Active Directory Service with Samba

Setting up Samba and Active Directory

**We're building an Active Directory server from Samba running on FreeBSD, free server software on a free operating system.** We have already set up the [needed DNS infrastructure](https://cromwell-intl.com/open-source/samba-active-directory/dns.html) on an existing BIND master (or primary) DNS server, [installed FreeBSD on a Raspberry Pi](https://cromwell-intl.com/open-source/samba-active-directory/freebsd-raspberry-pi.html), and set that up as a [BIND slave DNS server](https://cromwell-intl.com/open-source/samba-active-directory/slave-dns.html) for those same zones. Now we're ready to do the Samba work! [Jump back to the start](https://cromwell-intl.com/open-source/samba-active-directory/) for an overview of the project.

Recall that Active Directory is simply Microsoft's bundle of DNS, LDAP (more or less), and Kerberos (kind of). I had thought that Active Directory meant an enormously complicated collection of Microsoft-specific services, all of which communicate through arcane Microsoft-specific protocols. Not really.

Samba setup

In an earlier step we [installed Samba](https://cromwell-intl.com/open-source/samba-active-directory/freebsd-raspberry-pi.html#samba) on our FreeBSD system. Now it's time to configure it. But first, let's see if any Samba processes are running, or if any old database files are lying around to cause confusion. The egrep process *may* notice itself, as here. That doesn't matter, just make sure there aren't any of the ones we're looking for:

Cleaning up

# **ps axuww | egrep 'PID|samba|smbd|nmbd|winbindd'**

USER PID %CPU %MEM VSZ RSS TT STAT STARTED TIME COMMAND

root 755 0.0 0.5 6424 2280 0 S+ 16:30 0:00.08 egrep PID|samba|smbd|nmbd|winbind

# **smbd -b | grep /**

Built using: /nxb-bin/usr/bin/cc

SRCDIR: /wrkdirs/usr/ports/net/samba43/work/samba-4.3.13/source3

BUILDDIR: /wrkdirs/usr/ports/net/samba43/work/samba-4.3.13/source3

SBINDIR: /usr/local/sbin

BINDIR: /usr/local/bin

CONFIGFILE: /usr/local/etc/smb4.conf

LOGFILEBASE: /var/log/samba4

LMHOSTSFILE: /usr/local/etc/lmhosts

LIBDIR: /usr/local/lib

MODULESDIR: /usr/local/lib/shared-modules

LOCKDIR: /var/db/samba4

STATEDIR: /var/db/samba4

CACHEDIR: /var/db/samba4

PIDDIR: /var/run/samba4

SMB\_PASSWD\_FILE: /var/db/samba4/private/smbpasswd

PRIVATE\_DIR: /var/db/samba4/private

You may need to clean up some things:

# **pkill samba smbd nmbd winbindd**

# **rm -rf /usr/local/etc/smb4.conf \**

**/usr/local/etc/lmhosts \**

**/var/db/samba4/\***

Provisioning a domain

Make sure that you are certain about the DNS domain and Kerberos realm name, as you cannot change those. You would have to delete the definitions and start over with the correct names.

Here is one way to do the provisioning. You can also do it interactively, passing only the --use-rfc2307 option. **Include** [**RFC 2307**](https://tools.ietf.org/html/rfc2307) **support,** as this allows you to store Unix attributes like UID, home directory, etc., in Active Directory. Be patient, as this takes several minutes to run, especially on a single-CPU Raspberry Pi.

# **samba-tool domain provision --use-rfc2307 \**

**--realm=CORP.EXAMPLE.COM --domain=CORP \**

**--server-role=dc --dns-backend=BIND9\_DLZ**

Looking up IPv4 addresses

Looking up IPv6 addresses

Setting up share.ldb

Setting up secrets.ldb

Setting up the registry

Setting up the privileges database

Setting up idmap db

Setting up SAM db

Setting up sam.ldb partitions and settings

Setting up sam.ldb rootDSE

Pre-loading the Samba 4 and AD schema

Adding DomainDN: DC=corp,DC=example,DC=com

Adding configuration container

Setting up sam.ldb schema

Setting up sam.ldb configuration data

Setting up display specifiers

Modifying display specifiers

Adding users container

Modifying users container

Adding computers container

Modifying computers container

Setting up sam.ldb data

Setting up well known security principals

Setting up sam.ldb users and groups

Setting up self join

Adding DNS accounts

Creating CN=MicrosoftDNS,CN=System,DC=corp,DC=example,DC=com

Creating DomainDnsZones and ForestDnsZones partitions

Populating DomainDnsZones and ForestDnsZones partitions

See /var/db/samba4/private/named.conf for an example configuration include file for BIND

and /var/db/samba4/private/named.txt for further documentation required for secure DNS updates

Setting up sam.ldb rootDSE marking as synchronized

Fixing provision GUIDs

A Kerberos configuration suitable for Samba 4 has been generated at /var/db/samba4/private/krb5.conf

Setting up fake yp server settings

Once the above files are installed, your Samba4 server will be ready to use

Admin password: +Dyg<oqzjG~H$p%NI

Server Role: active directory domain controller

Hostname: freebsd

NetBIOS Domain: CORP

DNS Domain: corp.example.com

DOMAIN SID: S-1-5-21-1556128061-1363490377-3170584331

Record the automatically generated administrator password and the domain SID.

You probably want to change the password expiration. Otherwise you will be suddenly inconvenienced in 42 days.

# **samba-tool user setexpiry --noexpiry Administrator**

Expiry for user 'Administrator' disabled.

You may want to reset the password. If you do this without waiting for a day, change the minimum password age to zero:

# **samba-tool domain passwordsettings show**

Password informations for domain 'DC=corp,DC=example,DC=com'

Password complexity: on

Store plaintext passwords: off

Password history length: 24

Minimum password length: 7

Minimum password age (days): 1

Maximum password age (days): 42

Account lockout duration (mins): 30

Account lockout threshold (attempts): 0

Reset account lockout after (mins): 30

# **samba-tool domain passwordsettings set --min-pwd-age 0**

Minimum password age changed!

All changes applied successfully!

# **samba-tool user password --user=administrator**

Password for [CORP\administrator]:+Dyg<oqzjG~H$p%NI

New Password:*new-password-here*

Retype Password:*new-password-here*

Changed password OK

Remaining manual steps

Verify that /var/db/samba4/private/named.conf was automatically configured by uncommenting the appropriate dlz\_bind9\_\*.so shared library.

Then, if it hasn't already been done automatically, add a line at the end of /usr/local/etc/namedb/named.conf:

include "/var/db/samba4/private/named.conf";

I need to make a symbolic link to the new Kerberos configuration file:

# **ln -s /var/db/samba4/private/krb5.conf /etc/krb5.conf**

Now let's see if Samba starts cleanly:

# **/usr/local/etc/rc.d/samba\_server onestart**

Performing sanity check on Samba configuration: OK

Starting samba.

