新型可信计算赛题：userinfo\_access

## 一 题目描述：

本题考察学生对标记与访问控制规则的理解。

某公司的服务器（用实例server代表）上存储着公司员工的信息，包括员工姓名，id号，部门，职位，工作年限，个人薪水，电话号码，邮箱地址等。部门有研发（R）、财务（F）、营销（M）、行政（A）四个部门，员工信息格式如下：

表1 员工信息记录项格式

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 记录项名称 | 英文名 | 数据格式 | 长度 | 内容 | 备注 |
| 姓名 | name | ESTRING | <32 | 员工的姓名 |  |
| 序号 | ID | STRING | 8 | 员工的序号 |  |
| 部门 | department | STRING | 16 | 员工所属部门 | R,F,M,A四项选择 |
| 职位 | position | STRING | 8 | 员工职位 | ER、PM、DM、CTO、CFO、CMO、CAO、CEO 8项选择 |
| 工作年限 | YOF | INT | 4 | 员工工作时间（年） |  |
| 个人薪水 | salary | INT | 4 | 员工个人薪资标准（元） |  |
| 电话号码 | cell | STRING | 10 | 电话号码（10位） |  |
| 邮箱地址 | email | ESTRING | <64 | 邮箱地址 |  |

这里STRING表示定长字符串，ESTRING表示以0结尾的不定长字符串,INT则表示整数。

员工职位包括普通员工（ER)、项目经理（PM)、部门经理（DM)、CTO、CFO、CMO、CAO和CEO 共8种，各职位的级别、所在部门和管理权限如下所示。

表2 员工职位信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 职位名称 | 职位名称（英文） | 员工级别 | 部门 | 管理权限 |
| 普通员工 | ER | 1 | 所属部门 | 本部门 |
| 项目经理 | PM | 2 | 所属部门 | 本部门 |
| 部门经理 | DM | 3 | 所属部门 | 本部门 |
| 首席技术官 | CTO | 4 | R | R|M|A |
| 首席财务官 | CFO | 4 | F | F|M|A |
| 首席行政官 | CAO | 4 | A | M|A |
| 首席营销官 | CMO | 4 | M | M|A |
| 首席执行官 | CEO | 5 | 不属于任何部门 | R|F|M|A |

按照公司规定，员工名称、id号、部门、职位、工作年限为公开信息，个人薪水为本人、财务部门同级别或高级别员工可查询信息，电话号码为员工管理权限内同级别、高级别或低一级别员工可查询信息，邮箱地址则为员工管理权限内同级别、高级别、低一级别或低两个级别员工可查询信息。

目前，员工信息存放在server端。任意员工查询信息时，需首先通过user端登录server，输入员工名信息，即可列出所有同名员工的个人信息，没有任何访问控制措施，不符合公司规定。

小张是公司的信息安全官。他设计了一个变形的BLP访问控制方案来加强员工数据库的信息管理。他将数据库中的信息定为1-5五个级别，与员工级别对应，同时对应公司的研发（R）、财务（F）、营销（M）、行政（A）四个部门建立了四个安全类别。在安全标识上，则针对主体和客体采取了不同的标记方式：

主体标记中，安全级别设置为员工行政级别，安全类别设置为员工管理权限的组合。

客体标记则采用了员工信息与记录项客体属性按照如下说明方式合成计算的方式，首先，建立一个如下所示的记录项客体属性表：

表3 记录项客体属性表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性项 | 英文名 | 数据格式 | 长度 | 内容 | 备注 |
| 名称 | name | ESTRING | <32 | 记录项的名称 | 对应表1中的各种名称。 |
| 是否调整级别 | Isleveladjust | CHAR | 1 | 记录项内容是否应对级别进行调整 | 0为不需要，1为需要 |
| 是否自有类别 | Isselfdefine | char | 1 | 记录项内容是否有自己的专门类别 | 0为无，1为有 |
| 自有类别 | class | SFLAG | 2 | 记录项所属类别 | R,F,M,A的组合，包括全不选的情况（空集） |
| 固定级别 | Level\_fix | int | 4 | 记录项的固定信息级别 | 记录项固定的信息级别，当取值为0时，该项失效 |
| 调整级别 | Level\_adjust | int | 4 | 员工工作时间（年） | 当固定级别取值为0时，该项生效，生效时，该项取值范围为-5 — 5 |

同时，根据表2建立了一个职位-部门与员工信息基础安全属性的对应表。

当获取员工信息后，将根据员工个人所属部门、职位级别，查询获得员工信息基础安全属性，再与记录项客体属性表合并计算员工各记录项的客体级别。

如果记录项isselfdefine项为1，则其类别设置为记录项客体属性表中的自有类别（class)项内容，否则其类别为员工个人所属部门对应类别。

如果记录项isleveladjust为1，则记录项客体级别为员工职位级别加上调整级别（level\_adjust），否则记录项客体级别为客体属性表中的固定级别(level\_fix)。

