Cybersecurity

Subject: POSIX ACL

Here, we are going to play with access control mechanisms of the POSIX (Portable Operating System Interface) standard, prevalent in the whole Unix family of operating systems.

1. POSIX Access Control Lists

POSIX ACL (*Access Control Lists*) was introduced to extend the standard Unix permission mechanism that controls access to a file (or directory) for the owner (u), owning group (g), and other users (o), hereafter called *basic permissions*. The extension allows defining authorization entries (ACE, *Access Control Entry*) for any individual user and/or group. However, POSIX ACLs are still limited to `read`, `write` and `execute` permissions.

1.1 POSIX authorization

Permissions customarily used in Unix/Linux systems are limited to the following:

- `read`—reading a file / listing directory contents
- `write`—saving a file / directory *i*-node modification
- `execute`—executing a file / directory entry

1.2 Access control algorithm

The POSIX standard offers the ability to mask permissions using the *mask* field. The *effective* access permissions are the bit product of the assigned permissions (ACE) and the mask. The access control algorithm performs the following steps (until the first match):

1. if the user owns a file—use the owner's ACE permissions (user::),
2. if the user is explicitly listed on the ACL—use the user's *effective* permissions,
3. if one of the user's groups is the owning group—use the group's *effective* permissions (group::),
4. if the user's group is explicitly listed on the ACL—apply that group's *effective* permissions
   (if the matched group does not have sufficient effective rights—access is denied),
5. eventually, the ACE of other users (other::) determine the access rights.

To preserve compatibility, the user::, group:: and other:: rights from the ACL are mapped to the *basic permissions* u, g and o presented with the standard ls command. However, when the mask is set, the rights for the owning group (g) listed with ls correspond to the mask value, not the group:: entry.

1.3 POSIX ACL rights management

Linux has two commands that operate on ACLs, one—getfacl—is for reading extended permissions, the other—setfacl—for setting them. The getfacl command shows extended permissions for files and directories:

```
% ls -l
 -rw-r--r-- 1 user group 1000 2004-10-01 09:00 file.txt

% getfacl file.txt
# file: file.txt
# owner: user
# group: group
user::rw-
group::r--
other::r--

% getfacl file.txt --omit-header
user::rw-
group::r--
other::r--
```

The setfacl command allows to change, add, or remove permissions from the ACL.

Adding permissions:

```
% setfacl -m user:john:rwx plik.txt
% getfacl file.txt --omit-header
user::rw-
user: john:rwx
group::r--
mask::rwx
other::r--
```

Changing permissions:

```
% setfacl -m u: john:r file.txt
% getfacl file.txt --omit-header
user::rw-
user: john:r
group::r--
mask::r--
other::r--
```

Removing permissions:

```
% setfacl -x u: john file.txt
% getfacl file.txt --omit-header
user::rw-
group::r--
mask::r--
other::r--
```

A simple pipe can be used to copy an ACL from one file to another:

```
% getfacl file1.txt | setfacl -M- file2.txt
```

1.3.1 Default permissions

Default permissions are applied only to directories and allow to automatically grant extended permissions to newly created objects in a directory that has been granted default permissions. The x permission is treated in a different way and, although it may be set in the default permissions, it is not effectively granted to newly created regular files.

Adding default permissions:

```
% setfacl -d -m group:students:wx katalog
% getfacl katalog --omit-header
user::rwx
group::r-x
other::r-x
default:user::rwx
default:group::r-x
default:group:students:-wx
default:mask::rwx
default:other::r-x
```

Modification and removal of default permissions is done in the same way.

Copying extended permissions to default permissions can be performed as follows:

```
% getfacl --access directory | setfacl -d -M- directory
```

1.3.2 Mask

ACL rights explicitly granted to users (except the owner) and groups are constrained by the mask (the mask specifies the maximum effective permissions). You can set the mask value using the following command:

```
% setfacl -m mask::rwx file.txt
```

For example, blocking the write permission using a mask is achieved with the command:

```
% setfacl -m m::rx file.txt
```

Using the `setfacl` command (as well as `chmod` on an object with extended permissions) automatically changes the mask value when manipulating permissions for explicitly signed users, default group, and explicitly signed groups. To avoid automatic mask recalculation, `setfacl -n` option needs to be used.

1.3.3 ACL removal

Extended ACL permissions (`-b` option) and default permissions (`-k`) can be removed entirely.