[2017/02/18 11:55:02.721769, 0] ../lib/util/debug.c:947(reopen\_logs\_internal)

Unable to open new log file '/var/log/samba4/log.samba': No such file or directory

[2017/02/18 11:55:02.758508, 0] ../source4/smbd/server.c:371(binary\_smbd\_main)

samba version 4.3.13 started.

Copyright Andrew Tridgell and the Samba Team 1992-2015

# **/usr/local/etc/rc.d/samba\_server onestop**

Stopping samba.

Waiting for PIDS: 1063

It fails because /var/log is mounted on RAM on the Raspberry Pi. Logs disappear at every boot, along with the expected subdirectory. I have configured syslog to start, but then to send everything over the network to a log collector instead of saving it to disk. Samba, however, wants to save a copy locally. So, I added one line to the end of the [global] stanza in /usr/local/etc/smb4.conf. I also removed the dnsupdate from the end of the list of server services.

[global]

workgroup = CORP

realm = CORP.EXAMPLE.COM

netbios name = FREEBSD

server role = active directory domain controller

server services = s3fs, rpc, nbt, wrepl, ldap, cldap, kdc, drepl, winbindd, ntp\_signd, kcc

idmap\_ldb:use rfc2307 = yes

log file = /var/log/samba.%m

[netlogon]

path = /var/db/samba4/sysvol/corp.example.com/scripts

read only = No

[sysvol]

path = /var/db/samba4/sysvol

read only = No

Now let's try it again:

# **/usr/local/etc/rc.d/samba\_server onestart**

Performing sanity check on Samba configuration: OK

Starting samba.

[... wait a minute for everything to come up ...]

# **/usr/local/etc/rc.d/samba\_server onestatus**

samba is running as pid 564.

# **pstree**

-+= 00001 root /sbin/init --

|--= 00347 root /sbin/devd

|--= 00493 root /usr/sbin/syslogd -s

|--= 00505 bind /usr/local/sbin/named -u bind -c /usr/local/etc/namedb/named.conf

|--= 00553 root /usr/sbin/ntpd -g -c /etc/ntp.conf -p /var/run/ntpd.pid -f /var/db/ntpd.drift

|-+= 00564 root /usr/local/sbin/samba --daemon --configfile=/usr/local/etc/smb4.conf

| |-+- 00690 root samba: task[s3fs\_parent] (samba)

| | \-+= 00692 root /usr/local/sbin/smbd -D --option=server role check:inhibit=yes --foreground

| | |--- 00707 root /usr/local/sbin/smbd -D --option=server role check:inhibit=yes --foreground

| | |--- 00710 root /usr/local/sbin/smbd -D --option=server role check:inhibit=yes --foreground

| | \--- 01642 root /usr/local/sbin/smbd -D --option=server role check:inhibit=yes --foreground

| |--- 00691 root samba: task[dcesrv] (samba)

| |--- 00693 root samba: task[nbtd] (samba)

| |--- 00694 root samba: task wrepl server\_id[694] (samba)

| |--- 00695 root samba: task[ldapsrv] (samba)

| |--- 00696 root samba: task[cldapd] (samba)

| |--- 00697 root samba: task[kdc] (samba)

| |--- 00698 root samba: task[dreplsrv] (samba)

| |-+- 00699 root samba: task[winbindd\_parent] (samba)

| | \-+= 00703 root /usr/local/sbin/winbindd -D --option=server role check:inhibit=yes --foreground

| | |--- 00709 root /usr/local/sbin/winbindd -D --option=server role check:inhibit=yes --foreground

| | |--- 01164 root /usr/local/sbin/winbindd -D --option=server role check:inhibit=yes --foreground

| | \--- 01165 root /usr/local/sbin/winbindd -D --option=server role check:inhibit=yes --foreground

| |--- 00700 root samba: task[ntp\_signd] (samba)

| |--- 00701 root samba: task[kccsrv] (samba)

| \--- 00702 root samba: task[dnsupdate] (samba)

|-+= 00634 root /usr/sbin/sshd

| \-+= 01647 root sshd: cromwell [priv] (sshd)

| \-+- 01649 cromwell sshd: cromwell@pts/0 (sshd)

| \-+= 01650 cromwell -tcsh (tcsh)

| \-+= 01688 cromwell pstree

| \--- 01689 cromwell ps -axwwo user,pid,ppid,pgid,command

|--= 00689 root /usr/libexec/getty 3wire.115200 ttyu0

|--= 00685 root /usr/libexec/getty Pc ttyv0

|--= 00686 root /usr/libexec/getty Pc ttyv1

|--= 00687 root /usr/libexec/getty Pc ttyv2

\--= 00688 root /usr/libexec/getty Pc ttyv3

I also looked in the syslog output on the log collector, there were no new error messages about missing log directories. Locally I have multiple Samba log files, all of them ephemeral ones on the RAM-based /var/log file system. But all the content is going to the log collector and being saved there.

# **ls -l /var/log/log\* /var/log/samba\***

total 20

drwxrwxr-x 2 root operator 512 Feb 18 11:42 .snap

-rw-r--r-- 1 root wheel 266 Feb 18 12:27 log.wb-CORP

-rw-r--r-- 1 root wheel 133 Feb 18 12:27 log.winbindd-idmap

-rw-r--r-- 1 root wheel 3639 Feb 18 12:27 samba.%m

-rw-r--r-- 1 root wheel 154 Feb 18 12:27 samba.smbd

-rw-r--r-- 1 root wheel 342 Feb 18 12:27 samba.winbindd

All this seems good, so I added this line to /etc/rc.conf:

samba\_server\_enable="YES"

Now let's reboot and make sure that everything comes up.

Checking the network services

DNS

This is good, it's listening on IPv4 and IPv6 on all interfaces.

# **lsof -i tcp:53 -o -i udp:53**

COMMAND PID USER FD TYPE DEVICE OFFSET NODE NAME

named 508 bind 21u IPv6 0xc321f650 0t0 TCP localhost:domain (LISTEN)

named 508 bind 22u IPv4 0xc321f328 0t0 TCP localhost:domain (LISTEN)

named 508 bind 23u IPv4 0xc321f000 0t0 TCP freebsd.example.com:domain (LISTEN)

named 508 bind 24u IPv6 0xc321eca0 0t0 TCP [fe80:2::ba27:ebff:fe41:b9ae]:domain (LISTEN)

named 508 bind 25u IPv6 0xc321e978 0t0 TCP freebsd.example.com:domain (LISTEN)

named 508 bind 512u IPv6 0xc31373e8 0t0 UDP localhost:domain

named 508 bind 513u IPv4 0xc31373d4 0t0 UDP localhost:domain

named 508 bind 514u IPv4 0xc31373c0 0t0 UDP freebsd.example.com:domain

named 508 bind 515u IPv6 0xc31373ac 0t0 UDP [fe80:2::ba27:ebff:fe41:b9ae]:domain

named 508 bind 516u IPv6 0xc3137398 0t0 UDP freebsd.example.com:domain

Microsoft SMB/CIFS services

Three samba processes are providing five network services. Check /etc/services to figure out which is which.

