Впервые опубликовано на developerWorks 24.05.2006

Подготовка к экзамену LPI: Управление клиентом сети

Средний уровень администрирования (LPIC-2) тема 210

Введение

DHCP

Dynamic Host Configuration Protocol (DHCP) -- это преемник более старого протокола BOOTP. Главная роль DHCP-сервера состоит в присвоении клиентским машинам, которые могут соединяться или рассоединяться с сетью, IP-адресов. Большинство IP сетей, даже при наличии стабильной топологии и клиентских списков, используют DHCP, чтобы исключить конфликты при распрелелении IP-адресов.

Кроме этого, DHCP-сервер обеспечивает клиентов информацией о маршрутизации и о подсети, адресами DNS, и иногда и другой информацией. DHCP присваивания могут иметь различную длительность, как малую, так и бесконечно большую, в зависимости от конфигурации сервера и деталей запроса клиента. На самом деле DHCP совместим и с присваиванием фиксированных IP-адресов некоторым машинам (посредством их аппаратных MAC адресов), но в любом случае конфликты между машинами предотвращаются.

Формальной спецификацией DHCP является RFC 2131.

О NIS

Network Information Service (NIS) -- это протокол клиент-серверной службы каталога от Sun Microsystems' "Yellow Pages" (YP) для распространения данных системных конфигураций, таких как пользователи и имена хостов в компьютерной сети.

NIS/YP использует центральный каталог для хранения а пользователей, имен хостов, и многих других полезных в компьютерной сети вещей. Например, в большинстве сред UNIX, список пользователей (для идентификации) размещается в /etc/passwd. При использовании NIS добавляется другой "глобальный" список пользователей, который используется для идентификации пользователей на каждой машине.

В большинстве своем NIS вытесняется более общей и более безопасной LDAP, пригодной для более широкого использования.

Хорошим началом для добывания большей информации о NIS -- это "The Linux

NIS(YP)/NYS/NIS+ HOWTO"

О LDAP

Lightweight Directory Access Protocol (LDAP) -- клиент-серверный протокол для доступа к службам каталога, в обобенности основанных на X.500.

Каталог LDAP похож на базу данных, но склоняется к содержанию более подробной, аттрибутивной информацию. Поэтому LDAP обеспечивает достаточную гибкость для хранения любого вида информации, разделяемое по сети. Информация в каталоге считывается намного чаще, нежели записывается туда, так что она настроена на быстрый отклик при операции поиска или при запросах большого объёма данных.

LDAP имеет способность широко реплицировать информацию, чтобы повысить ее доступность и надежность, при этом уменьшая время доступа. При репликации информации из каталога всякие временные несоответствия между репликами будут синхронизоваться.

Формальной спецификацией LDAP являкется RFC 2251

О PAM

Linux-PAM (Pluggable Authentication Modules for Linux) -- это комплект разделяемых библиотек, которые позволяют локальному системному администратору выбирать, как приложения идентифицируют пользователей.

Использующее PAM приложение может в процессе выполнения переключаться между различными механизмами идентификации. Кроме того, можно полностью обновить локальную систему идентификации, не перекомпилируя приложения. Эта библиотека PAM сконфигурирована локально при помощи системного файла, /etc/pam.conf (или совокупностью конфигурационных файлов, расположенных в /etc/pam.d/) для идентификации запроса пользователя через локально доступные модули идентификации. Модули сами по себе обычно располагаются в каталоге /lib/security и имеют вид динамически загружаемых объектных файлов.

Руководство Linux-PAM System Administrators -- хорошее начало для получения дополнительной информации

Другие ресурсы

Как и другие приложения Linux, всегда полезно обращаться к man-страницам любых рассматриваемых здесь утилит. Linux Documentation Project содержит большое количество различных полезных документов, в особенности HOWTO. Также опубликовано множество книг по сетям Linux; я считаю, что книга O'Reilly's TCP/IP Network Administration, Крейга Ханта, вполне может помочь.

Конфигурация DHCP

Общее о протоколе

Подобно большинству сетевых протоколов, Dynamic Host Configuration Protocol (DHCP) -- это клиент/серверный интерфейс. Клиент DHCP является намного более простой программой, как по устройству, так и для настройки, нежели сервер DHCP. В сущности, вся работа клиента DHCP состоит в рассылке сообщения DHCPDISCOVER по локальной физической подсети, и ожидания ответа.

