SUSE Linux Enterprise Security Guide.
Аутентификация с помощью Pam

October 19, 2009

Linux использует PAM (подключаемые модули аутентификации) в процессе аутентификации в качестве слоя, являющегося посредником между пользователем и приложением. Модули PAM доступны во всей системе, так как могут быть затребованы любым приложением. В этой части описывается, каким образом работает модульный механизм аутентификации и каким образом он настроен.

Системные администраторы и программисты часто хотят ограничить доступ к различным частям системы или ограничить использование определенных функций приложения. При отсутствии PAM, приложения сначала должны быть всякий раз адаптированы к новому механизму аутентификации. например, LDAP, Samba или Kerberos. Этот процесс, однако, отнимает достаточно много времени и не застрахован от ошибок. Единственным способом избежать этим проблем является отделение приложений от механизма аутентификации и передача задач аутентификации централизованно управляемым модулям. Всякий раз, когда требуется новая схема аутентификации, необходимо найти или написать подходящий модуль PAM для использования его программой.

У каждой программы, заисящей от механизма PAM, есть свой собственный файл конфигурации в директории /etc/pam.d/programname. В этих файлах определяются модули, использующиеся аутентификации. Кроме того, существуют глобальные файлы конфигурации для модулей PAM в директории /etc/security, которые определяют точное поведение этих модулей (примеры включают pam_env.conf, and time.conf). Каждое приложение, использующее модуль PAM, в действительности вызывает ряд функций PAM, которые затем обрабатывают информацию в различных конфигурационных файлах и возвращают результат обработки вызвавшему приложению.

Для облегчения создания и поддержки модулей PAM были введены основные файлы конфигурации по умолчанию для функций модулей auth, account, password и session. Они получаются из настроек PAM каждого приложения. Обновляются глобальные модули конфигурации PAM в common-* - что позволяет обновлять все файлы конфигурации PAM, таким образом, администратору не требуется обновлять каждый отдельный файл конфигурации.

Глобальные общие конфигурационные файлы PAM поддерживаются с использованием инструмента pam-config. Этот инструмент автоматически добавляет новые модули в конфигурацию, изменяет конфигурацию существующих или удаляет модули или опции. Вмешательство вручную в файлы конфигурации PAM сведено на минимум или не требуется вовсе.

Структура файла конфигурации PAM

Каждая строка конфигурационного файла содержит максимум четыре столбца:

```
<Type of module> <Control flag> <Module path> <Options>
```

Модули PAM обрабатываются как стек. Различные типы модулей имеют различное назначение, например, один модуль проверяет пароль, другой - расположение, с которого осуществляется доступ к системе, а еще один читает настройки, определенные для пользователя. PAM известно четыре различных типа модулей:

auth - Назначение этого типа модуля - проверка подлинности пользователя. По традиции это происходит путем запроса пароля, но это также достигается с помощью пластиковой карты или биометрии (например, отпечатков пальцев или сканирования радужной оболочки).

account - Модуль этого типа проверяет, обладает ли пользователь необходимыми правами для использования требуемого сервиса. Например, такую проверку необходмо выполнять для того, чтобы гарантировать, что никто не сможет войти под этой учетной записью после истечения ее срока действия.

password - Назначение этого модуля состоит в том, чтобы разрешить изменение маркера аутентификации. В большинстве случаев, это пароль.

session - Модули этого типа ответственны за управление и настройку сессий пользователей. Они запускаются до и после аутентификации для регистрации попытки входа в системных логах и настройки определенного окружения пользователя (учетная запись почты, домашняя директория, ограничения системы и др.).

