|  |  |
| --- | --- |
| 项目编号 | 2015GZ0105 |
| 版 本 | V1.0.0.1 |

**终端安全防护**

**概要设计**

**终端安全防护**

**2015年12月14日**

**目录**

[1 项目概述 1](#_Toc438043488)

[1.1 前言 1](#_Toc438043489)

[1.2 开发方式 1](#_Toc438043490)

[1.3 术语与略缩语 1](#_Toc438043491)

[1.4 系统组成 3](#_Toc438043492)

[1.5 系统部署图 4](#_Toc438043493)

[1.6 总体框架 5](#_Toc438043494)

[2 服务器设计 5](#_Toc438043495)

[2.1 支持平台 5](#_Toc438043496)

[2.2 基本框架 6](#_Toc438043497)

[2.3 与客户端交互接口 7](#_Toc438043498)

[2.4 策略设计 7](#_Toc438043499)

[2.5 模块设计 8](#_Toc438043500)

[2.6 异常处理 22](#_Toc438043501)

[3 WEB控制台设计 23](#_Toc438043502)

[3.1 支持平台 23](#_Toc438043503)

[3.2 基本框架 23](#_Toc438043504)

[3.3 总体结构设计 27](#_Toc438043505)

[3.4 用户管理 29](#_Toc438043506)

[3.5 客户端管理 31](#_Toc438043507)

[3.6 策略管理 33](#_Toc438043508)

[3.7 日志管理 41](#_Toc438043509)

[3.8 系统设置 42](#_Toc438043510)

[4 客户端设计 44](#_Toc438043511)

[4.1 开发平台和工具 44](#_Toc438043512)

[4.2 支持平台 44](#_Toc438043513)

[4.3 分发与部署安装 44](#_Toc438043514)

[4.4 基本架构 44](#_Toc438043515)

[4.5 与服务器交互接口 51](#_Toc438043516)

[4.6 策略 61](#_Toc438043517)

[4.7 资产收集 62](#_Toc438043518)

[4.8 日志模块 63](#_Toc438043519)

[4.9 服务进程控制 63](#_Toc438043520)

[4.10 基线检测 65](#_Toc438043521)

[4.11 文件访问监控 67](#_Toc438043522)

[4.12 安全防护 73](#_Toc438043523)

[4.13 网络访问监控 75](#_Toc438043524)

# 项目概述

## 前言

## 开发方式

### 迭代开发

### 文档要求

每阶段的版本要先写清楚需求分析，经过讨论，条件允许的情况需要提供界面原型由售前确认。需求分析评审会决定最终用来开发的需求分析文档。

在需求分析基础上要写出概要设计，描述主要的算法、数据结构、协议、接口。经过开发组长会讨论后再进行编码。概要设计评审会决定最终用来编码的概要设计文档。

## 术语与略缩语

## 系统组成

系统分为三个模块：服务器、WEB控制台、客户端。

服务器和客户端各自是一个整体。控制台是一个集合，每个功能点都做一个独立的控制台。使用不同的控制台向服务器发送相应的策略配置，策略经服务器下发到客户端，来开启相应的功能模块。

客户端使用5011端口，服务器使用5012端口，采用TCP通信协议提供数据通信服务。

客户端定时主动连接服务器，与服务器保持心跳连接，以便实现对NAT网络的兼容。

系统所涉及的计算机，主要包含客户端、服务器、控制台，都统一采用GUID来标识一个实体的唯一性，该GUID要做到与软件环境无关，只与硬件环境有关。

## 系统部署图

图：系统部署图

系统采用客户端、服务器的C/S架构，Client与Server通过网络进行通信。

Client启动后主动与Server通信，并保持心跳连接。

Server接收Client的数据，并下发相应控制策略。

Client接收到控制策略，按照策略规定执行相应功能。当无法与Server通信时，

采用上一次的策略（或默认策略）执行功能。

## 总体框架

本图描述客户端、服务器、控制台间的协作关系。本系统设计时考虑了对NAT网络的支持。所以客户端可以和服务器不在一个网络内。客户端和服务器的网络链接都是采用客户端主动的方式。

