# Qscanner概要设计

## 概述

Qscanner的目标是提供一个本地服务程序， 为病毒样本系统提供全部第三方病毒扫描器的扫描服务。

该程序为内部后台使用。

## 名次解释

扫描器： 指第三方的病毒命令行扫描器

引擎： 同上

任务： 指调用所有第三方扫描器对某一个特定的样本进行扫描的过程。

## 层次图

共分为3层

**引擎层： 维护和控制所有第三方扫描器。**

**任务层： 接收样本服务器传送的样本， 并控制整体的扫描流程。**

**通信层: 负责QScanner与扫描器， QScanner与样本服务器之间的通信。**

## 架构图



## 进程视图

QScanner服务通过进程管道通信来连接扫描器组和样本服务器。

当QScanner与样本服务器连接时， 它是进程管道的客户端。

当QScanner与扫描器连接时， 它是进程管道的服务端。



## 引擎管理

引擎管理器主要负责

* 启动各个第三方扫描器引擎
* 获取扫描器的扫描结果，并分析
* 维护扫描器的信息
* 提供任务管理器调用接口
* 当扫描器崩溃或者假死时， 重启扫描器。



## 任务管理

任务管理主要负责

* 接收样本服务器发送过来的样本
* 启动引擎扫描
* 返回所有扫描器的扫描结果。



## 引擎控制

采用远程模块注入的方式注入一个DLL到第三方扫描器进程中。

该远程模块的工作内容如下：

* + - 作为进程管道的客户端连接QScanner。
    - 使用内联钩子技术Hook FindNextFile和WriteFile两个API



## 所需支持

* 需要各家安全厂商扫描器启动的参数

例： fprot -o filename -d scan.log

* 需要对结果进行识别的正则表达式

例：

'Fprot' : (\_AdapterFprotResult, r'(\d{4}-\d{1,2}-\d{1,2} \d{1,2}:\d{1,2}:\d{1,2}).\*?(?:\[Clean\]|\[Found.\*?<(.\*?)(?:\s\(.\*\))?>).\*?(\w+\_\w+-\w+\_\w{32}\.\w+)'),

* 需要样本服务器通信的数据格式。

样本服务器输入的是文件全路径，

返回样本服务器的是全部厂商的扫描结果，

具体的格式需要详细定义