Второй столбец содержит контрольные флаги для влияние на поведение зарущенных модулей:

required - Модуль с этим флагом должен быть обработан прежде, чем аутентификация будет продолжена. После неудачного завершения модуля с этим флагом все другие модули с флагом required будут обработаны до того, как пользователь получит сообщение об ошибке в процессе аутентификации.

requisite - Модули, имеющие этот флаг, также должны быть успешно обработаны, образом, аналогичным модулям с флагом required. Однако, в случае неудачи этот модуль сразу же сообщает об этом пользователю, никакие другие модули далее обрабатываться не будут. В случае же успеха, другие модули обрабатываются, точно так же, как модули с флагом required. Флаг requisite может использоваться в качестве основного фильтра для проверки существования определенных условий, которые являются основными дял корректной аутентификации.

sufficient - После того, как модуль с этим флагом был успешно обработан, вызвавшее его приложение немедленно получит сообщение об успехе и никакие модули больше не обрабатываются, при условии, что не было ни одной неудачной обработки модулем с флагом required. Неудача этого модуля не будет иметь никаких последствий, в том смысле, что все последующие модули будут обработаны в соответствующем порядке.

optional - Неудача или успех модуля с этим флагом ни на что не влияет. Это полезно для модулей, которые предназначены только для отображения сообщений (например, сообщить пользователю, что для него пришла почта), не предпринимая никаких действий.

include - Если указан этот флаг, файл, указанный в качестве аргумента, вставляется в это место.

Если модули расположены в директории по умолчанию /lib/security (для всех 64-битных платформ, поддерживаемых SUSE® Linux Enterprise Server, директория - /lib64/security), нет необходимости явно указывать путь до них. Четвертый столбец может содержать опцию для конкретного модуля, например, debug (включает отладку) или nollok (позволяет использование пустых паролей).

ЗАМЕЧАНИЕ: Для смешанных 32х и 64х-битных установок

При использовании 64-битной операционной системы возможно также включить возможность выполнения в окружении для 32-битных приложений. В этом случае, убедитесь, что вы установили обе версии соответствующих модулей PAM при установке новых модулей.

Конфигурация PAM для sshd

Чтобы показать как в теории работает PAM, рассмотрим конфигурацию PAM для sshd на практике:

```
#%PAM-1.0
auth required pam_nologin.so
auth include common-auth
account include common-account
password include common-password
session required pam_loginuid.so
session include common-session
# Enable the following line to get resmgr support for
# ssh sessions (see /usr/share/doc/packages/resmgr/README)
#session optional pam_resmgr.so fake_ttyname
```

Первый модуль, который вызывается - pam_nologin.so. Он проверяет, существует ли файл /etc/nologin. Если существует, никакой пользователь, кроме root, не сможет войти.

Типичная конфигурация PAM для приложения (sshd, в данном случае) содержит четвре включенных оператора, ображающихся к файлам конфигурации четврех типов модулей: common-auth, common-account, common-password и common-session. Эти четвре файла хранят конфигурацию по умолчанию для каждого типа модуля. Подключая их вместо добавления каждого модуля по отдельности к соответствующей конфигурации PAM, вы автоматически получаете обновленную онфигурацию PAM, если администратор изменит значению по умолчанию. Раньше было необходимо настраивать вручную все файлы конфигурации для всех приложений каждый раз, когда изменялся PAM или устанавливалось новое приложение. Сейчас же конфигурация PAM делается централизованно через конфигурационные файлы и все изменения автоматически наследуются настройками PAM каждого сервиса.

Первый включенный файл (common-auth) вызывает два модуля типа auth: pam_env.so and pam_unix2.so.

```
auth required pam_env.so
auth required pam_unix2.so
```

Первый из них, pam_env, загружает файл /etc/security/pam_env.conf для установки переменных окружения согласно указаниям в этом файле. Он может использоваться для установки переменной DISPLAY корректного значения, поскольку модуль pam_env в курсе места расположения, откуда имеет место быть попытка входа в систему. Второй из них, pam_unix2, проверяет логи и пароль пользователя согласно файлам /etc/passwd и /etc/shadow.

Весь стэк модулей auth обрабатывается прежде, чем sshd получит информацию об успешности входа в систему. Учитывая, что се модули стэка обладают контрольным флагом required, они все доолжны быть обработаны с положительным результатом до того, как sshd получит сообщение о успешном завершении.

Если один из модулей получит отрицательный результат, все равно будет обработан весь стэк и только затем sshd будет уведомлен об отрицательном результате.

