SMB Server on OmniOS or Solaris

First steps

published: 2023 Jul 28

Kernel based/ OS included Solarish SMB server

1. About

1.1 Basics
1.2 enable SMB shares
1.3 advanced SMB settings

2. Solarish ACL

2.1 Solaris ACL vs Windows ntfs ACL
2.2 ACL inheritance
2.3 Share ACL

3. Setup ACL

3.1 ACL for toplevel folder
3.2 ACL for folders below
3.3 ACL and classic Unix permissions
3.4 ACL and Macs
3.5 NFS v3 and SMB
3.6 Windows Active Directory

4. OmniOS server for Apple clients

5. more manuals and infos

1. About the ZFS/ kernelbased SMB server

1.1 Basics

When Sun invented ZFS on Solaris they added an integrated package of storage services like IB/FC/ iSCSI, NFS and SMB fully integrated into ZFS as part of the Unix operating system and maintained with the OS. No extra 3rd party applications are needed for a fullscale enterprise storage server, everything is in Solaris or its free forks like OpenIndiana or OmniOS (Solaris world=Solarish). Especially the multithreaded kernelbased Solarish SMB service is unique on Unix. While another Unix SMB server like SAMBA offers a way to access the Unix server world via SMB, Sun tried to implement an SMB server that behaves as much as possible like a real Windows server on Unix and on a Unix filesystem like ZFS.

While on Solaris like on any other Unix filesystem owner and permission settings must rely on Unix uid/gid credidentials, Sun decided to implement full support of native Windows SID credidentials into its SMB server as additional ZFS attributes. For lokal users the SID is generated from the Unix uid, for Windows Active Directory the native AD SID is used directly. This allows to backup/ restore a ZFS filesystem with all AD permissions intact without the need of flaky mappings between Unix uid and Windows SID after restore as Windows SID contain the AD name what makes them unique unlike the Unix uid/gid that are only simple numbers.

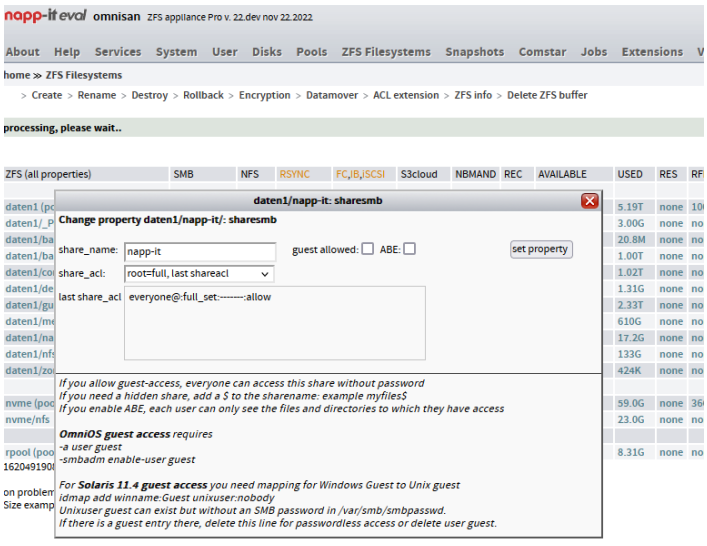
In the Unix world users with their uid can be added to a Unix group with a gid group number. Groups can only contain users not other groups. This is different to Windows where SMB groups can contain other groups. This is why you find SMB group management additionally to Unix group management on Solarish.

While the SMB server SAMBA does not know anything about ZFS and only offers regular folder sharing of any folder (regular folder or ZFS filesystem), the NFS and SMB services on Solaris are fully integrated into ZFS. This simplifies handling and allows to enable ZFS snaps = Windows previous versions without any additional settings as snaps like shares are a strict property of the same ZFS filesystem. If shares are independent from ZFS filesystems you must carefully setup or it won't work.

You can manage connected users and open files on the Solarish SMB server remote from Windows via computer management. SMB connect as a user that is member of the local SMB group administrators. Then connect computer management to your Solarish server.

