VCI-PLT 功能简要说明

北京宏博远达科技有限公司

2017 年 06 月

文档控制

1.1 创建更改记录

日期	作者	版本	更改参考
2017/06/10	熊超	1.0	

1.2 审阅人员

日期	职位	签字

1.3 分发人员

日期	职位	签字

目　录

2 概述

本文详细描述本发布版本所包含的功能简要说明。

3 功能说明

3.1 三员管理、用户组织、角色管理

本发布版本包含内置的三员角色（系统管理员、安全保密员、安全审计员）及三员用户（系统管理员(sysAdmin)、安全保密员（secAdmin）、安全审计员（auditAdmin），默认密码 123456），该三员角色和三员用不能删除。

3.1.1 默认功能

3.1.1.1 系统管理员

默认拥有 IP 密级配置、用户\机器密级停启用配置、用户（使用系统的用户）管理、部门管理、角色管理等模块的管理功能。

3.1.1.2 安全保密员

默认拥有 配置系统管理员（将普通用户添加到系统管理员角色）、配置安全保密员（将普通用户添加安全保密员角色）、配置安全审计员（将普通用户添加到安全审计员角色）、密码策略管理、功能模块权限配置、查看普通用户登录日志、普通用户授权日志、查看普通用户操作日志、UI 授权、数据授权

3.1.1.3 安全审计员

默认拥有 查看三员的登录日志、查看三员的授权日志、查看三员的操作日志。

可以为三员分配普通角色，从而让三员也有普通用户的功能权限。

也可以通过将普通用户添加到三员角色（只有安全保密员有此模块的权限），从而让普通‘提升’为三员，拥有三员角色的相关功能。

3.2 单点登录支持

VCI-PLT 发布版本中富客户端已经包含一个简单的单点登录实现（发布目录 /sso/login.jsp），其逻辑如下：

从 HTTP 头中读取 iv-user 参数，如果存在该参数，则取出该参数到系统中进行判断，如果存在则允许登录，如果不存在则进入富客户端的登录界面。

WEB 端单点登录可采用该方式进行处理。

3.3 密级管理

本发布版本支持三个维度的密级管理。

处理规则：

- 如果开启用户密级，关闭机器密级，此时在平台后台接口中所有的数据查询都按密级进行过滤控制（主要针对业务类型数据查询），如果被查询的业务类型上没有密级属性（secretgrade）,则忽略此规则。
- 如果用户密级、机器密级同时开启，则用户在登录（非单点认证登录）时会对登录客户端 IP 进行判定,当且仅该 IP 密级必须大于或等于用户密级时才允许登录，否则不允许登录。（登录认证通过后，后台的数据查询同样会增加用户密级与数据密级的过滤控制，规则同上）
- 如果启用机器密级，关闭用户密级，不作其它任何过滤规则控制。
- 用户密级、机密密级、数据密级进行比较控制时，都是使用密级枚举值对应的整数值进行大小判断，约定枚举值越大密级级别越高。

3.3.1 用户密级

内置 4 个枚举级别，可扩充（整数，该数据定义在枚举类型‘人员密级类型’（usersecurityenum）中）

- 10 内部
- 20 一般
- 30 重要
- 40 核心

3.3.2 机器密级

内置 3 个枚举级别，可扩充（整数，该数据定义在枚举类型‘IP 密级枚举’（ipsecurityenum）中）

- 10 非密
- 20 秘密

- 30 机密

3.3.3 数据密级

内置 4 个枚举级别，可扩充（整数，该数据定义到枚举类型‘密级枚举’（Enumsecretgrade）中）

- 10 非密
- 15 内部
- 20 秘密
- 30 机密

3.4 权限管理

权限管理支持基于角色的功能模块权限定义及控制。

出于安全考虑及细粒度的权限控制需求，对于使用平台定义出来的各个上下文，需要进行授权（UI 授权）后才能使用里面定义的组件及按钮（平台当前规则：一个上下文里可包含多个区域（导航、控制、操作，每种区域最多一个，最多同时包含三种区域），每个区域可以多个页签，每个页签里可以包含多个组件，每个组件上可以的多个功能按钮）

可通过修改 conf.properties 中的 ui.right.swith 选项，将其值修改为 off 来禁用对 UI 上下文中的组件及按钮的权限检查。

3.5 日志管理

3.5.1 系统日志

日志管理模块支持三类日志查询：登录日志、授权日志、操作日志的查询，每类日志又分管理员日志和普通日志。

3.5.1.1管理员日志

三员角色下的用户产生的日志。

系统管理员内置拥有查看普通用户的三类日志。

安全保密员内置拥有查看三员用户的三类日志。

3.5.1.2普通用户日志

普通角色下的用户产生的日志。

可以为普通用户授权三类日志查询模块的权限，也可以为普通用户授权三员产生的三类日志查询模块的权限。

3.5.2 异常\控制台日志

在代码中使用 System.out.println(xxx) 或异常时的 e1.printStackTrace(); 以及使用 log4j 输出的日志都将会写入到 发布环境根目录/logs/server_log.txt 中。

如果是以 windows 服务方式启动运行的，控制台日志将会被写入到发布环境根目录/logs/VCI_PLT_CORE_SERVICE.txt 中，使用 log4j 输出的日志将会写入到发布环境根目录/logs/server_log.txt 中。

3.6 工作流

3.7 消息管理

当前的发布版本中包含一个简易的消息模型（对象模型）