Lessons learned: ACLs with Samba on NFSv4 (Solaris)

23. April 2009

Nils Goroll

nils.goroll@hamburg.de
About myself

- First contact with Linux approx. 1992
- 1993: First commercial ISP in Braunschweig
- 1999-2008: Technical director at MCS (Sun-minded systems integrator and ISP)
- Since 2008: Freelancer
- First release of UPLEX brand today ;-)

- Age: 34, married, two kids
Lessons learned: ACLs with Samba on NFSv4 (Solaris)

- Simple use case for access control
Simple Access Control requirements case

- Users work on projects
  - One user might be involved in many projects
- One project = one directory tree
- Users need read-only (r/o) or write (r/w) access on project files and directories
- Exceptions should be possible
- Need inheritance for new files
Lessons learned: ACLs with Samba on NFSv4 (Solaris)

• ACL schemes intro:
  • UNIX / POSIX mode bits
  • POSIX (draft) ACLs
  • NFSv4 / ZFS ACLs
UNIX mode bits

- Unix legacy: POSIX mode bits, owner & group
  ```
  haggis:~/Projekte/SambaXP$ ls -alsd .
  4 drwxr-xr-x   2 slink    staff          2 Apr 22 11:44 .
  ```
- Cannot even implement the simple r/o – r/w differenciation
- Rudimentary inheritance (set group-id bit)
- Commonly seen „workarounds“:
  - Access control by group, only r/w access
  - group: r/w access, other: r/o access
  - No access control at all
Lessons learned: ACLs with Samba on NFSv4 (Solaris)

- ACL schemes intro:
  - POSIX (draft) ACLs
    - Usage
    - ACL mask
    - How to deploy for our use case
    - Shortcomings on NFSv3
  - NFSv4 / ZFS ACLs
POSIX ACLs (1)

• POSIX (draft) ACLs:
  • 3 mode bits (rwx) for arbitrary users/groups
  • ACL mask: Restrict permissions for everyone (other than owner)
  • Default permissions: Inheritance
  • Solaris:
    – setfacl(1) / getfacl(1)
    – acl(2) with SETACL / GETACL
POSIX ACLs (2)

elementary

- haggis:/mnt$ getfacl .

  # file: .
  # owner: slink
  # group: staff
  user::rwx
  user:acluser1:r--  #effective:r--
  user:acluser2:rwx  #effective:rwx
  group::r-x    #effective:r-x
  mask:rwx
  other:r-x
POSIX ACLs (3)

ACL mask

- haggis:/mnt$ setfacl -m mask:r-- .
haggis:/mnt$ getfacl .

# file: .
# owner: slink
# group: staff
user::rwx
user:acluser1:r-- #effective:r--
user:acluser2:rwx #effective:r--
group::r-x #effective:r--
mask:r--
other:r-x
POSIX (draft) ACLs for our use case

- Users need read-only (r/o) or write (r/w) access on project files and directories

- Solution 1:
  - Two groups per project
    - r/w group
    - r/o group
  - Users need to have many supplementary groups (note 16 group limit in NFSv2/3 with AUTH_SYS)

- Solution 2:
  - Add/remove users from/to ACLs
POSIX (draft) ACLs on NFSv3

• Shortcomings
  • Impossible to map full Windows ACL semantics to POSIX (draft) ACLs
  • NFS: Only ever got (properly) implemented in Solaris NFS Client/Server (true?)
Lessons learned: ACLs with Samba on NFSv4 (Solaris)

• ACL schemes intro:
  • NFSv4 / ZFS ACLs
    - usage
    - Access checking
    - Summary
Many access controls (edit of ls(1) man page)

```bash
ls -dV /sandbox/dir.1
```

```
drwxr-xr-x+  2 root     root           2 Jan 17 15:09 dir.1
user:marks:r-------------:fd-----:allow
  owner:--------------:-------:deny
  owner:@:rwxp---A-W-Co-:-------:allow
  group@:-w-p-------------:-------:deny
  group@:r-x--------------:-------:allow
  everyone@:-w-p---A-W-Co-:-------:deny
  everyone@:r-x---a-R-c--s:-------:allow
```

```
  (r)ead data +|||        ||||      :|||      |+  (I)nherited
  (w)rite data -+|||        ||||      :|||      +-  (F)ailed access (audit)
  e(x)ecute --+|||        ||||      :|||      ++-  (S)uccess access (audit)
  a(p)pend ---+|||        ||||      :|||      +--  (n)o propagate
  (d)elete ----+|||        ||||      :|||      +---- (i)nherit only
  (D)elete child -----+|||        ||||      :++++    (d)irectory inherit
  read (a)ttrib ------+|||        ||||      :++++    (f)ile inherit
  write (A)ttrib -------+|||        ||||      :++++
  (R)read xattr ----+|||        ||||      :++++
  (W)rite xattr --------+|||        ||||      :++++
  read a(c)l --------+|||        ||||      :++++
  write A(C)L -------+|||        ||||      :++++
  change (o)wner ------------+|||        ||||      :++++
  sync -------------+|||        ||||      :++++
```
NFSv4 / ZFS ACLs (2)