Сообщение DHCPDISCOVER может содержать параметры, которые содержат желаемое значение для сетевого адреса и продолжительности аренды. Если серверы получает сообщение DHCPDISCOVER, то он должен ответить соответствующему MAC адресу сообщением DHCPOFFER. Клиент, в свою очередь, отвечает сообщением DHCPREQUEST одному из предлагающих серверов, обычно первому (и единственному) серверу, приславшему ответ.

Главные параметры конфигурации, которые использует клиент, он получает в сообщении DHCPACK. При этом клиент получает собственный IP-адрес и с этого момента может взаимодействовать не только на уровне Data Link Layer (Ethernet), но и на уровне Network Layer (IP) как полноценный участник IP-сети.

Процесс на клиенте

Клиент DHCP обычно требует только настройки информации, которую он желает получать. Например, дистрибутив на основе Debian обычно использует клиента DHCP, dhclient, который сконфигурирован файлом /etc/dhcp3/dhclient.conf. Образец файла, распространяемые с пакетом dhcp3-client, имеет все опции, кроме одной, закомментированными. Единственная задействованная опция, скорее всего, имеет вид:

Listing 1. Опции для dhclient.conf

```
request subnet-mask, broadcast-address, time-offset, routers,
  domain-name, domain-name-servers, host-name,
  netbios-name-servers, netbios-scope;
```

В этом примере, конфигурации по умолчанию, клиент, по существу, говорит: "спрашивайте все, что возможно". В переговорных сообщениях, сообщение DHCPACK от сервера будет содержать информацию для всех таких запрошенных значений, которые клиент сможет использовать по мере получения. IP-адрес клиента содержится в списке, поскольку эта конфигурация всегда есть предмет

переговоров.

Наряду с указанием максимального времени и параметрами времени аренды (а также некоторыми другми), клиент может, но, в большинстве случаев, не обязан, наложить некоторые ограничения на IP-адрес, который он хочет использовать. Для того, чтобы исключить определенный адрес, можно записать `reject 192.33.137.209;`. Для указания явного адреса, который хочет использовать клиент, следует записать `fixed-address 192.5.5.213;`.

Клиент может отклонить предложение аренды сообщением DHCPDECLINE, но сервера попытаются удовлетворить запросы везде, где это возможно. DHCP-сервер может также сделать фиксированное присваивание определенного IP-адреса к запрашиваемому MAC-адресу; настройка IP-адресов по машинам чаще производится конфигурацией сервера, чем конфигурацией клиента.

Чтобы отслеживать полученные аренды, dhclient хранит список аренд, которые были присвоены в файле /var/lib/dhcp3/dhclient.leases (путь может отличаться в различных дистрибутивах); таким образом, не просроченная DHCP аренда не теряется, даже если система отсоединяется от физической сети и/или перезагружается.

Процесс на сервере

Сервер DHCP должен знать немного больше своих опций, поскольку он обеспечивает клиентов различной информацией во время DHCP аренды, и также должен быть уверен, что клиенты получили уникальные IP-адреса. Сервер DHCPобычно запускается как демон, dhcpd, и черпает информацию о своей конфигурации из /etc/dhcpd.conf (этот путь может варьироваться, в зависимости от дистрибутива). Один демон dhcpd может, тем не менее, поддерживать несколько подсетей, обычно когда к серверу подсоединяются многие физические сети; однако зачастую все-таки один сервер управляет одной подсетью. Listing 2 -- фактически полный пример конфигурации сервера.

