

Re: move audit/privilage check into VFS

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/arch/2007-04/msg00072.html>

- *From:* Robert Watson <rwatson@xxxxxxxxxxx>
 - *Date:* Tue, 24 Apr 2007 09:11:39 +0100 (BST)
-

On Mon, 23 Apr 2007, Howard Su wrote:

On 4/23/07, Robert Watson <rwatson@xxxxxxxxxxx> wrote:

Pawel and I have talked about this a bit in the past — `vaccess(9)` and `vaccess_acl_posix1e(9)` were really the first step in abstracting file system access control decisions, and aren't a bad step — they certainly cover a lot of the previously plentifuly replicated cases (countless `foo_access()` VOP implementations). However, I think we should be restrained and do a bit of experimentation — sometimes as much work could be done bundling up the common arguments to deliver them to a central access check as is done in having the access check appear in the calling code itself. Can we refine `VOP_ACCESS()` a bit further to get what we need, or do we need new common functions?

In FS dependent code, we don't only call `VOP_ACCESS`, but also check some flags like `ISUID`, `ISGID`, `NOUNLINK`, `APPEND`, etc. This sort of stuffs are so easy to regerssion when I work on `tmpfs` and it should be almost same code in all the FS. However VFS don't have this sort of information in `vnode` structure. Is this can be added?

I don't think I would add these to the `vnode` — remember that, for distributed file systems, these fields may change asynchronously, and that for at least one critical distributed file system (NFS) there is no asynchronous notification facility from the server. I like the `vaccess()` approach, in which the file system is responsible for determining the values of any relevant fields, and passing them into what is effectively a library routine that performs the check. This avoids having these access control checks perform VOP's, which has significant overhead, and allows the file system to optimize storage/retrieval of these volatile fields.

Robert N M Watson
Computer Laboratory
University of Cambridge

freebsd-arch@xxxxxxxxxxx mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-arch>

To unsubscribe, send any mail to "freebsd-arch-unsubscribe@xxxxxxxxxxx"