Визитка

ЕВГЕНИЙ БУШКОВ, системный администратор в Passware

Использование pGina
для аутентификации в системах MS Windows. Часть 2

В первой части статьи я описал основные настройки pGina и модуля LDAP Auth. Сейчас мы рассмотрим, как управлять доступом к компьютеру с pGina с помощью групп LDAP, а также познакомимся с другим плагином SSHAuth

Управление доступом к компьютеру с pGina

Любая операционная система Microsoft (рабочая станция или сервер) начиная с Windows XP/2003 имеет службу для удаленного управления (Terminal Services). Начиная с Windows Server 2008 R2 эта служба сменила название на Remote Desktop Services. для определенности будем считать любые рабочие станции и серверы под управлением Windows терминальными серверами (ТС). в данной статье в качестве примера я использую сервер с установленной Windows Server 2008 R2.

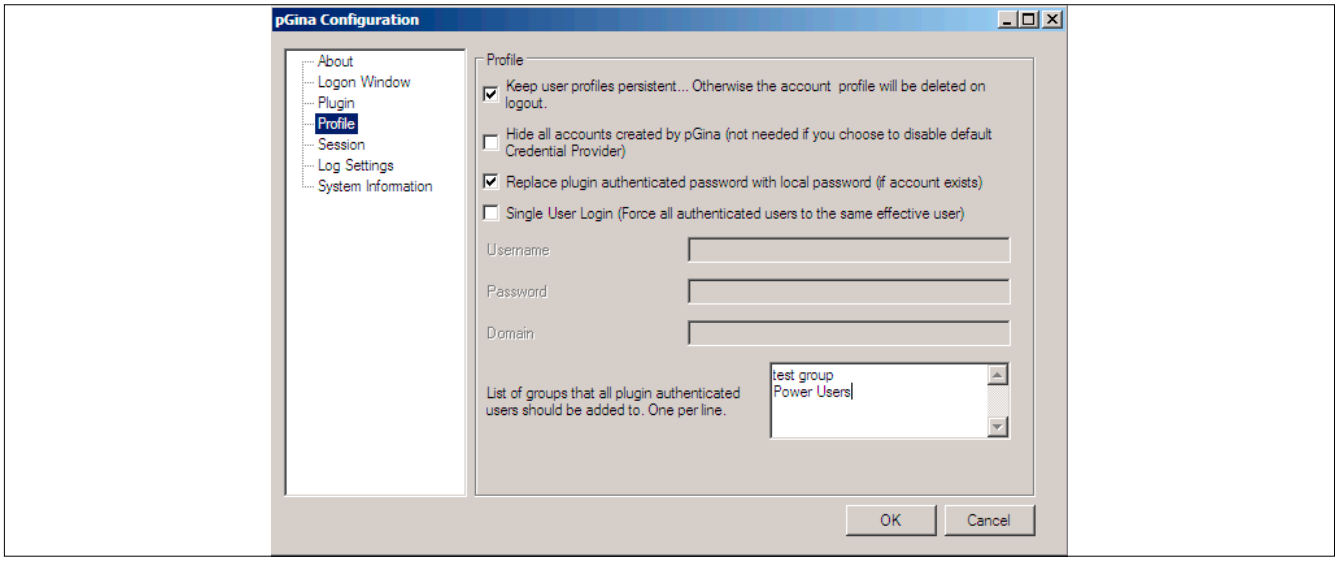
Для того чтобы пользователи могли подключаться к ТС с установленным сервисом аутентификации pGina по RDP-протоколу, используя программу Remote Desktop Connection, необходимо разобраться, как дать минимально необходимые права тем, кому разрешено работать удаленно, и как отказать в доступе всем остальным. Также нужно понять, как управлять правами на разделяемые ресурсы.

Далее я остановлюсь на четырех моментах и подробнее опишу, что может нам помочь в задаче управления правами ТС.

