。

7.1.5.4 排除规则

可配置排除规则来限制安全检测功能检测的 URL、HTTP 参数或者 JAR 包。

7.1.5.5 自定义用户代码

可配置自定义用户代码策略来精准识别用户代码和组件代码。

7.1.6 系统管理

安全管理中心应具有系统信息查看功能，并可对系统配置、升级管理、备份恢复、系统日志等重要功能进行操作。具体包括以下方面：

a) 系统信息：展示应用安全管理中心的服务器信息；
b) 系统配置：对应用安全管理中心进行磁盘清理、邮件告警、LDAP 等配置，并可根据用户的使用情况对漏洞状态进行自定义；
c) 升级管理：用于检测规则的更新升级；
d) 备份恢复：对系统的数据进行备份防止丢失，还原备份的数据；
e) 系统日志：记录应用安全管理中心服务器的安全日志，可查看系统异常情况。

7.2 安全管理功能要求

7.2.1 安全策略管理功能

安全管理中心应支持定制Web安全管理策略，并通过Web端提供的远程安全管理接口下发，实现对应用安全检测功能模块的开关和对各Web安全功能模块配置管理。

7.2.2 Web 日志收集功能

安全管理中心应支持从安全检测端处获取错误日志管理接口和获取Web端日志，实现日志的汇聚和集中管理分析。

7.2.3 检测管理功能

安全管理中心应支持对所控制的安全检测端进行各种管理操作，包括启动、暂停和终止安全检测功能。

7.2.4 Web 安全审计功能

安全管理中心应具备对安全检测端和管理中心的安全审计功能，实现对所控制的安全检测端管理操作的安全审计，以及对安全管理中心的各种重要操作的安全审计。

7.3 漏洞管理功能要求

7.3.1 漏洞状态管理功能

安全管理中心能够对发现的漏洞进行漏洞状态管理。漏洞状态包括：

a) 已发现；
b) 已确认；
c) 可疑；
d) 没问题；
e) 已修复。

当漏洞状态更改时，当前用户可以对当前漏洞状态进行备注或者评论，其他有权限访问该漏洞的安全管理中用户同样能够对漏洞状态进行备注，有助于漏洞的后期管理修复。

7.3.2 自定义 HOOK 点功能

安全管理中心能够配置自定义的 HOOK 点。基于智能动态污点跟踪的应用安全检测技术可以通过用户配置的 HOOK 点在智能动态污点跟踪的过程中增加自定义的插桩点，配置相应的安全检测规则后，进一步完善应用安全检测能力，使得应用安全检测策略与应用更加契合。

7.3.3 漏洞问题跟踪功能

安全管理中心能针对特定漏洞进行问题创建，并可将具体问题分配至相关人员。漏洞创建的问题能同步到漏洞管理平台中（如 JIRA 平台），同时，和漏洞管理平台的问题状态能双向同步管理。

8 安全管理接口通用要求

8.1 安全管理接口通信协议要求

安全管理中心与Web应用系统之间的安全管理接口应满足如下通信协议要求：

a) 安全管理中心和 Web 应用系统之间的安全管理命令请求和响应消息格式应满足 JSON 格式要求（ECMA-262）；
b) 安全管理中心和 Web 应用系统之间的安全管理命令请求和响应消息应分别封装在 HTTP 协议（RFC2616）请求消息和响应消息中进行传输；
c) 在将 JSON 格式的安全管理命令请求和响应消息封装到 HTTP 协议前，应对其进行加密和编码处理，以确保安全管理命令请求和响应消息的安全性。

8.2 安全连接密钥管理要求

应允许来自安全管理中心的安全管理连接请求，并采用安全连接密钥来验证安全连接的有效性，具体应满足如下要求：

a) 应在 Web 安全防护模块安装前或安装过程中，由安全管理中心为受保护的 Web 应用系统生成一个安全连接密钥，并通过安全方式（如邮件、短信或人工方式）传输到 Web 应用系统，并由双方安全保存，用来验证双方安全连接的有效性；
b) 安全管理中心在发送 JSON 格式的安全管理命令请求消息时，必须在消息中附加上与所管理的 Web 应用系统匹配的安全管理密钥，Web 应用系统应基于本地安全保存的安全连接密钥对安全

