Интеграция CentOS 8 с ADDS и настройка файлового сервера Samba

**Входные данные**

Файловый сервер (samba-srv.fdsdn.corp) на CentOS 8 с ip address – 192.168.0.108;

ADDS сервер (adds.fdsdn.corp) на Windows Server 2016 с ip address – 192.168.0.120, домен – fdsdn.corp.

Подготовка для присоединения сервера к домену fdsdn.corp

Важной частью для удачного присоединения linux сервера к домену является правильно настроенный hostname сервера.

# hostnamectl set-hostname samba-srv.fdsdn.corp

Проверить правильность настройки hostname можно таким образом:

# hostname -s

samba-srv

# hostname -f

samba-srv.fdsdn.corp

Не менее важной частью является настройка правильного DNS т.е. того, кто обслуживает домен AD. В нашем случает DNS установлен на контролере домена. Указать ip address DNS сервера можно с помощью **nmtui** или вручную изменить конфигурационный файл на linux сервере (**/etc/sysconfig/network-scripts/ifcfg-ens33**), где **ifcfg-ens33** файл конфигурации определенного сетевого интерфейса.

После проделанных настроек нужно удостовериться, что контроллер домена доступен с linux сервера по имени т.е. банально, сделать ping.

Далее важной частью является синхронизация времени linux сервера и контроллера домена т.к. для протокола Kerberos это очень важно. Для синхронизации времени в CentOS 8 используется chronyd. В нашем случае добавляем строчку в конфигурационный файл chronyd (**/etc/chrony.conf**), а настроенные по-умолчанию pools удаляем или комментируем (не забываем перезапустить chronyd):

server adds.fdsdn.corp iburst

Посмотреть успешную синхронизацию можно с помощью команды:

# timedatectl

Local time: Sun 2020-08-30 12:40:57 MSK

Universal time: Sun 2020-08-30 09:40:57 UTC

RTC time: Sun 2020-08-30 09:40:58

Time zone: Europe/Moscow (MSK, +0300)

**System clock synchronized: yes**

NTP service: active

RTC in local TZ: no

Или с помощью log файла:

# tail -n 10 /var/log/messages | grep chronyd

Aug 30 12:37:57 samba-srv chronyd[4264]: chronyd version 3.5 starting (+CMDMON +NTP +REFCLOCK +RTC +PRIVDROP +SCFILTER +SIGND +ASYNCDNS +SECHASH +IPV6 +DEBUG)

Aug 30 12:37:57 samba-srv chronyd[4264]: Frequency 8.798 +/- 0.055 ppm read from /var/lib/chrony/drift

Aug 30 12:37:57 samba-srv chronyd[4264]: Using right/UTC timezone to obtain leap second data

**Aug 30 12:38:01 samba-srv chronyd[4264]: Selected source 192.168.0.120**

Aug 30 12:38:01 samba-srv chronyd[4264]: System clock TAI offset set to 37 seconds

Также можно проверить какой источник выбран:

# chronyc sources

210 Number of sources = 1

MS Name/IP address Stratum Poll Reach LastRx Last sample

===============================================================================

^\* adds.fdsdn.corp 3 7 377 111 +29us[ +81us] +/- 175ms

**Заметка.**

Бывает возникает проблема, при которой не происходит синхронизации времени linux сервера с контроллером домена на Windows Server. Если ip связность между linux и ADDS в порядке, и firewall не препятствует то скорее всего контроллер домена не отвечает на запросы синхронизации. Чтобы заставить Windows Server отвечать на ntp запросы нужно проделать такие действия в CMD (тезисно):

> net stop w32time

> w32tm /config /reliable:yes

> net start w32time

Важным моментом является то, что при присоединении linux к домену с помощью winbind не происходит его динамической регистрации в DNS. Поэтому перед тем, как присоединять к домену сервер на linux, нужно в DNS создать A запись для данного сервера.

**Заметка.**

Бывает, при присоединении к домену выходит подобная ошибка:

# realm join -v fdsdn.corp

\* Resolving: \_ldap.\_tcp.fdsdn.corp

\* Performing LDAP DSE lookup on: 192.168.0.120

\* Successfully discovered: fdsdn.corp

Password for administrator:

\* Required files: /usr/sbin/oddjobd, /usr/libexec/oddjob/mkhomedir, /usr/sbin/sssd, /usr/bin/net

\* LANG=C LOGNAME=root /usr/bin/net -s /var/cache/realmd/realmd-smb-conf.KAEP3Y -U administrator ads join fdsdn.corp

Enter administrator's password:

**Failed to join domain: failed to lookup DC info for domain 'fdsdn.corp' over rpc: {Device Timeout} The specified I/O operation on %h was not completed before the time-out period expired.**

! Joining the domain fdsdn.corp failed

realm: Couldn't join realm: Joining the domain fdsdn.corp failed

Если все записи DNS для домена существуют, сетевое взаимодействие с контроллером домена существует и с DNS сервером настроено, резолвятся все DNS записи, то стоит попробовать открыть порт 445/tcp на контроллере домена для linux сервера, который вы хотите присоединить.

Присоединение CentOS 8 в ADDS (Winbind) на Windows Server 2016

Для присоединения системы CentOS 8 к AD необходимы два компонента. Один компонент, Samba Winbind, взаимодействует с AD и источником аутентификации, а другой компонент, realmd, обнаруживает доступные домены и настраивает базовые системные службы, в данном случае Samba Winbind, для подключения к домену AD.

Samba Winbind эмулирует клиента Windows в системе Linux и взаимодействует с серверами AD. Данный сервис предоставляет интерфейс для Name Servic Switch (NSS) и позволяет пользователям домена аутентифицироваться в AD при входе в локальную систему.