Как только все модули типа auth будет успешно обработаны, будет обрабатываться другой оператор, в данном случае, common-account, содержащий только один модуль - pam_unix2. Если pam_unix2 возвращает результат, что пользователь существует, sshd получает сообщение, сообщающее об успехе, и обрабатывается следующий стэк модулей (password).

```
account required pam_unix2.so
password requisite       pam_pwcheck.so  nullok cracklib
password required       pam_unix2.so    nullok use_authtok
```

Снова конфигурация PAM для sshd вызывает включенные операторы, обращающиеся к конфигурации по умолчанию для модулей password, расположенных common-password. Эти модули должны быть успешно обработаны (флаги requisite и required) всякий раз, когда приложение запрашивает изменение маркера аутентификации.

Изменение пароля или другого маркера аутентификации требует проверки на безопасность. Это достигается с помощью модуля pam_pwcheck. Модуль pam_unix2, использующийся после этого, переносит старые и новые пароли от pam_pwcheck, таким образом, пользователю не нужно аутентифицироваться всякий раз после смены пароля. Эта процедура лишает возможности обойти проверки, выполненные pam_pwcheck. Всякий раз, когда учетная запись или тип auth настроены на отправку сообщений об истекшем сроке пароля, также используются модули password.

```
session required pam_limits.so
session required pam_unix2.so
session optional pam_umask.so
```

И в качестве последнего этапа, модули типа session, связанные в файле common-session, вызываются для настройки сессии, согласно параметрам настроки пользователя. Модуль pam_limits загружает файл /etc/security/limits.conf, в котором есть возможность определения ограничений на использование различных системных ресурсов. Модуль pam_unix2 обрабатывается снова. Модуль pam_umask может использоваться для установки маски создания режима файла. Так как этот модуль несет флаг optional, неудачная обработка этого модуля не повлияет на успешное завершение всего стэка модулей session. Модули session вызываются повторно, когда пользователь выходит из системы.

Конфигурация модулей PAM

Некоторые модули PAM есть возможность настраивать. Соответствующие файлы конфигурации расположены в /etc/security. В этом разделе кратко описываются файлы конфигурации, относящиеся к примеру с sshd - pam_env.conf и limits.conf.

pam_env.conf

Этот файл может использоваться для определения стандартизированного окружения для пользователей, которое устанавливается всякий раз, когда вызывается модуль pam_env. Вместе с этим, предварительно установленные переменные окружения используют следующий синтаксис:

```
VARIABLE [DEFAULT=[value]] [OVERRIDE=[value]]
```

VARIABLE - Имя устанавливаемой переменной окружения.

[DEFAULT=[value]] - Значение по умолчанию, которое желает установить администратор.

[OVERRIDE=[value]] - Значения, которые могут быть запрошены и устанавливаться pam_env, переопределяя значение по умолчанию.

Типичным примером того, как pam_env может использоваться, является адаптация переменной DISPLAY, которая меняется всякий раз, когда происходит удаленный вход в систему.

```
REMOTEHOST DEFAULT=localhost OVERRIDE=@{PAM_RHOST}
DISPLAY DEFAULT=${REMOTEHOST}:0.0 OVERRIDE=${DISPLAY}
```

В первой строке устанавливается значение переменной REMOTEHOST в localhost, которое используется каждый раз, когда pam_env не может определить никакое другое значение. Переменная DISPLAY в свою очередь содержит значение REMOTEHOST. Найти больше информации можно в комментариях в файле /etc/security/pam_env.conf.

pam_mount.conf

Назначением pam_mount является монтирование каталогов пользователя в процессе входа в систему и отмонтирование их в случае выхода в окружение, когда все домашние каталоги пользователей хранятся на центральном файловом сервере. При использовании этого способа нет необходимости монтировать директорию /home целиком, чтобы все каталоги пользователей были доступны. Вместо этого, монтируется только каталог соответствующего пользователя.

После установки pam_mount в /etc/security становится доступен шаблон pam_mount.conf.xml. Описание его разлличных элементов может быть найдено на справочной странице man 5 pam_mount.conf.

Основная настройка этой возможности может быть сделана с помощью yast. Выберите Network Settings > Windows Domain Membership > Expert Settings для добавления соответствующего файлового сервера.

limits.conf

Ограничения системы могут быть установлены для пользователя или для группы в файле limits.conf, который читается модулем pam_limits. Этот файл позволяет устанавливать жесткие ограничения, которые вообще не могут быть привышены, и мягкие ограничения, которые врменно могут быть привышены. Чтобы узнать о синтаксисе и доступных опциях, прочитайте комментарии, входящие в файл /etc/security/limits.conf.