服务器

数据库

客户端

控制台

客户端

只在进行远程控制时进行链接

定时上报及策略跟新、上报日志等

配置策略等，保持链接

当客户端链接不上服务器时，客户端可以向临近的客户端询问服务器的IP地址

图：模块间关系示意图

# 服务器设计

## 支持平台

支持在以下平台下运行：

MicroSoft Windows xp 及以上所有版本操作系统。

包括：Windows 2000/XP/2003/7/2008/2008R2/Vista/8/2012/10。

## 基本框架

### 服务器自启动

* 服务器以服务的方式运行，操作系统启动后自动运行
* 和用户登录无关，即使服务器操作系统没有进行用户登录，也可以正常工作。
* 服务器可以支持10000个客户端。
* 服务器程序本身没有界面，它的所有设置通过控制台来进行。

### 服务器授权许可

* 授权期限

服务器启动时需要检测授权是否到期，到期后则自动转为试用版，只处理前五个注册的客户端的请求，试用时间为一个月（或无期限）。同时在管理员登陆WEB界面时自动跳转到注册页面。

授权到期后停止socket监听。

授权未到期时，可以有界面（如通过控制台）看到授权序列号、到期时间、授权许可数量、售后服务时间等。

* 授权控制

服务器在启动的时候将注册码信息加载入内存。并每隔一天时间检测一次授权时间是否已经到期，如果到期则只接收第一个注册的客户端的请求。在用登陆WEB控制台时根据注册码信息显示用户购买的功能模块的策略配置菜单。如果注册码过期则自动跳转到注册页面。如果管理员无法（或没有）设置某个功能模块的工作策略，客户端也就收不到相应功能模块的策略，客户端相应的功能模块也就无法工作，以此来达到对各功能模块的授权。

* 许可方式采用如下两种方式之一：

1、系统第一次运行时，检测出硬件特征码，该码报告给厂商，厂商根据此码生成许可序列号，输入后激活。

2、系统运行时，输入厂商提供的许可序列号，该号在服务器在线升级的时候，连接厂商的升级服务器做唯一性判断，如果发现在多台机器上安装，则通知该序列号失效或者限制该台机器的升级。

* 授权序列号中含有如下信息：

硬件特征码、许可日期、许可模块、允许客户端数量、售后服务时间。

### 服务端升级

* 直接卸载重新安装

卸载后重新安装，或者覆盖安装。

* 手动升级

通过在控制台上点击本地升级按钮（或菜单），自动选择控制台所在机器上的升级包，对服务器软件进行升级。

* 自动（在线）升级

设置升级地址后，服务器按照设置在指定的时间内去升级服务器上检测是否需要升级，然后在设置的下载升级包的时间内去下载升级包。然后再在指定时间内去实现更新操作。

在此设置的升级地址可以是官方网站，也可以是升级服务器的地址。

* 升级包内容

包括服务器程序的升级包、服务器数据库表的升级包、控制台程序升级包、客户端程序升级包、策略升级包。

## 与客户端交互接口

参照客户端与服务器的交到方式。

## 策略设计

### 策略格式

策略采用XML格式存储，和客户端的格式一致。

### 策略类别和组合

* 策略分为单机策略、组策略。
* 客户端单机策略， 或者该客户端所属的组被控制台更改时，服务器应该建立客户端策略需要更新的内存标志。
* 客户端连接服务器并请求策略更新时，服务器将该客户端的单机策略、所属组策略进行依序下发
* 控制台每次更新了组策略，则服务器自动将组内所管辖的所有的客户端的策略合成并存储在服务器硬盘上（合成的来源包括单机策略、组策略）。
* 客户端连接服务器并请求策略更新时，服务器将该客户端的单机策略、所属组策略进行组合下发。