Сначала, инсталлируем необходимые пакеты для присоединения сервера к контроллеру домена:

~# dnf install realmd oddjob-mkhomedir oddjob samba-winbind-clients \

samba-winbind samba-common-tools samba-winbind-krb5-locator

**Заметка.**

Так как наша задача состоит в том, чтобы после присоединения к домену настроить файловый сервер на базе Samba и организовать аутентификацию к ресурсам по доменным учетным записям, то нужно установить сам samba сервер.

~# dnf install samba samba-client

Можно переименовать конфигурационный файл по-умолчанию samba сервера (**/etc/samba/smb.conf**) т.к. при присоединении к домену данный файл будет автоматически создан.

~# mv /etc/samba/smb.conf /etc/samba/smb.conf.old

Присоединяем linux сервер к домену:

~# realm join --membership-software=samba --client-software=winbind fdsdn.corp \

-U denysb

В данном случае, denysb– это учетная запись, имеющая административные права (Domain Admins), с помощью которой присоединяется к домену данный сервер.

Использованная выше команда **realm** производит такие действия:

* Создает файл **/etc/samba/smb.conf** и заносит настройки для присоединения сервера к домену fdsdn.corp;
* Добавляет настройки winbind модуля для поиска пользователей и групп в файл **/etc/nsswitch.conf**;
* Обновляем конфигурационные файлы PAM (**/etc/pam.d**/)
* Запускает winbind сервис и настраивает его на автозагрузку.

По сути, выполняет команды по настройке аутентификации ОС (выполнять команды ниже не нужно):

Для NSS (nsswitch.conf):

# authconfig --enablewinbind --enablewinbindauth --update

Для PAM:

# authconfig --enablewinbindauth –update

Для автоматического создания домашних каталогов:

# authconfig --enablemkhomedir –update

В CentOS 8 утилита authconfig была заменена утилитой authselect поэтому вместо трех команд выше можно использовать вот эту:

# authselect select winbind with-mkhomedir

Далее отредактируем файл **/etc/krb5.conf**. В определенных ситуациях администраторы хотят разрешить пользователям домена аутентифицироваться в локальных службах, которые запущены от доменных учетных записей. При использовании Kerberos для проверки подлинности пользователей домена включите подключаемый модуль winbind\_krb5\_localauth для правильного сопоставления участников Kerberos с учетными записями Active Directory через службу winbind.

Добавляем или изменяем следующие строчки:

[plugins]

localauth = {

module = winbind:/usr/lib64/samba/krb5/winbind\_krb5\_localauth.so

enable\_only = winbind

}

Можно проверить, что сервис winbind запустился корректно и будет запускаться при запуске системы:

~# systemctl status winbind

● winbind.service - Samba Winbind Daemon

Loaded: loaded (/usr/lib/systemd/system/winbind.service; **enabled**; vendor preset: disabled)

Active: **active (running)** since Sun 2020-08-30 16:40:26 MSK; 5min ago

Docs: man:winbindd(8)

man:samba(7)

man:smb.conf(5)

…

На данном этапе мы ввели linux сервер в домен. Мы можем зайти на данный сервер по ssh как из-под доменных учетный записей, так и из-под локальных (хранящихся в файле **passwd**). Так же это актуально и для локального входа на сервер. Главное, что, если мы хотим зайти из-под доменной учетки, то нужно вводить логин, например, таким образом – **denysb@fdsdn.corp**.

**Настройка sudo**

Как было изложено выше мы можем попасть на сервер по ssh из-под доменных учетных записей. Теперь осталось дать данному пользователю права sudo.

Создаем файл **/etc/sudoers.d/fdsdn** (без точек, а то работать не будет) и делаем там запись:

%FDSDN\\Domain\ Admins ALL=(ALL) ALL

%FDSDN\\ssh\_users ALL=(ALL) ALL

Таким образом мы дали права на sudo пользователям в группе **Domain Admins** и для примера еще пользователям входящие в группу **ssh\_users**.

Выставляем минимальные права на файл:

~# chmod 0440 /etc/sudoers.d/fdsdn

**Примечание**

Вот какой samba файл был создан утилитой realm:

~# cat /etc/samba/smb.conf

[global]

kerberos method = secrets and keytab

template homedir = /home/%U@%D

template shell = /bin/bash

security = ads

idmap config FDSDN : range = 2000000-2999999

idmap config FDSDN : backend = rid

idmap config \* : range = 10000-999999

idmap config \* : backend = tdb

winbind use default domain = no

winbind refresh tickets = yes

winbind offline logon = yes

winbind enum groups = no

winbind enum users = no

workgroup = FDSDN

realm = FDSDN.CORP

Данный файл, как и следующий настраивать дополнительно не надо (кроме как добавлять share), но в больших инсталляциях (много доменных структурах) может понадобиться настройка ID mapping. Об этом подробно можно почитать в официальной документации RedHat([Samba id mapping](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/deploying_different_types_of_servers/assembly_using-samba-as-a-server_Deploying-different-types-of-servers#assembly_samba-id-mapping_assembly_setting-up-samba-as-an-ad-domain-member-server)).

Вот файл nsswitch.conf, до того, как утилита realm отработала:

~# cat /etc/nsswitch.conf

# Generated by authselect on Wed Feb 12 11:14:59 2020

# Do not modify this file manually.

# If you want to make changes to nsswitch.conf please modify

# /etc/authselect/user-nsswitch.conf and run 'authselect apply-changes'.

passwd: **sss files systemd**

group: **sss files systemd**

netgroup: sss files

automount: sss files

services: sss files

shadow: files sss

hosts: files dns myhostname

bootparams: files

ethers: files

netmasks: files

networks: files

protocols: files

rpc: files

publickey: files

aliases: files

А вот после присоединения к домену:

# Generated by authselect on Sun Aug 30 16:40:26 2020

# Do not modify this file manually.

# If you want to make changes to nsswitch.conf please modify

# /etc/authselect/user-nsswitch.conf and run 'authselect apply-changes'.

#

passwd: **files winbind systemd**

group: **files winbind systemd**

shadow: files sss

hosts: files dns myhostname

bootparams: files

ethers: files

netmasks: files

networks: files

protocols: files

rpc: files

services: files sss

netgroup: sss

publickey: files

automount: files sss

aliases: files

Проверки правильности ввода сервера в домен

Для начала проверим наличие доверительной учетной записи сервера на AD:

~# wbinfo -t

checking the trust secret for domain FDSDN via RPC calls succeeded

Теперь проверяем, может ли наш сервер получать списки пользователей и групп. Первая команда выводит список всех групп домена, вторая - всех пользователей:

~# wbinfo -g

~# wbinfo -u

Проверим авторизацию пользователя через winbind:

# wbinfo -a FDSDN\\denysb%'Yakagam@220987'

plaintext password authentication succeeded

challenge/response password authentication succeeded

Проверим, что операционная система может Can получить информацию о доменных пользователях и группах:

~# getent passwd FDSDN\\denysb

FDSDN\denysb:\*:2001104:2000513::/home/denysb@FDSDN:/bin/bash

~# getent group "FDSDN\\Domain Users"

FDSDN\domain users:x:2000513:

Проверка того, могут ли пользователи домена AD получить билет Kerberos. Для начала, установим утилиты для проверки:

~# dnf install krb5-workstation

Теперь, собственно, проверки:

~# kinit denysb@FDSDN.CORP

~# klist

Ticket cache: KCM:0:94936

Default principal: denysb@FDSDN.CORP

Valid starting Expires Service principal

09/02/2020 11:04:12 09/02/2020 21:04:12 krbtgt/FDSDN.CORP@FDSDN.CORP

renew until 09/09/2020 11:04:07

Проверим вывод доступных доменов:

~# wbinfo --all-domains

BUILTIN

SAMBA-SRV

FDSDN

Покинуть домен

Покинуть т.е. отсоединиться от домена:

# realm leave fdsdn.corp

с удаление учетной записи компьютера на контролере домена:

# realm leave fdsdn.corp --remove

Конфигурирование samba share с аутентификацией по доменных учетных записях

**Настройка samba share используя POSIX ACLs**

В данном примере создадим группу **smb\_users** в домене и занесем туда определенных пользователей. Данная группа будет иметь полные права на share **/share/ad-pos-scl**.

Как служба Linux, Samba поддерживает общие ресурсы с ACL POSIX. Они позволяют управлять разрешениями локально на сервере Samba с помощью утилит, таких как chmod.

Начинаем настройку:

~# mkdir /share/ad-pos-acl

Если SELinux в enforcing mode, установим samba\_share\_t контекст для нашей share:

~# semanage fcontext -a -t samba\_share\_t "/share/ad-pos-acl(/.\*)?"

~# restorecon -Rv /share/ad-pos-acl/

Далее устанавливаем права:

~# chown "root":"FDSDN\smb\_users" /share/ad-pos-acl/

~# chmod 2770 /share/ad-pos-acl/

Включение бита set-group-ID (SGID) в каталоге автоматически устанавливает данную группу по-умолчанию для всех новых файлов и подкаталогов, вместо обычного поведения установки ее для основной группы пользователя, который создал новая запись в каталоге.

Теперь заносим данную share в **/etc/samba/smb.conf**. Таким образом данный файл имеет вид:

[global]

kerberos method = secrets and keytab

template homedir = /home/%U@%D

workgroup = FDSDN

template shell = /bin/bash

security = ads

realm = FDSDN.CORP

idmap config FDSDN : range = 2000000-2999999

idmap config FDSDN : backend = rid

idmap config \* : range = 10000-999999

idmap config \* : backend = tdb

winbind use default domain = no

winbind refresh tickets = yes

winbind offline logon = yes

winbind enum groups = no

winbind enum users = no

**[ad-pos-acl]**

**path = /share/ad-pos-acl**

**read only = no**

Проверяем правильность файла **/etc/samba/smb.conf**:

~# testparm

Запускаем или перезапускаем samba:

~# systemctl restart smb

Таким образом пользователи входящие в группу – **smb\_users** имеют полный доступ к share. Единственное что следует упомянуть, что каждый пользователь из группы **smb\_users** имеет доступ на запись и изменение только тех объектов (директории и файлы), которые он сам создал. Если вас так устраивает, то можно считать, что настройка закончена, если же вам нужно, чтобы пользователи из данной папки могли иметь права на чтение и изменение всех объектов в шаре, что нужно добавить некоторые строки в конфиг **smb.conf**:

create mask = 0660

directory mask = 0771

force create mode = 0660

force directory mode = 0771

Т.е. файл будет иметь вид:

…

[ad-pos-acl]

path = /share/ad-pos-acl

read only = no

valid users = +FDSDN\"smb\_users"

**create mask = 0660**

**directory mask = 0771**

**force create mode = 0660**

**force directory mode = 0771**

Запускаем или перезапускаем samba:

~# systemctl restart smb

**Настройка samba share используя Extended POSIX ACLs**

Преимуществом расширенных прав над стандартными является то, что при расширенных прав можно делать разнообразный доступ для разных групп в отличии от стандартных прав.

В данном примере создадим группу **smb\_users** в домене и занесем туда определенных пользователей. Данная группа будет полный доступ (читать, писать, исполнять) на share **/share/ad-pos-scl**. Группа **Domain Admins** будет, также, иметь полный доступ к ней, а группа **Domain Users** будет иметь права на чтение и выполнять. Выполнять нужно для того, чтобы пользователи данной группы могли «входить» в данную директорию.

Начинаем настройку:

~# mkdir /share/ad-pos-ext-acl

Если SELinux в enforcing mode, установим samba\_share\_t контекст для нашей share:

~# semanage fcontext -a -t samba\_share\_t "/share/ad-pos-ext-acl(/.\*)?"

