# 用户插件预分析文档

## 用户插件概述

用户插件目的在于最大程度保证路由系统安全与稳定的情况下，但又尽可能给开发者一个最大权限的灵活的开发与运行环境。因此设计重点在权限管理上，难点是权限的定义、如何下发权限与下发多少权限。

极路由的核心机制是在SD中制造一套系统运行环境，制造一个监狱系统环境（也有很多称之为沙盒），将用户插件安装在此环境中，最大限度隔离用户插件对路由系统的伤害。另外，为每个插件分配不用的用户，减少插件间的相互影响。

讨论：除了此方案，是否还有其他可行方案？

之前做的插件管理和用户插件的区别。

极路由的用户插件是一整套平台策略下的结果，包含开发者权限申请、插件上传评审、云网管插件管理、插件下载安装（公开发布与私有发布）、云网管插件配置界面等一系列内容。

## 角色定义

1. 用户角色

用户：终端用户，最终路由的使用者。

开发者：第三方插件的开发者，即可使用监狱系统进行开发和插件运行。

超级用户：可拥有root权限的开发者，

1. 安装角色

用户：终端用户通只能查看到正式发布的插件，然后对其下载与安装。

拥有者：除了可以看到正式发布的插件，还能看到自己开发的插件，对其进行下载与安装。

讨论1：安装角色的定义是否符合我们的需求，整个功能需要云网管配合，考虑云网管与路由系统整个的运营思路。

## 插件类型

按照插件的几种功能与形态进行划分，这种划分并未定义在插件中，只是为了方便讨论。

1. 配置型插件

（1）这种插件本身不包含功能性代码，但是会改变系统的配置和整体运行环境。

（2）在之前的原始SDK中，此种插件并不能很好的完成功能，比如接口从LLLLW转换到WLLLL，便会很难实现。

（3）搭建在新路由平台上的系统，便可以在配置上灵活很多，可以完成此类型插件功能。

讨论2：该类型插件是整个插件系统中最为灵活，权限最难掌控的插件类型，需要讨论是否开放该种插件。

1. 功能插件

最为常见插件类型，其具有自己的一套的运行程序、配置文档、lib库等，它是基于路由系统之上的，不会修改路由系统配置。

讨论3：重点针对的插件类型，可以很好的定义其行为与权限，安全性也是最高的。

1. 系统插件

可对系统的内核模块与应用进行插件化处理。

讨论4：当前SDK的系统中，没有可存储的文件系统运行环境，此类型插件便不进行开发。

1. 已发布的正式插件
2. 已发布的私有插件

讨论5：插件需要发布需要云网管配合，可借用讨论1的结果。

1. 未发布的私有插件

此类型插件使用场景：

1. 调试插件。
2. 在云网管不支持发布私有插件的情况下，自行安装使用。
3. 路由爱好者使用。

讨论6：在此处我们可以看到，插件的用法是有很多的，需要讨论的是我们的核心需求到底是什么？

1. 如果供路由爱好者使用，此种插件是完全符合要求了。
2. 如果是提供给第三方合作厂商，不支持发布私有插件，远程安装升级需要另外考虑。

## 其他细节

当插件运行在监狱系统，这是一个受限制的运行环境，实际运行过程我们可能会遇到如果下问题

1. 监狱环境是无法满足插件功能需求的，常见于配置型插件。
2. 当插件与其他子系统进行交互的时候，比如防火墙、http服务等，应该如何开放权限和开发多少权限。
3. 插件的配置界面，极路由采用的方案是通过云网管来提供界面，而非本地添加界面。
4. 第三方合作厂商的插件是否纳入我们云网管的管理之内，要其申请权限、提交插件、审批插件，通过才准予发布并升级，还是由其自行管理插件。

另外从安全角度来看，防止越狱应当属于另一个话题。