# **egrep -w '135|136|137|138|139|445' /etc/services**

loc-srv 135/tcp epmap #Location Service

loc-srv 135/udp epmap #Location Service

profile 136/tcp #PROFILE Naming System

profile 136/udp #PROFILE Naming System

netbios-ns 137/tcp #NETBIOS Name Service

netbios-ns 137/udp #NETBIOS Name Service

netbios-dgm 138/tcp #NETBIOS Datagram Service

netbios-dgm 138/udp #NETBIOS Datagram Service

netbios-ssn 139/tcp #NETBIOS Session Service

netbios-ssn 139/udp #NETBIOS Session Service

microsoft-ds 445/tcp

microsoft-ds 445/udp

# **lsof -i tcp:135-139 -o -i udp:135-139 -o -i tcp:445 -o -i udp:445**

COMMAND PID USER FD TYPE DEVICE OFFSET NODE NAME

samba 691 root 41u IPv6 0xc321d328 0t0 TCP \*:loc-srv (LISTEN)

samba 691 root 42u IPv4 0xc321d000 0t0 TCP \*:loc-srv (LISTEN)

smbd 692 root 44u IPv6 0xc321f978 0t0 TCP \*:microsoft-ds (LISTEN)

smbd 692 root 45u IPv6 0xc3480978 0t0 TCP \*:netbios-ssn (LISTEN)

smbd 692 root 46u IPv4 0xc3480650 0t0 TCP \*:microsoft-ds (LISTEN)

smbd 692 root 47u IPv4 0xc3480328 0t0 TCP \*:netbios-ssn (LISTEN)

samba 693 root 24u IPv4 0xc31372e4 0t0 UDP \*:netbios-ns

samba 693 root 27u IPv4 0xc31372d0 0t0 UDP \*:netbios-dgm

samba 693 root 29u IPv4 0xc31372bc 0t0 UDP 10.1.1.255:netbios-ns

samba 693 root 30u IPv4 0xc31372a8 0t0 UDP freebsd.corp.example.com:netbios-ns

samba 693 root 31u IPv4 0xc3137294 0t0 UDP 10.1.1.255:netbios-dgm

samba 693 root 32u IPv4 0xc3137280 0t0 UDP freebsd.corp.example.com:netbios-dg

LDAP, LDAP/S

LDAP is needed for the directory service. We need both LDAP and Kerberos for Windows clients to move beyond old-style NetBIOS/LANMAN operation.

# **lsof -i tcp:389 -o -i udp:389 -o -i tcp:636 -o -i udp:636**

COMMAND PID USER FD TYPE DEVICE OFFSET NODE NAME

samba 695 root 24u IPv6 0xc3381328 0t0 TCP \*:ldap (LISTEN)

samba 695 root 29u IPv6 0xc3381000 0t0 TCP \*:ldaps (LISTEN)

samba 695 root 33u IPv4 0xc3380650 0t0 TCP \*:ldap (LISTEN)

samba 695 root 34u IPv4 0xc3380328 0t0 TCP \*:ldaps (LISTEN)

samba 696 root 24u IPv6 0xc313726c 0t0 UDP \*:ldap

samba 696 root 30u IPv4 0xc3137258 0t0 UDP \*:ldap

samba 696 root 32u IPv6 0xc3137244 0t0 UDP freebsd.corp.example.com:ldap

samba 696 root 33u IPv4 0xc3137230 0t0 UDP freebsd.corp.example.com:ldap

Kerberos

Finally, Kerberos authenticates users and issues service tickets.

# **lsof -i tcp:88 -o -i udp:88 -o -i tcp:464 -o -i udp:464**

COMMAND PID USER FD TYPE DEVICE OFFSET NODE NAME

samba 697 root 24u IPv6 0xc3382000 0t0 TCP \*:kerberos-sec (LISTEN)

samba 697 root 31u IPv6 0xc313721c 0t0 UDP \*:kerberos-sec

samba 697 root 33u IPv6 0xc3381ca0 0t0 TCP \*:kpasswd5 (LISTEN)

samba 697 root 34u IPv6 0xc3137208 0t0 UDP \*:kpasswd5

samba 697 root 35u IPv4 0xc3381978 0t0 TCP \*:kerberos-sec (LISTEN)

samba 697 root 36u IPv4 0xc31371f4 0t0 UDP \*:kerberos-sec

samba 697 root 37u IPv4 0xc3381650 0t0 TCP \*:kpasswd5 (LISTEN)

samba 697 root 38u IPv4 0xc31371e0 0t0 UDP \*:kpasswd5

samba 697 root 39u IPv6 0xc31371cc 0t0 UDP freebsd.corp.example.com:kerberos-sec

samba 697 root 40u IPv6 0xc31371b8 0t0 UDP freebsd.corp.example.com:kpasswd5

samba 697 root 41u IPv4 0xc31371a4 0t0 UDP freebsd.corp.example.com:kerberos-sec

samba 697 root 42u IPv4 0xc3137190 0t0 UDP freebsd.corp.example.com:kpasswd5

Testing

Let's see if we can get a list of the shares from another system:

$ **smbclient -L //freebsd -U administrator**

Enter administrator's password: \*\*\*\*\*\*\*\*\*\*\*\*

Domain=[CORP] OS=[Windows 6.1] Server=[Samba 4.3.13]

Sharename Type Comment

--------- ---- -------

netlogon Disk

sysvol Disk

IPC$ IPC IPC Service (Samba 4.3.13)

Domain=[CORP] OS=[Windows 6.1] Server=[Samba 4.3.13]

Server Comment

--------- -------

Workgroup Master

--------- -------

The netlogon and sysvol shares must exist on a domain controller. They were created automatically during the provisioning. Let's test the sysvol share:

$ **smbclient //freebsd/sysvol -U administrator -c 'dir'**

Enter administrator's password: \*\*\*\*\*\*\*\*\*\*\*\*

Domain=[CORP] OS=[Windows 6.1] Server=[Samba 4.3.13]

. D 0 Fri Feb 17 10:40:28 2018

.. D 0 Fri Feb 17 13:57:46 2018

corp.example.com D 0 Fri Feb 17 10:40:28 2018

30162908 blocks of size 1024. 25800308 blocks available

Let's see if we can get information about the AD server. I should be able to do that from the AD server itself *and* from another system, so long as its Samba configuration file /etc/samba/smb.conf is set up reasonably. Its [global] stanza should have a realm = *REALM\_NAME* line.

$ **net ads info**

LDAP server: 10.1.1.235

LDAP server name: freebsd.corp.example.com

Realm: CORP.EXAMPLE.COM

Bind Path: dc=CORP,dc=EXAMPLE,dc=COM

LDAP port: 389

Server time: Wed, 15 Mar 2017 14:16:53 EST

KDC server: 10.1.1.235

Server time offset: 0

Can I get Kerberos credentials from a command prompt on the AD server?