Первый момент

Чтобы получать доступ к разделяемым папкам (shared folders) Windows-сервера с установленными pGina и Auth LDAP, LDAP-аккаунтам нужно назначить доступ групп и пользователей к этим ресурсам так, как это делается обычно в среде Windows. Проблема здесь в том, что pGina не используется для аутентификации и проверки доступа к таким ресурсам. а обратиться к ресурсу от имени несуществующего аккаунта пользователя нельзя, то есть он должен быть вначале создан на компьютере Windows до того, как им можно будет воспользоваться. при заходе с помощью Remote Desktop Connection на сервер учетная запись пользователя может и не существовать на вашей Windows-системе – достаточ-

Рисунок 1. Входящие пользователи могут автоматически становиться членами определенных групп



но иметь рабочий логин LDAP. в случае успешной аутентификации и авторизации ее профиль будет создан, пароль задан. в описании нового аккаунта в поле Description будет запись pGinaUser – так вы сможете отличать созданные локально аккаунты от автоматически созданных pGina. Если такой пользователь уже был в системе, то его пароль будет автоматически синхронизирован с паролем из LDAP. После чего уже он сможет обращаться к разделяемым папкам этого сервера под своим логином и паролем.

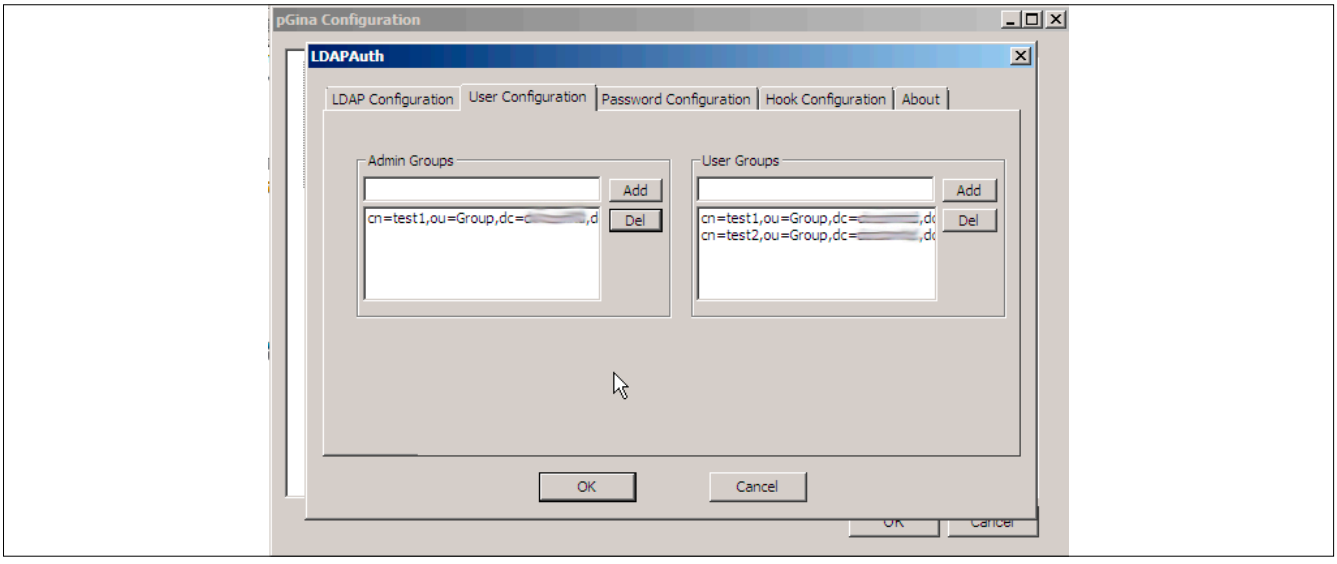
Второй момент

Для начала проверьте, что удаленный доступ вообще разрешен на терминальном сервере с pGina. для этого откройте окно Server Manager, найдите в нём ссылку Configure Remote Desktop. Если хотите разрешить доступ любым клиентам RDP, то в открывшемся окне System Properties на вкладке Remote выберите пункт Allow connections from computers running any version of Remote Desktop. Это позволит работать с удаленным доступом разным версиям Remote Desktop, программе rdesktop, а также другим RDP-клиентам. Проверьте также, что встроенный сетевой экран Windows (firewall) имеет правило, разрешающее соединения RDP: в окне Server Manager выбрать Configuration – Windows Firewall and Advanced Security – Inbound Rules. Далее найдите правило Remote Desktop (TCP-In) и, если в поле Enabled установлено значение No, нажмите справа мышкой Enable Rule.