1.2 Enable SMB shares

One of the most fascinating experiences is the easýness of the kernelbased SMB server. No need to adjust configfiles like smb.conf carefully or add mappings or other settings. Just open napp-it menu ZFS filesystems and click on off in the line of a filesystem under SMB. Now you can enable SMB sharing for this filesystem under a given name and select options like ABE (users only see files and folders they have permissions to) or guest access without the need to login.



1.3 Advanced SMB server settings

If you need some advanced server options like locking, signing, maxprotocol or encrypt, goto menu Services > SMB > properties . Some properties like global inheritance options are ZFS filesystem properties. Click on ACL on files in menu ZFS filesystems to modify. Windows alike is aclinherit=pass-through. You can also restrict chmod actions ex via NFS via the aclmode property.

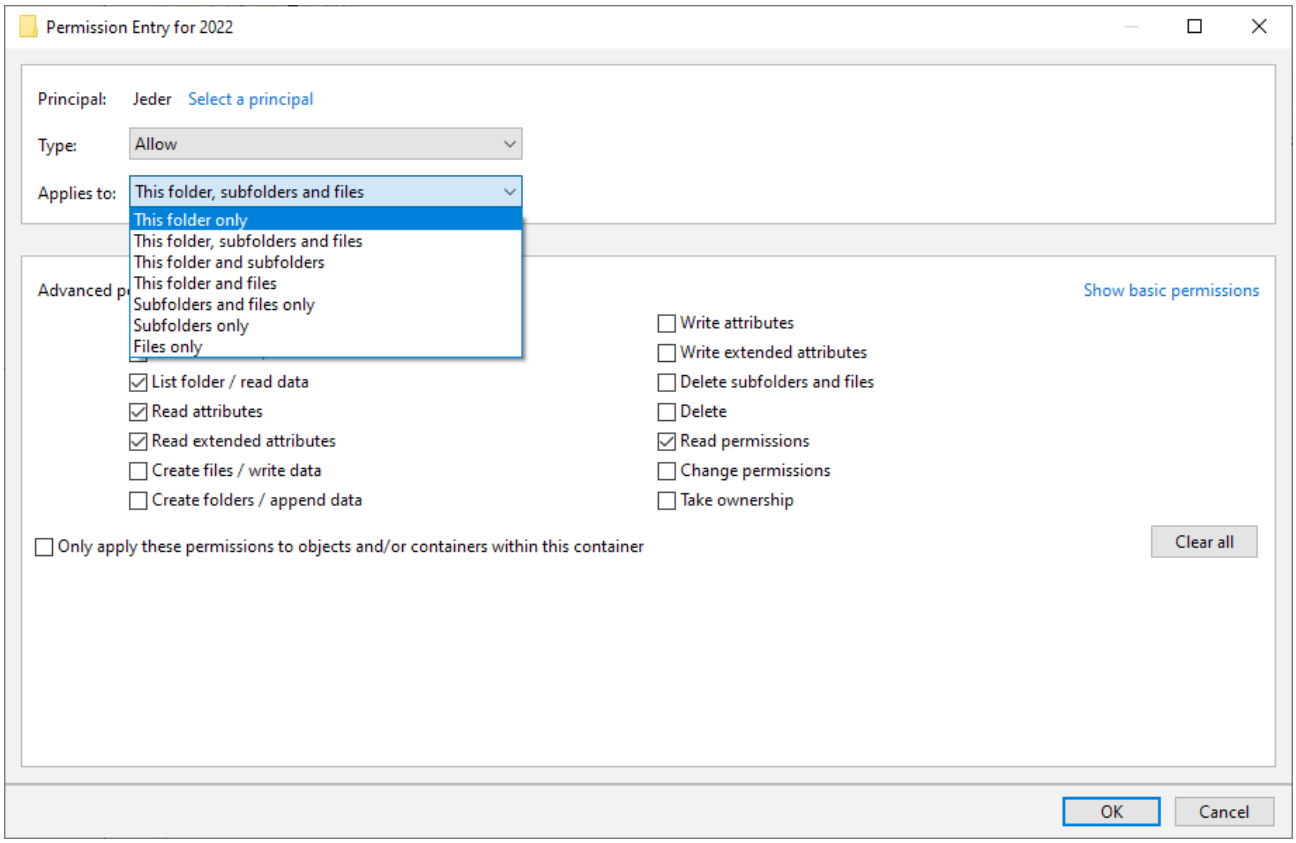
Be careful on nested ZFS filesystems below a share as lower filesystem can have different ZFS properties like casesensitivity, character sets or nbmand locking options what can cause problems. OmniOS allows nested shares, Solaris not to avoid problems.

