

BLGM 455 Bilgisayar Sistemleri ve Ağ Güvenliği

Lab1- Linux'ta Erişim Kontrolü

Samed Reyhanlı tarafından Türkçe'ye çevrilmiştir.

İçindekiler

Linux'ta Erişim Kontrolü	1
1. Görev.....	3
2. Fedora İşlem Sistemi	3
2.1 Bash Terminali.	3
2.2 Bazı Önemli ve Faydalı Komutların Listesi	3
3. Giriş kontrolüne giriş.....	5
4. Unix dosya izinleri ve inode'lar.....	5
Örnek 1	5
Örnek 2.....	6
Örnek 3.....	7
Örnek 4.....	7
Örnek 5.....	7
5. İzinlerin sekizli gösterimi	7
Örnek 6.....	8
6. Linux sistemindeki dosya izinlerini değiştirmek, kullanıcı, grupelemek, dosya sahibini ve grubunu değiştirmek	8
Örnek 7.....	9
Örnek 8.....	9
Örnek 9.....	10
Örnek 10.....	11
Örnek 11.....	11
Örnek 12.....	12
Örnek 14.....	12

Örnek 15	12
Örnek 16	13
Örnek 17	13
Örnek 18	13
Örnek 19	13
Örnek 20	14
Örnek 21	14
Örnek 22	14
Örnek 23	15
Örnek 24	15
Örnek 25	15
Çalışma 1	15
7. umask Komutu	15
Örnek 26	16
Çalışma 2	16
8. UID'yi(SUID)ayarlama	16
Örnek 27	16
Örnek 28	17
Örnek 29	17
Örnek 30	18
9. C' de SUID programı yazmak	18
Örnek 31	18
Örnek 32	19
Örnek 33	20
Örnek 34	20
Örnek 35	21
Çalışma 3	21
Çalışma 4	21
Çalışma 5	21
Çalışma 6	21
10. Linux'ta Genişletilmiş ACL'ler	22
Örnek 36	22

11. Sonuç.....	22
Kaynakça.....	23

Laboratuvar materyalimiz [1] 'e dayanmaktadır. Bölüm 1 görevi açıklar. Bölüm 2, 3 en önemli Linux komutlarını ve erişim kontrol kavramlarını kısaca tanıtmaktadır. Laboratuvarlarda Fedora Linux tabanlı işletim sistemi ile çalışacaksınız. Bölüm 4-10, örnekler (ekran görüntülerinde alex @ lenovo ile birlikte Kali Linux'ta) ve Fedora (ekran görüntülerinde linuxlab @ asus ile) işletim sistemlerinde hazırlanmış) ve tartışmaları içerir. Bölüm 11 ise labın sonucunu içeriyor.

1. Görev

Laboratuvar materyalinde gösterilen Örnek 1-36'yı (ekran görüntüleri) tekrarlayarak kendi örneklerinizi oluşturun (alex veya linuxlab yerine kendi kullanıcı adınızı kullanın, örn. "Chmod 701 / home / kullanıcı_ adınız /"). Sizin tarafınızdan çözülmesi beklenen 1-6 arası Tartışmalar vardır. Laboratuvar gruplar halinde yapılacaktır. Her grup, 1'den 36'ya olan örnekleri, 1'den 6'ya olan tartışmaların çözümlerini, açıklamalarını ve ekran görüntülerini içeren rapor. **Raporlama için son dersin web sayfasında duyurulacaktır. Lab süresince Örneklerinizi ve çözümlerinizi çalıştıracak ve bunları Laboratuvar Asistanı'nın sorularını yanıtlayarak açıklayacaksınız.**

2. Fedora İşletim Sistemi

Fedora, topluluk destekli Fedora Projesi tarafından geliştirilen bir Linux dağıtımdır. Açık kaynaklıdır ve ücretsizdir. Fedora işletim sistemi, kullanıcı ile işbirliği içinde geliştirildiği için kullanıcı dostudur.

2.1 Bash Terminali

Deneylerimiz için sık sık bash terminalini kullanacağız. Bu nedenle, bash terminalini açtığımızda ne göreceğimizi öğrenelim.

```
[linuxlab@asus ~]$
```

Terminali açtığımızda yukarıdaki gibi bir görüntü görüyoruz. "Linuxlab" kullanıcı adımızdır, "@" işaretinden sonra gelen "asus", kullandığımız makinenin adıdır ve "~" kullanıcının ev dizini anlamına gelir.

```
[root@asus linuxlab]#
```

Aynı şekilde, "root" bir kullanıcı adıdır (sistemde tüm yetkilere sahiptir), "asus" makine adıdır ve "#" yönetici (root) olarak oturum açtığımız anlamına gelir.

2.2 Bazı Önemli ve Yararlı Komutların Listesi

- Fedora, dnf paket yöneticisini kullanır. Yükleme istiyorsanız herhangi bir açık kaynak programı veya aracı, dnf kullanmanız gerekir. Örneğin, metinlerinizi veya program kodlarınızı düzenlemek için bir metin düzenleyici kurmanız gerekir. Terminalde "dnf install kwrite" komutu kullanılarak bir metin editörü "KWrite" kurulabilir.

```
[linuxlab@asus ~]$ sudo dnf install kwrite
[sudo] password for linuxlab:
```

- Sudo komutu - Sudo komutu, Fedora sisteminizi yönetmeyi kolaylaştırır. Fedora'daki bazı komutlar yalnızca ayrıcalıklı bir kullanıcı veya yönetici tarafından çalıştırılmayı bekler. Sudo komutu, bir komutu root olarak bilinen yönetici gibi çalıştırmanıza izin verir. Örneğin, "sudo chmod 666somefile".

```
[linuxlab@asus ~]$ sudo chmod 666 somefile
[linuxlab@asus ~]$
```

- Su komutu - kullanıcıyı değiştirir. Örneğin, "su öğrenci" komutu mevcut kullanıcı ile öğrenci kullanıcı arasında geçiş yapmak için kullanılabilir.

```
[linuxlab@asus ~]$ su student
Password:
[student@asus linuxlab]$
```

- Ls komutu - dosya ve dizinlerin izin içeriğini listeler. Örneğin, "ls /home/linuxlab" komutu, /home/linuxlab dizininin içeriği terminal penceresinde gösterecektir.

```
[linuxlab@asus ~]$ ls /home/linuxlab
Desktop  Downloads  Pictures  somefile  Videos
Documents Music      Public    Templates
[linuxlab@asus ~]$
```

