

NAME

vfs_zfsacl - ZFS ACL samba module

SYNOPSIS

vfs objects = zfsacl

DESCRIPTION

This VFS module is part of the **samba**(7) suite.

The zfsacl VFS module is the home for all ACL extensions that Samba requires for proper integration with ZFS.

Currently the zfsacl vfs module provides extensions in following areas :

⊕

ACL Interfaces with configurable options for ZFS

NOTE: This module follows the posix-acl behaviour and hence allows permission stealing via chown. Samba might allow at a later point in time, to restrict the chown via this module as such restrictions are the responsibility of the underlying filesystem than of Samba.

This module makes use of the smb.conf parameter **acl map full control = acl map full control**. When set to yes (the default), this parameter will add in the FILE_DELETE_CHILD bit on a returned ACE entry for a file (not a directory) that already contains all file permissions except for FILE_DELETE and FILE_DELETE_CHILD. This can prevent Windows applications that request GENERIC_ALL access from getting ACCESS_DENIED errors when running against a filesystem with NFSv4 compatible ACLs.

ZFS has multiple dataset configuration parameters that determine ACL behavior. Although the nuances of these parameters are outside the scope of this manpage, the "aclmode" and "aclinherit" are of particular importance for samba shares. For datasets that are intended solely as Samba shares, "aclmode = restricted" and "aclinherit = passthrough" provide inheritance behavior most consistent with NTFS ACLs. A "restricted" aclmode prevents chmod() on files that have a non-trivial ACL (one that cannot be expressed as a POSIX mode without loss of information). Consult the relevant ZFS manpages for further information.

This module is stackable.

Since Samba 4.0 all options are per share options.

OPTIONS

nfs4:mode = [simple | special]

Controls substitution of special IDs (OWNER@ and GROUP@) on ZFS. The use of mode simple is recommended. In this mode only non inheriting ACL entries for the file owner and group are mapped to special IDs.

The following MODEs are understood by the module:

⊕

- use OWNER@ and GROUP@ special IDs for non inheriting ACEs only.

⊕

- use OWNER@ and GROUP@ special IDs in ACEs for all file owner and group ACEs.

nfs4:acedup = [dontcare|reject|ignore|merge]

This parameter configures how Samba handles duplicate ACEs encountered in ZFS ACLs. ZFS allows/creates duplicate ACE for different bits for same ID.

Following is the behaviour of Samba for different values :

⊕

(default) - copy the ACEs as they come

⊕

(deprecated) - stop operation and exit with error on ACL set op

⊕

(deprecated) - don't include the second matching ACE

⊕

- bitwise OR the 2 ace.flag fields and 2 ace.mask fields of the 2 duplicate ACEs into 1 ACE

nfs4:chown = [yes|no]

This parameter allows enabling or disabling the chown supported by the underlying filesystem. This parameter should be enabled with care as it might leave your system insecure.

Some filesystems allow chown as a) giving b) stealing. It is the latter that is considered a risk.

Following is the behaviour of Samba for different values :

⊕

- Enable chown if as supported by the under filesystem

⊕

(default) - Disable chown

`zfsacl:denymissingspecial = [yes|no]`

Prevent users from setting an ACL that lacks NFSv4 special entries (owner@, group@, everyone@). ZFS will automatically generate these these entries when calculating the inherited ACL of new files if the ACL of the parent directory lacks an inheriting special entry. This may result in user confusion and unexpected change in permissions of files and directories as the inherited ACL is generated.

⊕

⊕

(default)

`zfsacl:block_special = [yes|no]`

Prevent ZFS from automatically adding NFSv4 special entries (owner@, group@, everyone@). ZFS will automatically generate these these entries when calculating the inherited ACL of new files if the ACL of the parent directory lacks an inheriting special entry. This may result in user confusion and unexpected change in permissions of files and directories as the inherited ACL is generated. Blocking this behavior is achieved by setting an inheriting everyone@ that grants no permissions and not adding the entry to the file's Security Descriptor

⊕

(default)

⊕

`zfsacl:map_dacl_protected = [yes|no]`

If enabled and the ZFS ACL on the underlying filesystem does not contain any inherited access control entires, then set the SEC_DESC_DACL_PROTECTED flag on the Security Descriptor

returned to SMB clients. This ensures correct Windows client behavior when disabling inheritance on directories.

Following is the behaviour of Samba for different values :

⊕

- Enable mapping to SEC_DESC_DACL_PROTECTED

⊕

(default)

EXAMPLES

A ZFS mount can be exported via Samba as follows :

```
[samba_zfs_share]
  vfs objects = zfsacl
  path = /test/zfs_mount
  nfs4: mode = simple
  nfs4: acedup = merge
```

VERSION

This man page is part of version 4.16.11 of the Samba suite.

AUTHOR

The original Samba software and related utilities were created by Andrew Tridgell. Samba is now developed by the Samba Team as an Open Source project similar to the way the Linux kernel is developed.