管理中心发送的 JSON 格式安全管理命令请求消息进行验证，只有安全连接密钥匹配成功，才真正执行安全管理中心下发的安全管理命令。

8.3 会话密钥安全管理要求

8.3.1 会话加密密钥安全管理

安全管理中心和Web应用系统在将JSON格式的安全管理命令请求和响应消息封装到HTTP请求和响应消息之前，应能采用会话加密密钥对安全管理命令请求和响应消息进行数据加密和解密，并满足以下要求：

a) 用来解密数据的会话加密密钥由安全管理中心生成，并通过安全方式下发到 Web 应用系统，由 Web 应用系统安全保存；
b) 对从安全管理中心接收到的加密的安全管理命令请求消息，Web 应用系统应能够使用先前获取和存储的会话加密密钥进行数据解密处理；
c) 对需要返回到安全管理中心的安全管理命令响应消息，Web 应用系统应能够对安全管理命令响应消息采用会话加密密钥进行加密处理，确保数据的保密性。

8.3.2 会话签名密钥安全管理

安全管理中心和Web应用系统在将JSON格式的安全管理命令请求和响应消息封装到HTTP请求和响应消息之前，宜采用会话签名密钥对安全管理命令请求和响应消息进行数据签名，以确保消息的完整性，并满足以下要求：

a) 用来签名数据的会话签名密钥由安全管理中心生成，并通过安全方式下发到 Web 应用系统，由 Web 应用系统安全保存；
b) 对从安全管理中心接收到的附加了签名数据的安全管理命令请求消息，Web 应用系统应能够使用先前获取和存储的会话签名密钥进行签名验证处理，以验证消息的完整性；
c) 对需要返回到安全管理中心的安全管理命令响应消息，Web 应用系统宜采用会话签名密钥对其进行签名处理，以确保响应消息的保密性。

8.4 安全管理命令消息解封和封装要求

Web应用系统应支持对安全管理中心发送的安全管理命令请求消息进行安全解封，以及对将要发送到安全管理中心的安全管理请求响应消息进行安全封装，具体要求如下：

a) 对于接收到的安全管理命令请求消息，Web 应用系统首先采用 base64(RFC3548)解码，然后获取本地存储的会话加密密钥，采用双方协商的加密算法对消息进行解密，得到 json 格式的安全管理命令请求消息；
b) 如果解密后的 JSON 格式的安全管理命令请求消息中包含了签名数据，Web 应用系统应获取本地存储的会话签名密钥，采用双方协商的签名算法对签名进行验证，如果验证失败，则直接丢弃该安全管理命令；
c) Web 应用系统应从消息中抽取安全连接密钥，并和本地的安全连接密钥进行比较，只有比对成功才能执行安全管理命令，否则拒绝执行；
d) 对于 Web 应用系统需要发送的安全管理命令响应消息，如果 Web 应用系统收到了安全管理中心下发的会话签名密钥，则应采用存储于本地的会话签名密钥对消息进行签名，然后对签名数据进行 base64 编码，并将编码后的签名数据封装在 HTTP 响应消息中。
e) 对于 Web 应用系统需要发送的安全管理命令响应消息，应采用存储于本地的会话加密密钥对消息进行加密，然后对加密后的数据进行 base64 编码，然后封装在 HTTP 响应消息中。

9 安全建设验证要求

9.1 源代码安全审计要求

Web应用系统上线前，应对其进行安全审计，避免源代码中存在的脆弱性问题。

9.2 安全功能测试要求

9.2.1 测试覆盖

应对Web应用系统编制测试覆盖文档，技术要求如下：

a) 表明测试文档中所标识的测试与功能规范中所描述的 Web 应用系统安全功能间的对应性；
b) 表明上述对应性是完备的，并证实功能规范中的所有安全功能接口都进行了测试。

9.2.2 测试深度

应对Web应用系统进行测试深度分析，技术要求如下：

a) 证实测试文档中的测试与 Web 应用系统设计中的安全功能子系统和实现模块之间的一致性；
b) 证实 Web 应用系统设计中的所有安全功能子系统、实现模块都已经进行过测试。

9.2.3 功能测试

应测试Web应用系统的安全功能，将结果文档化并提供测试文档。测试文档应包括以下内容：

a) 测试计划，标识要执行的测试，并描述执行每个测试的方案，这些方案包括对于其它测试结果的任何顺序依赖性；
b) 预期的测试结果，表明测试成功后的预期输出；
c) 实际测试结果和预期的测试结果一致。