- cd komutu - mevcut dizini değiştirir. Örneğin, "cd Desktop" komutu mevcut dizini / Desktop dizini olarak değiştirir. (aynı şekilde "cd.." komutu da bir önceki dizine dönmek için kullanılabilir.)

```
[linuxlab@asus ~]$ cd Desktop
[linuxlab@asus Desktop]$ cd ..
[linuxlab@asus ~]$
```

- mkdir - Bu komut yeni bir dizin oluşturur. "MyLabWorks" adıyla yeni bir dizin oluşturmak için örnek kullanım: "mkdir myLabWorks".

```
[linuxlab@asus ~]$ mkdir myLabWorks
[linuxlab@asus ~]$ ls
Desktop  Downloads  myLabWorks  Public  Templates
Documents Music      Pictures    somefile  Videos
[linuxlab@asus ~]$
```

- rm komutu - bu komut, bazı dosya veya dizinleri silmek için kullanılabilir. Örneğin, "test.txt" dosyasını "rm test.txt" ile silebiliriz.

```
[linuxlab@asus ~]$ ls
Desktop  Downloads  myLabWorks  Public  Templates  Videos
Documents Music      Pictures     somefile test.txt
[linuxlab@asus ~]$ rm test.txt
[linuxlab@asus ~]$ ls
Desktop  Downloads  myLabWorks  Public  Templates
Documents Music      Pictures     somefile  Videos
[linuxlab@asus ~]$
```

3. Erişim kontrolüne giriş

Erişim kontrolü, hangi eylemlere izin verildiğini belirleyerek ve uygulayarak yetkilendirmeyi zorlar. Bazı terminoloji: özne, bir kullanıcı veya program gibi eylemler gerçekleştiren aktif bir varlıktır ve bir nesne, bir dosya veya ağ kaynağı gibi erişilebilen (genellikle pasif) kaynaktır. Erişim kontrolü, bir güvenlik politikası uygulayarak, hangi eylemlere izin verilip verilmediğini sınırlayarak öznelerin nesnelere erişimine aracılık eder. Politika, bir dizi kural olarak resmi veya gayri resmi olarak nelere izin verildiğini ifade eder. Erişim kontrol mekanizması, bir politikayı uygulayan kod veya şeydir. Erişim kontrol modeli, bir politika veya politika türleri hakkında temsil ve muhakeme etme yoludur.

4. Unix dosya izinleri ve inode'lar

Geleneksel Unix güvenlik modeli, kullanıcıların "sahip oldukları" kaynaklara kimin erişebileceğini yapılandırmasına olanak tanıyan isteğe bağlı erişim denetimi (discretionary access control - DAC) modeline dayanmaktadır. Her kullanıcı, oluşturdukları dosyalara hangi kullanıcıların erişebileceğini kontrol edebilir. Bu, kullanıcıların bir sistem yöneticisini dahil etmeden izinler vermesini sağlar. Bu, geleneksel olarak Windows ve Unix gibi çoğu tüketici işletim sisteminde yerleşik olarak bulunan güvenlik türüdür. Unix dosya izinleri, erişim kontrol listesinin (access control list - ACL) kısaltılmış haliyle ACL, kullanır. ACL, her öznenin bir listesini ve her bir nesneye nasıl erişebileceğini içerir (Windows dosya erişimini bu şekilde yönetir). Örneğin, bir dosya şu ACL'ye sahip olabilir: "Joe okuyabilir, Frank yazabilir, Alice okuyabilir ve Eve okuyabilir". Unix, yalnızca aşağıdaki üç tür özne için kurallar tanımlayarak izinleri basitleştirir:

- Dosyanın sahibi olan kullanıcı (u)
- Dosyanın grubu (g)
- Diğer kullanıcılar (o)

ls komutu dosyaların izinlerini görüntülemek için kullanılır, -l parametresi ayrıntılı çıktı sağlar,

örneğin, ls -l / bin / ls, ls komutunun izinlerini görüntüler

örnek 1

```
alex@Lenovo:~$ ls -l /bin/ls
-rwxr-xr-x 1 root root 134792 Oct 2 2017 /bin/ls
alex@Lenovo:~$
```

File path

Owner can read, write, execute	Group can read and execute	Others can read and execute	The file has this many names (hard links)	File owner is root	File group is root	File size (bvt)	Last modified
--------------------------------	----------------------------	-----------------------------	---	--------------------	--------------------	-----------------	---------------

The meaning for a regular file (as is the case for /bin/ls, the first symbol is – (dash)):

- r: Dosyanın içeriğini okuyun
- w: Dosyanın içeriğini değiştirin
- x: Dosyayı bir işlem olarak yürüt (İlk birkaç bayt, bunun ne tür bir yürütülebilir dosya olduğunu (bir program veya komut dosyası mı olduğunu) açıklar)

Bir dizin için (ilk sembol d'dir):

- r: Dizinde hangi dosyaların olduğuna bakma
- w: Dizine dosya ekleme, yeniden adlandırma veya silme
- x: dosya sahiplerini ve boyutlarını görüntüleme ('stat') dizine girebilme (cd) ve dizindeki dosyalara erişebilme
- t (x yerine), AKA "Sticky bit, yapışkan bit": kullanıcıların sahip olmadıkları dosyaları silmelerini veya yeniden adlandırmalarını engeller.