Listing 2. Конфигурационные опции dhcpd.conf

```
# default/max lease in seconds: day/week
default-lease-time 86400;
max-lease-time 604800;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.2.255;
option routers 192.168.2.1;
# DNS locally, fallback to outside local domain
option domain-name-servers 192.168.2.1, 151.203.0.84;
option domain-name "example.com";
# Set the subnet in which IP address are pooled
subnet 192.168.2.0 netmask 255.255.255.0 {
   range 192.168.2.100 192.168.2.199;
}
# Assign a fixed IP to a couple machines by MAC address
group {
   use-host-decl-names true;
   host first.example.com {
      hardware ethernet 00:00:c1:a2:de:10;
      fixed-address 192.168.2.200;
   }
```

```
host second.example.com {
   hardware ethernet 00:dd:cc:a1:78:34;
   fixed-address 192.168.2.201;
}
```

Когда клиент высылает широковещательное сообщение серверу с такой конфигурацией, он либо получит в аренду 192.168.2.200, или 192.168.2.201, если он имеет указанный MAC-адрес, либо получит в аренду доступный адрес в диапазоне с 192.168.2.100 по 192.168.2.199.

Клиент также может использовать сообщение DHCPINFORM, чтобы сообщить серверу, что он уже имеет назначенный (вручную) IP-адрес, но хотел бы получить другую конфигурационную информацию. Обратите внимание, что сообщать серверу о том, что клиент использует определенный IP-адрес -- совсем не то же, что запрашивание определенного IP-адреса; в последнем случае сервер может дать согласие на запрос, а может и не дать, в зависимости от существующих арендах. А в первом случае сервер не не имеет голоса в принятии решения, и, по сути, аренда не предоставляется (однако сервер попытается избежать присваивания новым клиентам тех IP-адресов, которые используются).

Когда истекает срок действия аренды, клиенты и сервера должны договориться о новых арендах, а параметры конфигурации остаются в силе. Более короткие аренды могут использоваться там, где конфигурационная информация на сервере вероятно изменяется (например, на динамическом DNS посредством внешнего WAN). Клиент может элегантно прекратить аренду путем послания сообщения DHCPRELEASE, но для корректной работы не требуется и этого (клиенты иногда сбоят, перезагружаются, или, в конце концов, разъединяются и не могут послать такое сообщение).

При отсутствии высвобождающего сообщения аренда поддерживается сервером, пока истекут условия, на которых она была получена, так что перезагруженная машина зачастую продолжает использовать свою предшествующую аренду (которая хранится в dhclient.leases как на сервере, так и на клиенте).

Конфигурация NIS

Когда надо использовать NIS

Большая часть утилит, связанных с NIS, все еще имеют префикс yp, из-за того, что она была первоначально известна как "Sun Yellow Pages"; проблемы с использованием торговой марки вынудили сменить имя на NIS. NIS используется, когда нужно совместно использовать информацию, такую как пользователи и группы (содержимым /etc/passwd и /etc/group, соответственно), внутри сети, предлагая пользователям права доступа на любой машине внутри домена NIS.

NIS действует способом, аналогичным тому, как DNS определяет домены, где распределена информация, и разрешает master и slave серверам иерархически распределять информацию внутри домена. Фактически, NIS можно использовать вместо DNS, путем распределения информации о доменном имени, обнаруженной в

/etc/hosts, но на практике так делают редко. NIS присуща определенная гибкость, так что любой тип информации может быть, в принципе, помещен в базу данных NIS (которая имеет формат DBM, и хотя утилита makedbm из пакета NIS сервера преобразовывает простые файлы в этот формат, как правило, "за сценой").

Также существует служба, которая называется NIS+, предназначенная заменить NIS, включая шифрование данных и идентификацию; однако NIS+ не имеет обратной совместимости с NIS и не является широко используемым.

Перед тем, как начать

Для запуска любой из утилит NIS, необходимо запустить демона /sbin/portmap, который переводит номера программ RPC в номера портов протокола TCP/IP (или UDP/IP), поскольку клиенты NIS делают вызова RPC. Ьольшинство дистрибутивов Linux запускают /sbin/portmap в своих скриптах инициализации системы, однако следует проверить, что этот демон запущен, с помощью

```
% ps -ax | grep portmap.
```

Если он еще не выполняется, установите /sbin/portmap и внесите его в скрипты запуска вашей системы.

Утилиты клиента NIS (демон ypbind)

Клиент NIS содержит утилиты **ypbind**, **ypwhich**, **ypcat**, **yppoll** и **ypmatch**. Демон ypbind должен быть запущен от root, и обычно он запускается одним из скриптов запуска системы (хотя это и не обязательно).