Третий момент

После того как мы проверили, что удаленный вход в систему разрешен, нужно определить, как назначать необходимые пользователям права для соединения. по умолчанию разрешение работать с удаленным доступом имеют только члены группы Administrators. Кроме того, такие права могут иметь члены группы Remote Desktop Users. Следовательно, авторизацию можно организовать добавлением нужных пользователей в эту группу. Если пользователь уже имеет учетную запись в системе, то ее можно добавить в список членов группы Remote Desktop Users, а если не имеет, то нечего добавлять – его учетную запись нужно вначале создать. Она создается при первом успешном входе пользователя в систему, поэтому нам необходимо сделать так, чтобы пользователь, не имеющий учетной записи, после успешного прохождения аутентификации автоматически добавлялся в эту группу, тем самым получая способность успешно авторизоваться в системе. Как один из вариантов решения, можно добавить в группу Remote Desktop Users группу Authenticated Users, тогда при успешной аутентификации LDAP аккаунт получит доступ в систему. Решение не является безопасным (мы дали возможность входа на ТС всем пользователям LDAP) и годится для небольшой организации, где вы можете непосредственно контролировать подключение к серверу. Рассмотрим другой вариант автоматического назначения прав доступа. в окне конфигурации pGina на вкладке profile есть возможность добавить группы, в которые автоматически при входе в систему должны добавляться пользователи (поле List of groups that all plugin authenticated users should be added to. One per line.). Группы записываются без кавычек по одной в строчку. Пробуем добавить туда Remote Desktop Users. к сожалению (или к счастью), такой трюк не срабатывает, у меня это вызывало нефатальный сбой системы аутентификации Windows с выдачей на экран ошибки. Ну и хорошо, это даже правильно, что нам не позволяют обойти систему безопасности таким образом. Тем не менее при добавлении других групп в это поле, например, встроенная группа Power Users или любая другая, все они исправно добавляются в поле Member Of учетной записи пользователя. У читателя может возникнуть вопрос: а нельзя ли использовать отдельную группу, где бы были перечислены все те, кому доступ закрыт? не получится. Ситуация здесь та же, что была описана чуть ранее: пока пользователь не зашел первый раз на Windows-сервер, у него нет аккаунта, соответственно нечего добавлять в группу с запрещенным доступом. Можно обязывать новых пользователей регистрироваться на сервере для создания их учетных записей или поручить это кому-нибудь из ИТ-отдела, что неудобно, сложно, и я этот вариант не рассматриваю.

Рисунок 2. Вкладка User Configuration плагина LDAP Auth



Теперь мы выяснили, как дать доступ LDAP-аккаунтам, еще не имеющим учетных записей в Windows-системе с pGina. Напомню, что мы использовали вариант с добавлением специальной группы Authenticated Users в группу Remote Desktop Users. Попутно узнали, как в случае надобности добавлять всех входящих пользователей в определенные группы, не имеющие отношения к управлению удаленным доступом. но как все-таки разрешать доступ в систему только определённым пользователям, а не всем прошедшим аутентификацию?

Четвертый момент

Можно настроить проверку членства в группах LDAP, но для этого, возможно, придется модифицировать настройки самого LDAP.

Напомню, что LDAP – это облегченный протокол для доступа к сервису каталогов. Модель LDAP основывается на «элементах» (entries), каждый из которых имеет тип, набор атрибутов и характерное имя (или DN – Distinguished Name). DN состоит из пар атрибут=значение, разделенных запятыми. Атрибутами могут быть имя пользователя, адрес его электронной почты и т.п.