Her dosyanın izinleri dosyanın inode'unda saklanır. Bir inode, Unix dosya sistemlerindeki bir dosyayı tanımlayan bir veri yapısıdır. Bir inode, bir inode numarası içerir ve Unix dosya izinlerini ve dosya için erişim zamanlarını içeren özniteliklerle birlikte dosyanın disk üzerindeki konumunu tanımlar. Bu dosya için inode numarasını görüntüleyin:

```
ls -li / bin / ls
```

Örnek 2

```
alex@Lenovo:~$ ls -li /bin/ls
1970324837335149 /bin/ls
alex@Lenovo:~$
```

Dosyalardaki bağlantılar için dosya türü, l de olabilir (bağlantıların yardımıyla bir ve aynı verilere, sabit bağlantılar için bir ve aynı inode'a bağlı farklı dosya adları kullanılarak erişilebilir).

ls programına sabit bir bağlantı oluşturun:

```
mkdir / bin / tmp (sabit bağlantı için yeni bir dizin oluşturun)
```

```
ln / bin / ls / bin / tmp / ls
```

Şimdi yeni dosya adınız / bin / tmp / ls için ayrıntıları görüntüleyin:

```
ls -li / bin / tmp / ls
```

Örnek 3

```
[linuxlab@asus ~]$ sudo ln /bin/ls /bin/tmp/ls
[linuxlab@asus ~]$ ls -l /bin/tmp/ls
-rwxr-xr-x. 2 root root 157896 May 29 19:33 /bin/tmp/ls
[linuxlab@asus ~]$ ls -l /bin/ls
-rwxr-xr-x. 2 root root 157896 May 29 19:33 /bin/ls
[linuxlab@asus ~]$ ls -i /bin/ls
1311751 /bin/ls
[linuxlab@asus ~]$ ls -i /bin/tmp/ls
1311751 /bin/tmp/ls
[linuxlab@asus ~]$
```

Böylece, her iki dosyanın da aynı inode'u paylaştığını görüyoruz. Dolayısıyla bu dosyalardan birinin değiştirilmesi diğerini de etkileyecektir.

Dosya adlarından birini silmek, bağlantı sayacını azaltır. Sadece

bağlantı sayacı 0'a ulaştığında inode gerçekten kaldırılır::

```
rm / bin / tmp / ls
```

Örnek 4

```
[linuxlab@asus ~]$ rm /bin/tmp/ls
rm: remove write-protected regular file '/bin/tmp/ls'? y
rm: cannot remove '/bin/tmp/ls': Permission denied
[linuxlab@asus ~]$ ls -ld /tmp/
drwxrwxrwt. 14 root root 360 Oct 6 18:38 /tmp/
[linuxlab@asus ~]$
```

İzin reddedildi! İlginç bir şekilde, bu durumda normal bir kullanıcı olarak / bin / ls'ye bağlantı oluşturabiliriz, ancak yapışkan bit / tmp / dizini için ayarlandığı için bu bağlantıyı silemeyiz.

```
ls -ld / tmp /
```

İzinlerdeki "t" harfine dikkat edin ve yukarıda açıklanan anlama bakın.

Bağlantıyı kök olarak silebilirsiniz:

```
sudo rm / bin / tmp / ls
```

Örnek 5

```
alex@Lenovo:~$ sudo rm /tmp/ls
[sudo] password for alex:
alex@Lenovo:~$ dir /tmp/
alex@Lenovo:~$
```

/ Bin / tmp // bin / tmp / ls kaldırıldıktan sonra hiçbir içerik görmüyoruz.

5. İzinlerin sekizli gösterimi

stat komutu, inode'dan daha fazla bilgi görüntülemek için kullanılabilir:

```
stat / bin / ls
```

Örnek 6

```
alex@Lenovo:~$ stat /bin/ls
File: /bin/ls
Size: 134792      Blocks: 264      IO Block: 4096   regular file
Device: 2h/2d   Inode: 1970324837335149  Links: 1
Access: (0755/-rwxr-xr-x)  Uid: (  0/   root)   Gid: (  0/   root)
Access: 2018-09-25 10:22:10.000000000 +0300
Modify: 2017-10-02 20:51:20.000000000 +0300
Change: 2018-10-02 16:19:53.572589600 +0300
Birth: -
alex@Lenovo:~$
```

Unix file permissions

File's owner

File's group

Bu bilgilere bir göz atın. Çıktı, dosyaya en son ne zaman erişildiği, değiştirildiği ve inode'un en son ne zaman değiştirildiği ile birlikte erişim haklarını içermektedir.

stat komutu çıktısı, bilginin saklandığı formatı ve daha "insan tarafından okunabilir" bir çıktıyı içerir. Bildiğimiz gibi, kullanıcı hesaplarına sistem tarafından UID'ler tarafından atıfta bulunulur, bu durumda dosya kök kullanıcıya ait olduğundan UID 0'dır. Benzer şekilde, GID ile tanımlanan gruplar da 0 olarak tanımlanır. Gerçek izinler, bu durumda "0755" olmak üzere dört sekizli (0-7 rakamları) olarak saklanır. Bu, (artık tanıdık) insan dostu çıktı "-rwxr-xr-x" anlamına geliyor. Şimdilik ilk sekizliyi görmezden geleceğiz, bu normalde 0'dır, bunun özel anlamına daha sonra geri döneceğiz. Diğer üç sekizlinin her biri basitçe rwx için ikiliyi temsil eder, her biri 0 veya 1 olarak temsil edilir. Üçten ilki kullanıcıyı, sonra grubu ve sonra diğer izni temsil eder. Dönüştürmeyi yapmanın kolay ve hızlı bir yolu, şunları hatırlamaktır:

- r = 4
- w = 2
- x = 1

Ve üç sekizlinin her birini üretmek için bunları bir araya getirin. Örneğin, rwx = binary 111 = (4 + 2 + 1) = 7.

- Aynı şekilde, r-x = binary 101 = (4 + 1) = 5.
- Böylelikle, "-rwxr-xr-x" = 755.

6. Linux sistemindeki dosya izinlerini değiştirme, kullanıcı, grup ekleme, dosya sahibini ve grubunu değiştirme

Şimdi, iki kullanıcı arasında geçiş yapma fırsatına ihtiyacımız var. Kullanıcıların listesi şu şekilde görüntülenebilir:

```
cat /etc/passwd
```

/etc/passwd dosyasının içeriğini gösteriyor.