$ **klist**

klist: No ticket file: /tmp/krb5cc\_1000

$ **kinit administrator@CORP.EXAMPLE.COM**

administrator@CORP.EXAMPLE.COM's Password: \*\*\*\*\*\*\*\*\*\*\*\*

$ **klist**

Credentials cache: FILE:/tmp/krb5cc\_1000

Principal: administrator@CORP.EXAMPLE.COM

Issued Expires Principal

Feb 17 16:43:02 2018 Feb 18 00:43:01 2018 krbtgt/CORP.EXAMPLE.COM@CORP.EXAMPLE.COM

$ **klist -v**

Credentials cache: FILE:/tmp/krb5cc\_1000

Principal: administrator@CORP.EXAMPLE.COM

Cache version: 4

Server: krbtgt/CORP.EXAMPLE.COM@CORP.EXAMPLE.COM

Client: administrator@CORP.EXAMPLE.COM

Ticket etype: aes256-cts-hmac-sha1-96, kvno 1

Ticket length: 985

Auth time: Feb 17 16:43:02 2018

End time: Feb 18 00:43:01 2018

Ticket flags: pre-authent, initial, forwardable

Addresses: addressless

The above is on the AD server, so klist is from the Heimdal Kerberos implementation. I'll do this from another system where klist is from the MIT implementation. ***Unfortunately, the two implementations have different syntax.***

$ **klist**

klist: No credentials cache found (filename: /tmp/krb5cc\_1000)

$ **kinit administrator@CORP.EXAMPLE.COM**

administrator@CORP.EXAMPLE.COM's Password: \*\*\*\*\*\*\*\*\*\*\*\*

$ **klist**

Ticket cache: FILE:/tmp/krb5cc\_1000

Default principal: administrator@CORP.EXAMPLE.COM

Valid starting Expires Service principal

03/16/17 14:48:35 03/17/17 00:48:35 krbtgt/CORP.EXAMPLE.COM@CORP.EXAMPLE.COM

renew until 03/17/17 14:48:33

$ **klist -ef**

Ticket cache: FILE:/tmp/krb5cc\_1000

Default principal: administrator@CORP.EXAMPLE.COM

Valid starting Expires Service principal

03/16/17 14:48:35 03/17/17 00:48:35 krbtgt/CORP.EXAMPLE.COM@CORP.EXAMPLE.COM

renew until 03/17/17 14:48:33, Flags: RIA

Etype (skey, tkt): aes256-cts-hmac-sha1-96, aes256-cts-hmac-sha1-96

[The MIT Kerberos Administrator's How-To Guide](http://www.kerberos.org/software/adminkerberos.pdf) [MIT Kerberos Documentation](https://web.mit.edu/kerberos/krb5-current/doc/) [BIND9\_DLZ Back End Troubleshooting](https://wiki.samba.org/index.php/BIND9_DLZ_DNS_Back_End#Troubleshooting) [Red Hat Enterprise Linux Windows Integration Guide](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/pdf/Windows_Integration_Guide/Red_Hat_Enterprise_Linux-7-Windows_Integration_Guide-en-US.pdf)

It's working!

If it didn't there is troubleshooting information at [samba.org](https://wiki.samba.org/).

The MIT [**How-To Guide**](http://www.kerberos.org/software/adminkerberos.pdf) starts with a very good overview of Kerberos. MIT has further Kerberos [documentation](https://web.mit.edu/kerberos/krb5-current/doc/) if you need it. The Heimdal project has some good [documentation](https://www.h5l.org/manual/HEAD/info/heimdal/Setting-up-a-realm.html) about their implementation, although you won't be interacting with it directly.

Now we need to create users and groups

**We need to define users and groups in the LDAP directory tree and Kerberos database.** All of our testing has used Administrator, the only user defined so far. You use **samba-tool** to define AD groups and users. We will manage the Kerberos cryptography requirements with **ktutil.**

Yes, for standalone Kerberos you would use kadmin to manage the Kerberos database in /var/heimdal/heimdal.db, a Berkeley Btree database file.

However, Samba manages its own Kerberos database under /var/db/samba4/private/. **You do all interaction with the Samba database, both the LDAP directory and the Kerberos database, using the samba-tool command.** The exception to this is the use of ktutil to interact with the Kerberos keytab. You will see that in the [next step](https://cromwell-intl.com/open-source/samba-active-directory/deployment.html).

The next page takes us through a [practical deployment](https://cromwell-intl.com/open-source/samba-active-directory/deployment.html).

# Deploying Users and Groups on a Samba-based Active Directory Server

## Practical Deployments — Groups and Users in a Small Realm

We need to create groups and users for Linux systems that will use this server for user information and authentication. Our goal is to have four users: jane, joe, frank, and alice. All of them are members of the Linux group users. The first two are also members of the Linux group wheel.   
  
We will do this in a small Kerberos realm, where the realm is the same as the DNS domain. That is, all hosts are directly in the example.org DNS domain instead of subdomains like east.example.org and west.example.org. So, hosts host1.example.org, host2.example.org, and so on, in the EXAMPLE.ORG realm (and the EXAMPLE domain).

## Samba Deployment

I deployed the Samba domain this way:

# **samba-tool domain provision --use-rfc2307 \**

**--realm=EXAMPLE.ORG --domain=EXAMPLE \**

**--server-role=dc --dns-backend=BIND9\_DLZ \**

**--adminpass=P@55w0rd --krbtgtpass=P@55w0rd --machinepass=P@55w0rd**

See the previous page on [Samba deployment](https://cromwell-intl.com/open-source/samba-active-directory/samba.html) for more details on this step.

Unfortunately, this broke the standard BIND back end. The named service would not start. This server was to be a BIND slave server for all names \*.example.org and all addresses 10/8, and all addresses 192.168/16, and fc00::/16. That is, the example.org, 10.in-addr.arpa, 168.192.in-addr.arpa, and 0.0.0.0.0.0.0.0.0.0.0.0.0.0.c.f.ip6.arpa zones.

It would start if I commented out the definition of the example.org zone in /usr/local/etc/namedb/named.conf. But, I need that zone!

I discovered what was happening by modifying /var/db/samba4/private/named.conf to add a debugging option. Let's go for debug level 10, at or beyond the maximum amount of output:

[ ... lines deleted ...]

database "dlopen /usr/local/lib/shared-modules/bind9/dlz\_bind9\_10.so -d 10";

[ ... lines deleted ...]

Now I will start the process manually, with the overall debugging also dialed all the way up.

|  |  |
| --- | --- |
| -f | = run in the foreground, do not daemonize |
| -g | = run in the foreground, force all logging to stderr |
| -d 10 | = debugging level 10 |

# **named -f -g -d 10**

[ ... much output deleted ...]

24-Aug-2018 19:48:34.649 samba\_dlz:

24-Aug-2018 19:48:34.652 samba\_dlz:

24-Aug-2018 19:48:34.655 samba\_dlz: ldb: ldb\_trace\_response: DONE

24-Aug-2018 19:48:34.658 samba\_dlz: error: 0

24-Aug-2018 19:48:34.660 samba\_dlz:

24-Aug-2018 19:48:34.666 samba\_dlz: Failed to configure zone 'example.org'

24-Aug-2018 19:48:34.713 load\_configuration: already exists

24-Aug-2018 19:48:34.716 loading configuration: already exists

24-Aug-2018 19:48:34.718 exiting (due to fatal error)

Everything I found on the topic explained that "of course" you should not do DNS with BIND, you should use samba-tool to define the zone and add the records. Then run DNS out of Samba.