9.2.4 独立测试

应提供一组与其自测安全功能时使用的同等资源，以用于安全功能的抽样测试。

9.2.5 脆弱性评定

基于已标识的潜在脆弱性，Web应用系统能够抵抗较强的攻击。

附 录 A
（资料性）
安全管理中心和 agent 之间的接口技术规范

A.1 通用接口规范定义

A.1.1 通信协议定义

agent和安全管理中心使用https和wss（websocket+ssl）协议通信，https和ssl可以保证通信的安全，https协议一般用于agent内核的下载和认证，wss协议一般用于业务数据的传输

A.1.2 通用认证流程定义

不同的接口会有不同的验证方式，针对不同的情况，安全管理中心会提供三种认证方式。
authentication token认证：用于注册前还未拿到api token时的认证。
register token认证:用于注册接口的认证,该接口调用成功后,管理中心会返回一个api token。
api token认证：用于注册后的业务数据传输和websocket链接认证等接口的认证。

A.1.3 通用请求头定义

A.1.3.1 https请求头定义

本请求头定义适用于所有https接口，定义的参数全部用于https的header部分，详细参数定义见表A.1

表 A.1 https 请求头定义参数表

参数名	必要性	参数类型	说明
token	必须	String	不同的接口使用不同的token，如果是注册前下载内核则用authentication token，如果是注册接口则用register token，如果是注册后的业务数据传输和websocket链接认证等接口则用api token。
app_key	必须	String	应用标示，用于标示agent属于哪一个应用

A.1.3.2 wss请求头定义

本请求头定义适用于所有websocket接口，websocket接口传输二进制数据，其中传输数据的前两个字节为header部分，后续的数据为body部分，header部分的详细参数定义见表A.2

表 A.2 weboskcet header 定义参数表

参数名	必要性	参数类型	说明
data_type	必须	value	数据类型，表明body里的数据是哪个类型的数据，该字节的值的选项说明详见数据类型表A.3
data_flag	必须	bit	数据标示，用于标示数据的特征，该字节有8个位，标示了8种不

			同的数据特征，详见表A.4

表 A.3　weboskcet　header 定义参数表

值可选项	说明
1	标示着body里的数据为三方组件上报数据
2	标示着body里的数据为agent信息上报数据
3	标示着body里的数据为漏洞上报数据
4	标示着body里的数据为日志上报数据

表 A.4　weboskcet　header 定义参数表

位偏移量	说明
0	保留
1	保留
2	保留
3	保留
4	保留
5	保留
6	保留
7	值为1时，说明数据是经过gzip压缩的，值为0值时，说明数据没有经过gzip压缩

A.1.4　通用数据格式定义

Agent和管理中心通过接口交换数据全部采用json序列化协议传输，下文中提到的报文的请求和应答报文格式如无特别说明，均只指的是http或websocket的body里json报文的格式。

A.1.5　通用应答码定义

本应答码定义适用于所有http接口，应答码均指的是http协议的应答报文的response　code，其详细内容说明如表A.5所示

表 A.5　通用应答码说明表

应答码	成功应答	说明
200	是	请求成功
400	否	请求报文格式错误
404	否	请求的处理过程中，某些资源找不到
409	否	权限验证失败
5xx	否	网络或服务器发生错误，请联系管理员

A.2　http接口详细规范定义

A.2.1　agent引擎下载

A.2.1.1 功能描述

agent引擎包含了所有agent核心功能包括漏洞扫描、污点跟踪、容器检测等，它是一个agent运行的必要文件，是agent的心脏。

A.2.1.2 请求报文格式定义

本接口的请求报文无参数。

A.2.1.3 应答报文格式定义

心跳接口的应答报文格式参数说明如表A.6所示

表 A.6 agent 引擎下载应答报文参数表

参数名	必要性	参数类型	说明
bytes	必须	String	agent引擎的内容

A.2.1.4 请求报文格式示例

```
{}
```

A.2.1.5 应答报文格式示例

```
{
"bytes":"FJDSKLAJFOIEWJFKLEWJFKDLSAJFIOQJEWGREHNGKJRHLEWFUEIWPQHFJDEAJDKSAFNDJKS"
}
```

A.2.2 agent注册接口规范

A.2.2.1 功能描述

注册接口是agent启动后必须最先向安全管理中心发起的，该接口主要用于通知安全管理中心agent上线了，同时获取一个用于后续其他接口验证身份的authentication token。