Örnek 7

```
alex@Lenovo: ~  
alex@Lenovo:~$ cat /etc/passwd  
root:x:0:0:root:/root:/bin/bash  
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin  
bin:x:2:2:bin:/bin:/usr/sbin/nologin  
sys:x:3:3:sys:/dev:/usr/sbin/nologin  
sync:x:4:65534:sync:/bin:/bin/sync  
games:x:5:60:games:/usr/games:/usr/sbin/nologin  
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin  
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin  
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin  
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin  
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin  
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin  
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin  
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin  
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin  
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin  
gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin  
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin  
_apt:x:100:65534:./nonexistent:/usr/sbin/nologin  
systemd-timesync:x:101:102:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin  
systemd-network:x:102:103:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin  
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin  
alex:x:1000:1000:,,,:/home/alex:/bin/bash  
alex1:x:1001:1001:./home/alex1:/bin/sh  
bob:x:1002:1002:bob,,,:/home/bob:/bin/bash  
messagebus:x:104:110:./nonexistent:/usr/sbin/nologin  
alex@Lenovo:~$ ls -l /etc/passwd  
-rw-r--r-- 1 root root 1357 Sep 29 18:03 /etc/passwd  
alex@Lenovo:~$
```

/ Etc / passwd'nin her kullanıcının okunabildiğini görüyoruz. Şimdi şifresi student olan student isimli yeni bir kullanıcı oluşturalım. Sudo'nun yardımıyla süper kullanıcı haklarını kullanmak için bunları yapmamız gerekiyor. Sudo yapmasına izin verilen kullanıcıların hakları (sudo grubundan) / etc / sudoers dosyasında listelenmiştir.

vedi / etc / sudoers

```
alex@Lenovo: ~  
cat: /etc/sudoers: Permission denied  
alex@Lenovo:~$ sudo cat /etc/sudoers  
[sudo] password for alex:  
#  
# This file MUST be edited with the 'visudo' command as root.  
#  
# Please consider adding local content in /etc/sudoers.d/ instead of  
# directly modifying this file.  
#  
# See the man page for details on how to write a sudoers file.  
#  
Defaults        env_reset  
Defaults        mail_badpass  
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"  
  
# Host alias specification  
  
# User alias specification  
  
# Cmnd alias specification  
  
# User privilege specification  
root    ALL=(ALL:ALL) ALL  
  
# Allow members of group sudo to execute any command  
%sudo   ALL=(ALL:ALL) ALL  
  
# See sudoers(5) for more information on "#include" directives:  
  
#includedir /etc/sudoers.d
```

Sudo grup kullanıcılarının root ile aynı izinlere sahip olduğunu görüyoruz. Gruplar ve üyeleri aşağıdaki komutla görüntülenebilir.

```
cat /etc/group
```

Örnek 9



```
Select alex@Lenovo: ~
alex@Lenovo:~$ cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:alex
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:
fax:x:21:
voice:x:22:
cdrom:x:24:alex
floppy:x:25:
tape:x:26:
sudo:x:27:alex
audio:x:29:
dip:x:30:alex
www-data:x:33:
backup:x:34:
operator:x:37:
list:x:38:
irc:x:39:
src:x:40:
```

Alex kullanıcısının sudo grubuna ait olduğu gibi adm (sistem görevlerini izleyebilir), cdrom (CDROM'a erişebilir) ve dip (çevirmeli bağlantı için ppp, dip vb. araçları kullanabilir) gruplarına da ait olduğunu görüyoruz. Aşağıdaki satır yazılarak yeni bir kullanıcı oluşturulur

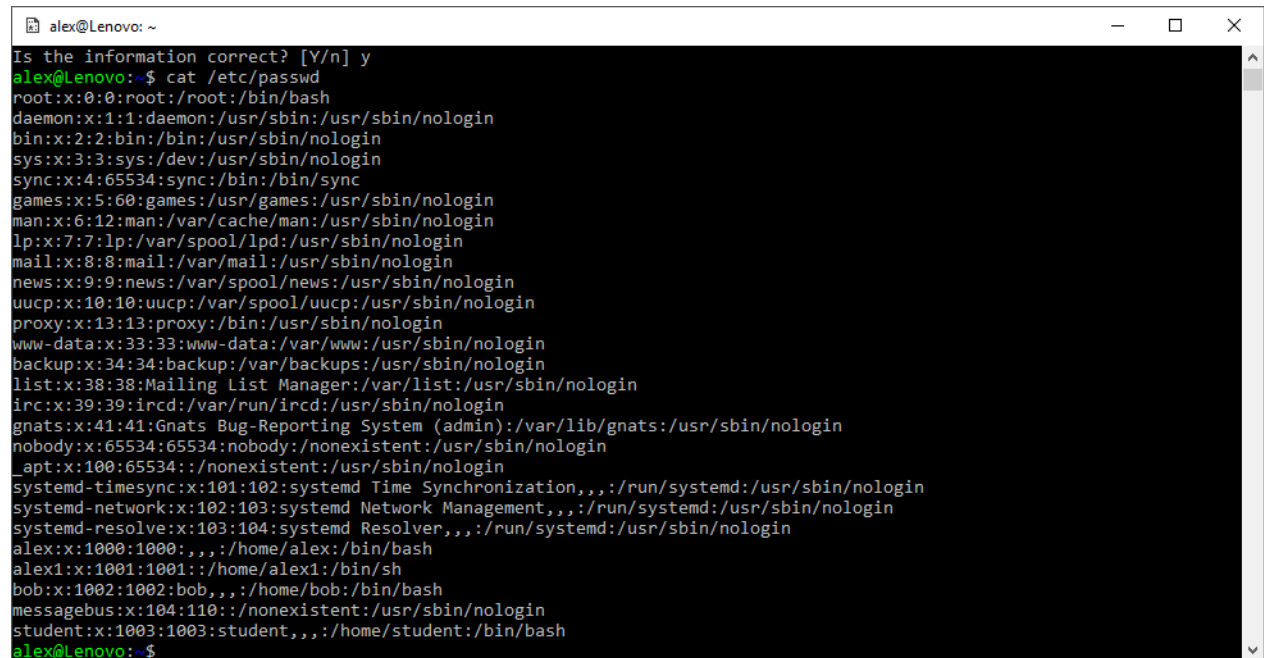
```
sudo adduser student
```

Örnek 10

```
alex@Lenovo:~$ sudo adduser student
[sudo] password for alex:
Adding user `student' ...
Adding new group `student' (1003) ...
Adding new user `student' (1003) with group `student' ...
Creating home directory `/home/student' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for student
Enter the new value, or press ENTER for the default
  Full Name []: student
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
alex@Lenovo:~$
```

Yeni kullanıcı student'in kullanıcı listesinde olup olmadığını kontrol edin:

Örnek 11



```
alex@Lenovo: ~
Is the information correct? [Y/n] y
alex@Lenovo:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534:/:nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:102:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:102:103:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
alex:x:1000:1000:,,,:/home/alex:/bin/bash
alex1:x:1001:1001:~/home/alex1:/bin/sh
bob:x:1002:1002:bob,,,:/home/bob:/bin/bash
messagebus:x:104:110:/:nonexistent:/usr/sbin/nologin
student:x:1003:1003:student,,,:/home/student:/bin/bash
alex@Lenovo:~$
```

Ana dizininizde "mysecret" adlı bir dosya oluşturun:

```
cat > ~/mysecret
```

Örnek 12

```
alex@Lenovo:~$ cat ? ~/mysecret
cat: '?': No such file or directory
cat: /home/alex/mysecret: No such file or directory
alex@Lenovo:~$ cat > ~/mysecret
It is my secret
Here it is
My secret is here
alex@Lenovo:~$
```

Bir dizi içerik satırı girin. Bir "secret, sır" girmeyi bitirdiğinizde (başkalarının görebileceği) Ctrl-D tuşlarına basın. İlk amacınız, "mysecret" dosyanızın aynı sistemdeki diğer kullanıcılar tarafından görülmemesini sağlamaktır. Önce yeni oluşturduğunuz dosyanın izinlerini görüntüleyin:

```
ls -l ~/mysecret
```

Örnek 13

```
alex@Lenovo:~$ ls -l ~/mysecret
-rw-rw-rw- 1 alex alex 45 Oct  2 19:24 /home/alex/mysecret
alex@Lenovo:~$
```

Oh hayır! O kadar gizli değil ...

Chmod komutu, bir dosyadaki izinleri ayarlamak için kullanılabilir. chmod, izinleri mutlak sekizlik değerlere veya görelî değişikliklere göre ayarlayabilir. Örneğin, bir dosya üzerinde izinleri octet'e dayalı ayarlamak için chmod'u kullanabilirsiniz: 770, sahibine ve grubuna rwx verirken diğerlerine izin vermez. Örnek:

```
chmod 770 /home / tmp / somefile
```

Örnek 14

```
alex@Lenovo:~$ mkdir /home/tmp
mkdir: cannot create directory '/home/tmp': Permission denied
alex@Lenovo:~$ sudo mkdir /home/tmp
alex@Lenovo:~$ touch /home/tmp/somefile
touch: cannot touch '/home/tmp/somefile': Permission denied
alex@Lenovo:~$ sudo touch /home/tmp/somefile
alex@Lenovo:~$ ls -l /home/tmp/somefile
-rw-r--r-- 1 root root 0 Oct  2 19:42 /home/tmp/somefile
alex@Lenovo:~$ chmod 770 /home/tmp/somefile
chmod: changing permissions of '/home/tmp/somefile': Operation not per
alex@Lenovo:~$ sudo chmod 770 /home/tmp/somefile
alex@Lenovo:~$ ls -l /home/tmp/somefile
-rwxrwx-- 1 root root 0 Oct  2 19:42 /home/tmp/somefile
alex@Lenovo:~$
```

Böylece, sudo komutu kullanılarak / home / tmp / somefile için izinler 644 (rw-) 'den 770'e (rwxrwx ---) değiştirilir. Veya görelî değişiklikler yapabilirsiniz: u-x, sahibin (kullanıcının) dosyayı yürütme yeteneğini kaldırır. Örnek:

```
chmod u-x / home / tmp / somefile
```

Örnek 15

```
alex@Lenovo:~$ sudo chmod u-x /home/tmp/somefile
alex@Lenovo:~$ ls -l /home/tmp/somefile
-rw-rwx--- 1 root root 0 Oct  2 19:42 /home/tmp/somefile
alex@Lenovo:~$
```

Aynı şekilde, o + w başkalarının dosyaya yazma yeteneğini de ekler

Örnek:

```
chmod o+w /home/tmp/somefile
```

Örnek 16

```
alex@Lenovo:~$ sudo chmod o+x /home/tmp/somefile
alex@Lenovo:~$ ls -l /home/tmp/somefile
-rw-rwx--x 1 root root 0 Oct  2 19:42 /home/tmp/somefile
alex@Lenovo:~$
```

mysecrets dosyanıza kendinize okuma-yazma izinleri vermek ve diğer herkese dosya için hiç bir izin vermemek için chmod kullanın

```
chmod 660 ~/ mysecrets
```

Örnek 17

```
alex@Lenovo:~$ chmod 660 ~/mysecrets
alex@Lenovo:~$ ls -l ~/mysecrets
-rw-rw---- 1 alex alex 33 Sep 29 12:36 /home/alex/mysecrets
alex@Lenovo:~$
```

Dosyaya student kullanıcısı adına erişmeyi deneyin:

```
cat / home / alex / mysecrets
```

Örnek 18

```
alex@Lenovo:~$ su student
Password:
student@Lenovo:/home/alex$ cat /home/alex/mysecrets
cat: /home/alex/mysecrets: Permission denied
student@Lenovo:/home/alex$
```

Bir "~/ myshare" dosyası oluşturun ve herkese okuma-yazma erişimi verin. İzinleri doğru şekilde ayarlayıp ayarlamadığınızı test edin. Ayrıca diğer kullanıcılara "~/myshare" dosyasını bulmaları için gerekli diğer izinleri verin.

Örnek 19

```
[student@asus linuxlab]$ su linuxlab
Password:
[linuxlab@asus ~]$ touch ~/myshare
[linuxlab@asus ~]$ chmod 666 ~/myshare
[linuxlab@asus ~]$ ls -l ~/myshare
-rw-rw-rw-. 1 linuxlab linuxlab 0 Oct  8 20:44 /home/linuxlab/myshare
[linuxlab@asus ~]$ kwrite ~/myshare
[linuxlab@asus ~]$ cat ~/myshare
It is to be shared with
all the people
[linuxlab@asus ~]$ sudo chmod 701 /home/linuxlab
[sudo] password for linuxlab:
[linuxlab@asus ~]$ su student
Password:
[student@asus linuxlab]$ cat /home/linuxlab/myshare
It is to be shared with
all the people
[student@asus linuxlab]$ █
```

“mygroupshare” oluşturun, grubunuzdaki herkese yalnızca okuma erişimi verin. İzinleri doğru şekilde ayarlayıp ayarlamadığınızı test edin.