To stretch the metaphor much too far, this would have Samba's LDAP and Kerberos as the tail wagging the dog of DNS. No thanks, I'll keep DNS service on BIND. What I did was simply edit the main named.conf and edit out the reference to include the named.conf that was a part of Samba.

On FreeBSD, that meant editing /usr/local/etc/namedb/named.conf and putting /\*...\*/ comment delimiters around the line including /var/db/samba4/private/named.conf.

### Defining LDAP and CIFS Service Principals

I had to define a **Service Principal Name** or **SPN** for each of the LDAP and CIFS services on the new AD server.

# **samba-tool spn add ldap/freebsd.example.org Administrator**

# **samba-tool spn add cifs/freebsd.example.org Administrator**

# **samba-tool spn list Administrator**

administrator

User CN=Administrator,CN=Users,DC=example,DC=org has the following servicePrincipalName:

ldap/freebsd.example.org

cifs/freebsd.example.org

# **samba-tool domain exportkeytab /etc/krb5.keytab**

# **ls -l /etc/krb5.keytab**

-rwxr-xr-x 1 root 1001 1632 Aug 24 14:23 /etc/krb5.keytab

## Standards? What Standards?

Many Linux distributions have predefined groups wheel and users. CentOS, Red Hat Enterprise, Fedora/Pidora, and Mageia Linux (among others) have these:

CentOS:$ **egrep 'users|wheel' /etc/group**

wheel:x:10:cromwell

users:x:100:

Not all Linux distributions have both. Raspbian, derived from Debian, does not have wheel:

Raspbian:$ **egrep 'users|wheel' /etc/group**

users:x:100:

OpenBSD has both groups, although with different UIDs:

OpenBSD:$ **egrep 'users|wheel' /etc/group**

wheel:\*:0:root,cromwell

users:\*:10:

FreeBSD has only wheel:

FreeBSD:$ **egrep 'users|wheel' /etc/group**

wheel:\*:0:root,cromwell

The rest of this will assume the non-Debian Linux model, wheel/10 and users/100.

## Defining Groups and Users

I will add a group wheel, and users jane, joe, frank, and alice. All users will belong to the already existing group users, and jane and joe will also belong to group wheel.

[RFC 2307](https://tools.ietf.org/html/rfc2307) requires that I specify an NIS domain for the group.

These commands allow account creation to be partially automated with scripts. **Beware:** A user running the ps command at the right time would see the password. At least three solutions are possible:

1. **The most likely solution** is that you are not going to have untrusted users who can run commands on the AD server. The problem is avoided.
2. Use the --random-password option in place of the explicit password method, and capture the script's output to record the password.
3. Run the script manually, and type the new password twice when it asks for it mid-way through.

In this example, let's assume there are no untrusted users on the AD server, and we will set passwords in the commands.

# **samba-tool group add wheel \**

**--gid-number=10 \**

**--nis-domain=example**

Added group wheel

# **samba-tool user create jane janePassword7 \**

**--uid=jane --uid-number=10001 --gid-number=100 \**

**--unix-home=/home/jane --home-directory=/home/jane \**

**--login-shell=/bin/bash \**

**--gecos='Jane User' --given-name=Jane --surname=User**

User 'jane' created successfully

# **samba-tool user create joe joePassword7 \**

**--uid=joe --uid-number=10002 --gid-number=100 \**

**--unix-home=/home/joe --home-directory=/home/joe \**

**--login-shell=/bin/bash \**

**--gecos='Joe User' --given-name=Joe --surname=User**

User 'joe' created successfully

# **samba-tool user create frank frankPassword7 \**

**--uid=frank --uid-number=10002 --gid-number=100 \**

**--unix-home=/home/frank --home-directory=/home/frank \**

**--login-shell=/bin/bash \**

**--gecos='Frank User' --given-name=Frank --surname=User**

User 'frank' created successfully

# **samba-tool user create alice alicePassword7 \**

**--uid=alice --uid-number=10002 --gid-number=100 \**

**--unix-home=/home/alice --home-directory=/home/alice \**

**--login-shell=/bin/bash \**

**--gecos='Alice User' --given-name=Alice --surname=User**

User 'alice' created successfully

# **samba-tool user list**

Administrator

dns-freebsd

alice

frank

jane

joe

krbtgt

Guest

# **samba-tool group addmembers users joe,jane,frank,alice**

Added members to group users

# **samba-tool group addmembers wheel joe,jane**

Added members to group wheel

# **samba-tool group listmembers users**

jane

S-1-5-11

frank

alice

S-1-5-4

joe

Domain Users

# **samba-tool group listmembers wheel**

jane

joe

# **samba-tool domain exportkeytab /etc/krb5.keytab**

# **ls -l /etc/krb5.keytab**

-rwxr-xr-x 1 root 1001 2837 Aug 24 14:39 /etc/krb5.keytab

## Discovering the Kerberos Realm

Let's discover the Kerberos realm from another machine.

# **cat /etc/samba/smb.conf**

[global]

security = ads

realm = EXAMPLE.ORG

workgroup = EXAMPLE

passdb backend = tdbsam

kerberos method = secrets and keytab

template shell = /bin/bash

winbind offline logon = true

winbind use default domain = Yes

winbind enum users = Yes

winbind enum groups = Yes

idmap config \* : backend = rid

idmap config \* : range = 10000-20000

# **realm discover EXAMPLE.ORG**

example.org

type: kerberos

realm-name: EXAMPLE.ORG

domain-name: example.org

configured: no

server-software: active-directory

client-software: sssd

required-package: oddjob

required-package: oddjob-mkhomedir

required-package: sssd

required-package: adcli

required-package: samba-common-tools

### Joining a Host to the Domain

Without the ldap/freebsd.example.org principal, attempts to join a host to the domain returned error messages about **"Server not found in Kerberos database"**.

adcli: couldn't connect to EXAMPLE.ORG domain: Couldn't authenticate to active directory: SASL(-1): generic failure: GSSAPI Error: Unspecified GSS failure. Minor code may provide more information (**Server not found in Kerberos database**)

However, I have defined the ldap/freebsd.example.org principal. Now I can join a host to the domain. I'm doing the below on centos7.example.org:

# **adcli join -v EXAMPLE.ORG -U Administrator**

\* Using domain name: EXAMPLE.ORG

\* Calculated computer account name from fqdn: CENTOS7

\* Calculated domain realm from name: EXAMPLE.ORG

\* Discovering domain controllers: \_ldap.\_tcp.EXAMPLE.ORG

\* Sending netlogon pings to domain controller: ldap://[fc00::ba27:ebff:fe41:b9ae]