If you use Apple systems you can enable Bonjour/mdns in menu Services > Bonjour and autostart, allow Time-Machine backups and display OmniOS via a nice Xserve icon.

2. Solarish ACL

2.1 Solarish nfsv4 ACL vs Windows ntfs ACL
https://docs.oracle.com/cd/E23824_01/html/821-1448/gbacb.html

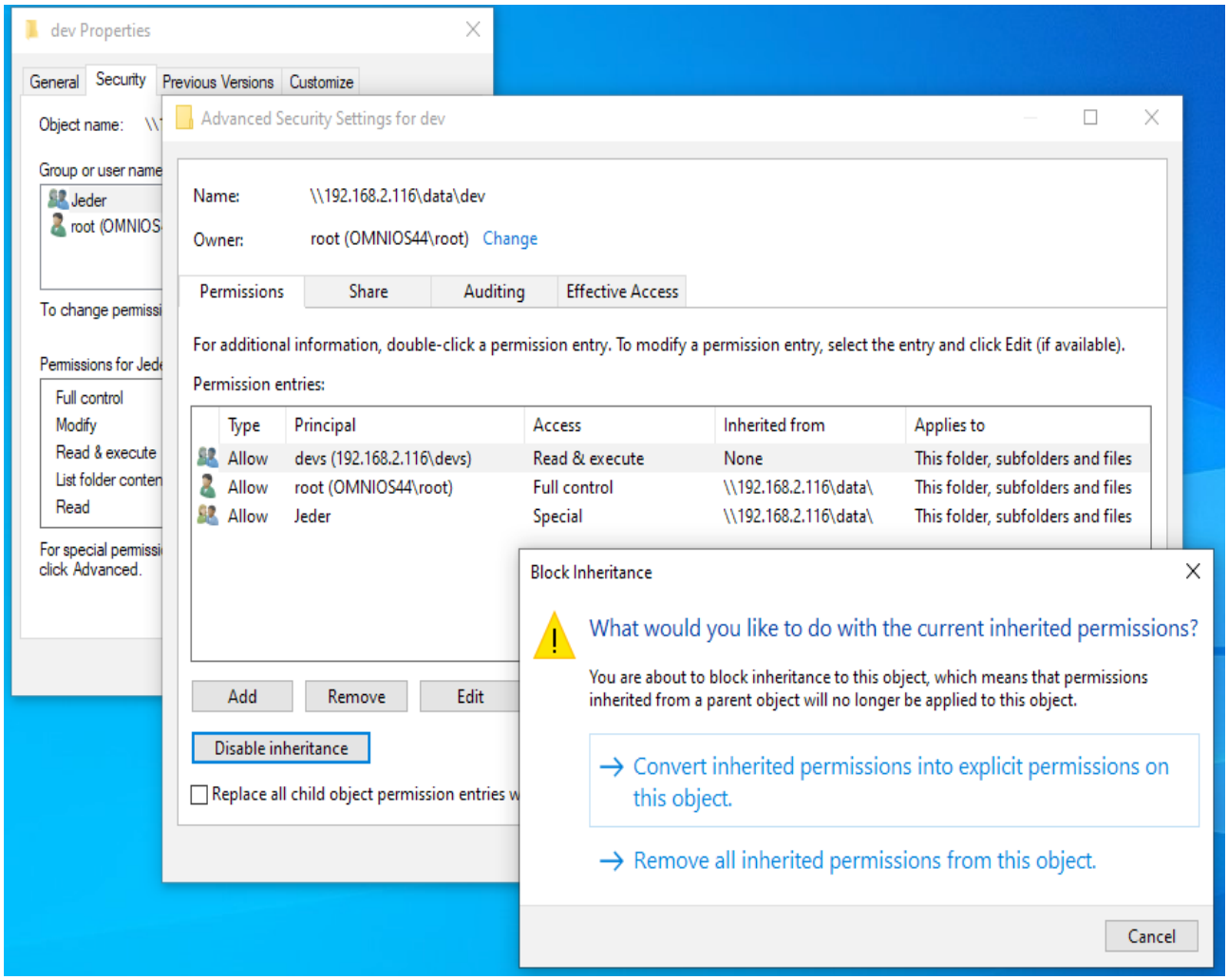
Basically it is ok to say they are quite identical regarding options. Both offer fine granular permission options with inheritance. Main difference is that Windows processes first deny then allow rules while Solarish respects order of rules where the first match defines access. Without deny rules you can mostly ignore differences.