~# restorecon -Rv /share/ad-pos-ext-acl/

Теперь заносим данную share в **/etc/samba/smb.conf**. Важным моментом при настройке share является добавление наследования расширенных acl:

inherit acls = yes

Таким образом данный файл имеет вид:

[global]

kerberos method = secrets and keytab

template homedir = /home/%U@%D

workgroup = FDSDN

template shell = /bin/bash

security = ads

realm = FDSDN.CORP

idmap config FDSDN : range = 2000000-2999999

idmap config FDSDN : backend = rid

idmap config \* : range = 10000-999999

idmap config \* : backend = tdb

winbind use default domain = no

winbind refresh tickets = yes

winbind offline logon = yes

winbind enum groups = no

winbind enum users = no

[ad-pos-acl]

path = /share/ad-pos-acl

read only = no

**[ad-pos-ext-acl]**

**path = /share/ad-pos-ext-acl**

**read only = no**

**inherit acls = yes**

Запускаем или перезапускаем samba:

~# systemctl restart smb

Проверяем что файловая система поддерживает расширенные ACL:

~# tune2fs -l /dev/md126 | grep "Default mount options:"

Default mount options: user\_xattr **acl**

Теперь будем давать права, о которых говорили выше.

По-умолчанию мы имеем такие настройки:

~# getfacl /share/ad-pos-ext-acl/

getfacl: Removing leading '/' from absolute path names

# file: share/ad-pos-ext-acl/

# owner: root

# group: root

user::rwx

group::r-x

other::r-x

Теперь нужно отключить автоматическую выдачу прав primary пользователю и группе:

~# setfacl -m group::--- /share/ad-pos-ext-acl/

~# setfacl -m default:group::--- /share/ad-pos-ext-acl/

~# getfacl /share/ad-pos-ext-acl/

getfacl: Removing leading '/' from absolute path names

# file: share/ad-pos-ext-acl/

# owner: root

# group: root

user::rwx

group::---

other::r-x

**default:user::rwx**

**default:group::---**

**default:other::r-x**

Теперь выдадим права, о которых мы говорили выше:

~# setfacl -m group:"FDSDN\Domain Admins":rwx /share/ad-pos-ext-acl/

~# setfacl -m group:"FDSDN\Domain Users":r-x /share/ad-pos-ext-acl/

~# setfacl -m group:"FDSDN\smb\_users":rwx /share/ad-pos-ext-acl/

~# setfacl -R -m other::--- /share/ad-pos-ext-acl/

Теперь нужно включить наследования разрешений, установленных на предыдущем шаге, новыми объектами файловой системы, созданными в этом каталоге:

~# setfacl -m default:group:"FDSDN\Domain Admins":rwx /share/ad-pos-ext-acl/

~# setfacl -m default:group:"FDSDN\Domain Users":r-x /share/ad-pos-ext-acl/

~# setfacl -m default:group:"FDSDN\smb\_users":rwx /share/ad-pos-ext-acl/

~# getfacl /share/ad-pos-ext-acl/

getfacl: Removing leading '/' from absolute path names

# file: share/ad-pos-ext-acl/

# owner: root

# group: root

user::rwx

group::---

group:FDSDN\\domain\040admins:rwx

group:FDSDN\\domain\040users:r--

group:FDSDN\\smb\_users:rw-

mask::rwx

other::---

default:user::rwx

default:group::---

default:group:FDSDN\\domain\040admins:rwx

default:group:FDSDN\\domain\040users:r--

default:group:FDSDN\\smb\_users:rw-

default:mask::rwx

default:other::r-x

Таким образом у нас получилось то, что пользователи в группах **Domain Admins** и **smb\_users** имеют полный доступ к share (читать, писать, заходить) т.е. создавать и редактировать папки и файлы, а пользователи в группе **Domain Users** могут только зайти на share и почитать там документы и пройтись по директориям.

**Настройка samba share используя Windows ACLs**

Только пользователи и группы, имеющие привилегию – **SeDiskOperatorPrivilege**, могут настраивать разрешения для общих ресурсов, использующих Windows ACL. Например, чтобы предоставить привилегию группе **FDSDN\Domain Admins**:

~# net rpc rights grant "FDSDN\Domain Admins" SeDiskOperatorPrivilege \

-U "FDSDN\denysb"

Чтобы вывести всех пользователей и группы, которые имеют привелегию – **SeDiskOperatorPrivilege**:

# net rpc rights list privileges SeDiskOperatorPrivilege \

-U "FDSDN\denysb"

Enter FDSDN\denysb's password:

SeDiskOperatorPrivilege:

BUILTIN\Administrators

FDSDN\domain admins

Начинаем настройку:

~# mkdir /share/ad-win-acl

Если SELinux в enforcing mode, установим samba\_share\_t контекст для нашей share:

~# semanage fcontext -a -t samba\_share\_t "/share/ad-win-acl(/.\*)?"

~# restorecon -Rv /share/ad-win-acl/

Чтобы наша share поддерживаkf списки управления доступом Windows, необходимо включить эту функцию в Samba. Чтобы включить его глобально для всех общих ресурсов, добавьте следующие настройки в раздел [global] файла **/etc/samba/smb.conf**. Также можно включить это индивидуально для определенной share (как мы и сделаем) заодно не забудем добавить конфиг share – **ad-win-acl**:

vfs objects = acl\_xattr

map acl inherit = yes

store dos attributes = yes

Таким образом данный файл имеет вид:

[global]

kerberos method = secrets and keytab

template homedir = /home/%U@%D

workgroup = FDSDN

template shell = /bin/bash

security = ads

realm = FDSDN.CORP

idmap config FDSDN : range = 2000000-2999999

idmap config FDSDN : backend = rid

idmap config \* : range = 10000-999999

idmap config \* : backend = tdb

winbind use default domain = no

winbind refresh tickets = yes

winbind offline logon = yes

winbind enum groups = no

winbind enum users = no

#log level = 3

[ad-pos-acl]

path = /share/ad-pos-acl

read only = no

[ad-pos-ext-acl]

path = /share/ad-pos-ext-acl

read only = no

inherit acls = yes

**[ad-win-acl]**

**path = /share/ad-win-acl**

**read only = no**

**vfs objects = acl\_xattr**

**map acl inherit = yes**

**store dos attributes = yes**

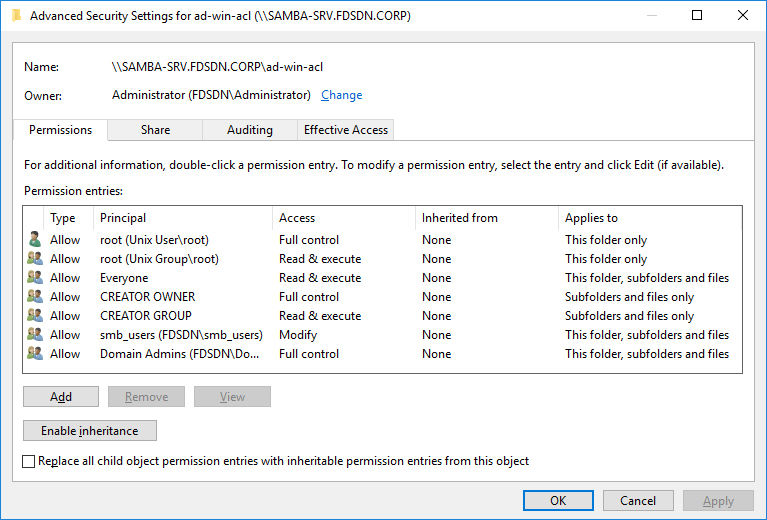
Перезагружаем или запускаем samba:

# systemctl restart smb

Теперь осталось дать права. Это можно сделать с помощью **Computer Management** из любого компьютера, который находится в домене из-под учетной записи пользователя, который входит в группу, которой выдана привилегия **SeDiskOperatorPrivilege**. В прошлых шагах мы выдали эту привилегию доменной группе **Domain Admins**. Так же можно управлять правами с помощью утилиты **smbcacls**.

Чтобы получилось выдать права на share через Computer Management нужно через эту утилиту изменить владельца данной share (например, на **Administrator**), выдать полные права, например, группе **Domain Admins** и добавить остальные права, если нужно, в нашем случае нам нужно дать права на изменение группе **smb\_users**.

Вот пример:



Управление доступом с помощью утилиты smbcacl можно найти в официальной документации по RedHat ([Managing ACLs on an SMB share using smbcacls](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/deploying_different_types_of_servers/assembly_using-samba-as-a-server_Deploying-different-types-of-servers#assembly_managing-acls-on-an-smb-share-using-smbcacls_assembly_configuring-file-shares-on-a-samba-server)).

Автоматическое монтирование шар при входе пользователя с помощью pam\_mount

**Настройка серверной части (Samba).**

Нам нужно организовать автоматическое монтирование шар при логине на компьютеры используя доменные учетные записи. Причем чтобы монтировались только те шары, доступ к которым есть у определенного пользователя.

Автоматическое монтирование делается на стороне клиентского компьютера. Мы это рассмотрим ниже. Предоставление шары и возможность видеть только те шары, к которым у пользователей есть доступ делается на стороне сервера.

Понятно, что нужно сервер Samba завести в домен. Как это делается мы рассмотрели ранее. Единственное что можно добавить, так это то, что заведен он должен быть с помощью winbind. Так же нужно установить samba сервер.

Начнем с входных данных. У нас есть три группы (rdp\_users, smb\_users, ssh\_users), в которые входят определенные пользователи. Пользователи каждой группы имеют доступ только к своим шарам (rdp-users-folder, smb-users-folder, ssh-users-folder). Для этого содаем необходимые директории:

~# mkdir -p /samba-shares/{rdp-users-folder,smb-users-folder,ssh-users-folder}

Теперь нужно сделать необходимые права. К директории rdp-users-folder имеют доступ только пользователи группы rdp\_users, к smb-users-folder – только smb\_users, а к ssh-users-folder – только ssh\_users. Группе “Domain Admins” сделаем полные права. Будем использовать “extendet acl”.

Предоставим права на директорию /samba-shares группе “Domain Admins” полные права, а “Domain Users” на чтение и выполнение (значит, что могут заходить в папку). Это делается для того, чтобы у всех в домене был доступ к “начальной” шаре. Кроме того, “включим” наследования только для “Domain Admins”, чтобы права только для аминов были наследованы и другим директориям.

~# setfacl -m group::--- /samba-shares

~# setfacl -m default:group::--- /samba-shares

~# setfacl -m other::--- /samba-shares

~# setfacl -m default:other::--- /samba-shares

~# setfacl -m group:"FDSDN\Domain Admins":rwx /samba-shares

~# setfacl -m group:"FDSDN\Domain Users":r-x /samba-shares

~# setfacl -m default:group:"FDSDN\Domain Admins":rwx /samba-shares

~# getfacl /samba-shares/

getfacl: Removing leading '/' from absolute path names

# file: samba-shares/

# owner: root

# group: root

user::rwx

group::---

group:FDSDN\\domain\040admins:rwx

group:FDSDN\\domain\040users:r-x

mask::rwx

other::---

default:user::rwx

default:group::---

default:group:FDSDN\\domain\040admins:rwx

default:mask::rwx

default:other::---

Далее предоставим только необходимые правила тем директория, которые находятся внутри директории /samba-shares.