\* Sending netlogon pings to domain controller: cldap://10.1.1.235

\* Received NetLogon info from: freebsd.example.org

\* Wrote out krb5.conf snippet to /tmp/adcli-krb5-43jV32/krb5.d/adcli-krb5-conf-sUDJ1j

Password for Administrator@EXAMPLE.ORG: \*\*\*\*\*\*\*\*\*\*\*\*

\* Authenticated as user: Administrator@EXAMPLE.ORG

\* Looked up short domain name: EXAMPLE

\* Using fully qualified name: centos7.example.org

\* Using domain name: EXAMPLE.ORG

\* Using computer account name: CENTOS7

\* Using domain realm: EXAMPLE.ORG

\* Calculated computer account name from fqdn: CENTOS7

\* Generated 120 character computer password

\* Using keytab: FILE:/etc/krb5.keytab

\* Computer account for CENTOS7$ does not exist

\* Found well known computer container at: CN=Computers,DC=example,DC=org

\* Calculated computer account: CN=CENTOS7,CN=Computers,DC=example,DC=org

\* Created computer account: CN=CENTOS7,CN=Computers,DC=example,DC=org

\* Set computer password

\* Retrieved kvno '1' for computer account in directory: CN=CENTOS7,CN=Computers,DC=example,DC=org

\* Modifying computer account: dNSHostName

\* Modifying computer account: userAccountControl

\* Modifying computer account: operatingSystem, operatingSystemVersion, operatingSystemServicePack

\* Modifying computer account: userPrincipalName

\* Discovered which keytab salt to use

\* Added the entries to the keytab: CENTOS7$@EXAMPLE.ORG: FILE:/etc/krb5.keytab

\* Cleared old entries from keytab: FILE:/etc/krb5.keytab

\* Added the entries to the keytab: host/CENTOS7@EXAMPLE.ORG: FILE:/etc/krb5.keytab

\* Cleared old entries from keytab: FILE:/etc/krb5.keytab

\* Added the entries to the keytab: host/centos7.example.org@EXAMPLE.ORG: FILE:/etc/krb5.keytab

\* Cleared old entries from keytab: FILE:/etc/krb5.keytab

\* Added the entries to the keytab: RestrictedKrbHost/CENTOS7@EXAMPLE.ORG: FILE:/etc/krb5.keytab

\* Cleared old entries from keytab: FILE:/etc/krb5.keytab

\* Added the entries to the keytab: RestrictedKrbHost/centos7.example.org@EXAMPLE.ORG: FILE:/etc/krb5.keytab

The next time, instead of:  
Computer account for CENTOS7$ does not exist  
I will see:  
Found computer account for CENTOS7$ at: CN=CENTOS7,CN=Computers,DC=example,DC=org

Now I can plan the user and group deployment.

## Listing the Keytab

Let's see the principals and keys defined in the Kerberos keytab file.

# **ktutil list**

FILE:/etc/krb5.keytab:

Vno Type Principal Aliases

1 arcfour-hmac-md5 FREEBSD$@EXAMPLE.ORG

1 aes256-cts-hmac-sha1-96 FREEBSD$@EXAMPLE.ORG

1 aes128-cts-hmac-sha1-96 FREEBSD$@EXAMPLE.ORG

1 des-cbc-md5 FREEBSD$@EXAMPLE.ORG

1 des-cbc-crc FREEBSD$@EXAMPLE.ORG

1 arcfour-hmac-md5 Administrator@EXAMPLE.ORG

1 aes256-cts-hmac-sha1-96 Administrator@EXAMPLE.ORG

1 aes128-cts-hmac-sha1-96 Administrator@EXAMPLE.ORG

1 des-cbc-md5 Administrator@EXAMPLE.ORG

1 des-cbc-crc Administrator@EXAMPLE.ORG

1 arcfour-hmac-md5 CENTOS7$@EXAMPLE.ORG

1 aes256-cts-hmac-sha1-96 CENTOS7$@EXAMPLE.ORG

1 aes128-cts-hmac-sha1-96 CENTOS7$@EXAMPLE.ORG

1 des-cbc-md5 CENTOS7$@EXAMPLE.ORG

1 des-cbc-crc CENTOS7$@EXAMPLE.ORG

1 arcfour-hmac-md5 dns-freebsd@EXAMPLE.ORG

1 aes256-cts-hmac-sha1-96 dns-freebsd@EXAMPLE.ORG

1 aes128-cts-hmac-sha1-96 dns-freebsd@EXAMPLE.ORG

1 des-cbc-md5 dns-freebsd@EXAMPLE.ORG

1 des-cbc-crc dns-freebsd@EXAMPLE.ORG

1 arcfour-hmac-md5 alice@EXAMPLE.ORG

1 aes256-cts-hmac-sha1-96 alice@EXAMPLE.ORG

1 aes128-cts-hmac-sha1-96 alice@EXAMPLE.ORG

1 des-cbc-md5 alice@EXAMPLE.ORG

1 des-cbc-crc alice@EXAMPLE.ORG

1 arcfour-hmac-md5 frank@EXAMPLE.ORG

1 aes256-cts-hmac-sha1-96 frank@EXAMPLE.ORG

1 aes128-cts-hmac-sha1-96 frank@EXAMPLE.ORG

1 des-cbc-md5 frank@EXAMPLE.ORG

1 des-cbc-crc frank@EXAMPLE.ORG

1 arcfour-hmac-md5 jane@EXAMPLE.ORG

1 aes256-cts-hmac-sha1-96 jane@EXAMPLE.ORG

1 aes128-cts-hmac-sha1-96 jane@EXAMPLE.ORG

1 des-cbc-md5 jane@EXAMPLE.ORG

1 des-cbc-crc jane@EXAMPLE.ORG

1 arcfour-hmac-md5 joe@EXAMPLE.ORG

1 aes256-cts-hmac-sha1-96 joe@EXAMPLE.ORG

1 aes128-cts-hmac-sha1-96 joe@EXAMPLE.ORG

1 des-cbc-md5 joe@EXAMPLE.ORG

1 des-cbc-crc joe@EXAMPLE.ORG

1 arcfour-hmac-md5 krbtgt@EXAMPLE.ORG

1 aes256-cts-hmac-sha1-96 krbtgt@EXAMPLE.ORG

1 aes128-cts-hmac-sha1-96 krbtgt@EXAMPLE.ORG

1 des-cbc-md5 krbtgt@EXAMPLE.ORG

1 des-cbc-crc krbtgt@EXAMPLE.ORG

Look at the outdated DES and RC4 cipher support! Let's disable those:

# **foreach cipher ( arcfour-hmac-md5 des-cbc-md5 des-cbc-crc )**

foreach? **ktutil remove --enctype=$cipher**

foreach? **end**

# **ktutil list**

FILE:/etc/krb5.keytab:

Vno Type Principal Aliases

1 aes256-cts-hmac-sha1-96 FREEBSD$@EXAMPLE.ORG

1 aes128-cts-hmac-sha1-96 FREEBSD$@EXAMPLE.ORG

1 aes256-cts-hmac-sha1-96 Administrator@EXAMPLE.ORG

1 aes128-cts-hmac-sha1-96 Administrator@EXAMPLE.ORG

1 aes256-cts-hmac-sha1-96 CENTOS7$@EXAMPLE.ORG

1 aes128-cts-hmac-sha1-96 CENTOS7$@EXAMPLE.ORG

1 aes256-cts-hmac-sha1-96 dns-freebsd@EXAMPLE.ORG

1 aes128-cts-hmac-sha1-96 dns-freebsd@EXAMPLE.ORG

1 aes256-cts-hmac-sha1-96 alice@EXAMPLE.ORG

1 aes128-cts-hmac-sha1-96 alice@EXAMPLE.ORG

1 aes256-cts-hmac-sha1-96 frank@EXAMPLE.ORG

1 aes128-cts-hmac-sha1-96 frank@EXAMPLE.ORG

1 aes256-cts-hmac-sha1-96 jane@EXAMPLE.ORG

1 aes128-cts-hmac-sha1-96 jane@EXAMPLE.ORG

1 aes256-cts-hmac-sha1-96 joe@EXAMPLE.ORG

1 aes128-cts-hmac-sha1-96 joe@EXAMPLE.ORG

1 aes256-cts-hmac-sha1-96 krbtgt@EXAMPLE.ORG

1 aes128-cts-hmac-sha1-96 krbtgt@EXAMPLE.ORG

You can add the --keys option if you want to see the keys.

## Getting a User Ticket Remotely

CentOS7:$ **klist**

klist: No credentials cache found (filename: /tmp/krb5cc\_0)

CentOS7:$ **kinit frank**

Password for frank@EXAMPLE.ORG: frankPassword7

CentOS7:$ **klist -ef**

Ticket cache: FILE:/tmp/krb5cc\_0

Default principal: frank@EXAMPLE.ORG

Last modified: 2018-08-24 19:37:10 UTC

Valid starting Expires Service principal

08/17/2018 23:17:09 08/18/2018 09:17:09 krbtgt/EXAMPLE.ORG@EXAMPLE.ORG

renew until 08/18/2018 23:17:05, Flags: RIA

Etype (skey, tkt): aes256-cts-hmac-sha1-96, aes256-cts-hmac-sha1-96

## Local Authentication through Active Directory

### Configuration file changes

I had /etc/samba.smb.conf set up as shown [above](https://cromwell-intl.com/open-source/samba-active-directory/deployment.html#smbconf).

The file /etc/krb5.conf contains this:

includedir /etc/krb5.conf.d/

includedir /var/lib/sss/pubconf/krb5.include.d/

[libdefaults]

default\_realm = EXAMPLE.ORG

rdns = false

dns\_lookup\_realm = false

dns\_lookup\_kdc = true

[realms]

EXAMPLE.ORG = {

kdc = freebsd.example.org

}

EXAMPLE = {

kdc = freebsd.example.org

}

[domain\_realm]

example.org = EXAMPLE.ORG

.example.org = EXAMPLE.ORG

The file /etc/nsswitch.conf contains these lines:

[... lines deleted ...]

passwd: files winbind

shadow: files winbind

group: files winbind

[... lines deleted ...]

hosts: files dns myhostname

[... lines deleted ...]

Four PAM files were changed:  
fingerprint-auth-ac  
password-auth-ac  
smartcard-auth-ac  
system-auth-ac   
The file /etc/pam.d/fingerprint-auth-ac contains this:

#%PAM-1.0

# This file is auto-generated.

# User changes will be destroyed the next time authconfig is run.

auth required pam\_env.so

auth sufficient pam\_fprintd.so

auth required pam\_deny.so

account required pam\_unix.so broken\_shadow

account sufficient pam\_localuser.so

account sufficient pam\_succeed\_if.so uid < 1000 quiet

account [default=bad success=ok user\_unknown=ignore] pam\_winbind.so cached\_login

account required pam\_permit.so

password required pam\_deny.so

session optional pam\_keyinit.so revoke

session required pam\_limits.so

-session optional pam\_systemd.so

session [success=1 default=ignore] pam\_succeed\_if.so service in crond quiet use\_uid

session required pam\_unix.so

session optional pam\_winbind.so cached\_login

The file /etc/pam.d/password-auth-ac contains this:

#%PAM-1.0

# This file is auto-generated.

# User changes will be destroyed the next time authconfig is run.

auth required pam\_env.so

auth sufficient pam\_unix.so nullok try\_first\_pass

auth requisite pam\_succeed\_if.so uid >= 1000 quiet\_success

auth sufficient pam\_winbind.so cached\_login use\_first\_pass

auth required pam\_deny.so

account required pam\_unix.so broken\_shadow

account sufficient pam\_localuser.so

account sufficient pam\_succeed\_if.so uid < 1000 quiet

account [default=bad success=ok user\_unknown=ignore] pam\_winbind.so cached\_login

account required pam\_permit.so

password requisite pam\_pwquality.so try\_first\_pass local\_users\_only retry=3 authtok\_type=

password sufficient pam\_unix.so sha512 shadow nullok try\_first\_pass use\_authtok

password sufficient pam\_winbind.so use\_authtok

password required pam\_deny.so

session optional pam\_keyinit.so revoke

session required pam\_limits.so

-session optional pam\_systemd.so

session [success=1 default=ignore] pam\_succeed\_if.so service in crond quiet use\_uid

session required pam\_unix.so

session optional pam\_winbind.so cached\_login

The only differences between fingerprint-auth-ac and smartcard-auth-ac are changes within two lines, #5 and #14:

# **diff fingerprint-auth-ac smartcard-auth-ac**

5c5

< auth sufficient pam\_fprintd.so

---

> auth [success=done ignore=ignore default=die] pam\_pkcs11.so nodebug wait\_for\_card

14c14

< password required pam\_deny.so

---

> password required pam\_pkcs11.so

The only difference between password-auth-ac and system-auth-ac is that the second file has an added line inserted after line 4:

# **diff password-auth-ac system-auth-ac**

4a5

> auth sufficient pam\_fprintd.so

The file /etc/pam.d/postlogin-ac contains the following, which I believe is the original content:

#%PAM-1.0

# This file is auto-generated.

# User changes will be destroyed the next time authconfig is run.

session [success=1 default=ignore] pam\_succeed\_if.so service !~ gdm\* service !~ su\* quiet

session [default=1] pam\_lastlog.so nowtmp showfailed

session optional pam\_lastlog.so silent noupdate showfailed

The file /etc/sysconfig/authconfig contains these lines:

[... lines deleted ...]

USEKERBEROS=no

USELDAP=no

USELDAPAUTH=no

[... lines deleted ...]

USESSSD=yes

USESSSDAUTH=no

USESYSNETAUTH=no

USEWINBIND=yes

USEWINBINDAUTH=yes

WINBINDKRB5=no

### Testing the Configuration

First, see if winbind can talk to the AD server. Ask for lists of users and groups, and for AD server info. You should see the groups and users we set up earlier. My examples are highlighted here.