Элементы LDAP упорядочены в виде иерархического дерева, имеющего вершину, ветви (дочерние элементы, способные иметь собственных предков), листья (конечные элементы дерева). Традиционно такая конструкция отражала географическое положение элементов (страна, город и т.д.) или организационное (отдел, подразделение и т.д.). Наиболее распространено сегодня структурирование на основе имен интернет-доменов. Например, элемент с DN cn=test1, ou=Group,dc=domain,dc=ru может иметь вершину дерева dc=domain,dc=ru, которая является предком для элемента ou=Group, который, в свою очередь, является предком для дочернего элемента cn=test1. Каждый элемент может иметь несколько специальных атрибутов objectClass, которые определяют, каким правилам должен подчиняться данный элемент. Все возможные атрибуты и объектные классы дерева LDAP описываются схемой, схем может быть несколько – для каждой задачи своя.

Элемент, содержащий атрибут objectClass со значением posixGroup или groupOfNames, является групповым. Объектный класс posixGroup известен давно и некоторыми считается устаревшим. Он описан в схеме nis.schema, и элемент с таким классом должен иметь два атрибута – cn и gidNumber, а также может иметь несколько атрибутов, среди которых самым интересным является memberUid. Класс groupOfNames описан в схеме core.schema, и элемент, использующий этот класс, должен иметь атрибуты member, cn и может иметь некоторые другие атрибуты, для нас не представляющие важности. Это означает, что группа (элемент с атрибутом objectClass: groupOfNames) должна иметь хотя бы одного члена (member). Оба класса имеют тип STRUCTURAL, а это значит, что никакой элемент не может иметь оба этих класса одновременно [2]. а нам хотелось бы иметь в элементе хотя бы два атрибута вместе – gidNumber и member. gidNumber содержит идентификатор группы, что может быть полезным для определения групповых прав файловых систем. а member нам нужен для того, чтобы иметь возможность получать ответы на запросы типа «в какие группы входит данный пользователь?». в Интернете можно найти черновой вариант схемы rfc2307bis.schema. Эта схема очень походит на схему nis.schema, но в ней класс posixGroup имеет тип AUXILIARY, что означает: элемент может иметь оба наших класса. Подробнее об описании и администрировании LDAP можно почитать в руководстве [3].

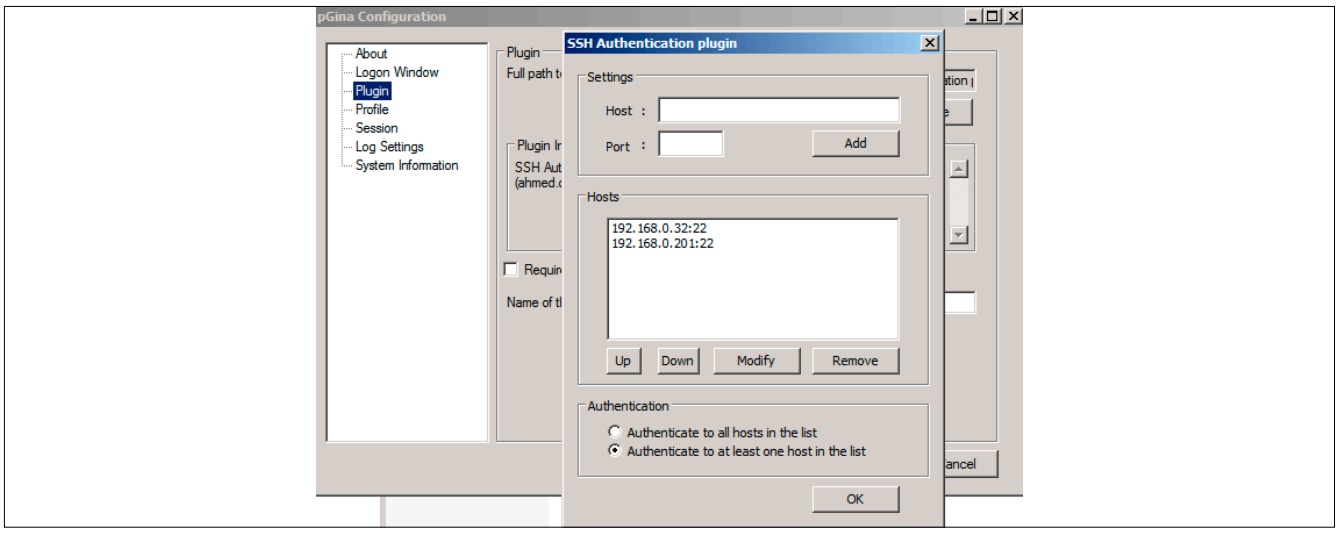
Теперь в своем дереве LDAP вы можете поддерживать как минимум три разных типа групп. Первая группа:

```
dn: cn=users,ou=Group,dc=domain,dc=ru
objectClass: groupOfNames
cn: users
member: uid=username,ou=People,dc=domain,dc=ru
```