Örnek 20

```
[linuxlab@asus ~]$ cat > ~/mygroupshare
It is my group share
People in my group can access it
[linuxlab@asus ~]$ cat ~/mygroupshare
It is my group share
People in my group can access it
[linuxlab@asus ~]$ █
```

Aşağıdaki komut ile alex grubuna student kullanıcıyı ekleyin

```
sudo usermod -a -G alex student
```

Örnek 21

```
alex@Lenovo:~$ ls -l ~/mygroupshare
-rw-rw-rw- 1 alex alex 54 Oct  3 20:11 /home/alex/mygroupshare
alex@Lenovo:~$ sudo usermod -a -G alex student
[sudo] password for alex:
```

```
alex@Lenovo:~$ cat /etc/group
```

```
alex:x:1000:student
```

```
alex@Lenovo:~$ su student
Password:
student@Lenovo:/home/alex$ cat /home/alex/mygroupshare
It is my group share
People in my group can access it
student@Lenovo:/home/alex$ ls -l /home/alex/mygroupshare
-rw-rw-rw- 1 alex alex 54 Oct  3 20:11 /home/alex/mygroupshare
student@Lenovo:/home/alex$
```

Aşağıdaki komutu kullanarak mygroup isminde yeni bir grup oluşturun

sudo groupadd mygroup

Örnek 22

```
[linuxlab@asus ~]$ sudo groupadd mygroup
[sudo] password for linuxlab:
[linuxlab@asus ~]$
```

Aşağıdaki komutu kullanarak mygroupshare'in grubunu

alex'ten mygroup olarak değiştirin

Sudo chown: mygroup ~ / mygroupshare

Bir dosyanın sahibini ve grubunu değiştirmek için aşağıdaki komutu kullanın chown new_owner:newgroup file

Örnek 23

```
alex@Lenovo:~$ sudo chown :mygroup ~/mygroupshare
alex@Lenovo:~$ ls -l ~/mygroupshare
-rw-rw-rw- 1 alex mygroup 54 Oct  3 20:11 /home/alex/mygroupshare
alex@Lenovo:~$
```

mygroupshare grubumu alex olarak değiştirin ve alex grubuna mygroupshare için yalnızca : okuma erişimi verin

sudo chown: alex ~ / mygroupshare

sudo chmod g-w ~ / mygroupshare

Örnek 24

```
alex@Lenovo:~$ sudo chown :alex ~/mygroupshare
```

```
alex@Lenovo:~$ ls -l ~/mygroupshare
-rw-rw-rw- 1 alex alex 54 Oct  3 20:11 /home/alex/mygroupshare
alex@Lenovo:~$ sudo chmod g-w ~/mygroupshare
alex@Lenovo:~$ ls -l ~/mygroupshare
-rw-r--rw- 1 alex alex 54 Oct  3 20:11 /home/alex/mygroupshare
alex@Lenovo:~$
```

"staff" adında yeni bir grup oluşturun ve sizin ve sınıf arkadaşınızın(diğer kullanıcı) birlikte üzerinde işbirliği yapabileceğiniz (her ikisi de düzenleyebilir) bir dosya oluşturun. İzinleri doğru şekilde ayarlayıp ayarlamadığınızı test edin. Her iki kullanıcının da dosyayı düzenleyebilmesi gerekir, ancak diğer kullanıcıların yazma erişimine sahip olmaması gerekir.

Örnek 25

```
alex@Lenovo:~$ dir
accessmysecrets accessmysecrets.c acl.txt mygroupshare mysecret mysecrets myshare newfile test
alex@Lenovo:~$ mkdir tmp
alex@Lenovo:~$ dir
accessmysecrets accessmysecrets.c acl.txt mygroupshare mysecret mysecrets myshare newfile test tmp
alex@Lenovo:~$ touch tmp/test1 tmp/test2 tmp/test3
alex@Lenovo:~$ dir tmp
test1 test2 test3
alex@Lenovo:~$
```

Çalışma 1.

mkdir testi

touch testi / test1 test / test2 test / test3

Yeni "test" dizininde bulunan tüm dosyalar için özyinelemeli(recursively) izinleri ayarlamak için tek bir chmod komutu kullanın.

İpucu: "man chmod"

7. umask Komutu

Yeni oluşturulan dosyamızın, herkesin dosyayı okuyabileceği anlamına gelen izinlerle başladığını unutmayın. Bu, kullanıcı dosyası oluşturma modu maskesi (umask) ayarlayarak önlenir. Her işlemin bir umask'i vardır: bu yeni oluşturulan dosyaların izinlerini belirleyen bir sekizlik(octal)'dır Dosyalar için varsayılan "666" ve yeni yürütülebilir dosyalar için "777" izinlerini kaldırarak çalışır.(mantıksal NOT'a göre). Yani, "000" umask "666" izinlerine sahip yeni dosyalar ile sonuçlanacaktır. "022" olan bir umask (varsayılan değerdir) "644" verir, yani "rw- -r- -r-". umask sistem komutu, mevcut işlem(proses) için umask ayarlama kullanılır.

Mevcut umask değerini kontrol edin:

Umask

Örnek 26

```
alex@Lenovo:~$ umask -s
-bash: umask: -s: invalid option
umask: usage: umask [-p] [-S] [mode]
alex@Lenovo:~$ umask -S
u=rwx,g=rwx,o=rwx
alex@Lenovo:~$ umask -p
umask 0000
alex@Lenovo:~$ umask
0000
alex@Lenovo:~$
```

Çalışma 2.

umask yerleşik komutunu kullanarak, umask'ınızı yeni dosyalara yalnızca sizin tarafınızdan rw erişilebilecek şekilde ayarlayın (grubunuz veya başkaları için değil):

umask XXX

burada XXX, kullanılacak yeni umasktir.

Yeni bir dosya oluşturun izinlerini kontrol ederek yeni umask değerinizi test edin: yeni dosya adına dokunun

ls -l yenisosyaadı

İzinler "rw -" olarak mı okunuyor? Değilse, umask değerini değiştirin ve tekrar deneyin.