ACL and inheritance options in Windows for a folder named 2022. If you select acl properties for a file instead a folder inheritance and folder options are missing. Compared to the classic Unix permissions „read, write, execute" fine granular ntfs/nfs4 ACL are far superiour especially when combined with inheritance. I cannot think about complex permission restrictions in a larger organisation without ACL and especially inheritance.

2.2 The inheritance mystery

ACL inheritance is the real killer feature of a modern storage server. Without inheritance permission settings in a filesystem with many subfolders and different access options for different user/ groups can be a real mess. With inheritance you start at the toplevel folder and assign permissions to that folder. You can decide if the rule applies only to this folder or also to deeper objects like folders or files. The same applies to subfolders but there you can decide to disable further inheritance from upper folders with the option to copy or remove inherited permissions.



In this screen you see settings for folder /data/devs where the first ACL is a explicit entry for this folder and the other two are inherited from toplevel folder. If you click on „Disable inheritance" you get the option to convert inherited settings to explicit settings for this folder or to remove them.

To restore inheritance settings for this sub folder, goto toplevel folder security settings and select „Replace all child object permission entries with inheritable entries from this object".

2.3 Share ACL

Usually you assign permissions individually to files and folders. For additional global or temporary restrictions you can use share ACL. These are ACL to a share control file /pool/filesstem/.zfs/shares/filesystem. This file is created when you enable a share and deleted when you disable a share. This is why share ACL are normally not persistent. Napp-it store share ACL as additional ZFS properties to allow a restore when you re-enable a share.

Set share ACL in napp-it menu ZFS filesystems > ACL on shares

3. Setup basic ACLs

3.1 Set ACL for the filesystem itself (toplevel of a share) via Windows Properties > Security (SMB connect as root)

Start with a setting that allows root full access:
add root with full access and inheritance to files and folders.

If you want to cleanup a filesystem with ACLs, use „Replace all child object permission entries with inheritable entries from this object". You can also set ACL recursively in napp-it menu ZFS filesystems > folder ACL > reset ACL ex root=full. Now only root has full access to the share and all folders and files.

3.2 Set ACL for regular folders in a share

Next step is to allow everyone@ at least access to the toplevel of the share. Add read permissions for everyone@ (Solarish reads set) but restrict to this folder only. Now everyone can access the share and nothing else.

If there are already folders ex staff for whom you want restricted access for staff members, set ACL for this folder, add SMB group staff=modify with inheritance to files and folders. Now staff can work with this folder but cannot modify ACL. If you want additionally read access for SMB group „students" add this group with read access and inheritance to files and folders. If there is a subfolder below staff ex research where you do not want staff, disable inheritance for this folder (delete entries) then add allow SMB group research with modify permissions and inheritance to files and folders to allow access to folders below research.

A special case is if you want allow creation of new files and folders (only). Add the ACL that allows file/ folder creation. The owner is the creator of the file. On Unix the owner and root always have full access. To make this visible in Windows you can add a permission that allows owner full access.

3.3 ACL and classic Unix permissions like 755

If you do a chmod like 755, you set an according explicit ACL for this folder and delete all inheritance settings as they are only available on modern ACLs. This is mostly unwanted so I would say, never use classic Unix permissions on Solaris, always use ACL.