~# setfacl -m group:"FDSDN\rdp\_users":rwx /samba-shares/rdp-users-folder

~# setfacl -m default:group:"FDSDN\rdp\_users":rwx /samba-shares/rdp-users-folder

~# getfacl /samba-shares/rdp-users-folder/

getfacl: Removing leading '/' from absolute path names

# file: samba-shares/rdp-users-folder/

# owner: root

# group: root

user::rwx

group::---

group:FDSDN\\domain\040admins:rwx

group:FDSDN\\rdp\_users:rwx

mask::rwx

other::---

default:user::rwx

default:group::---

default:group:FDSDN\\domain\040admins:rwx

default:group:FDSDN\\rdp\_users:rwx

default:mask::rwx

default:other::---

~# setfacl -m group:"FDSDN\smb\_users":rwx /samba-shares/smb-users-folder

~# setfacl -m default:group:"FDSDN\smb\_users":rwx /samba-shares/smb-users-folder

~# getfacl /samba-shares/smb-users-folder/

# getfacl /samba-shares/smb-users-folder/

getfacl: Removing leading '/' from absolute path names

# file: samba-shares/smb-users-folder/

# owner: root

# group: root

user::rwx

group::---

group:FDSDN\\domain\040admins:rwx

group:FDSDN\\smb\_users:rwx

mask::rwx

other::---

default:user::rwx

default:group::---

default:group:FDSDN\\domain\040admins:rwx

default:group:FDSDN\\smb\_users:rwx

default:mask::rwx

default:other::---

~# setfacl -m group:"FDSDN\ssh\_users":rwx /samba-shares/ssh-users-folder

~# setfacl -m default:group:"FDSDN\ssh\_users":rwx /samba-shares/ssh-users-folder

~# getfacl /samba-shares/ssh-users-folder/

getfacl: Removing leading '/' from absolute path names

# file: samba-shares/ssh-users-folder/

# owner: root

# group: root

user::rwx

group::---

group:FDSDN\\domain\040admins:rwx

group:FDSDN\\ssh\_users:rwx

mask::rwx

other::---

default:user::rwx

default:group::---

default:group:FDSDN\\domain\040admins:rwx

default:group:FDSDN\\ssh\_users:rwx

default:mask::rwx

default:other::---

Далее в конфигурационный файл /etc/samba/smb.conf добавляем настройки шары. Таким образом он будет иметь вид:

[global]

kerberos method = secrets and keytab

template homedir = /home/%U@%D

workgroup = FDSDN

template shell = /bin/bash

security = ads

realm = FDSDN.CORP

idmap config FDSDN : range = 2000000-2999999

idmap config FDSDN : backend = rid

idmap config \* : range = 10000-999999

idmap config \* : backend = tdb

winbind use default domain = no

winbind refresh tickets = yes

winbind offline logon = yes

winbind enum groups = no

winbind enum users = no

#log level = 3

winbind expand groups = 1

**[samba-shares]**

**path = /samba-shares**

**read only = no**

**inherit acls = yes**

**hide unreadable = yes**

**access based share enum = yes**

Настройка “access based share enum = yes” нужна для того, чтобы при монтировании шары (/samba-srv/samba-shares) на клиентском ПК были видны только те директории, на которые для залогиненного пользователя есть права хотя бы на чтение. Т.е., например, если пользователь smbuser1 смонтирует данную шару, то несмотря на то, что в этой шаре есть три директории он увидит только одну (“smb-users-folder”).

**Настройка клиентской части.**

Понятное дело, что компьютер должен быть заведен в домен. Как это делается тут рассказываться не будет. Эту информацию можно найти в интернете. Единственное что можно добавить, так это то, что заведен он должен быть с помощью winbind.

Для автоматического монтирования разделяемых файловых ресурсов (шар) при входе пользователя используется pam модуль pam\_mount. На клиентской машине (Ubuntu Desktop 20.04) нужно установить определенные пакеты:

~# apt install cifs-utils libpam-mount

Для начала, проверим возможность монтирование шар вручную (для этого достаточно cifs-utils). Логинимся на клиентскую систему, например, под пользователем smbuser1@fdsdn.corp. И выполняем команду mount (напомню, что данная команда выполняется только с sudo):

~# sudo mount -t cifs -o username=smbuser1@fdsdn.corp,dir\_mode=0777,file\_mode=0666 //samba-srv/samba-shares /home/smbuser1@FDSDN/mounts/

Если все работает штатно, то система запросит пароль для sudo, а потом запросит пароль для доступа к шаре. После этого произойдет монтирование.

Кстати, есть вариант сделать так, чтобы не вводить пароль для шары. Для этого нужно, например, в домашнюю директорию пользователя поместить файл с паролем. Создаем файл “.smbcred” в домашнем каталоге данного пользователя с таким содержимым:

username=smbuser1

password=AhuenniyP@ssW00rD

domain=fdsdn.corp

После этого командой монтируем с указанием данного файла:

~# sudo mount -t cifs -o username=smbuser1@fdsdn.corp,dir\_mode=0777,file\_mode=0666,credentials=/home/smbuser1@FDSDN/.smbcred //samba-srv/samba-shares /home/smbuser1@FDSDN/mounts/

Чтобы так же убрать запрос пароля при выполнении команд sudo mount, можно добавить строчку в файл /etc/sudoers или создать дополнительный файл (например, /etc/sudoers.d/fdsdn) и добавить туда строчку:

%FDSDN\\Domain\ Users ALL=(root) NOPASSWD:/usr/bin/mount

Таким образом при выполнении команды монтирования у нас не будут спрашивать пароль ни в случае команд sudo mount ни в случае запроса пароля на шару, что облегчает монтирование шар для пользователей.

Но нам это не совсем интересно т.к. в нашем случае нужно организовать автоматическое монтирование при логине. Вот за это будет отвечать установленный модуль libpam-mount.

После того как мы установили libpam-mount у нас появился файл /etc/security/pam\_mount.conf.xml.

<?xml version="1.0" encoding="utf-8" ?>

<!DOCTYPE pam\_mount SYSTEM "pam\_mount.conf.xml.dtd">

<!--

See pam\_mount.conf(5) for a description.