Настройка PAM, используя pam-config

Инструмент pam-config поможет настроить файлы конфигурации PAM в /etc/pam.d/common-*-pc, так же, как и нескольких настроек приложений. Для получения списка всех поддерживаемых модулей используйте команду pam-config –list-modules. Используйте команду pam-config для поддержки ваших файлов конфигурации PAM. Добавьте новые модули в конфигурацию PAM, удалите другие модули или измените опции для этих модулей. При изменении глобальных файлов конфишурации PAM, нет никакой необходимости в ручной настройке PAM для отдельных приложений.

Простой реальный пример использования pam-config содержит следующее:

1. Автоматическое создание новой конфигурации PAM в Unix-стиле. Позвольте pam-config создать самую простую установку, которую при желании можно расширить позднее. Команда pam-config –create создаст простую конфигурацию аутентификации Unix. Уже существующие файлы конфигурации не будут перезаписаны, а будут сделаны резервные копии с расширением *.pam-config-backup.

2. Добавление нового метода аутентификации. Добавление нового метода аутентификации (например, LDAP) к стэку модулей PAM сводится к выполнению простой команды pam-config –add –ldap. LDAP добавляется везде, где потребуется порседством всех файлов конфигурации PAM common-*-pc.

3. Добавление отладки для тестовых целей. Для того, чтобы убедиться, что новая процедура аутентификации работает, как и было запланировано, включите режим отладки для всех операций, относящихся к PAM. Команда pam-config –add –ldap-debug включает режим отладки операций PAM, относящихся к LDAP. Вывод отладки можно найти в /var/log/messages.

4. Запрос установки. До того, как принять окончательно новую установку PAM, все ли опции, когда планировались, она содержит. Команда pam-config –query –module отобразит список как типов, так и опций запрашивающего модуля PAM.

5. Удаление опций отладки. Наконец, если вы полностью удовлетворены производительностью установку, удалите опции отладки. Команда pam-config –delete –ldap-debug отключит отладку для аутентификации LDAP. В случае, если были включены опции отладки для других модулей, можно использовать пдобные команды для их отключения.

Когда вы создаете файлы конфигурации PAM с нуля при помощи команды pam-config –create, создаются символические ссылки с common-* на файлы common-*-pc. pam-config просто изменяет файлы конфигурации common-*-pc. Удаление этих символических ссылок эффективно отключает pam-config, поскольку pam-config просто воздействует на файлы common-*-pc и эти файлы не будут действовать без символических ссылок.

Для получения более подробной информации о команде pam-config и доступных опциях обращайтесь к man-странице команды pam-config - pam-config(8).

Дополнительная информация

В директории /usr/share/doc/packages/pam установленной системы можно найти следующую дополнительную документацию:

READMEs - В этом каталоге имеются несколько общих файлов README. В поддиректории modules находятся файлы README для доступных модулей PAM.

The_Linux-PAM_System_Administrators'_Guide - Этот документ включает все, что системный администратор должен знать о PAM. В нем обсуждается диапазон тем от синтакиса файлов конфигурации до аспектов безопасности PAM. Документ доступен в виде PDF, HTML и простого текста.

The_Linux-PAM_Module_Writers'_Manual - Этот документ обобщает тему с точки зрения разработчика, с информацией о том, как писать совместимые со стандартами модули PAM. Он доступен в виде PDF, HTML и простого текста.

The_Linux-PAM_Application_Developers'_Guide - Этот документ включает ебя все, что необходимо разработчику приложений, желающего использовать библиотеки PAM. Он доступен в виде PDF, HTML и простого текста.

The_PAM_Manual_Pages - PAM в целом, так же как и отдельные модули поставляются со страницами руководств, которые обеспечивают хороший краткий обзор функциональности, обеспеченной соответствующим компонентом.

Торстен Кукук разработал множество модулей PAM и выложил некоторую информацию на http://www.suse.de/~kukuk/pam/.