# **wbinfo -u**

administrator

dns-freebsd

alice

frank

jane

joe

krbtgt

guest

# **wbinfo -g**

allowed rodc password replication group

enterprise read-only domain controllers

denied rodc password replication group

read-only domain controllers

group policy creator owners

ras and ias servers

domain controllers

enterprise admins

domain computers

cert publishers

dnsupdateproxy

domain admins

domain guests

schema admins

domain users

dnsadmins

wheel

# **net ads lookup**

Information for Domain Controller: 10.1.1.235

Response Type: LOGON\_SAM\_LOGON\_RESPONSE\_EX

GUID: 0af1eaef-4599-4ccc-9d27-2190c1365867

Flags:

Is a PDC: yes

Is a GC of the forest: yes

Is an LDAP server: yes

Supports DS: yes

Is running a KDC: yes

Is running time services: yes

Is the closest DC: yes

Is writable: yes

Has a hardware clock: yes

Is a non-domain NC serviced by LDAP server: no

Is NT6 DC that has some secrets: no

Is NT6 DC that has all secrets: yes

Runs Active Directory Web Services: no

Runs on Windows 2012 or later: no

Forest: example.org

Domain: example.org

Domain Controller: freebsd.example.org

Pre-Win2k Domain: EXAMPLE

Pre-Win2k Hostname: FREEBSD

Server Site Name : Default-First-Site-Name

Client Site Name : Default-First-Site-Name

NT Version: 5

LMNT Token: ffff

LM20 Token: ffff

Also try net ads status -U administrator | less, and be ready for a lot of output

Now, a bigger step, is nsswitch set up correctly? Ask it for the passwd and groups lists. For each, you should see the contents of the local file followed by those in the AD database.

# **getent passwd**

[... lines with all but the last two entries of /etc/passwd deleted ...]

tcpdump:x:72:72::/:/sbin/nologin

cromwell:x:1000:1000::/home/cromwell:/usr/bin/tcsh

dirsrv:x:389:389:389-ds-base:/usr/share/dirsrv:/sbin/nologin

administrator:\*:10500:10513:Administrator:/home/EXAMPLE/administrator:/bin/bash

dns-freebsd:\*:11101:10513:dns-freebsd:/home/EXAMPLE/dns-freebsd:/bin/bash

alice:\*:11108:10513:Alice User:/home/EXAMPLE/alice:/bin/bash

frank:\*:16777220:10513:Frank User:/home/EXAMPLE/frank:/bin/bash

jane:\*:16777221:10513:Jane User:/home/EXAMPLE/jane:/bin/bash

joe:\*:11106:10513:Joe User:/home/EXAMPLE/joe:/bin/bash

krbtgt:\*:10502:10513:krbtgt:/home/EXAMPLE/krbtgt:/bin/bash

guest:\*:10501:10514:Guest:/home/EXAMPLE/guest:/bin/bash

# **getent group**

[... lines with all but the last two entries of /etc/group deleted ...]

tcpdump:x:72:

cromwell:x:1000:

dirsrv:x:389:

wbpriv:x:88:

allowed rodc password replication group:x:10571:

enterprise read-only domain controllers:x:10498:

denied rodc password replication group:x:10572:

read-only domain controllers:x:10521:

group policy creator owners:x:10520:

ras and ias servers:x:10553:

domain controllers:x:10516:

enterprise admins:x:10519:

domain computers:x:10515:

cert publishers:x:10517:

dnsupdateproxy:x:11103:

domain admins:x:10512:

domain guests:x:10514:

schema admins:x:10518:

domain users:x:10513:

dnsadmins:x:11102:

wheel:x:11104:

### User authentication

At this point you should be able to refer to users defined within the AD database. In the following, cromwell is defined locally and the others are in AD:

# **echo ~cromwell**

/home/cromwell

# **echo ~jane**

/home/EXAMPLE/jane

# **echo ~frank**

/home/EXAMPLE/frank

# **id cromwell**

uid=1000(cromwell) gid=1000(cromwell) groups=1000(cromwell),4(adm),7(lp),10(wheel),11(cdrom),63(audio),190(systemd-journal)

# **id jane**

uid=16777221(jane) gid=10513(domain users) groups=10513(domain users),11104(wheel),16777219(BUILTIN\users)

# **id frank**

uid=16777220(frank) gid=10513(domain users) groups=10513(domain users),16777219(BUILTIN\users)

I specified home directories in /home/username with both the --unix-home= and --home-directory= options, but they still end up as /home/DOMAIN/username.

I was able to log in on the text console as user frank using the password set when creating the account. The event left the following tracks in /var/log/messages:

Aug 24 18:20:35 centos7 dbus-daemon: dbus[703]: [system] Activating via systemd: service name='net.reactivated.Fprint' unit='fprintd.service'

Aug 24 18:20:35 centos7 dbus[703]: [system] Activating via systemd: service name='net.reactivated.Fprint' unit='fprintd.service'

Aug 24 18:20:35 centos7 systemd: Starting Fingerprint Authentication Daemon...

Aug 24 18:20:35 centos7 dbus-daemon: dbus[703]: [system] Successfully activated service 'net.reactivated.Fprint'

Aug 24 18:20:35 centos7 dbus[703]: [system] Successfully activated service 'net.reactivated.Fprint'

Aug 24 18:20:35 centos7 systemd: Started Fingerprint Authentication Daemon.

Aug 24 18:20:35 centos7 fprintd: Launching FprintObject

Aug 24 18:20:35 centos7 fprintd: \*\* Message: D-Bus service launched with name: net.reactivated.Fprint

Aug 24 18:20:35 centos7 fprintd: \*\* Message: entering main loop

Aug 24 18:20:40 centos7 systemd: Created slice user-16777220.slice.

Aug 24 18:20:40 centos7 systemd: Starting user-16777220.slice.

Aug 24 18:20:40 centos7 systemd-logind: New session 45 of user frank.

Aug 24 18:20:40 centos7 systemd: Started Session 45 of user frank.

Aug 24 18:20:40 centos7 systemd: Starting Session 45 of user frank.

Aug 24 18:21:05 centos7 fprintd: \*\* Message: No devices in use, exit

The following appears at the end of /var/log/secure. The user frank is not defined in the local files, so pam\_unix fails but then pam\_winbind succeeds.

Aug 24 18:20:39 centos7 login: pam\_unix(login:auth): authentication failure; logname=LOGIN uid=0 euid=0 tty=tty3 ruser= rhost= user=frank

Aug 24 18:20:39 centos7 login: pam\_winbind(login:auth): getting password (0x00000210)

Aug 24 18:20:39 centos7 login: pam\_winbind(login:auth): pam\_get\_item returned a password

Aug 24 18:20:39 centos7 login: pam\_winbind(login:auth): user 'frank' granted access

Aug 24 18:20:39 centos7 login: pam\_winbind(login:account): user 'frank' granted access

Aug 24 18:20:40 centos7 login: pam\_unix(login:session): session opened for user frank by LOGIN(uid=0)

Aug 24 18:20:40 centos7 login: LOGIN ON tty3 BY frank

All that's left is an overview of what has been added or changed.