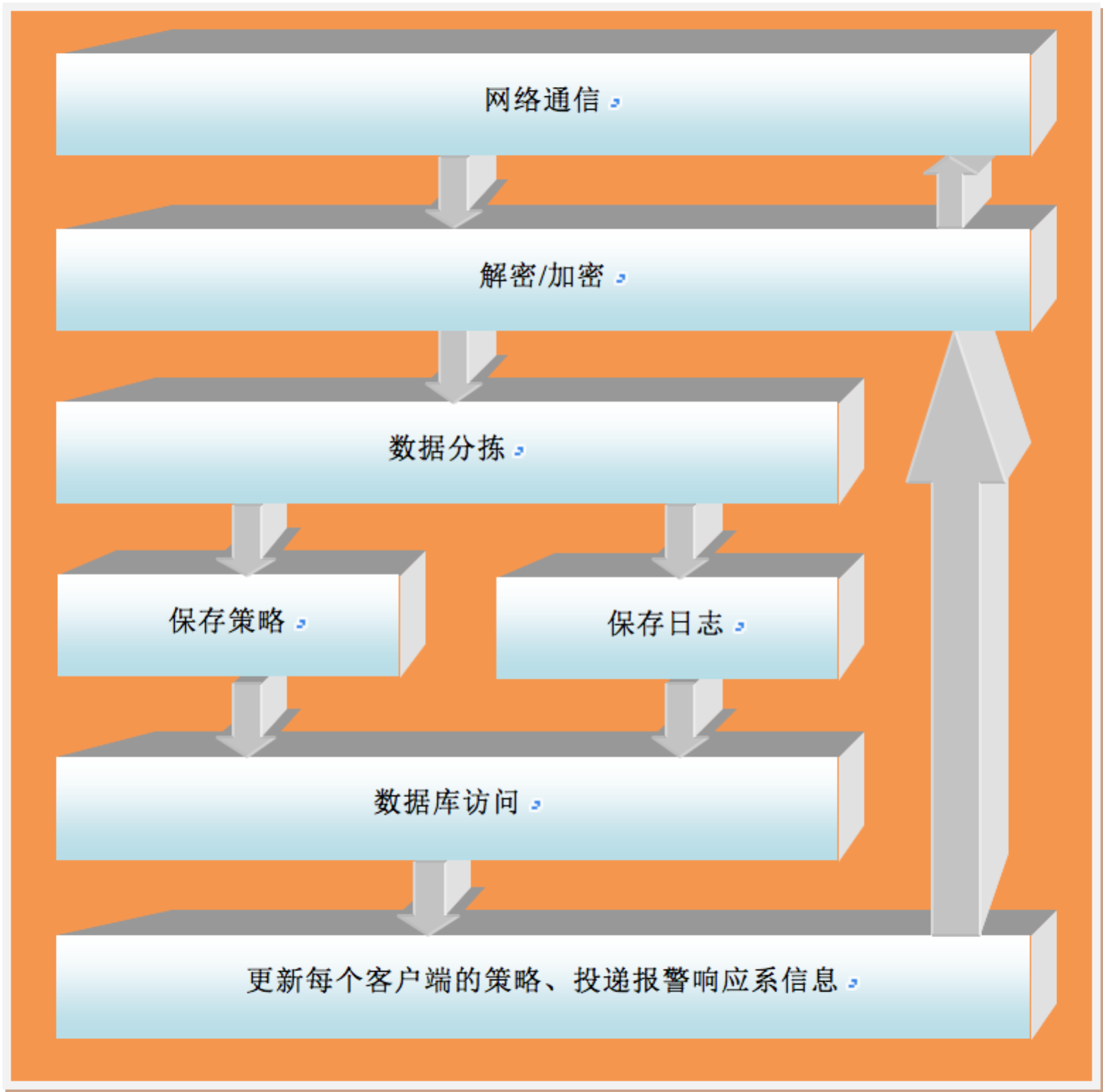
### 策略下发

当新的策略创建或者已有的策略被修改时，服务器先主动链接所有客户端，把新的策略下发到每一个客户端。如果客户端是处于NAT网络后的话，那么就要等客户端主动链接到服务器上时，服务器借助这个链接把策略发下去。

## 模块设计

服务器由初始化模块、通信监听模块、通信模块、客户端数据处理模块、WEB控制台数据处理模块、资产管理模块、管理管理模块、日志管理模块组成。

模块层次划分如下图：



图：服务器模块层次划分

### 模块协作



图：服务器模块协作示意图

### 服务器初始化模块



图：服务器运行流程图

1） 服务器启动。

2） 服务器唯一标识符的生成、授权验证。

3） 服务器系统配置信息加载。

4） 客户端状态信息加载。

5） 初始化线城池、socket连接池、控制台数据处理链表、客户端数据处理链表。

6） 启动监听端口

#### 接口设计

CreateGuid()

|  |  |
| --- | --- |
| 定义 | Public String CreateGuid（） |
| 功能 | 获取服务器唯一标识符。 |
| 参数 | 无 |
| 返回值 | 服务器唯一标识符 |