A.2.2.2 请求报文格式定义

本接口的请求报文无参数。

A.2.2.3 应答报文格式定义

注册接口的应答报文格式参数说明如表A.7所示

表 A.7 注册接口应答报文参数表

参数名	必要性	参数类型	说明
token	必须	String	api__token 用于其他接口验证权限

A.2.2.4 请求报文格式示例

```
{}
```

A.2.2.5 应答报文格式示例

{“token”:”FJKDLSAJOFIWENSAIFREWIQOHFEJKWOITHUIN”}

A.3 websocket接口详细规范定义

A.3.1 agent信息上报

A.3.1.1 功能描述

信息上报接口会上报一些主机信息、应用信息等数据。

A.3.1.2 请求报文格式定义

信息上报接口的请求报文格式参数说明如表A.8所示

表 A.8 信息上报接口请求报文参数表

参数名	必要性	参数类型	说明
LangType	必须	String	Agent语言类型，一般有Java，Php，dotnet等
LangVersion	必须	String	语言版本，比如java的1.8.0_220
AppAddr	必须	String	被检测应用的ip
AppPath	必须	String	被检测应用的路径
AppPackage	必须	String	被检测应用的包名或者命名空间
AppName	必须	String	被检测应用的应用名
ServerType	必须	String	被检测应用的容器类型
ServerVersion	必须	String	被检测应用的容器版本
AgentVersion	必须	String	Agent的版本
OsName	必须	String	被检测应用所在服务器的操作系统类型
OsVersion	必须	String	被检测应用所在服务器的操作系统版本
UserName	必须	String	被检测应用所属的用户名
UserHome	必须	String	被检测应用所属的用户的Home目录
UserDir	必须	String	被检测应用所属的用户的当前目录

A.3.1.3 应答报文格式定义

本接口的应答报文无参数。

A.3.1.4 请求报文格式示例

```
{
        "LangType":"Java",
        "LangVersion":"1.8.0_122",
        "AppAddr":"192.168.1.199",
        "AppPath":"/home/user/user_app",
        "AppPackage":"cn.com.example.user",
        "AppName":"user_service",
```

```
        "ServerType":"tomcat",
        "ServerVersion":"8.5.56",
        "AgentVersion":"1.14.2",
        "OsName":"centos",
        "OsVersion":"7.5",
        "UserName":"user",
        "UserHome":"/home/user",
        "UserDir":"/home/user/user_app/bin"
}
```


A.3.1.5 应答报文格式示例


```
{}
```


A.3.2 agent三方组件信息上报

A.3.2.1 功能描述

三方组件上报接口会上报一些三方组件的信息，包括md5、路径、名字、版本。

A.3.2.2 请求报文格式定义

三方组件上报接口的请求报文格式参数说明如表A.9所示

表 A.9 三方组件上报接口请求报文参数表

参数名	必要性	参数类型	说明
Components	必须	Component[]	Component的格式如表A.10所示

表 A.10 Component 参数表

参数名	必要性	参数类型	说明
Type	必须	String	三方组件的类型，如java的“jar”
Group	必须	String	三方组件的第一标识，如java的groupId
Name	必须	String	三方组件的第二标识，一般是组件的名字
Version	必须	String	三方组件的第三标识，一般是组件的版本号
Path	必须	String	三方组件在服务器中的绝对路径

A.3.2.3 应答报文格式定义

本接口的应答报文无参数。

A.3.2.4 请求报文格式示例


```
{
        "Components":[
                {
                        "Type":"Java",
```

```
                "Group":"org.apache.commons",
                "Name":"commons-lang3",
                "Version":"3.4",
                "Path":"/home/user/user_app/lib/commons-lang3-3.4.jar"
            },
            {
                "Type":"Java",
                "Group":"mysql",
                "Name":"mysql-connector-java",
                "Version":"5.1.6",

"Path":"/home/user/user_app/lib/mysql-connector-java-5.1.6.jar"
            }
        ]
    }
```