3.4 ACL/SMB and Macs

Macs cannot see or modify ntfs/NFSv4 ACL but respect settings. You must use Windows or napp-it to set ACL. SMB problems on Macs are common, google ex „OSX problems smb" and avoid some releases like 10.14. and try the offered workarounds in other releases. Up from OmniOS 151032 there are „Apple extensions" for the kernelbased SMB server that you want to disable in some cases, see

https://illumos.topicbox.com/groups/omnios-discuss/T889d4225f9982c49-M196e9bc4f2f1275b783f0949/problem-with-smb-mount-on-macos-smb2-aapl

3.5 NFS and SMB

You often use NFS as a network filesystem ex for ESXi. You can share the same filesystem via SMB to allow ZFS snap access and restore via Windows previous verssion. Only problem is that NFS (v3) has no authentication or authorisation. NFS can only restrict access based on a client uid or a client ip, both can be faked. Depending on clients files are created with user nobody or the client uid. NFS should be only used in secure networks. To allow NFS access you should set an acl everyone@=full or modify with inheritance to files and folders. On NFS write problems reset recursively to this setting. For concurrent access with SMB set nbmand blocking setting to on (ZFS properties). To hinder NFS to modify ACL via chmod set aclmode to restricted (napp-it menu ZFS filesystems > Folder ACL)

3.6 Windows Active Directory (AD)

The Solarish kernelbased SMB server supports AD out of the box, more advanced than other solutions as it use Windows AD sid that includes the AD server as extended ZFS file attributes for advanced ntfs alike ACL for ex S-1-5-21-4125707049-143087068-3943788208-2147483748 instead simple Unix uid/gid numbers like 1002 for Posix ACL like SAMBA. This means that you can backup/ restore files with permissions intact on Solarish without the need of any id mappings to assign a Unix uid to a Windows sid when you restore files.
In workgroup mode the Solarish SMB server use also a Windows sid that is generated from the local user uid.

To use Windows AD, you must join your Solarish server to your AD server with napp-it menu Services > SMB > Active Directory. This will sync time and set the AD server as DNS server (both mandatory). If you want to set ACL for AD users from Windows, your Windows client must be an AD member as well.