GetCheckResult(String strSn,String strHostCode)

|  |  |
| --- | --- |
| 定义 | Public boolean GetCheckResult(String strSn,String strHostCode) |
| 功能 | 验证注册码是否合法（是否存在、是否过期、是否与硬件特征码对应）。 |
| 参数 | String license |
| 返回值 | License是否合法 |

getLicenseInfo（String liscense）

|  |  |
| --- | --- |
| 定义 | Public licenseBean getLicenseInfo（String license） |
| 功能 | 返回注册码包含的信息。 |
| 参数 | license |
| 返回值 | License包含的信息 |

LoadServerConfigInfo（）

|  |  |
| --- | --- |
| 定义 | Public int LoadServerConfigInfo（） |
| 功能 | 加载系统配置信息。 |
| 参数 | 无 |
| 返回值 | 返回加载系统配置信息的结果 |

LoadClientStatusInfo（）

|  |  |
| --- | --- |
| 定义 | Public List LoadClientStatusInfo（） |
| 功能 | 获取数据库中客户端信息，加载到内存。 |
| 参数 | 无 |
| 返回值 | 返回加载客户端链表 |

Init（）

|  |  |
| --- | --- |
| 定义 | Public List Init（） |
| 功能 | 初始化线城池、socket连接池、控制台数据处理链表、客户端数据处理链表、客户端状态链表。 |
| 参数 | 无 |
| 返回值 | 无 |

StartListenPort（int port，boolean isblocking）

|  |  |
| --- | --- |
| 定义 | Public int StartListenPort（int port，boolean isblocking） |
| 功能 | 调用服务器监听模块。 |
| 参数 | Port端口，isblocking是否阻塞 |
| 返回值 | Int 启动结果 |

### 监听模块

通信模块实现功能如下：

1） 启动监听端口

2） 创建线程池，根据注册序列号确定线程池大小

3） 监听客户端的链接请求

4） 把链接请求放入连接链表当中，如果对应得线程仍然处于活动，则直接启动，否则重新创建工作线程，放入连接池当中。

#### 接口设计

ListenPort（int port ,boolean isblocking）

|  |  |
| --- | --- |
| 定义 | Public boolean ListenPort（int port ,boolean isblocking） |
| 功能 | 启动监听端口。 |
| 参数 | Port要开启的监听端口，isblocking工作模式包括阻塞、非阻塞 |
| 返回值 | boolean |

newFixedThreadPool(POOL\_SIZE)

|  |  |
| --- | --- |
| 定义 | Public int newFixedThreadPool(POOL\_SIZE) |
| 功能 | 创建线程池 |
| 参数 | POOL\_SIZE，根据客户端个数及cpu个数及连接池的大小来创建线程池的大小 |
| 返回值 | Int |

executorService.execute(Runnable workThread)

|  |  |
| --- | --- |
| 定义 | Public int executorService.execute(Runnable workThread) |
| 功能 | 启动工作线程，java自带库函数，在这里通常放置Runnable,在系统中调用采用Thread实现这个Runnable |
| 参数 | Runnable workThread |
| 返回值 | Int |

### 通信模块

通信模块实现以下功能：

1） 实现客户端与服务器通信交互

2） 处理心跳通信

3） 处理传输通信

#### 线程调度流程图



图：通信模块线程调试流程

#### 工作线程流程图



图：工作线程流程图

#### 接口设计

HanderSocketQueenThread(LinkedList<Socket> socketQueue,**int** threadID)

|  |  |
| --- | --- |
| 定义 | Public HanderSocketQueenThread(LinkedList<Socket> socketQueue,**int** threadID) |
| 功能 | 将一个连接放入工作线程当中，主要处理心跳通信，如果为传输通信，那么将数据放入到数据处理线程当中处理。 |
| 参数 | Socket，threadID |
| 返回值 | 无 |

send(Protocol)