A.3.2.5 应答报文格式示例


```
    {}
```


A.3.3 agent漏洞上报

A.3.3.1 功能描述

漏洞上报接口主要用于上报漏洞数据。

A.3.3.2 请求报文格式定义

漏洞上报接口的请求报文格式参数说明如表A.11所示

表 A.11 漏洞上报接口请求报文参数表

参数名	必要性	参数类型	说明
VulnsList	必须	Vulns[]	漏洞数据列表，数据格式如表A.12所示
RequestType	可选	String	被检测应用的请求流量类型，如http，dubbo，none等
ContextPath	必须	String	J2ee里的ContextPath，缺省值“/”
Request	可选	Request	被检测应用的请求，数据格式如表A.14所示
Response	可选	Response	被检测应用的应答，数据格式如表A.16所示

表 A.12 Vulns 参数表

参数名	必要性	参数类型	说明
VulnsId	必须	String	漏洞Id
Processs	必须	Process[]	漏洞传播的节点列表，数据格式如表A.13所示

表 A.13 Process 文参数表

参数名	必要性	参数类型	说明
Type	必须	String	传播节点的类型(1-污染源节点,2-传播节点,3-漏洞执行节点)
Source	必须	String	标识污染源传入位置（0-方法所在对象本身，1-第一个参数，2-第二个参数，以此类推）
Target	必须	String	标识污染源传出位置（-1-方法返回值，0-方法所在对象本身，1-第一个参数，2-第二个参数，以此类推）
CodeSnippet	必须	String	该节点的方法的片段
ThisVlue	必须	String	方法所在对象的值
returnValue	必须	String	返回值的值
ParamValues	必须	String	参数的值
Stacks	必须	String	调用栈

表 A.14　Request 参数表

参数名	必要性	参数类型	说明
Method	可选	String	请求方法
Protocol	可选	String	请求的协议
QueryString	可选	String	请求的参数
Scheme	可选	String	请求的域
Host	可选	String	请求的主机ip
Uri	可选	String	请求的地址
RemoteIp	必须	String	远程地址
ContentType	可选	String	请求数据的类型
HeaderList	必须	Header[]	请求头列表，数据格式如表A.15所示
Body	必须	String	请求数据

表 A.15　Header 参数表

参数名	必要性	参数类型	说明
Key	必须	String	键值对的键名
Value	必须	String	键值对的值

表 A.16　Response 参数表

参数名	必要性	参数类型	说明
HeaderList	必须	Header[]	应答头列表，数据格式如表A.15所示
Status	必须	Int	应答码
Body	必须	String	应答报文数据

A.3.3.3　应答报文格式定义

本接口的应答报文无参数。

A.3.3.4 请求报文格式示例


```
    {{
            "VulnsList":[
                    {
                            "VulnsId":"201",
                            "Processs":[
                                    {
                                            "Type":"Rule",
                                            "Source":"1",
                                            "Target":"",

"CodeSnippet":"java.io.FileInputStream.<init>()",
                                            "ThisVlue":"FileInputStream@913c4",
                                            "ReturnValue":"",
                                            "ParamValues":"FileInputStream@1b0d3",

"Stacks":"java.io.FileInputStream.<init>(FileInputStream.java:123),org.owasp.benchmark.testcode.BenchmarkTest00001.doPost(BenchmarkTest00001.java:69),javax.servlet.http.HttpServlet.service(HttpServlet.java:660),javax.servlet.http.HttpServlet.service(HttpServlet.java:741),org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:231),org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166),org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:52),org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:193),org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166),org.owasp.benchmark.helpers.filters.DataBaseFilter.doFilter(DataBaseFilter.java:28),org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:193),org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166),org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:199),org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:96),org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.java:543),org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:139),org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:81),org.apache.catalina.valves.AbstractAccessLogValve.invoke(AbstractAccessLogValve.java:688),org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:87),org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:343),org.apache.coyote.http11.Http11Processor.service(Http11Processor.java:609),org.apache.coyote.AbstractProcessorLight.process(AbstractProcessorLight.java:65),org.apache.coyote.AbstractProtocol$ConnectionHandler.process(AbstractProtocol.java:818),org.apache.tomcat.util.net.NioEndpoint$SocketProcessor.doRun(NioEndpoint.java:1623),org.apache.tomcat.util.net.SocketProcessorBase.run(SocketProcessorBase.java:49),java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142),java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617),org.apache.tom
```