What happens when you join an AD server:

```
##############################
1. sync date with AD server ex
##############################
ntpdate 192.168.1.10

#########################################################################
2. edit /etc/resolv.conf with AD server=nameserver and domain local.de
#########################################################################
search local.de
domain local.de
nameserver 192.168.2.124


######################
3. check /etc/pam.conf
######################
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the „License“).
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets „[]“ replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright 2010 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
```

```
# PAM configuration
#
# Unless explicitly defined, all services use the modules
# defined in the „other" section.
#
# Modules are defined with relative pathnames, i.e., they are
# relative to /usr/lib/security/$ISA. Absolute path names, as
# present in this file in previous releases are still acceptable.
#
# Authentication management
#
# login service (explicit because of pam_dial_auth)
#
login   auth requisite          pam_authtok_get.so.1
login   auth required           pam_dhkeys.so.1
login   auth required           pam_unix_cred.so.1
login   auth required           pam_unix_auth.so.1
login   auth required           pam_dial_auth.so.1
#
# rlogin service (explicit because of pam_rhost_auth)
#
rlogin  auth sufficient         pam_rhosts_auth.so.1
rlogin  auth requisite          pam_authtok_get.so.1
rlogin  auth required           pam_dhkeys.so.1
rlogin  auth required           pam_unix_cred.so.1
rlogin  auth required           pam_unix_auth.so.1
#
# Kerberized rlogin service
#
krlogin  auth required          pam_unix_cred.so.1
krlogin  auth required          pam_krb5.so.1
#
# rsh service (explicit because of pam_rhost_auth,
# and pam_unix_auth for meaningful pam_setcred)
#
rsh     auth sufficient         pam_rhosts_auth.so.1
rsh     auth required           pam_unix_cred.so.1
#
# Kerberized rsh service
#
krsh    auth required           pam_unix_cred.so.1
krsh    auth required           pam_krb5.so.1
#
# Kerberized telnet service
#
ktelnet  auth required          pam_unix_cred.so.1
ktelnet  auth required          pam_krb5.so.1
#
# PPP service (explicit because of pam_dial_auth)
#
ppp     auth requisite          pam_authtok_get.so.1
ppp     auth required           pam_dhkeys.so.1
ppp     auth required           pam_unix_cred.so.1
ppp     auth required           pam_unix_auth.so.1
ppp     auth required           pam_dial_auth.so.1
```

```
# GDM Autologin (explicit because of pam_allow).  These need to be
# here as there is no mechanism for packages to amend pam.conf as
# they are installed.
#
gdm-autologin auth  required    pam_unix_cred.so.1
gdm-autologin auth  sufficient  pam_allow.so.1
#
# Default definitions for Authentication management
# Used when service name is not explicitly mentioned for authentication
#
other       auth requisite              pam_authtok_get.so.1
other       auth required               pam_dhkeys.so.1
other       auth required               pam_unix_cred.so.1
other       auth required               pam_unix_auth.so.1
#
# passwd command (explicit because of a different authentication module)
#
passwd      auth required               pam_passwd_auth.so.1
#
# cron service (explicit because of non-usage of pam_roles.so.1)
#
cron        account required        pam_unix_account.so.1
#
# cups service (explicit because of non-usage of pam_roles.so.1)
#
cups        account       required       pam_unix_account.so.1
#
# GDM Autologin (explicit because of pam_allow) This needs to be here
# as there is no mechanism for packages to amend pam.conf as they are
# installed.
#
gdm-autologin account  sufficient  pam_allow.so.1
#
# Default definition for Account management
# Used when service name is not explicitly mentioned for account management
#
other       account requisite       pam_roles.so.1
other       account required        pam_unix_account.so.1
#
# Default definition for Session management
# Used when service name is not explicitly mentioned for session management
#
other       session required        pam_unix_session.so.1
#
# Default definition for Password management
# Used when service name is not explicitly mentioned for password management
#
other       password required       pam_dhkeys.so.1
other       password requisite       pam_authtok_get.so.1
other       password requisite       pam_authtok_check.so.1
other       password required       pam_authtok_store.so.1
#
# Support for Kerberos V5 authentication and example configurations can
# be found in the pam_krb5(7) man page under the „EXAMPLES“ section.
#
# smb settings set by napp-it installer
other   password required   pam_smb_passwd.so.1 nowarn
```

```
#################################################################
4. edit /etc/krb5/krb5.conf (care about lower/uppercase) example
# oDmain is local.de, AD server is 192.168.2.124
#################################################################
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the „License“).
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets „[]“ replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright 2007 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# ident       „%Z%%M%       %I%       %E% SMI“
#

# krb5.conf template
# In order to complete this configuration file
# you will need to replace the __<name>__ placeholders
# with appropriate values for your network and uncomment the
# appropriate entries.
#
[libdefaults]
        default_realm = LOCAL.DE


[realms]
        LOCAL.DE = {
                kdc = 192.168.2.124
                admin_server = 192.168.2.124
                kpasswd_server = 192.168.2.124
                kpasswd_protocol = SET_CHANGE
        }

[domain_realm]
        .local.de = LOCAL.DE

[logging]
        default = FILE:/var/krb5/kdc.log
        kdc = FILE:/var/krb5/kdc.log
```

```
        kdc_rotate = {

# How often to rotate kdc.log. Logs will get rotated no more
# often than the period, and less often if the KDC is not used
# frequently.

                period = 1d

# how many versions of kdc.log to keep around (kdc.log.0, kdc.log.1, ...)

                versions = 10
        }

[appdefaults]
        kinit = {
                renewable = true
                forwardable= true

       }

############################
5. set lmauth level ex to 4
############################
sharectl set -p lmauth_level=4 smb



################################################
6. join ad  with AD adminuser to domain local.de
################################################
smbadm join -u administrator local.de


################################################
7. restart SMB server
################################################
svcadm reload smb/server
```

Other methods for AD or LDAP directory services integration:

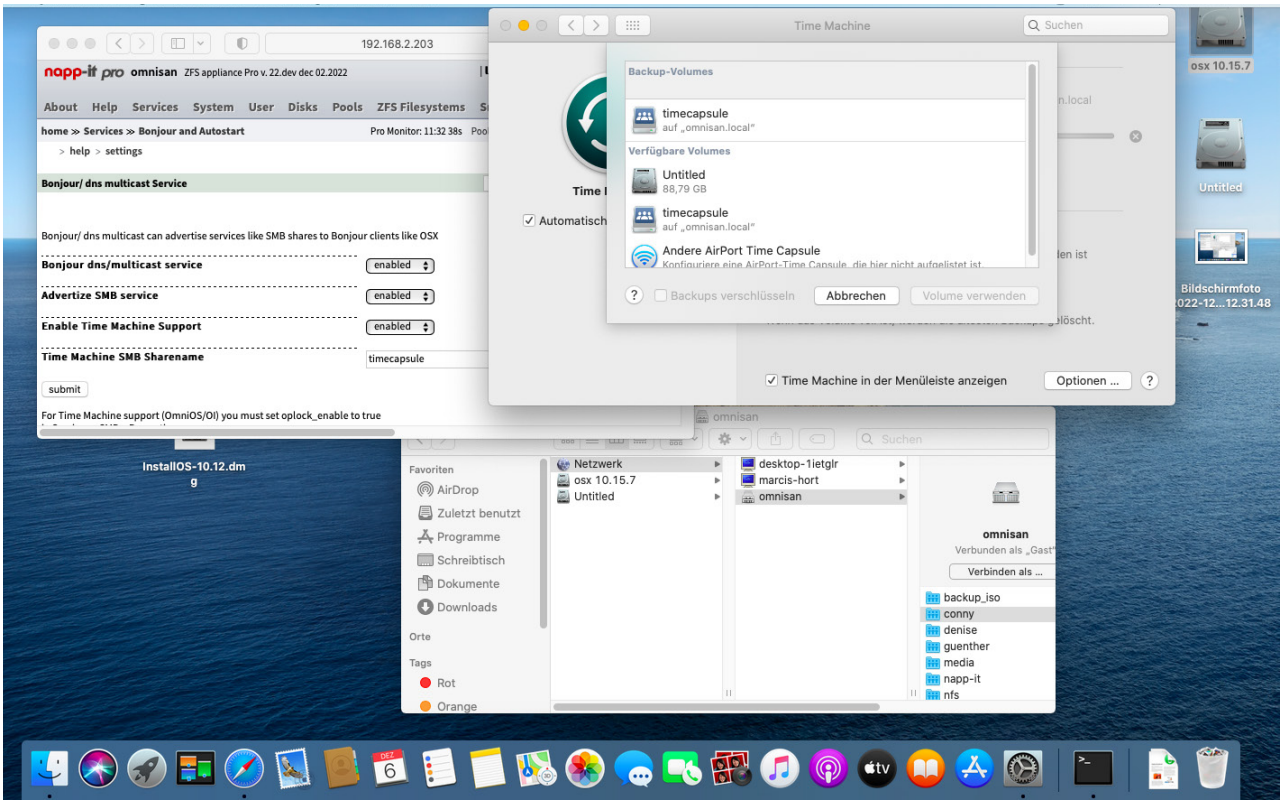
There is another method that also provides Unix uid(gid from your AD server:
https://omnios.org/setup/ad-connect.html

You can also use LDAP
https://omnios.org/info/openldap-client-auth.html

4. OmniOS server for Apple clients

OS X and the Solaris based (OmniOS) multithreaded kernelbased SMB server

OmniOS is a very good and fast SMB server for Macs. It supports SMB from v.1 to v.3.1.1 with Apple extensions, Bonjour/mdns discovery in the local LAN segment and Time Machine via SMB. The OmniOS server is displayed with a nice Xserve icon.



OSX.png

If you use Macs with non Apple servers, you often find annoying incompatibilities as Apple introduces new features or remove others without caring for already available products. While Linux and Windows work well with a wide range of SMB services, OS X often does not. This is why OmniOS added Apple extensions to allow OS X features for Macs like Time Machine. On remaining problems, first google for solutions ex „OSX smb problem". As you will find thousands of hits, reduce optionally for ex to „OS X 10.15 smb problem" (or NFS).