• An ACL consists of many ACEs
• Simple access check: ACEs are checked top to bottom, first match action taken

Check for user **slink** write access:

```
-rw-r--r---+ 1 root   root          0 Nov 24 21:38 ./log.smbd
user: **slink**:--wxp-------------------------:deny
user:slink:rwxpdDaARWc--s:--------I:allow
  group@:-----dDaARWc--s:--------I:allow
  owner@:--x---------------------:deny
  owner@:rw-p---A-W-Co-:--------:allow
  group@:--wxp---------------------:deny
  group@:r------------------------:allow
everyone@:--wxp---A-W-Co-:--------:deny
everyone@:r-----a-R-c--s:--------:allow
```
NFSv4 / ZFS ACLs

Usage

- `chmod(1), ls(1)`
- `acl(2)` with `ACE_SETACL / ACE_GETACL(CNT)`
- Tip: Use edited `ls -V` output to create simple shell-scripts

```bash
chmod A=
user:acluser1:rwxpdDaARWC--s:fdi---I:allow,
user:acluser2:r-x-------------s:------I:allow,
group@:r--------------s:fdi---I:allow,
$*
```
NFSv4 / ZFS ACLs

Access checking

- NFSv4 / ZFS ACLs are similar to Windows ACLs: „Photocopy of the specs“ (Jeremy)
- Samba will do the mapping for ACL display and edit by the CIFS client + some access checks
  - Will remain an area of conflict (see also Jeremy's presentation)
- ZFS: Access checking is done by the O/S.
- NFSv4: Access checking is done on the Server, Client O/S should not implement additional checking
NFSv4 / ZFS ACLs

Summary

• Very powerful & flexible
  • Can implement „arbitrary“ access control schemes (but shouldn't, really)
  • More than we need to implement use case requirements

• Additional complexity -> Script your Policy!

• Vendors are not required to implement the full feature set
  • NFSv4 RFC leaves a lot of room for interpretation (NFSv4.1 draft often also acts as a guide on how to interpret NFSv4 RFC)
Lessons learned: ACLs with Samba on NFSv4 (Solaris)

- Basic Samba Setup on NFSv4 / ZFS ACLs
• ZFS: Limit effects of chmod(2) and create mode bits
  • `pfexec zfs set aclmode=passthrough`<ds>
  • `pfexec zfs set aclinherirt=passthrough`<ds>

• Samba: needs `vfs_zfsacl.c` module, depends on `nfs4_acls.c`. Same config for ZFS and NFSv4:

```
[share]
  acl check permissions = False
  ea support = yes
  store dos attributes = yes
  map readonly = no
  map archive = no
  map system = no
  vfs objects = zfsacl
  nfs4: mode = special
  nfs4: acedup = merge
```
Lessons learned: ACLs with Samba on NFSv4 (Solaris)

• Lesson 1
  • Strange mode bits created by M$ Office
    – Inaccessible files
M$ Office
strange mode bits

• Directory:

```
drwxrwx---+  2 userA group1 10 Nov 24 17:25 .
    owner@:rwxpdDaARWc--s:fd-------:allow
    group@:rwxpdDaARWc--s:fd-------:allow
    group:group2:rwxpdDaARWcCos:fd-------:allow
```

• UserA creates a M$ word file and inherits ACEs:

```
-rwxrwx---+  1 userA group3 79258 Nov 24 17:26 f.xlsx
    owner@:rwxpdDaARWc--s:-------:allow
    group@:rwxpdDaARWc--s:-------:allow
    group:group2:rwxpdDaARWcCos:-------:allow
```

• UserB overwrites the file:

```
----rwx---+  1 userB group3 35067 Nov 24 17:55 f.xlsx
    user:userA:rwxpdDaARWc--s:-------:allow
    group@:rwxpdDaARWc--s:-------:allow
    group:group2:rwxpdDaARWcCos:-------:allow
```
Stranger mode bits

- Samba thinks the file is read-only

  open_file_ntcreate: write access requested for file
test/t.rtf on read only file
[2008/11/24 17:04:45, 3] smbd/error.c:(106)
  error packet at smbd/nttrans.c(805) cmd=162 (SMBntcreateX)
  NT_STATUS_ACCESS_DENIED
Strange mode bits