|  |  |
| --- | --- |
| 定义 | Public int send(Protocol) |
| 功能 | 通用发送数据接口。 |
| 参数 | Protocol协议包 |
| 返回值 | int |

receive()

|  |  |
| --- | --- |
| 定义 | Public Protocol receive() |
| 功能 | 通用的接收数据（包括包头和数据内容）接口。 |
| 参数 | 无 |
| 返回值 | Protocol |

receiveHeader ()

|  |  |
| --- | --- |
| 定义 | Public Protocol receiveHeader () |
| 功能 | 通用的接收数据包头接口。 |
| 参数 | 无 |
| 返回值 | Protocol |

receiveContent (Protocol protocol)

|  |  |
| --- | --- |
| 定义 | Public Protocol receiveContent (Protocol protocol) |
| 功能 | 通用的接收传输通讯内容数据接口。 |
| 参数 | 无 |
| 返回值 | Protocol |

sendOK(TcpTransferLayer tranfer,String guid)

|  |  |
| --- | --- |
| 定义 | Public void sendOK(TcpTransferLayer tranfer,String guid) |
| 功能 | 通用的回复ok数据包接口。 |
| 参数 | Socket，客户端唯一标识guid |
| 返回值 | 无 |

close()

|  |  |
| --- | --- |
| 定义 | Public void close() |
| 功能 | 通用关闭socket接口。 |
| 参数 | 无 |
| 返回值 | 无 |

### 客户端数据处理模块

实现功能：处理客户端上传的资产信息、日志信息等，保存到数据库

#### 模块流程图



图：客户端数据处理流程图

#### 接口设计

HanderSocketDataThread(protocol,client)

|  |  |
| --- | --- |
| 定义 | HanderSocketDataThread(Protocol protocol,TransferLayer client) |
| 功能 | 传输通信数据处理线程。 |
| 参数 | protocol,TransferLayer |
| 返回值 | 无 |

dataProcess(TcpTransferLayer client)

|  |  |
| --- | --- |
| 定义 | dataProcess(TcpTransferLayer client) |
| 功能 | Dt发生变化后，服务器组织相关策略、消息等内容，统一打包发送给客户端。 |
| 参数 | 客户端唯一标识 |
| 返回值 | 无 |

public String getDownloadPolicyContent(String guid)

|  |  |
| --- | --- |
| 定义 | public String getDownloadPolicyContent(String guid) |
| 功能 | Dt发生变化后，服务器组织相关策略。 |
| 参数 | guid客户端唯一标识 |
| 返回值 | 策略xml内容 |

processPolicy(PolicyManageBL pmBl, String guid)

|  |  |
| --- | --- |
| 定义 | processPolicy(PolicyManageBL pmBl, String guid) |
| 功能 | Dt发生变化后，策略处理业务。 |
| 参数 | pmBl 策略数据封装bean，guid客户端唯一标识 |
| 返回值 | 无 |

processMessage(String guid)

|  |  |
| --- | --- |
| 定义 | processPolicy(PolicyManageBL pmBl, String guid) |
| 功能 | Dt发生变化后，消息处理业务。 |
| 参数 | pmBl 策略数据封装bean，guid客户端唯一标识 |
| 返回值 | 无 |

deleteMemoryPolicy(String guid)

|  |  |
| --- | --- |
| 定义 | deleteMemoryPolicy(String guid) |
| 功能 | 当服务器收到客户端返回的ok后，删除掉对应客户端的内存信息。 |
| 参数 | 客户端唯一标识 |
| 返回值 | 无 |

### 控制台数据处理模块

实现功能：

处理来自控制台的数据。

#### 处理流程图



图：控制台数据处理流程图

#### 接口设计

WebDataProcessThread（）

|  |  |
| --- | --- |
| 定义 | Public int WebDataProcessThread（） |
| 功能 | 启动web数据处理线程。 |
| 参数 | 无 |
| 返回值 | int |

serverProcess(WebDataBean databean)

|  |  |
| --- | --- |
| 定义 | Public int serverProcess(WebDataBean databean) |
| 功能 | 控制台与服务器交互的统一接口，服务器将将接收到策略、消息、及其它指令性内容放在内存链表中，由服务器的处理线程去处理。 |
| 参数 | int command，object databean |
| 返回值 | int |