Другие утилиты зависят от служб ypbind, но выполняются на пользовательском уровне. Старые версии ypbind рассылают по локальной сети запрос на соединение, но это позволяет злонамеренному NIS-серверу отвечать на запрос и давать клиентам неправильную имформацию о пользователях и группах. Предпочтитается с помощью файла /etc/yp.conf настраивать определенные серверы, к которым может подключиться ypbind. Когда настроено несколько серверов (или когда, несмотря на опасность, используется рассылка), ypbind может переключаться между серверами каждые 15 минут, в зависимости от того, кто из них откликается быстрее всего. Эти разные сервера должны быть связаны друг с другом как master/slave, клиенту не нужно ни знать не заботиться об этом. Например, конфигурация ypbind может иметь такой вид:

Listing 3. /etc/yp.conf

```
      ypserver 192.168.2.1
      ypserver 192.168.2.2
      ypserver 192.168.1.100
      ypserver 192.168.2.200
```

Перед запуском /usr/sbin/ypbind, следует задать вашей сети доменное имя NIS. Оно может быть любым именем, на использование которого настроен NIS, и, вообще говоря, должно отличаться от доменного имени. Это доменное имя NIS задается примерно так: `% ypdomainname my.nis.domain`.

Утилиты для клиента NIS (другие конфигурации)

Если вы желаете использовать NIS для поиска доменного имени, следует изменить /etc/host.conf так, чтобы включить NIS в порядок поиска; например, чтобы проверить имя сначала в /etc/hosts, затем в NIS, а затем в DNS:

Listing 4. Изменение порядка lookup

```
% cat /etc/host.conf
order hosts,nis,bind
```

Для подключения NIS-распределенных пользователей, следует изменить клиентский файл /etc/passwd, добавив туда +::::::.

Информация базы данных NIS ведет себя как шаблон для попыток входа с такой "незаполненным" содержимым. При желании можно настроить информацию о пользователе, например, так:

Listing 5. Подробный /etc/passwd

```
+user1:::::::
+user2:::::::
+user3:::::::
+@sysadmins:::::::
-ftp
+:*::::::/etc/NoShell
```

Это даст разрешение на вход для `user1`, `user2` и `user3`, а также всем членам сетевой группы sysadmin, и обеспечивает хранение учетных записей всех пользователей в базе данных NIS.

Источники NIS настаиваются в /etc/nsswitch.conf. Имя этого файла предполагает, что он предназначен строго для сервера имен, однако там описывается разные типы данных. В основном эта конфигурация описывает порядок, в котором происходит поиск источников информации. Имя `nis` в этой последовательности означает, что информацию получают с NIS-сервера; имя `files` значит, что нужно использовать соответствующий локальный файл конфигурации. Имя `dns` используется для опции `hosts`.

Кроме того, можно указать, что делать, если начальные источники не содержат требуемой информации: `return` означает отказ (а если NIS еще доступен, `continue` значит, что нужно попробовать обратиться за данными к следующему

источнику). Например:

Listing 6. /etc/nsswitch.conf

```
      passwd:         compat
      group:          compat
      shadow:         compat
      hosts:          dns [!UNAVAIL=return] files
      networks:       nis [NOTFOUND=return] files
      ethers:         nis [NOTFOUND=continue] files
      protocols:      nis [NOTFOUND=return] files
      rpc:            nis [NOTFOUND=return] files
      services:       nis [NOTFOUND=return] files
```

Утилиты пользователя NIS-клиента

Утилиты **ypwhich**, **ypcat**, **yppoll**, и **ypmatch** используются на пользовательском уровне для запросов информации NIS:

- ypwchich выдает имя NIS-сервера.
- ypcat выдает значения всех ключей базы данных NIS.
- yppoll выдает версию и master server карты NIS.
- ypmatch выдает значения одного или более ключей из карты NIS.

За более подробной информацией по каждой утилите обратитесь к соответствующим man-страницам.