Вторая группа:

```
dn: cn=users,ou=Group,dc=domain,dc=ru
objectClass: posixGroup
cn: users
gidNumber: 100
memberUid: username
```

Рисунок 3. Окно настроек плагина SSHAuth



Третья группа:

```
dn: cn=users,ou=Group,dc=domain,dc=ru
objectClass: groupOfNames
objectClass: posixGroup
cn: users
member: uid=username,ou=People,dc=domain,dc=ru
memberUid: username2
gidNumber: 100
```

Как видим, в последнем примере можно даже сочетать атрибуты member и memberUid. к сожалению, некоторые приложения и даже отдельные утилиты работают только с определенными типами групп. Одним из способов решения является использование третьего типа группы (комбинированного), который содержит оба класса: и groupOfNames, и posixGroup. Нужно еще иметь в виду, что схема nis.schema позволяет создавать новые группы первого и второго типа, а схема rfc2307bis.schema – первого и третьего.

Таким образом, для одновременной поддержки и первого, и третьего типов групп скачайте и добавьте в файл /etc/openldap/slapd.conf схему rfc2307bis.schema[4], закомментируйте схему nis.schema в любом доступном редакторе (vi, nano, mc и т.п.):

```
#include         /etc/openldap/schema/nis.schema
include /etc/openldap/schema/rfc2307bis.schema
include /etc/openldap/schema/core.schema
```

Что именно, какой тип группы, схемы и как использовать в вашем каталоге, решать вам. Более того, вы можете создавать свои собственные классы и группы, если для этого есть особые причины. Далее я буду использовать только первый тип группы (то есть в этом случае не нужно устанавливать новую схему rfc2307bis.schema), атрибутов которой вполне хватает для работы с плагином LDAP Auth.

В LDAP существуют дополнительные программные компоненты, называемые оверлеями, которые отслеживают запросы к базе данных (backend) LDAP, изменяя или дополняя их функциональность. для дальнейшей настройки нам потребуется оверлей memberof, для этого LDAP должен быть собран с поддержкой оверлеев. в Gentoo для этого достаточно иметь следующую строчку в файле /etc/portage/package.use:

```
net-nds/openldap overlays
```

Добавляем пару строк в slapd.conf:

```
moduleload      /usr/lib/openldap/openldap/memberof.so
overlay memberof
```

Теперь создадим пару тестовых групп:

```
group1.ldif:
dn: cn=test1,ou=Group,dc=domain,dc=ru
description: a group with my account
objectClass: groupOfNames
cn: test1
member: uid=user1,ou=People,dc=domain,dc=ru
group2.ldif:
dn: cn=test2,ou=Group,dc=domain,dc=ru
description: a group with another account
objectClass: groupOfNames
cn: test2
member: uid=user2,ou=People,dc=domain,dc=ru
```

Добавим эти группы в наш каталог:

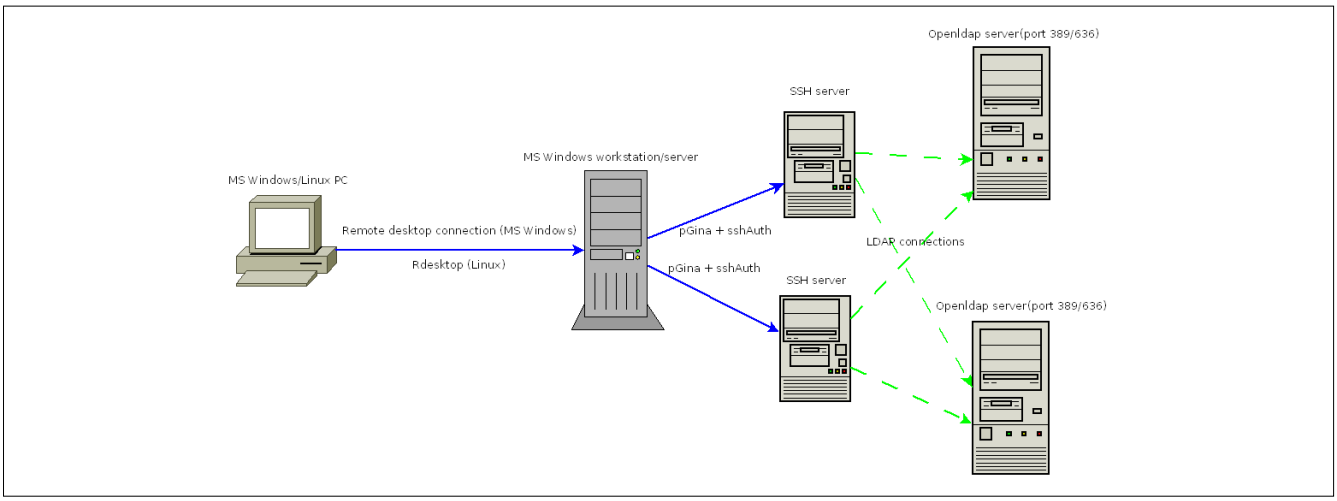
```
# ldapadd -ZZ -x -h ldap.domain.ru -D ↵
       'cn=root,dc=domain,dc=ru' -W -f group1.ldif
# ldapadd -ZZ -x -h ldap.domain.ru -D ↵
       'cn=root,dc=domain,dc=ru' -W -f group2.ldif
```

Поиск с помощью ldapsearch производится с добавлением слова memberof к запросу. Проверяем работу оверлея:

```
#ldapsearch -LLL uid=user1 memberof
dn: uid=user1,ou=People,dc=domain,dc=ru
memberOf: cn=test1,ou=Group,dc=domain,dc=ru
```

Теперь все приготовления со стороны нашего LDAP-сервера сделаны, осталось внести изменения в настройки pGina. Открываем конфигуратор и далее окно со значениями параметров плагина LDAP Auth. Нам понадобится прописать в поле Group Attr: вышеупомянутое слово memberOf – атрибут для определения членства пользователя в группах. Далее перейдем на вкладку User Configuration. Прописываем полностью характерное имя группы, членам которой мы хотим разрешать доступ на сервер, в поле ниже заглавия User Groups и добавляем ее в список таких групп кнопкой

Рисунок 4. Взаимодействие связки pGina + SSHAuth с клиентами и серверами ssh и LDAP



Add. Доступ здесь определяется следующим образом: если поле групп пустое, доступ разрешается всем (тем, кто ввел правильный логин и пароль). Если групп прописано несколько, то доступ определяется по принципу «или»: то есть, если пользователь входит в одну группу, но не входит в другую, доступ разрешается.

Проверяем настройки, выполняя соединение Remote Desktop от имени пользователя user1, который входит в группу test1, но не входит в группу test2, и убеждаемся в работоспособности настроек.

Теперь у нас появилась еще одна дополнительная возможность – членам определенных групп LDAP разрешать административный доступ на стороне сервера Windows. Если пользователь входит в группу, прописанную в поле Admin Groups на этой же вкладке настроек LDAP Auth, то при заходе на сервер он автоматически добавится в группу Administrators.

Проверяем это с помощью имеющейся группы test1 – пользователь user1 получает административные права. не забывайте только, что, если понадобится лишить пользователя user1 назначенных таким образом прав, недостаточно удалить группу test1 из поля Admin Groups, нужно отредактировать аккаунт этого пользователя в оснастке Local Users and Groups и там удалить ненужную ему группу. Это же следует помнить при описанном ранее редактировании поля List of groups that all plugin... на вкладке profile настроек pGina.