updateClientStatus(AgentStatus agentStatus)

|  |  |
| --- | --- |
| 定义 | public void updateClientStatus(AgentStatus agentStatus) |
| 功能 | 当服务器收到控制台发送的策略或消息等信息，更新服务器内部保存的客户端的dt。 |
| 参数 | AgentStatus |
| 返回值 | 无 |

noteClient(AgentStatus agentStatus)

|  |  |
| --- | --- |
| 定义 | noteClient(AgentStatus agentStatus) |
| 功能 | 服务器主动通知客户端心跳。 |
| 参数 | AgentStatus 客户端当前状态信息 |
| 返回值 | 无 |

### 资产管理模块

功能实现：获取客户端注册信息、硬件特征码，

服务器处理客户端上报资产信息，保存到数据库中

#### 接口设计

GetCheckResult(String strSn,String strHostCode)

|  |  |
| --- | --- |
| 定义 | public boolean GetCheckResult(String strSn,String strHostCode) |
| 功能 | 解析客户端发送的软件信息。 |
| 参数 | byte[] info 客户端软件信息，guid客户端唯一标识 |
| 返回值 | 解析软件xml内容 |

private boolean CheckCheckCode(String strSN)

|  |  |
| --- | --- |
| 定义 | private boolean CheckCheckCode(String strSN) |
| 功能 | 解析客户端发送的软件信息。 |
| 参数 | byte[] info 客户端软件信息，guid客户端唯一标识 |
| 返回值 | 解析软件xml内容 |

getSoftwareList(byte[] info, String guid)

|  |  |
| --- | --- |
| 定义 | public Collection getSoftwareList(byte[] info, String guid) |
| 功能 | 解析客户端发送的软件信息。 |
| 参数 | byte[] info 客户端软件信息，guid客户端唯一标识 |
| 返回值 | 解析软件xml内容 |

insertSoftware(Software software)

|  |  |
| --- | --- |
| 定义 | public void insertSoftware(Software software) |
| 功能 | 插入客户端的软件信息。 |
| 参数 | Software信息bean |
| 返回值 | 无 |

### 策略管理模块

功能实现：服务器处理控制台发送的策略信息，保存到内存策略链表。

#### 接口设计

public String getDownloadPolicyContent(String guid)

|  |  |
| --- | --- |
| 定义 | public String getDownloadPolicyContent(String guid) |
| 功能 | Dt发生变化后，服务器组织相关策略。 |
| 参数 | guid客户端唯一标识 |
| 返回值 | 策略xml内容 |

processPolicy(PolicyManageBL pmBl, String guid)

|  |  |
| --- | --- |
| 定义 | processPolicy(PolicyManageBL pmBl, String guid) |
| 功能 | Dt发生变化后，策略处理业务。 |
| 参数 | pmBl 策略数据封装bean，guid客户端唯一标识 |
| 返回值 | 无 |

### 日志管理模块

功能实现：服务器处理客户端上传得日志，保存到数据库当中。

#### 接口设计

LogXml(String content)

|  |  |
| --- | --- |
| 定义 | public LogXml(String content) |
| 功能 | 服务器解析客户端上报的xml内容。 |
| 参数 | 客户端唯一标识 |
| 返回值 | 消息xml内容 |

Insert(List LogList)

|  |  |
| --- | --- |
| 定义 | public boolean Insert(List LogList) |
| 功能 | 插入客户端上报的日志信息。 |
| 参数 | 日志记录 |
| 返回值 | 无 |

## 异常处理

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **出错情况** | **错误等级** | **程序输出信息** | **程序处理**  **办法** | **用户处理**  **办法** |
| 接收客户端上传信息 | 1 | 接收客户端上传信息不完整 | 不改变服务器维护的状态信息，并记录日志 | 客户端重新上传 |
| 服务器启动异常 | 1 | 记录日志 | 记录日志 | 检查端口开启情况，防火墙开启情况、数据库情况 |
| 用户不能正常登录 | 2 | 提示用户不能登录的原因 | 记录日志 | 查询数据库工作情况 |

# WEB控制台设计

## 支持平台

# 客户端设计