Утилиты сервера NIS (ypinit, ypserv)

Для предоставления клиентам баз данных NIS NIS-сервер использует демона ypserv, который сконфигурирован файлом /etc/ypserv.conf. Как уже упоминалось, внутри домена можно запускать два NIS-сервера, master и slave. Набор баз данных NIS инициализируется на master сервере (только во время первого запуска; после этого используется `make -C /var/yp`) с помощью команды:

```
 % /usr/lib/yp/ypinit -m.
```

slave сервер в действительности -- просто NIS-клиент, который получает базы данных от master сервера и выполняет ypserv. Чтобы скопировать информацию с master сервера на локально запущенный slave сервер, выполняют

```
% /usr/lib/yp/ypinit -s my.nist.domain.
```

На master сервере, базы данных NIS составляется из данных, размещенных в (некоторых из) следующих знакомых конфигурационных файлов:

- /etc/passwd,
- /etc/group,

- /etc/hosts,
- /etc/networks,
- /etc/services,
- /etc/protocols,
- /etc/netgroup,
- /etc/rpc.

То, какие базы данных экспортируются, задается в /var/yp/Makefile, который также распространяет изменения при новой сборке.

Slave серверы оповещаются (посредством программы yppush) о любых изменениях в картах NIS, если они пересобираются, и автоматически возвращают необходимые изменения, чтобы синхронизировать свои базы данных. NIS-клиентам не требуется делать этого, поскольку они непрерывно общаются с NIS-сервером и считывают информацию, хранящуюся в его DBM базах данных.

Конфигурация LDAP

Когда используется LDAP

В принципе, по своим целям Lightweight Directory Access Protocol похож на NIS. Оба распределяют некоторую стуктурированную информацию о настойках сети от клиента к серверу; однако LDAP идет дальше в иерархическом структурировании того, каким клиентам какую информация нужно предоставлять, перенаправляя запросы другим LDAP-серверам, куда необходимо, и выстраивая механизмы безопасности. Более того, LDAP дает клиентам механизмы и утилиты для обновления информации, содержащейся на LDAP-серверах, которые, в свою очередь, распространяют эти данные тем другим клиентам, что будут запрашивать их (конечно, это зависит от прав доступа).

Установка

Перед запуском OpenLDAP (the Free Software реализация, широко используемая на Linux, хотя существуют и коммерческие разработки), вам следует установить или проверить, установлены ли следующие необходимые библиотеки:

- OpenSSL Transport Layer Security (TLS) можно получить на OpenSSL Project (или через способы установки вашего дистрибутива Linux).
- Поддержка Kerberos Authentication Services необязательна, но их наличие очень желательно. Пойдет либо MIT Kerberos, либо Heimdal Kerberos.
- Simple Authentication и Security Layer может быть установлена как часть основного дистрибутива, но также может быть получено Cyrus SASL as well.
- Рекомендуется иметь Sleepycat Software Berkeley DB, хотя, вероятно, другие реализации DBM тоже подойдут.
- Posix нити и TCP wrappers если и не обязательны, то желательны.

После того, как эти предварительные условия соблюдены, скачайте библиотеку OpenLDAP и совершите более-менее привычный танец:

Listing 7. Обычное камлание для установки OpenLDAP

```
% ./configure
% make depend
% make
% make test  # make sure things went OK
% su root -c 'make install'
```

После базовой установки, следует настроить конфигурацию slapd, обычно расположенную в /usr/local/etc/openldap/slapd.conf. Установка должна содержать компоненты вашего домена:

Listing 8. Компоненты домена, включаемые в slapd.conf

```
database bdb
suffix "dc=eng,dc=uni,dc=edu,dc=eu"
rootdn "cn=Manager,dc=eng,dc=uni,dc=edu,dc=eu"
rootpw <secret>
directory /usr/local/var/openldap-data
```

Для того, чтобы найти значение `<secret>` используйте утилита `slappasswd`, а затем берите эту base64-зашифрованную строку в качестве вашего "`<secret>`":

Listing 9. Раскрытие вашего "секрета"

```
% slappasswd
New password: *******
Re-enter new password: ********
{SSHA}YzPqL5Jio2+17NFIy/pAz8pqS5Ko13fH
```

За большей информацией о правах доступа, репликации и других опциях, которые задаются в slapd.conf, обратитесь к manpages.

