Each process has three user IDs: the real user ID (realuid, or ruid), the effective user ID (effective uid, or euid),and the saved user ID (saved uid, or suid). The real uididentifies the owner of the process, the effective uid isused in most access control decisions, and the saved uid stores a previous user ID so that it can be restored later.

When a process is created by fork, it inherits the three user IDs from its parent process. When a process executes a new file by exec it inherits the uid, euid from its parent process but get suid from parent's euid .unless the SUID bit of the new file is set, in which case the effective uid and saved uid are assigned the user ID of the owner of the new file.

seteuid() It sets the effective uid while leaving the real uid and saved uid unchanged. However, when the current effective uid is not zero, only allow the parameter new euid only to be equal to any of the current three user IDs. when the current effective uid is zero, the new euid can be any value u want.

setuid() It sets all of the three IDs or only for euid. If the current euid is not zero, only allow the parameter newuid to be equal to either the real uid or saved uid, and the current ruid, suid are not changed. If the effective uid is zero, a successful setuid(newuid) call sets all three user IDs to newuid.

setreuid() It modifies the real uid and effective uid, and the saved uid is set as euid. When the current euid is not zero, allow the parameter neweuid only to bee qual to any of the three current user IDs. The newrudi must be equal to current euid or ruid. If the current euid is zero, the new euid and new ruid can be any values u want.

setresuid() system call to be allowed, . If the current euid is zero, the new suid, new euid and new ruid can be any values u want. When the current euid is not zero, each of the three parameters must be equal to one of the current three user IDs of the process.

Example：

1. Drop privilege temporarily

The program is running as a Setuid program.

uid\_t ruid, euid;

void drop\_privs()

{

if (seteuid(ruid) < 0)

fatal("drop\_privs");

}

void regain\_privs()

{

if (seteuid(euid) < 0)

fatal("regain\_privs");

}

int main(int argc, char \*\*argv)

{

ruid = getuid();

euid = geteuid();

drop\_privs();

...

regain\_privs();

do\_privileged\_stuff();

drop\_privs();

...

}

1. Drop privilege permanently

The program is running as a Setuid program.

uid\_t ruid, euid;

void drop\_privs\_permanently()

{

if (setresuid(ruid，ruid, ruid) < 0)

fatal("drop\_privs");

}

int main(int argc, char \*\*argv)

{

ruid = getuid();

euid = geteuid();

do\_privileged\_stuff();

drop\_privs\_permanently();

...

}