PAM可插拔认证模块

> 在Linux上，需要使用框架性服务的有名称解析和认证服务，例如用户登录程序(login)输入用户名，主机需要将用户、名解析成用户ID，以便后续的认证，这个过程需要名称解析；用户输入密码需要验证密码的正确性，这个过程用到了认 证的服务。

nsswitch

> 名称解析的背后，一定有名称和值的一个文件或者数据库，用来解析名称，这样的文件或数据库，我们称为解析库。解析库的存储方式可以是文本文件，MySQL、NIS、LDAP、DNS等。应用程序的运行需要名称解析时，可以选择任意一种解析库的存储方式，这带来一个问题就是，程序员开发程序的过程中，必须提供对这几种解析库存储方式的支持。这无疑对程序的通用性大打折扣。所以，nsswitch作为一个通用框架对所有程序提供一个统一接口，由nsswitch负责对不同类型解析库的支持。

> nsswitch通用框架以库的形式存在/usr/lib64/libnss3.so，nsswitch通用框架还提供了与各种类型的解析库交互的库，例如：/usr/lib64/libnss_db.so、/usr/lib64/libnss_nis-2.17.so、/usr/lib64/libnss_dns.so等。

配置文件

> /etc/nsswith.conf

```
# Example:
#passwd:    db files nisplus nis
#shadow:    db files nisplus nis
#group:     db files nisplus nis
passwd:     files sss
shadow:     files sss
group:      files sss
```

对于每一种方式的名称解析，从左至右一次查找解析，如果查找成功能，则直接返回结果，如果没有查找成功，则继续想后查询。

查询结果的状态有：`success` | `notfound` | `unavail` | `tryagain`

对每一种结果处理的动作有：`return` | `continue`

```
#示例
#在files和nis中查询结果都为notfound时，直接返回，不在查找dns
host：files nis [NOTFOUND=return] dns
```

PAM

> Pluggable Authentication Modules(可插拔认证模块)，主要作用是用于系统级的用户认证。和nsswitch类似，pam也是提供了一个框架，使的程序开发与认证方式细节分离，而是程序运行时调用认证模块完成认证工作。pam将应用程序和鉴别模块分离开来，pam充当应用程序和鉴别模块联系的中间层。

> 认证的过程需要认证库，认证库可是以文本文件、MySQL、LDAP等存储形式。 pam提供了很多支持各种存储形式的库支持。库文件/usr/lib64/sercurity/目录内。

配置文件及语法格式

> 模块配置文件通常是/etc/pam.d/NAME.conf文件，NAME是使用模块的应用程序的名称，每个应用程序一个单独的配置文件。每个配置文件中每一行定义一种检查规则。通常使用pam认证的应用程序会指定对其负责认证的pam认证配置文件，例如：vsftpd服务中，在他配置文件中会有定义：pam_service_name=vsftpd。

```
#配置文件语法格式
module_type      control      module-path      module-arguments
```

> `module_type`：指明程序所用PAM底层模块的类型

1. auto：鉴别类模块，用户、密码的认证和授权检查
2. account：账户类模块，与账号管理相关认证
3. password：口令类，例如修改密码
4. session：会话类模块，用户获得服务之前或使用服务完成之后，要进行的一些附加性操作

`control`：规定如何处理模块的成功和失败情况，配置文件中各模块的地位与出错时的处理由control栏的取值决定，单个应用程序可调用多个模块。

1. required:只有当对应于应用程序的所有带required标记的模块全部成功后，该程序才能通过鉴别。如果任何带required标记的模块出现了错误，PAM并不立刻将错误消息返回给应用程序，而是在所有模块都调用完毕后才将错误消息返回调用它的程序。
2. requisite：Requisite--它与required相仿，只有带此标记的模块返回成功后，用户才能通过鉴别，不同之处在于其一旦失败就不再执行堆中后面的其它模块，并且鉴别过程到此结束。
3. sufficient：只要标记为sufficient的模块一旦成功，那么PAM便立即向应用程序返回成功而不必尝试任何其他模块。当标记为sufficient的模块失败时，sufficient模块当做optional对待。
4. optional：它表示即便该模块失败，用户仍能通过鉴别。在PAM体系中，带有该标记的模块失败后将继续处理下一模块。
5. include：包含其他文件中通类型的规则。

`module-path`：模块路径，如果模块在/usr/lib64/sercurity/下，可使用相对路径。

`module-arguments`：模块参数

pam模块应用举例参考