Реализация системы управления доступом к информации в виде встраиваемых модулей аудентификации

А.В. Галатенко <agalat@castle.nmd.msu.ru>, А.В. Савочкин <saw@msu.ru>
Московский государственный университет им. М.В. Ломоносова
Центр телекоммуникаций и технологий Интернет
Тел.: (095) 939-2829, факс: (095) 939-46-47

В системах распределенного хранения и обработки информации важной компонентой является подсистема управления доступом. В докладе рассматриваются общие вопросы построения подсистемы аутентификации и авторизации, а также ее реализацию в рамках проекта PNIAM (Pluggable Non Interactive Authentication Modules, встраиваемые неинтерактивные модули аутентификации).

Современные системы управления доступом должны удовлетворять условиям целостности и гибкости. Целостность, в частности, обеспечивает то, что при наличии нескольких сервисов, обеспечивающих доступ к одной и той же информации (например, WWW и FTP), контроль прав доступа будет согласованным. Гибкость позволяет администратору оперативно реагировать на изменения в составе ресурсов и их пользователей.

В соответствии с изложенными требованиями разумно реализовывать систему управления доступом на основе принципов централизованности и модульности.

Под централизованностью понимается
наличие единых, общих механизмов, не зависящих от конкретных приложений, которым требуется аутентификация и авторизация. Централизованность аутентификации и авторизации
позволяет строить продуманные, гибкие и целостностные схемы управления доступом. Естественным способом реализовать принцип централизованности является выделение функций аутентификации, авторизации и протоколирования в отдельную библиотеку, предоставляющую абстрактный интерфейс приложениям.

Для достижения максимальной гибкости системы управления доступом независимые аутентификационные сервисы реализуются в виде отдельных динамически загружаемых модулей. Система динамической загрузки модулей позволяет заменять модули без перекомпиляции приложений и уменьшает потребление ресурсов.

В 1995 году эти три принципа были заложены в основу проекта PAM (Pluggable Authentication Modules, встраиваемые модули аутентификации). Преимущества централизованной, модульной, динамической схемы аутентификации были быстро оценены; PAM получил широкое распространение, став, например, частью популярной ОС RedHat. Однако по прошествии почти четырех лет стали очевидными и некоторые недостатки PAM.

* Разделение задач аутентификации и определения прав доступа и привилегий в некоторых случаях может иметь отрицательные последствия. Такое разделение оставляет теоретическую возможность несовместности процедур аутентификации и приобретения полномочий, или некорректных результатов второй.
* Неприспособленность PAM к выполнению задач аутентификации при отсутствии возможности интерактивного взаимодействия с пользователем. Существующая спецификация PAM разрешает модулям формировать запросы к пользователю без каких-либо ограничений на их количество и содержание. Это означает, что использование PAM в тех случаях, когда информационный обмен между сервером и клиентом строго фиксирован, затруднительно. Начало работ по решению этой проблемы было положено совместно с Andrew Morgan. Некоторые результаты этой работы

можно найти в [1, 2].
 * Для дальнейшего распространения PAM необходимо упрощение прикладного программного интерфейса.

PNIAM - это свободно распространяемая динамически загружаемая библиотека для ОС на базе ядра Linux,
которая обеспечивает единообразное и настраиваемое выполнение процедур аутентификации пользователей, авторизации доступа к сервисам и сопутствующих процедур (протоколирование и др.). Основными сущностями сервисов PNIAM являются приложение (клиентская сторона) и библиотека со списком динамически загружаемых модулей (серверная сторона). Обмен информацией между клиентской и серверной сторонами осуществляется посредством структурированных поименованных элементов (в состав элемента входит поле "name", задающее тип информации (имеется список предопределенных типов, но по необходимости можно вводить и новые), и поле "data", определяющее собственно значение; в качестве примера может послужить элемент с name=USER и data=root). Таким образом, в отличие от PAM, в PNIAM обмен информацией стандартизован, что позволяет реализовывать неинтерактивные схемы аутентификации с тем же успехом, что и интерактивные.

Приложение может получить один из четырех сервисов:
- аутентификация (pniam_authenticate);
- авторизация (pniam_authoorize);
- протоколирование (pniam_account_start - начало, pniam_account_end - конец);
- модификация пароля (pniam_change).
Библиотека PNIAM получает запрос на один из сервисов, и вызывает соответствующие динамически загруженные модули; каждый модуль принимает решение об успехе или неуспехе соответствующего сервиса и, возможно, запрашивает дополнительную информацию, после чего библиотека анализирует результаты работы модулей и принимает итоговое решение. Список используемых модулей и параметры их вызова определяются администратором системы с помощью конфигурационного файла.

К настоящему моменту реализована сама библиотека PNIAM и целый ряд PNIAM-модулей. Кроме того, имеется несколько PNIAM-приложений, работающих в самых разных сферах (начиная от простейших утилит типа login и su и заканчивая обработкой электронной почты). В ближайшее время планируется расширение базы модулей и приложений, а также подключение к проекту новых разработчиков. Более подробную информацию о PNIAM можно найти в [3].

Библиография:

1. http://kernel.org/pub/linux/libs/pam/
2. Morgan, A., "Pluggable Authentication Modules", IETF Internet Drafts, August, 1998.
3. http://www.msu.ru/pniam/pniam.html