8. Set UID (SUID)

Bazen bir kullanıcının her zaman sahip olmaması gereken izinleri gerektiren şeyleri yapabilmesi gerekir. Örneğin, parolanızı değiştirmek için passwd komutu kullanılır. Bu, / etc / shadow dosyasına okuma ve yazma erişimi gerektirir. Açıkçası, her kullanıcının bu tür bir erişime sahip olmaması gerekir! Ayrıca, ping komutunun ham ağ erişimine ihtiyacı var ... Yine her kullanıcının yapabileceği bir şey değil. Unix'in çözümü, Set UID (SUID) komutudur. SUID kullanılarak, işlemlere başka bir kullanıcı olarak çalışma izni verilebilir. Örneğin, passwd'yi çalıştırdığımızda, program aslında root olarak çalışır (çoğu Unix sisteminde). Aslında, her prosesin birden çok kimliği vardır, bunlara şunlar dahildir:

- real UID (RUID): komutu çalıştıran kullanıcı
- effective UID (EUID): sürecin işlenme şekli

effective UID'nin nasıl belirlendiğine bir göz atın:

```
ls -l /usr /bin /passwd
```

Örnek 27

```
alex@Lenovo:~$ ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 63736 Jul 27 11:07 /usr/bin/passwd
alex@Lenovo:~$
```

Dosya izinlerindeki "s", UID dosyasının effective UID olarak kullanılacağı anlamına gelir.

stat /usr /bin /passwd komutunu çalıştırın

Sonra başka bir kullanıcıya (bob) geçin ve passwd'yi çalıştırın

Örnek 28

```
alex@Lenovo:~$ stat /usr/bin/passwd
File: /usr/bin/passwd
Size: 63736          Blocks: 128          IO Block: 4096    regular file
Device: 2h/2d   Inode: 1970324837335145  Links: 1
Access: (4755/-rwsr-xr-x)  Uid: (  0/   root)   Gid: (  0/   root)
Access: 2018-09-25 10:22:10.000000000 +0300
Modify: 2018-07-27 11:07:37.000000000 +0300
Change: 2018-09-25 10:23:44.512141000 +0300
Birth: -
alex@Lenovo:~$ su bob
Password:
bob@Lenovo:/home/alex$ passwd
Changing password for bob.
(current) UNIX password:
```

Sekizli izinlerdeki (4755'teki) 4 rakamı, dosyanın UID = root ile yürütüleceği anlamına gelir. Bu nedenle SUID biti, inode'daki ilk (kullanıcı) izin sekizlisinde saklanır.

Başka bir bash sekmesinden çalışan işlemleri görüntüleme

```
ps -af
```

Örnek 29

```
alex@Lenovo:~$ ps -af
UID      PID  PPID  C  STIME TTY      TIME  CMD
alex      4    3    0  09:32 tty1    00:00:00 -bash
root     46    4    0  10:27 tty1    00:00:00 su bob
bob      47   46    0  10:27 tty1    00:00:00 bash
root     55   47    0  10:36 tty1    00:00:00 su alex
alex     56   55    0  10:36 tty1    00:00:00 bash
root     60   56    0  10:38 tty1    00:00:00 su bob
bob      61   60    0  10:38 tty1    00:00:00 bash
root     64   61    0  10:38 tty1    00:00:00 passwd
alex     66   65    0  10:38 tty2    00:00:00 -bash
alex     83   66    0  10:52 tty2    00:00:00 ps -af
alex@Lenovo:~$
```

bob tarafından başlatılmasına rağmen passwd'nin UID = root ile çalıştığını görüyoruz. Aşağıdaki komutu kullanarak / home / içinde çalışan bütün programları SUID ile bulun

```
sudo find /home -perm -4000 -type f -print
```

ve stat kullanarak bulguları kontrol edin:

Örnek 30

```
alex@Lenovo:~$ sudo find /home -perm -4000 -type f -print
[sudo] password for alex:
/home/alex/accessmysecrets
alex@Lenovo:~$ stat accessmysecrets
  File: accessmysecrets
  Size: 17032          Blocks: 40          IO Block: 4096   regular file
Device: 2h/2d  Inode: 6192449487819206  Links: 1
Access: (4711/-rws--x--x)  Uid: ( 1000/   alex)   Gid: ( 1000/   alex)
Access: 2018-09-29 19:28:36.212452600 +0300
Modify: 2018-09-29 19:28:36.328067700 +0300
Change: 2018-09-29 19:54:57.398230400 +0300
 Birth: -
alex@Lenovo:~$
```

Program erişim sırları (accessmysecrets) aşağıda açıklanmıştır.

9.C'de SUID programı yazmak

Dosyaya doğrudan erişimi paylaşmaksızın, programı çalıştıran herkese "mysecret" dosyanızın içeriğine erişim vermek için bir SUID programı oluşturacaksınız. "~ / Mysecrets" e yalnızca sahip tarafından erişilebilir olduğundan emin olun: ls -la o dosya için "rw-----" göstermelidir.

Örnek 31

```
alex@Lenovo:~$ chmod 600 mysecrets
alex@Lenovo:~$ ls -l mysecrets
-rw----- 1 alex alex 33 Sep 29 12:36 mysecrets
alex@Lenovo:~$
```

Yeni bir "accessmysecret.c" dosyası oluşturarak bir C programı oluşturun:
vi accessmysecret.c

Unutmayın, vi modaldir. Ekleme moduna girmek için "i" ye basın ve ardından şu kodu girin:

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>
#include <errno.h>
int main()
{
printf(" UID GID \n"
"Real %d Real %d \n"
"Effective %d Effective %d \n",
getuid (), getgid (),
geteuid(), getegid());
FILE *fp = fopen("mysecrets", "r");
if (fp == NULL) {
printf("Error: Could not open file");
exit(EXIT_FAILURE);
}
char c;
while ((c=getc(fp)) != EOF) {
putchar(c);
}
putchar('\n');
return EXIT_SUCCESS;
}
```

Örnek 32

```
alex@Lenovo: ~
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>
#include <errno.h>

int main(){
printf("    UID        GUID \n Real    %d    %d \n Effective    %d    %d \n",
    getuid(), getgid(), geteuid(), getegid());
FILE *fp=fopen("mysecrets","r");
if(fp==NULL){
printf("Error: Could not open file");
exit(EXIT_FAILURE);
}
char c;
while((c=getc(fp))!=EOF) putchar(c);
putchar('\n');
return EXIT_SUCCESS;
}