-->

<pam\_mount>

<!-- debug should come before everything else,

since this file is still processed in a single pass

from top-to-bottom -->

<debug enable="0" />

<!-- Volume definitions -->

<!-- pam\_mount parameters: General tunables -->

<!--

<luserconf name=".pam\_mount.conf.xml" />

-->

<!-- Note that commenting out mntoptions will give you the defaults.

You will need to explicitly initialize it with the empty string

to reset the defaults to nothing. -->

<mntoptions allow="nosuid,nodev,loop,encryption,fsck,nonempty,allow\_root,allow\_other" />

<!--

<mntoptions deny="suid,dev" />

<mntoptions allow="\*" />

<mntoptions deny="\*" />

-->

<mntoptions require="nosuid,nodev" />

<!-- requires ofl from hxtools to be present -->

<logout wait="0" hup="no" term="no" kill="no" />

<!-- pam\_mount parameters: Volume-related -->

<mkmountpoint enable="1" remove="true" />

</pam\_mount>

Нужно его привести к такому виду (жирным указано то, что мы добавили, а остальное было по-умолчанию):

<?xml version="1.0" encoding="utf-8" ?>

<!DOCTYPE pam\_mount SYSTEM "pam\_mount.conf.xml.dtd">

<!--

See pam\_mount.conf(5) for a description.

-->

<pam\_mount>

<!-- debug should come before everything else,

since this file is still processed in a single pass

from top-to-bottom -->

<debug enable="1" />

<!-- Volume definitions -->

<!-- <logout wait="microseconds" hup="yes/no" term="yes/no" kill="yes/no" /> -->

**<logout wait="50000" hup="yes" term="yes" kill="yes" />**

<!-- pam\_mount parameters: General tunables -->

**<volume**

**fstype="cifs"**

**server="samba-srv.fdsdn.corp"**

**path="samba-shares"**

**mountpoint="/home/%(DOMAIN\_USER)@%(DOMAIN\_NAME)/mounts"**

**options="uid=%(USERUID),gid=%(USERGID),file\_mode=0770,dir\_mode=0770"**

**/>**

<!--

<luserconf name=".pam\_mount.conf.xml" />

-->

<!-- Note that commenting out mntoptions will give you the defaults.

You will need to explicitly initialize it with the empty string

to reset the defaults to nothing. -->

<mntoptions allow="nosuid,nodev,loop,encryption,fsck,nonempty,allow\_root,allow\_other" />

<!--

<mntoptions deny="suid,dev" />

<mntoptions allow="\*" />

<mntoptions deny="\*" />

-->

<mntoptions require="nosuid,nodev" />

<!-- requires ofl from hxtools to be present -->

<logout wait="0" hup="no" term="no" kill="no" />

<!-- pam\_mount parameters: Volume-related -->

<mkmountpoint enable="1" remove="true" />

</pam\_mount>

Таким образом, кто бы то ни зашел на данный компьютер у него будет смонтирована шара – samba-shares, которая располагается на сервере – samba-srv.fdsdn.corp. Смонтируется она в /home/%(DOMAIN\_USER)@%(DOMAIN\_NAME)/mounts, где %(DOMAIN\_USER) – доменное имя пользователя без доменной части (например, smbuser1), а %(DOMAIN\_NAME) – название домена (например, FDSDN).

Как было сказано выше, нужно что бы клиентский компьютер был присоединен к домену с помощью “winbind”, но если все-таки вы присоединили ПК к домену с помощью “sssd”, то файл должен быть слегка изменен в части монтирования, т.е. (жирным выделено то, что изменено):

<volume

fstype="cifs"

server="samba-srv.fdsdn.corp"

path="samba-shares"

**mountpoint="/home/%(DOMAIN\_USER)/mounts"**

options="uid=%(USERUID),gid=%(USERGID),file\_mode=0770,dir\_mode=0770"

/>

Автоматическое монтирование домашних каталогов (перемещаемые профили) для линукс компьютеров с помощью pam\_mount

**Настройка серверной части (Samba).**

Т.к настройка очень похожа, то тут мы покажем ее без дополнительных объяснений. Единственное – повторюсь, что samba сервер должен быть присоединен к домену с помощью “winbind”. Пример будет состоять в том, что у нас 3 пользователя (smbuser1@fdsdn.corp, sshuser1@fdsdn.corp, rdpuser1@fdsdn.corp) и им необходимы перемещаемые профили т.е. когда они логиняться на ПК (под линуксом), то происходит автоматическое монтирование домашнего каталога, который на самом деле находится на сервере.

Создадим необходимые директории и права:

~# mkdir -p /samba-homes/{smbuser1@FDSDN,sshuser1@FDSDN,rdpuser1@FDSDN}

~# setfacl -m group::--- /samba-homes

~# setfacl -m default:group::--- /samba-homes

~# setfacl -m other::--- /samba-homes

~# setfacl -m default:other::--- /samba-homes

~# setfacl -m group:"FDSDN\Domain Admins":rwx /samba-homes

~# setfacl -m group:"FDSDN\Domain Users":r-x /samba-homes

~# setfacl -m default:group:"FDSDN\Domain Admins":rwx /samba-homes

~# setfacl -m user:"FDSDN\rdpuser1":rwx /samba-homes/rdpuser1@FDSDN

~# setfacl -m user:"FDSDN\smbuser1":rwx /samba-homes/smbuser1@FDSDN

~# setfacl -m user:"FDSDN\sshuser1":rwx /samba-homes/sshuser1@FDSDN

~# setfacl -m default:user:"FDSDN\rdpuser1":rwx /samba-homes/rdpuser1@FDSDN

~# setfacl -m default:user:"FDSDN\smbuser1":rwx /samba-homes/smbuser1@FDSDN

~# setfacl -m default:user:"FDSDN\sshuser1":rwx /samba-homes/sshuser1@FDSDN

Добавляем в конфиг (/etc/samba/smb.conf) для samba сервера такие настройки:

[samba-homes]

path = /samba-homes/%U@%D

read only = no

inherit acls = yes

hide unreadable = yes

При такой настройки (path), при монтировании клиентом ресурса “samba-homes” будет монтироваться не /samba-homes, а /samba-homes/<username>@<domain> (пример, /samba-homes/rdpuser1@fdsdn.corp).

