Setuid bit diğer kullanıcıların, dosyalarınıza erişmek ve dosyalarınızı yazmak için izini olan sınırlı bir yoldur.Mesela, zannedelim ki ben log1.c adında bir logging programı yazmak istiyorum.Şimdi ben Log file Lectureda diğerlerine bu programı kullanmaları için izin verdiğimi söyledim ve log file ı kurmak zorundayım.Bu durumda **/blugreen/homes/plank/cs360/notes/Setuid/log_file** herkes tarafında yazılabilir olmalıdır.

```
UNIX> chmod 0666 /blugreen/homes/plank/cs360/notes/Setuid/log_file
```

```
Bununla birlikte herhangi biri tarafından değiştirilebilir ve güvensiz
olduğu için kesecektir.Yani herkese güvenmediğiniz ortamda iyi bir çözüm
değildir.
```

```
Bu işlem için Unix i destekleyen şeylerden biri de setuid bittir.Dosya
sahibi, çalıştırılabilir olan programın kullanıcı kimliğini
ayarlamalıdır.Bitin Is-ı komutuna("stat" modun bir parçası) set edilip
edilmediğini görebilirsiniz.
```

```
UNIX> cd /blugreen/homes/plank/cs360/notes/Setuid
UNIX> ls -l log1 log2
-rwxr-xr-x  1 plank       24576 Sep 30 09:46 log1
-rwsr-xr-x  1 plank       24576 Sep 30 09:46 log2
UNIX>
```

Log1 , log1.c olarak derlendi ve Log2 tarafından

```
UNIX> cp log1 log2
UNIX> chmod 04755 log2
```

oluşturuldu.

Bu setuid bitini set eder.Farkedeceksiniz ki **log_file** , **chmod()**'d 0644 olduğundan , hesabınızdan çağırdığınızda log1 başarısız olur ama log2 başarılı olacaktır.

Bu işi yapmak için makine içine "vertex" oturum açmanız gerekir ki bunun için ilk olarak ssh vertex yapılır.

```
UNIX> echo "$USER"
booth
UNIX> cd /blugreen/homes/plank/cs360/notes/Setuid
UNIX> cat log_file
plank      Mon Feb 28 11:28:58 1994
booth      Mon Feb 28 11:37:24 1994
elmore     Tue Mar  8 09:45:19 1994
hamner     Tue Mar  8 21:14:06 1994
UNIX> log1
Can't write log file /blugreen/homes/plank/cs360/notes/Setuid/log_file
UNIX> date
Fri Sep 30 10:13:04 EDT 1994
UNIX> log2
UNIX> cat log_file
plank      Mon Feb 28 11:28:58 1994
booth      Mon Feb 28 11:37:24 1994
elmore     Tue Mar  8 09:45:19 1994
hamner     Tue Mar  8 21:14:06 1994
booth      Fri Sep 30 10:13:10 1994
UNIX>
```

Yani bu diğer kullanıcıların dosyalarınıza kısıtlı bir şekilde karışmaması için size izin verir.. Eğer fazla kullanıcıya erişim verirseniz setuid bit kullanırken çok dikkatli olmanız gerekir. Örneğin **log3.c** bakalım:

```
#include < stdio.h >

main(int argc, char **argv)
{
  char s[1000];
  FILE *f;
  int i;

  if (argc == 2) {
    sprintf(s, "/blugreen/homes/plank/cs360/notes/Setuid/%s", argv[1]);

    f = fopen(s, "w");
    if (f == NULL) {
      perror(s);
      exit(1);
    }
    i = 1;
    while(i > 0) {
      i = fread(s, sizeof(char), 1000, stdin);
      if (i > 0) {
        fwrite(s, sizeof(char), i, f);
      }
    }
    fclose(f);
  }
}
```

Bu dosya bir kullanıcının **/blugreen/homes/plank/cs360/notes/Setuid** dizinini oluşturmasını ve

Standart giriş içeriğini ayarlamanızı sağlar. Kullanıcı dosya adını belirler.Ben bu dasyanın setuid bitini set ederken kötü huylu bir kullanıcı "**log3 ../../../.cshrc**" tarafından çağrı yaparak benim **.cshrc** dosyamı silebilir. Sonra o kişi istediği her şeyi bu dosyadan değiştirebilir.Böylece iyi huylu bir program gibi görünür ama gerçekte büyük bir güvenlik zaafiyeti vardır.Bu yüzden program yazarken setuid bit kullanımında çok dikkatli olmalısın.

Bizim sistemimizde , setuid bit sadece çalıştırılabilir dosyanın olduğu dosya sisteminin bulunduğu makine üzerine açtıysanız çalışır. Bu yüzden log2 programını kullanmak için "/blugreen/homes/plank/..." dizinini içeren makine gibi , makine içinde “vertex” oturum açmanız gerekir. Bunun nedeni güvenliktir.Benim dosya sistemim uzaktan etkideyken setuid bit söz konusu olduğunda diğer makineler güvenilir değildirler.