Запуск демона slapd daemon весьма схож с запуском любого другого демона; проверить, запущен ли он, можно с помощью `ldapsearch`:

Listing 10. Проверка на то, запущен ли slapd

```
su root -c /usr/local/libexec/slapd
ldapsearch -x -b '' -s base '(objectclass=*)' namingContexts
```

Если все прошло успешно, вы увидите что-нибудь вроде этого:

Listing 11. Отклик работающего slapd

```
dn:
namingContexts: dc=eng,dc=uni,dc=edu,dc=eu
```

Добавление данных в файл LDIF

Формат данных, используемых в LDAP, является двоичным, а для экспорта и импорта данных в базу данных LDAP используется ASCII сериализация, которая называется LDAP Data Interchange Format (LDIF). Двоичные данные в LDIF представляются base64 кодированном виде. OpenLDAP содершит утилиты, экспортирующие данные от LDAP-сервером в LDIF (**ldapsearch**), импортирующие данные из LDIF на LDAP-серверы (**ldapadd**), и применяющие набор изменений, описанных в LDIF, к LDAP-серверам (**ldapmodify** и **ldapdelete**).

Более того, LDIF -- один из форматов, использующихся в импортировании и экспортировании данных адресной книги для Mozilla Application Suite и других пользовательских программ прикладного уровня. Даже Microsoft Windows 2000 Server и Windows Server 2003 содержат утилиту LDIF, **LDIFDE**, для передачи данных из Active Directory и обратно.

Чтобы вручную добавить информацию на LDAP-сервер, сеачала создайте LDIF-файл:

Listing 12. Создание примерного файла LDIF, example.ldif

```
dn: dc=example,dc=com
objectclass: dcObject
objectclass: organization
o: Example Company
dc: example

dn: cn=Manager,dc=example,dc=com
objectclass: organizationalRole
cn: Manager
```

Затем следует выполнить

```
% ldapadd -x -D "cn=Manager,dc=example,dc=com" -W -f example.ldif
```

, чтобы добавить его.

Очевидно, вам нужно поменять имя домена example.com на ваше. Как правило, структуры и имена доменов LDAP совпадают с обычными DNS именами. Вам понадобится указать значение `rootpw`, которое вы указали в slapd.conf.

Выполнение запросов в базах данных LDAP

Имеется утилита **slurpd** (Standalone LDAP Update Replication Daemon), реплицирующая всю информацию базы данных; однако для отдельных данных

используется либо программа типа ldapsearch, либо, что более вероятно, поддержка LDAP, встраивается в некоторые приложения, которые запускает пользователь. Утилита **slapcat** также пригодна для сброса базы данных LDAP в LDIF. Например, многие Mail User Agents (MUAs) могут использовать LDAP для выделения адреса и контактной информации.

Внутри приложений, включая и те, что вы напишете самостоятельно, используя компилируемые или скриптовые языки, к ресурам LDAP можно обращаться при помощи специальных URL для LDAP. Они имеют вид l`dap://host:port/dn?attr ibutes?scope?filter?extens ions`.

Большая часть этих полей является необязательными. Имя хоста по умочанию -- это localhost; порт, использующийся по умолчанию, 389. Умолчательное значение имени, распознаваемого как root, -- пустая строка. Если требуется информация для идентификации, она указывается в добавочных частях URL.

Вдобавок к LDAP URL, многие серверы и клиенты LDAP также поддерживают нестандартные, но все-же широко используемые LDAPS URL. LDAPS URL используют соединения SSL вместо обычных текстовых соединений, и их портом по умолчанию является 636:
`ldaps://host:port/dn?attributes?scope?filter?extensions`.

Идентификация PAM

Когда используется PAM

Первое, что следует уяснить о Pluggable Authentication Modules (PAM) -- это, что это не есть ни приложение, ни протокол. Правильнее говорить, что это -- коллекция библиотек, которые могут включаться в приложения во время компиляции, чтобы использовать функции этих модулей. Если в приложении включены PAM, политика безопасности этого приложения может задаваться системным администратором, без изменения или обновления самого приложения. Многие утилиты Linux, в особенности демоны и сервера могут использовать PAM.