Данная настройка сервера справедлива, если **клиент** будет присоединен к домену с помощью “winbind”. Если же ваши клиенты присоединены к домену с помощью “sssd”, то настройка несколько отличается, но малость. Нужно, просто, создать домашние каталоги для перемещаемых профилей с определенными именами. Вот пример:

~# mkdir -p /samba-sssd homes/rdpuser1\_fdsdn.corp

~# mkdir -p /samba-sssd homes/sshuser1\_fdsdn.corp

~# mkdir -p /samba-sssd homes/smbuser1\_fdsdn.corp

Далее чуть-чуть изменяются настройки samba сервера (жирным отмечено что отличается):

[samba-sssd-homes]

**path = /samba-sssd-homes/%U**

read only = no

inherit acls = yes

hide unreadable = yes

В остальном, настройки те же. Так же есть отличия в настройки pam\_mount на клиенте, но это мы опишем ниже.

**Настройка клиентской части.**

Клиент, тоже, как и сервер, должен быть присоединен к домену с помощью “winbind”.

Как и в предыдущем примере, будем использовать pam\_mount. Тезисно опишем шаги (опять же, будем использовать Ubuntu Desktop 20.04):

~# apt install cifs-utils libpam-mount

После того как мы установили libpam-mount у нас появился файл /etc/security/pam\_mount.conf.xml, в который нужно определенные строчки (по аналогии с примером выше):

<volume

fstype="cifs"

server="samba-srv.fdsdn.corp"

path="samba-homes"

mountpoint="/home/%(DOMAIN\_USER)@%(DOMAIN\_NAME)"

options="uid=%(USERUID),gid=%(USERGID),file\_mode=0770,dir\_mode=0770"

/>

Приведем пример, когда у нас клиентская часть присоединена к домену с помощью “sssd”. Вот так выглядит часть файла pam\_mount.xml (жирным показано то, что изменено):

<volume

fstype="cifs"

server="samba-srv.fdsdn.corp"

path="samba-sssd-homes"

**mountpoint="/home/%(DOMAIN\_USER) "**

options="uid=%(USERUID),gid=%(USERGID),file\_mode=0770,dir\_mode=0770"

/>

При автоматическом монтировании домашних каталогов нужно, чтобы клиенты были присоединены к домену либо с помощью “winbind”, либо с помощью ”sssd” т.к. название домашних директорий на стороне samba сервера при “winbind” и “sssd” разные.

Рабочий пример автоматического монтирования

Пример файла pam\_mount.xml, когда нужно смонтировать домашние каталоги при входе и еще смонтировать туда какой-нибудь ресурс (объединим два выше приведенных примера в один). Пример для клиентов присоединённых к домену с помощью “winbind” :

<?xml version="1.0" encoding="utf-8" ?>

<!DOCTYPE pam\_mount SYSTEM "pam\_mount.conf.xml.dtd">

<!--

See pam\_mount.conf(5) for a description.

-->

<pam\_mount>

<!-- debug should come before everything else,

since this file is still processed in a single pass

from top-to-bottom -->

<debug enable="1" />

<!-- Volume definitions -->

<!-- <logout wait="microseconds" hup="yes/no" term="yes/no" kill="yes/no" /> -->

**<logout wait="50000" hup="yes" term="yes" kill="yes" />**

<!-- pam\_mount parameters: General tunables -->

**<volume**

**fstype="cifs"**

**server="samba-srv.fdsdn.corp"**

**path="samba-homes"**

**mountpoint="/home/%(DOMAIN\_USER)@%(DOMAIN\_NAME)"**

**options="uid=%(USERUID),gid=%(USERGID),file\_mode=0770,dir\_mode=0770"**

**/>**

**<volume**

**fstype="cifs"**

**server="samba-srv.fdsdn.corp"**

**path="samba-shares"**

**mountpoint="/home/%(DOMAIN\_USER)@%(DOMAIN\_NAME)/mounts"**

**options="uid=%(USERUID),gid=%(USERGID),file\_mode=0770,dir\_mode=0770"**

**/>**

<!--

<luserconf name=".pam\_mount.conf.xml" />

-->

<!-- Note that commenting out mntoptions will give you the defaults.

You will need to explicitly initialize it with the empty string

to reset the defaults to nothing. -->

<mntoptions allow="nosuid,nodev,loop,encryption,fsck,nonempty,allow\_root,allow\_other" />

<!--

<mntoptions deny="suid,dev" />

<mntoptions allow="\*" />

<mntoptions deny="\*" />

-->

<mntoptions require="nosuid,nodev" />

<!-- requires ofl from hxtools to be present -->

<logout wait="0" hup="no" term="no" kill="no" />

<!-- pam\_mount parameters: Volume-related -->

<mkmountpoint enable="1" remove="true" />

</pam\_mount>

Ну, и приведем полностью конфигурационный файл samba сервера:

[global]

kerberos method = secrets and keytab

template homedir = /home/%U@%D

workgroup = FDSDN

template shell = /bin/bash

security = ads

realm = FDSDN.CORP

idmap config FDSDN : range = 2000000-2999999

idmap config FDSDN : backend = rid

idmap config \* : range = 10000-999999

idmap config \* : backend = tdb

winbind use default domain = no

winbind refresh tickets = yes

winbind offline logon = yes

winbind enum groups = no

winbind enum users = no

log level = 3

winbind expand groups = 1

**[samba-shares]**

**path = /samba-shares**

**read only = no**

**inherit acls = yes**

**hide unreadable = yes**

**access based share enum = yes**

**[samba-homes]**

**path = /samba-homes/%U@%D**

**read only = no**

**inherit acls = yes**

**hide unreadable = yes**