Solution

- [Link](http://lists.samba.org/archive/samba/2008-November/145094.html)
- [Link](https://bugzilla.samba.org/show_bug.cgi?id=6050)

- map read only = Permissions
- map system = no
- map archive = no
- Patch for `can_delete_file_in_directory()` and `can_access_file()`, not yet integrated
ACL compatibility wrapper

• On ZFS, POSIX draft ACLs are not implemented

haggis:~/Projekte/SambaXP$ getfacl .
File system doesn't support aclent_t style ACL's.
See acl(5) for more information on Solaris ACL support.
ACL compatibility wrapper

- **NFSv4 ACLs** *should* behave the same on an NFS client as on ZFS, but..

  ```
  haggis:~/Projekte/SambaXP$ pfexec share -o rw=localhost $PWD
  haggis:~/Projekte/SambaXP$ pfexec mount localhost:$PWD /mnt
  haggis:~/Projekte/SambaXP$ cd /mnt
  haggis:/mnt$ getfacl .
  # file: .
  # owner: slink
  # group: staff
  user::rwx
  group::r-x #effective:r-x
  mask:rwx
  other:r-x
  ```

- ... the Solaris NFSv4 client has a magic compatibility wrapper
ACL compatibility wrapper breaks Samba

- Wrapper breaks Samba NFSv4 ACL support, because VFS modules are cascaded and if the POSIX module succeeds, the zfsacl module never gets called:

  acl("acl1/acl2/Neuer Ordner", GETACL, 4, 0x085FB088) = 4
  acl("acl1/acl2/Neuer Ordner", SETACL, 4, 0x085FC9D8) = 0

  (These should be ACE_(GET|SET)ACL[CNT] calls for NFSv4 ACLs!)

- Easy solution: Put some dummy functions into the vfs_zfsacl module. Recent Samba Versions contain the fix:  https://bugzilla.samba.org/show_bug.cgi?id=5446
Lessons learned: ACLs with Samba on NFSv4 (Solaris)

• Lesson 3
  • NFSv4 implementation details: Examples for what can go wrong
    - ACL Inheritance broken (fixed)
    - Wrong mode bits (not fixed)
Example bug 1
Inheritance broken

- fd=open(fname,O_RDWR|O_CREAT|O_EXCL,perm) leads to:
  # ls -dV test
  -rw-r------ 1 root root 8 May 6 14:08 test
  owner@:rw-p--aARWcC-s:--------:allow
  owner@:--------o:--------:deny
  group@:r------a-R-c--s:--------:allow
  group@:-wxp---A-W-Co:--------:deny
  everyone@:---------a--c------:allow
  everyone@:----------a--c------:deny

- fd=creat(fname2,perm) leads to: (Inheritance OK!)
  # ls -dV test.creat
  -rwxr-xr-x+ 1 root root 8 May 6 14:08 test.creat
  user:acluser3:rwxp---ARW-Co:--------:allow
  user:acluser1:rwxp---ARW-Co:--------:allow
  user:acluser2:rwxp---ARW-Co:--------:allow
  owner@:rwxp---A-W-Co:--------:allow
  group@:r-x:--------:allow
  everyone@:r-x:--------:allow

- Bugfix release is available!
Example bug 1
Wrong mode

• Create mode is not applied to inherited ACL

  NFS:  Op = 18 (OPEN)
  NFS:  test.creat
  NFS:  Open Type = CREATE
  NFS:  Method = GUARDED
  NFS:  Mode = 0640

  -rwx-------+ 1 root  root  8 Sep 16 2008
  base.creat
    owner@:rwxp--a-R-c--s:--------:allow
    user:acluser1:rwxp--a-R-c--s:--------:allow
    user:acluser2:r-x---a-R-c--s:--------:allow
    everyone@:--------a---c--s:--------:allow

• Not fixed yet
Lessons learned: ACLs with Samba on NFSv4 (Solaris)

• Summary
Some vendors claim, their NFSv4 implementation was already mature

- My experience: **Not true**
- Have found fundamental flaws over the last two years
- Full-featured NFSv4 production installations are still rare – **complex topic**!
- Don't assume that NFSv4 was a fixed, unambiguous standard

My advice: Prototype your installation

- Test exactly the features you need in the final environment
There still are potholes on the NFSv4 ACL road.

But: very interesting technology

- allows proper solutions to fundamental access control requirements
- NFS & Samba for cross-platform ACL support
- NFS4.1 / pNFS is an even more interesting perspective
Lessons learned: ACLs with Samba on NFSv4 (Solaris)

DISCUSSION