Быстрый способ выяснить, является ли данное приложение возможно использующим PAM, состоит в использовании `ldd`, чтобы выяснить, какие библиотеки используются. Например, мне интересно, является ли использующей PAM программа, выполняющая вход в систему:

Listing 13. Вход PAM-задействован?

```
% ldd /bin/login | grep libpam
libpam.so.0 => /lib/libpam.so.0 (0xb7fa8000)
libpam_misc.so.0 => /lib/libpam_misc.so.0 (0xb7fa5000)
```

Использование `libpam.so` и `libpam_misc.so` программой login не полностью

гарантирует, что средства PAM действительно используются этими утилитами, и используются правильно, но это -- хороший знак. Подобным образом, мне, возможно, интересно узнать то же про мои сервера Apache и FTP:

Listing 14. А как насчет серверов Apache и FTP?

```
% ldd /usr/sbin/apache2 | grep libpam
% ldd /bin/login | grep libpam
libpam.so.0 => /lib/libpam.so.0 (0xb7fa8000)
libpam_misc.so.0 => /lib/libpam_misc.so.0 (0xb7fa5000)
```

Теперь я точно знаю, что моя сборка Apache не поддерживает PAM (хотя есть версии, для которых это не так).

Для более тщательной проверки, работают ли PAM с данным приложением, для данной программы можно создать конфигурационный файл PAM. Например, для проверки утилиты входа следует создать файл /etc/pam.d/login (однако обратите внимание, что он, возможно, уже имеется в вашей системе, и содержит более важные настройки, так что не удаляйте имеющийся файл):

Listing 15. Проверка входа на PAM с помощью etc/pam.d/login

```
auth      required    pam_permit.so
auth      required    pam_warn.so
```

Теперь выполнение правильного поддерживающего PAM `login` позволит осуществит вход, но при этом занесет действия в системный журнал. Если syslog покажет вход, значит для этого приложения PAM задействован. Читатель, наверно, заметит, что это -- наверно, худшая конфигурация, которую можно придумать для `login`, так как теперь он даст доступ к оболочке каждому. Подметив это, уясните, что настраивать PAM следует с определенной осторожностью.

Настройка PAM

PAM работает с двумя видами конфигурационных файлов.

Предпочтительный способ настройки PAM состоит в использовании файлов из каталога /etc/pam.d/, которые имеют те же имена, что и службы, чья безопасность является желаемой. Более старый и менее рекомендуемый способ состоит в использовании одного файла, /etc/pam.conf, для указания политики безопасности для всех приложений. С точки зрения возможности поддержки работы, создание по конфигурационному файлу на приложение проще, вместо файлов можно создать символические ссылки для "копирования" политика безопасности от одного приложения на другое. Оба стиля конфигурации в основном совпадают. Единый файл /etc/pam.conf содержит строки вида:

Listing 16. Оба конфмигурационных файла содержат

```
<service>  <module-type>  <control-flag>  <module-path>  <args>
```

В конфигурационных файлах отдельных приложений первое поле опускается, поскольку оно такое же,как и имя файла. Тестовая конфигурация входа, которую мы видели, в старом стиле выглядит как:

Listing 17. /etc/pam.conf

```
      login     auth      required    pam_permit.so
      login     auth      required    pam_warn.so
```

Поле `<module-type>` может принимать одно из четырех значений:

- auth (аутентификация),
- account (неидентификационные права доступа, основанные на состоянии пользователя в системе),
- session (совершать действия до/после запуска службы), and
- password (обновление tokenов идентификации пользователя).

Поле `<control-flag>` используется для "stack" модулей, которые позволяют вам вести довольно утонченный контроль того, когда этот метод требуется, требуется ли он вообще, и когда принимается какой-нибудь другой исход. Эти опции таковы:

- required,
- requisite,
- sufficient,
- optional, and
- include.

Я расскажу об этом ниже.

Поле `<module-path>` уже появлялось в примерах. Оно обозначает разделяемую библиотеку - либо указывая явный путь к ней (если значние начинается с "/", либо только имя (при этом сама библиотека ищется в каталогах по умолчанию). Например, в Listing 17, можно было бы написать `/lib/security/pam_warn.so`. Полем `<args>` может быть все, что угодно, в зависимости от того, какой именно модуль требуется для настройки этой операции, хотя несколько аргументов общего типа должны поддерживаться большинством модулей PAM. Обратите внимание, что модули PAM являются расширяемыми. Если кто-нибудь пишет новый модуль PAM, его можно просто поместить в /lib/security и все ваши приложения смогут использовать его, как только их конфигурационный файл будет обновлен.

