

Deploying hardened internet-accessible systems with read-only AFS volumes

Troy Benjegerdes
Scalable Computing Lab
DOE Ames Laboratory
<troy@scl.ameslab.gov>

AFS as root filesystem

- This gets easier each year
- Debian etch mkinitramfs script provide a really nice way to manage this
- Works well on cluster nodes, when the afs server is on the same network

What do we do?

- High performance computing research
 - Middleware research (schedulers, libraries, etc)
 - Network performance evaluation (NetPIPE)
 - HPC Parallel filesystems (PVFS on InfiniBand)
- Often means changing 1 line in a config file on N nodes
- Single system image? Doesn't scale.
- Replicating node images?
 - Why check 9GB for a 1k change
 - Whoops, node 25 was down last time, its out of sync

Some background..

- Started with local, moved to NFS
- NFS works, except when NFS server hiccups
 - Images are annoying to manage.. must log into the 1 node allowed to mount the filesystem r/w
- Try AFS.. 10 IBM power5 compute nodes, 6 Opteron PVFS servers
- Issues with AFS-root
 - Upgrading libc... Just reboot.
 - Only AFS admins can change UID's, or setuid bit on files

Allow volume owners to change file UID

```
--- a/src/viced/afsfileprocs.c    Fri Dec 09 16:48:36 2005 -0500
+++ b/src/viced/afsfileprocs.c    Fri Dec 09 19:06:51 2005 -0500
@@ -829,7 +829,7 @@ Check_PermissionRights(Vnode * targetptr
     if (CHOWN(InStatus, targetptr) || CHGRP(InStatus, targetptr)) {
     if (readonlyServer)
         return (VREADONLY);
-     else if (VanillaUser(client))
+     else if (VanillaUser(client) && !VolumeOwner(client, targetptr))
         return (EPERM); /* Was EACCES */
     else
         osi_audit(PrivilegeEvent, 0, AUD_ID,
@@ -855,7 +855,8 @@ Check_PermissionRights(Vnode * targetptr
     || CHGRP(InStatus, targetptr)) {
     if (readonlyServer)
         return (VREADONLY);
-     else if (VanillaUser(client))
+     /* Allow volume owner to chown */
+     else if (VanillaUser(client) && !VolumeOwner(client, targetptr))
         return (EPERM); /* Was EACCES */
     else
         osi_audit(PrivilegeEvent, 0, AUD_ID,
```

Debian initramfs-tools package

- <http://source.scl.ameslab.gov/hg/mkinitramfs-openafs>
- `/etc/mkinitramfs/hooks/openafs`
 - Copy `afsd`, config files, etc to `initrd`
- `/etc/mkinitramfs/scripts/openafs`
 - Run in the `initrd/initramfs` to start `afsd` and mount `/afs` in the right place
- `/etc/mkinitramfs/scripts/openafs-premount/openafs`
 - Optionally mount a disk cache partition (otherwise use `memcache`)

Cluster deployment

- 10 IBM Power5 compute nodes
 - 32 bit Debian etch base system, with support for 64 bit compilers and libc
- 6 Opteron storage servers with 2 Areca raid controllers with 8 disks each
 - Debian etch amd64
 - PVFS filesystem using InfiniBand interconnect
 - Network booting storage servers
 - Try out read-only replicated volume for availability
 - 'make install' of pvfs plus vos release takes < 30 seconds to update all 6 storage servers

FISMA attacks!

- Federal Information Security Management act of 2002
 - DOE management is interested in labs doing **something**
- Need to reduce internet-visible system exposure, and consolidate systems
- ssh gateway(s) with AFS and NFS homedirs
- Need an alternative to standard hardening approaches

AFS to the rescue

- We've got afs booted cluster nodes, why not the front-end gateway
- No disks, no place for an attacker to hide
- Mimimize compliance induced downtime
 - If an internet accessible gateway is compromised, power cycle it, or quarantine it and boot another node to take over
 - AFS as a service virtualization tool
- 1 public ssh gateway in production

Things that aren't so great

- SetUID
- AFS ACL's and unix permissions
 - auditing for unix permissions doesn't help a lot
- Root filesystem needs to be readable without tokens
 - LDAP & Kerberos for user authentication makes this less of a problem
- Explaining this to managers

What could be (or is) great

- All modifications to filesystem require a kerberos-authenticated user
- System administration is performed on the r/w volume, not on internet accessible system
- r/w volume server can be completely isolated via firewalls from internet accessible host
- AFS fileserver provides external logging and auditing
- r/o volume servers can have AFS-aware IDS systems to **externally** audit all filesystem access attempts

What could be done to make this better

- Afsrootd – modified afsd, started from initrd/initramfs instead of afsd
 - Put sensitive files in tmpfs.. passwd, keytab
 - Disable SUID completely, use afsrootd to grant root privileges
 - OR put suid files in tmpfs
- Make this concept FISMA buzzword compliant
 - Map the concepts to FISMA-speak
 - Convince pointy-haired bosses
 - \$profit\$

Credits

- DOE MICS
- OpenAFS developers
- Debian & Ubuntu initramfs maintainers

Questions?