cat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:61),java.lang.Thread.run(Thread.java:745)"
                                        }
                                ]
                        }
                ],
                "RequestType":"http",
                "ContextPath":"/benchmark",
                "Request":{
                        "Method":"POST",
                        "Protocol":"HTTP/1.1",
                        "QueryString":"",
                        "Scheme":"http",
                        "Host":"localhost",
                        "Uri":"https://localhost:8443/benchmark/sqli-00/BenchmarkTest00100",
                        "RemoteIp":"127.0.0.1",
                        "ContentType":"application/x-www-form-urlencoded",
                        "HeaderList":[
                                {
                                        "Key":"Connection",
                                        "Value":"keep-alive"
                                }
                        ],
                        "Body":"Y21kPWlwY29uZmln"
                },
                "Response":{
                        "HeaderList":[
                                {
                                        "Key":"Cache-Control",
                                        "Value":"private"
                                }
                        ],
                        "Status":200,

"Body":"PGhlYWQgaWQ9ImN0bDAwX0hlYWQxIj48bGluayBocmVmPSJBcHBfVGhlbWVzL0RlZmF1bHQvMDAucmVzXQuY3NzIiB0eXBlPSJ0ZXh0L2NzcyIgcmVsPSJzdHlsZXNoZWV0IiAvPjxsaW5rIGhyZWY9IkFwcF9UaGVtZXMvRGVmYXVsdC8wMS45NjBfMjRfY29sLmNzcyIgdHlwZT0idGV4dC9jc3MiIHJlbD0ic3R5bGVzaGV1dCIgLz48bGluayBocmVmPSJBcHBfVGhlbWVzL0RlZmF1bHQvMDIudGV4dC5jc3MiHR5cGU9InRleHQvY3NzIiByZWw9InN0eWxlc2hlZXQiIC8"
                }
        }

A.3.3.5 应答报文格式示例

{}

A.3.4 agent日志收集上报

A.3.4.1 功能描述

日志收集上报接口主要用于上报自己的日志文件，以便分析和排查问题。

A.3.4.2 请求报文格式定义

日志收集接口的请求报文格式参数说明如表A.17所示

表A.17 日志收集接口请求报文参数表

参数名	必要性	参数类型	说明
bodys	必须	String	日志包的base64编码数据

A.3.4.3 应答报文格式定义

本接口的应答报文无参数。

A.3.4.4 请求报文格式示例

```
{
"bodys":"FJDSKLAJFOIEWJFKLEWJFKDLSAJFIOQJEWGREHNGKJRHLEWFUEIWPQHFJDEAJDKSAFNDJKS"
}
```

A.3.4.5 应答报文格式示例

{}

附 录 B
（资料性）
数据交换说明

B.1 内容详细定义规范

B.1.1 漏洞信息上报格式规范

漏洞信息上报格式提供了Agent检测到的漏洞的数据，其格式如表B.1：

表 B.1 漏洞信息上报格式

参数名	必要性	参数值	说明
key	必须	String	漏洞的唯一标示
vulId	必须	String	漏洞id
url	必须	String	请求url
location	必须	String	漏洞发生的位置
payload	必须	String	漏洞执行的payload
stages	必须	Stage[]	payload传播的详细过程，Stage的结构见表B.2

表 B.2 漏洞信息上报格式中的 Stage 的格式

参数名	必要性	参数值	说明
source	必须	String	源payload
target	必须	String	源payload
signature	必须	String	传播执行的方法

B.1.2 资产信息上报格式规范

资产信息上报格式提供了Agent的一些环境信息，其格式如表B.3：

表 B.3 资产信息上报格式

参数名	必要性	参数值	说明
Agent_id	必须	String	Agent的唯一id
app_name	必须	String	应用名称

app_version	必须	String	应用版本
app_location	必须	String	应用安装位置
lang_version	必须	String	应用的编程语言版本
lang_type	必须	String	应用的编程语言
user_name	必须	String	应用登录用户名称
user_home	必须	String	应用登录用户主目录
host_addr	必须	String	主机的ip地址
os_type	必须	String	操作系统类型
os_version	必须	String	操作系统版本
os_name	必须	String	操作系统名字
third_component	必须	Component[]	三方组件列表

表 B.4　资产信息上报中 Component 的格式

参数名	必要性	参数值	说明
Md5	必须	String	组件的md5值
path	必须	String	组件的全路径

参 考 文 献

[1] Schneier, B., "Applied Cryptography: Protocols,Algorithms, and Source Code in C", 2nd Edition, 1996.