Пример прав доступа PAM.

Чтобы увидеть, как работает `<control-flag>`, давайте сконструируем довольно сложный пример. Первое, что следует сделать -- это создать специальную службу OTHER. Если это сделано, и для этой службы не определена никакая политика PAM, используется политика OTHER. Безопасное default будет пожоже на Listing 18:

Listing 18. /etc/pam.d/other

```
      auth      required    pam_warn.so
      auth      required    pam_deny.so
      @include safe-account
      @include safe-password
      @include safe-session
```

В этом примере, попытка идентификации сначала журналируется, а затем получает отказ. `@include` просто включает содержимое какого-нибудь файла, например /etc/pam.d/safe-account и подобных, в то время как эти "безопасные" определения должны содержать похожие предупредить-и-отклонить инструкции для других `<module-type>`-ов.

Теперь давайте настроим доступ к нашему гипотетическому секретному-db приложению. Будучи довольно озабоченным доступом, пользователь должен представить либо соответствующий отпечаток сетчатки глаза, либо отпечаток пальца, либо же ввести пароль. Пароль, однако, должен храниться либо в локальных конфигурациях /etc/passwd и /etc/shadow, или быть доступными посредством одного или двух внешних серверов баз данных.

Ни один из модулей безопасности, которые приводятся в этом примере, в действительности не существуют (насколько мне известно), кроме `pam_unix.so`, который есть доступ к паролю в устаревшем UNIX-стиле.

Listing 19. /etc/pam.d/classified-db

```
      auth      optional    pam_unix.so
      auth      optional    pam_database1.so    try_first_pass
      auth      optional    pam_database2.so    use_first_pass
      auth      requisite   pam_somepasswd.so
      auth      sufficient  pam_fingerprint.so  master=file1 7points
      auth      required    pam_retinaprint.so
```

Путь через эту конфигурацию весьма сложен. Вначале мы пытаемся идентифицировать пароль тремя `optional` типами модулей. Поскольку они `optional` (необязательны), отказ одного ни останавливает процесса идентификации, ни удовлетворяет его. Сначала пробуется стандартный пароль UNIX (пользователя просят ввести пароль). После этого пароль проверяется в database1, но вначале используется общий аргумент модуля `try_first_pass`, чтобы убедиться, совпадает ли пароль UNIX с паролем в базе данных; дополнительный пароль вводится только в случае несовпадения. Для базы данных `database2`, однако, мы лишь пытаемся идентифицировать, используя пароль UNIX, введенный пользователем (общий аргумент `use_first_pass`).

После проверки пароля в нескольких опциональных (`optional`) системах паролей, у нас есть гипотетический `pam_somepasswd.so`, которому нужно определить, увенчалась ли успехом какая-нибудь из ранее проведенных проверок паролей

(возможно, с использованием семафоров; но вопрос, как это делать остается открытым для гипотетических модулей). Но поскольку эта проверка является `requisite` (необходимый), то если он не срабатывает, дальнейшая идентификация не производится, и докладывается о неудаче.
Последние две идентификационные проверки (если дело доходит до них) являются `sufficient`. То есть, удовлетворение одной из них возвращает вызываемому приложению состояние успеха. То есть вначале мы делаем проверку отпечатка пальца, используя `pam_fingerprint.so` (обратите внимание, что гипотетические аргументы передаются модулю). И только если здесь будет неудача -- может быть, из-за отсутствия сканера отпечатков пальцев, или же из-за плохого отпечатка -- предпринимается попытка сканировать сетчатку глаза. Далее, если сканирование сетчатки имело успех, этого `sufficient` (достаточно). Однако, чтобы продемонстрировать все возможные опции, мы использовали также `required`, что означает, что даже если сканирование сетчатки дало успех, следует продолжить проверку другими методами (но в примере других нет, так что `sufficient` будет делать то же самое).

Также существует способ указать еще более тонко настроенные составные флаги для `<control-flag>` при помощи взятых в скобки `[value1=action1 ...]` множеств, но обычно хватает основных ключевых слов.