In general it is strongly recommended to use a current OmniOS stable/LTS and an OSX releases under support. If you are on an older OSX, update to newest ex 10.15.7 Catalina (or newer if possible as 10.15 support ended October 2022, often possible for Macs from 2012 or newer https://support.apple.com/en-us/HT210222), as newer OSX versions often have less problems and it may be easier to find help. If your Mac is basically supported but About this Mac > Softwareupdate does not offer an update, download OS X 10.15 installer and start update manually ex from https://support.apple.com/en-us/HT211683

The usual way to SMB connect OmniOS is

Finder: Goto > Connect to server: smb://serverip (this connects SMB2/3)
If Bonjour/mdns is enabled, you find the shares also under network

Common problems and solutions.

On problems with smb://, try cifs
Finder: Goto > Connect to server: cifs://serverip (this connects SMB1)

Other fixes for SMB (NFS) problems on current Macs

Google „OSX SMB problem" and restrict findings to last year.
Check/ask for Apple or OSX at https://illumos.topicbox.com/latest or https://www.illumos.org/issues

Apple SMB extensions
Mostly you are happy about them. On some (locking) problems you can disable

Put the following on a new file inside /etc/system.d/ and reboot
set smbsrv:smb2_aapl_use_file_ids=1

If you ever want to disable completly the smb aapl mode, you can use:
set smbsrv:smb2_aapl_extensions=0

Other sharing options:

AFP
Finder: Goto > Connect to server: afp://serverip (netatalk is available via pkgin but obsolet now, end of live)

iSCSI
You need an initiator software on your Mac (not included in OS). An initiator uses a ZFS volume like a local disk
Google for osx+initiator

NFS
Finder: Goto > Connect to server: nfs://serverip/pool/filesystem (ex nfs://192.168.1.1/tank/data)

NFS problems
NFS (v3) does not offer authentication or authorization like SMB. Use it only on secure networks and prefer SMB. Access restrictions are possible based on fakeable client ip or on client uid (or user nobody). Normally you enable a NFS share with „on" in the sharing dialog. If you instead enter „root=@192.168.1.0/24" you have access from those clients with another uid. This is similar to Unix no_root_sqash setting that maps root to nobody on NFS..

Set nbmand (ZFS property) to on when using a ZFS filesystem over SMB and NFS concurrently. Prefer an ACL like everyone@=modify for all files. For SMB you can additionally restrict access with Share ACL

Timemachine

On OmniOS:
- create a filesystem ex timecapsule with smbshare=on

Menu napp-it Services > Bonjour and Autostart
- enable Bonjour/mdns
- enable support for smb and Time Machine for this share timecapsule

OSX (Timemachine settings)
- select timecapsule as backup target

5. more manuals and infos

5.1 Solaris

https://docs.oracle.com/cd/E23824_01/html/821-1449/smbenvironmentoverview.html
https://docs.oracle.com/en/operating-systems/solaris/oracle-solaris/11.4/manage-smb/managing-smb-file-sharing-and-windows-interoperability-oracle-solaris-11.4.pdf

History: Sun Solaris CIFS Server
http://nfsv4bat.org/Documents/ConnectAThon/2008/cifs-server.pdf

5.2 Illumos/ OpenIndiana/ OmniOS/ older Solaris

https://docs.oracle.com/cd/E36784_01/html/E36832/smbservertasks.html

5.3 more manuals

https://omnios.org/ (click on documentation, upper left)
https://www.napp-it.org/manuals/index_en.html

Topicbox forum, newest features and questions (topicbox search ex for SMB or Apple)
look at illumos-dev, illumos-discuss and omnios-discuss

Illumos issues (bugtracker)

5.4 Forums

https://gitter.im/omniosorg/Lobby (OmniOS Lobby)
https://forums.servethehome.com/index.php?forums/solaris-nexenta-openindiana-and-napp-it.26/
https://hardforum.com/threads/opensolaris-derived-zfs-nas-san-omnios-openindiana-solaris-and-napp-it.1573272/
https://www.hardwareluxx.de/community/threads/zfs-stammtisch.570052/

more napp-it manuals, see https://www.napp-it.org/manuals/index_en.html