任意员工读取自己或其它员工信息时，服务器在数据发给用户前，需对信息的各个记录项进行访问控制检查，确认公司规定下允许读取的记录项，符合公司规定的记录项可以原样返回，不符合公司规定的记录项应该清除其内容：ESTRING格式返回长度为0的字符串,STRING格式的字符串内容全部填\*，INT格式的取值为0。

此外，还有一个例外规则：当读取内容为薪水时，本人可以作为可信对象，读取自己的薪水。

现在，小张已经构建了系统的实现架构，完成了标记模块user\_label和访问控制模块record\_acl,完成了主体策略文件subject.list，并为记录项客体信息文件record.list填写了记录项name-salary的六条策略。现在他有急事请假，请你帮助他完成record.list中的剩余策略，并在src/record\_acl.c中的except\_rule函数中实现例外规则，以实现公司的安全需求。

## 二 题目环境：

本题目为“兵棋“模式新型可信计算赛题中访问控制方向的应用赛题，基于可信软件基原型Cube-1.3搭建，通过分布式消息驱动机制模拟应用流程与攻防行为。

选手以赛题平台所显示的用户身份登入系统。题目提供测试环境。该环境中有一服务实例cube-userinfo-access/instance/server和一个用户实例cube-userinfo-access/instance/user,其中服务实例存储记录项object.list，并提供对记录项的读写支持，user实例执行用户登录和读记录项操作。cube-userinfo-access目录下,record\_access.sh是带访问控制的读写示例。

在server实例目录下，有一些文件用来存储策略或记录项，。如下所列：

表4 数据文件表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 文件名 | 类型 | 格式 | 记录内容 | 备注 |
| subject.list | 策略文件 | （MAC\_LABEL,  SUBJECT） | 员工的管理权限 |  |
| record.list | 策略文件 | (MAC\_LABEL,  RECORD) | 员工信息记录项的安全属性 |  |
| object.list | 数据文件 | （RECORD\_DEFINE,  RECORD) | 员工个人信息集合 |  |

主体读访问控制实现方式如图1所示：



图1 读访问控制实现方式

图1则展示了用户进行读操作时的标记与访问控制方式。这里有两个监控点,第一个监控点根据连接信息为读命令添加主体标记，而后读命令送入record\_server模块，读出记录数据并保留主体标记。record\_server模块输出的数据被拦截后转入访问控制模块record\_acl，record\_acl需根据主体标记清除记录数据中主体无权读取的内容。

选手通过修改记录项安全属性策略record.list,即可达成变形BLP模型的安全要求。通过在src/record\_acl/record\_acl.c中的except\_rule函数中添加代码，完成后在源码目录下用make编译，即可实现例外规则。完成后，可执行record\_access.sh来自行测试。观察脚本内容，可知测试的主体是instance/user目录下的login.msg,login1.msg和login2.msg中的用户，客体则是instance/user目录下的read.msg和read1.msg中的用户名对应的记录项。观察record\_access.sh的输出结果，可知不同用户读取数据时，实际获取的数据内容。

选手确定自己解题成功后，即可执行 sh player.sh来提交自己的解答，系统会自动执行测试过程，并在屏幕上输出解题结果。如解题结果正确，系统将提供flag，选手提交flag即可获得分数。

当前系统员工信息情况在object.list中，其内容如下表所示：

表5 员工个人信息表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| name | ID | department | position | YOF | salary | cell | email |
| zhao | 1001 | —— | CEO | 5 | 12000 | 10001 | zhao@corp.com |
| qian |  |  |  |  |  |  |  |
| sun |  |  |  |  |  |  |  |
| li |  |  |  |  |  |  |  |
| zhou |  |  |  |  |  |  |  |
| wu |  |  |  |  |  |  |  |
| zheng |  |  |  |  |  |  |  |
| wang |  |  |  |  |  |  |  |

## 三 解题要求

选手修改instance/server目录下策略文件record.list，补全表1中所有记录项对应的客体安全属性，并修改src/record\_acl/record\_acl.c中except\_rule，以在数据读取行为符合例外规则时返回1,并在record\_acl目录下用make命令编译。而后执行sh record\_access.sh自行测试，观察输出，如输出符合第一节中访问控制要求，则可执行sh player.sh ，提交答案，等待赛题裁决机制的输出，如输出flag，则提交flag，否则如果题目给出错误提示，则修改策略文件和代码，对代码重新编译后再次提交进行测试。

## 四 题目提示

本题目代码实现机制比较复杂，但题目已实现了机制和策略的分离，因此我们并不需要阅读代码中的细节只需正确配置策略和实现例外规则，观察输出是否符合预期即可解出题目，建议聚焦有访控要求的记录项，参考record.list中已有项来设置其它项的策略值。编写代码时，则注意record\_acl.c中except\_rule函数及其被调用位置的提示信息，根据题目要求，确定其内部逻辑并实现。