Использование плагина SSHAuth

Мы рассмотрели работу плагина LDAP Auth совместно с pGina, но это не единственный доступный способ аутентификации. Кратко опишу, как настроить плагин SSHAuth для аутентификации на ssh-серверах. Одним из главных недостатков LDAP Auth является возможность взаимодействия только с одним LDAP-сервером, что не всегда приемлемо для отказоустойчивой работы в случае, если у вас имеется несколько таких серверов и настроена репликация между ними. Неплохо было бы, чтобы разработчики добавили эту функциональность в будущем. Зато у SSHAuth такая возможность есть – можно использовать несколько ssh-серверов. Чтобы настроить аутентификацию с помощью ssh, скачайте и установите плагин [5] – он имеет собственный инсталлятор. Запомните путь установки. Откройте конфигуратор pGina и на вкладке Plugin, с помощью кнопки Browse укажите путь к только что установленной библиотеке dll плагина. Затем нажмите Configure.

Как видите, интерфейс очень простой, настроек минимум: добавляем IP-адреса и номера портов нужных нам ssh-серверов. Осталось только выбрать ниже, проходить аутентификацию на любом ssh-сервере из списка или требовать ее прохождения на всех серверах. Нажмите два раза «OK» и проверьте работу плагина с помощью вашего логина, используемого при работе с ssh.

При кажущей простоте данного инструментария можно организовать схожую с LDAP Auth функциональность, добавив настройки к сервису ssh. для работы ssh-сервера с LDAP нужно иметь настроенную конфигурацию с модулями pam_ldap, nss_ldap. О настройке этих модулей для Gentoo (для других дистрибутивов настройки похожи) можно почитать здесь [6]. На рисунке (см. рис. 4) показано, что pGina (совместно с SSHAuth) может использовать несколько серверов ssh для проверки доступа. Эти серверы могут проверять аккаунты своих локальных пользователей (посредством /etc/passwd), а кроме того, могут быть настроены для аутентификации на других LDAP-серверах (показано пунктирными связями). Такая схема работы обеспечивает хорошую отказоустойчивость системы.

Когда вы настроите возможность аутентификации ssh в LDAP, можно подумать об ограничении доступа. Например, это можно сделать с помощью PAM-модуля pam_access, указав его в настройках вашего PAM, отвечающих за вход в систему:

```
account required pam_access.so
```

После чего нужно отредактировать файл /etc/security/access.conf, добавив в него ограничения доступа, например:

```
+ : pginausers : 192.168.0.3
- : ALL EXCEPT wheel: ALL
```

То есть разрешить членам группы pginausers проходить аутентификацию с IP-адреса Windows-сервера с pGina и членам группы wheel с любого адреса, а всем другим запретить.

Другой (или дополнительный) способ ограничения доступа к ssh – использование директив DenyUsers, AllowUsers, DenyGroups, AllowGroups настроек самого сервиса ssh, например:

```
AllowGroups pginausers
```

В отличие от LDAP Auth с данным плагином можно использовать несколько серверов ssh для аутентификации. Минусом является то, что между pGina и LDAP появляется посредник (ssh) с его особенностями настройки группового доступа. Какой способ предпочесть, выбирать вам.

***

В этой части статьи мы рассмотрели, как можно управлять доступом к компьютеру с установленным pGina. Данный материал, надеюсь, может дать пищу для размышлений и тем, кто ищет способы перевода своей корпоративной сети на «свободные рельсы» (Open Source), используя рассмотренную программу как один из его пунктов. У меня не было возможности протестировать работу pGina в большой компьютерной системе из сотен компьютеров. Тем не менее в сети малого офиса связка pGina+LDAP Auth успешно работает более года. Можно использовать описанные в статье плагины, возможно, существуют другие, и, наконец, если хватает знаний, можно написать собственный плагин для работы с pGina. Успехов в освоении pGina и других Open Source-проектов! EOF

1. Бушков Е. Использование pGina для аутентификации в системах MS Windows. //«Системный администратор», №1-2, 2011 г. – С. 24-28.
2. http://www.zytrax.com/books/ldap/ch3.
3. http://www.openldap.org/doc/admin24.
4. http://simonraven.kisikew.org/src/ldap/rfc2307bis.schema.
5. http://sourceforge.net/projects/sshauth.
6. http://www.gentoo.org/doc/en/ldap-howto.xml.