Cal Poly Pomona

Groupwork Assignment 2

ZFS and Identity Management

Slava Maslennikov, Henry Ammermann, Jonathan Quiros

39/40

CS499-02

Dr. Rich

March 8, 2016

Contents

1 Functionality

ZFS in Cal Poly Pomona's identity management platform provides numerous vital functions, the most obvious being storage for students, staff and faculty. Every currently affiliated person on campus has their own space on ZFS, accessible from machines on campus as well as their own computers. Although ZFS is natively a *nix-born file system, first introduced with OpenSolaris in 2005, it is made available to campus users through Samba in order to be accessible from all operating systems used on campus: MS Windows, OS X, and Linux. When accessing campus *nix servers, however, the share is mounted as $HOME (the home directory) using NFS.

ZFS on campus also has a ZFSWeb component. Every currently affiliated human on campus has a directory made to be world-accessible through HTTP. Its outside address is in the form of http://www.cpp.edu/~[uid]/, where uid is user's user ID, e.g. carich for Dr. Rich. Up until recently, this was a non-optional feature, however now it must be enabled through the identity management front-end on CalPoly's website.

Furthermore, ZFS as a file system has numerous important features such as access list support and high limit volume and file size (256 zebibytes and 16 exbibytes, respectively). Those are important for a file system used for

campus-wide user shares. Access lists allow picking and choosing access to files and directories on a per-person or a per-group level, while the high volume size limit means we won't run into the software space limitation any time soon.

2 Identity Information

The ZFS module has several variables that must be kept sane and up to date. They include the owner's user or group ID, the name of the share, the quota, and the snap quota. The share is tied to the user/group ID, i.e. the unique key. If it changes, it is automatically changed in ZFS by the identity management suite. The quota and snap quota aren't automatically changed, but they can be manually.

The ZFS server stores just that: user/group ID, share name, quota, snap quota. Authentication when mounting the share is done either through Kerberos for NFS mounting or on Active Directory for Samba.

3 Invocation

ZFS is a file system and a logical volume manager, so many of the other supporting modules used for identity management will use ZFS to modify the user' or group's file system. In this case, the ZFS module does not have a direct API, instead, Identity Management uses REST calls to perform certain functions. The calls are pre-coded in a separate subroutine in the code, which, in turn, get triggered by other subroutines.

The access is controlled by ZFS itself. The Identity Management suite has its own account that it uses for making system changes for which it logs into ZFS over REST before making any other REST calls.

4 Supporting Modules

Inside of Identity Management, ZFS is managed by the Identity::ZFS and Identity::REST::ZFS modules. Identity::ZFS has a dependency on Identity, but this isn't used for anything other than logging. The two ZFS modules have several outside dependencies (Unix::Syslog, POSIX, etc), though none of them are ZFS-specific. Instead, all the manipulation of the underlying ZFS filesystem is done by calling the ZFS command-line tools directly from

inside the Identity::REST::ZFS module.

Furthermore, POSIX is specifically used within ZFS to send a SIGALRM signal to check for timeout on the connection. LWP, the World-Wide Web library for Perl, is used for the same purpose. Lastly, the external module XML-Simple is used to parse the XML data.

5 Underlying Services

ZFS provides neither authentication nor authorization, but it does verify file permissions and ACLs based on the user. Every currently affiliated person at Cal Poly Pomona has a share on ZFS. It's not optional whether to have the share or not, but how many people actually utilize it is questionable. The main purpose of ZFS is storage, but it can also be used as a web hosting service as outlined in section 1.

Though ZFS provides neither authentication nor authorization, file permissions can be set up by using either standard unix file ownership and permissions, or Access Lists. These can be used to restrict file access (and separately, writing) to just the file's owner, the file's group, the world, or a more complicated combination of these.

6 Service Connection Management

The ZFS module uses REST calls to modify object variables. The password for IDMGMT user is stored in '/etc/security/secrets/zfs-idmgmt' to better conceal it without showing it in plaintext in the code. There are four ZFS servers on campus, an array of whose hostnames is stored in code. All REST methods are pre-defined in a subroutine, so when one is called, the code is continuously reused.