~
~
~
~
~
~
"accessmysecrets.c" 22 lines, 461 characters
```

Değişikliklerinizi kaydedin ve çıkın (Esc, ": wq"). Başka herhangi bir metin düzenleyici (örneğin, editor) kullanabilirsiniz:

```
alex@Lenovo: ~
GNU nano 2.9.8 accessmysecrets.c
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>
#include <errno.h>

int main(){
printf("    UID        GUID \n Real    %d    %d \n Effective    %d    %d \n",
    getuid(), getgid(), geteuid(), getegid());
FILE *fp=fopen("mysecrets","r");
if(fp==NULL){
printf("Error: Could not open file");
exit(EXIT_FAILURE);
}
char c;
while((c=getc(fp))!=EOF) putchar(c);
putchar('\n');
return EXIT_SUCCESS;
}

[ Read 22 lines ]
^G Get Help      ^O Write Out    ^M Where Is     ^K Cut Text     ^J Justify     ^C Cur Pos     M-U Undo       M-A Mark Text
^X Exit          ^R Read File    ^N Replace     ^U Uncut Text  ^T To Spell    ^G Go To Line  M-E Redo       M-G Copy Text
```

```
alex@Lenovo:~$ vi accessmysecrets.c
alex@Lenovo:~$ editor accessmysecrets.c
alex@Lenovo:~$
```

Programı derleyin (bir yürütülebilir dosya oluşturmak için C kodunu kullanır):

```
gcc accessmysecrets.c -o accessmysecrets
```

Dosya için izinleri (chmod kullanarak) setuid olarak ayarlayın:

```
chmod u + s accessmysecrets
```

SUID içeren izinleri kontrol edin:

```
ls -l accessmysecrets
```

Örnek 33

```
alex@Lenovo:~$ gcc accessmysecrets.c -o accessmysecrets
alex@Lenovo:~$ chmod u+s accessmysecrets
alex@Lenovo:~$ ls -l accessmysecrets
-rws--x--x 1 alex alex 17032 Oct  4 11:54 accessmysecrets
alex@Lenovo:~$ stat mysecrets
  File: mysecrets
  Size: 33          Blocks: 0          IO Block: 4096   regular file
Device: 2h/2d  Inode: 50665495807921258  Links: 1
Access: (0600/-rw-----)  Uid: ( 1000/   alex)   Gid: ( 1000/   alex)
Access: 2018-09-29 12:35:50.285271800 +0300
Modify: 2018-09-29 12:36:26.692635300 +0300
Change: 2018-10-04 11:46:09.273720700 +0300
 Birth: -
alex@Lenovo:~$
```

Programı çalıştır:

```
./accessmysecrets
```

Örnek 34

```
alex@Lenovo:~$ accessmysecrets
bash: accessmysecrets: command not found
alex@Lenovo:~$ ./accessmysecrets
  UID      GUID
  Real    1000    1000
  Effective      1000          1000
it is my secret
I want to define
alex@Lenovo:~$
```

Programın real(gerçek) ve effective(etkili) kimliğini çıkardığına dikkat edin.

Başka bir kullanıcıya geçin ve programı çalıştırın:

```
/home /kullanıcı adınız /accessmysecrets
```

Örnek 35

```
alex@Lenovo:~$ su student
Password:
student@Lenovo:/home/alex$ /home/alex/accessmysecrets
      UID      GUID
Real    1003      1003
Effective      1000      1003
it is my secret
I want to define
student@Lenovo:/home/alex$
```

Etkin kimliğin programın sahibine ait olduğunu unutmayın. Sırlar dosyasına doğrudan erişiminiz olmasa bile, mysecrets dosyasının içeriğini de görmelisiniz.

Çalışma 3

Başka bir kullanıcıya geçin ve SUID accessmysecrets programını kullanarak sahip kullanıcının dosyalarından herhangi birine okuma erişimi elde edin!

İpucu: Bu kodla ilgili bir güvenlik sorunu var.

Başka bir ipucu: Sabit bağlantıları düşünün.

Çözüm:

Dosyayı açarken mutlak bir dosya adı kullanmamaktan kaynaklanan bir güvenlik sorunu var, "/home/user/mysecrets" yerine "mysecrets" açılıyor. Unutmayın, herhangi bir kullanıcı bir dosyaya sabit bağlantı oluşturabilir (bu nedenle istedikleri yerde SUID programının "kopyasını" oluşturabilirler).

Saldırganın yazabileceği bir dizinde SUID programına sabit bir bağlantı oluşturun, ardından SUID kullanıcısının sahip olduğu herhangi bir dosyaya sabit bir bağlantı oluşturun ve programla aynı dizinde "mysecrets" olarak adlandırın, ardından çalıştırdığınızda program dosyanın içeriğini yazacaktır.

Bu güvenlik açığından aşağıdaki şekilde yararlanabilirsiniz:

```
su - student
```

```
ln /home/user/accessmysecrets /tmp/access
```

```
ln /home/user/someotherfile /tmp/mysecrets
```

```
/tmp/access
```

Çalışma 4

Yukarıdaki güvenlik açığını düzeltmek için programı değiştirin.

Çalışma 5

Programı, başkalarına mysecrets dosyasının yalnızca ilk satırının görüntüleneceği şekilde değiştirin.

Çalışma 6

Programı, komut dosyası UID'yi kontrol edecek ve yalnızca belirli bir kullanıcı için devam edecek şekilde değiştirin (örneğin, kullanıcı root ise).

İpucu: "man getuid

10. Linux Genişletilmiş ACL'ler

Standart Unix izinlerini araştırdık. Modern Linux sistemleri (ve diğer bazı Unix tabanlı sistemler) artık daha eksiksiz (ve karmaşık) ACL desteğine sahip. Daha önce bahsedildiği gibi, bir nesneye (kaynağa) bir erişim kontrol listesi (ACL) eklenir ve erişime izin verilen tüm özneler (kullanıcılar / aktif varlıklar), yetkilendirilmiş erişim türüyle birlikte listeler.

Setfacl komutunu kullanarak mysecrets dosyanızda bir dosya ACL'si ayarlayın:

```
setfacl -m u:student:r ~/mysecrets
```

Örnek 36

```
[linuxlab@asus ~]$ sudo setfacl -m u:student:r ~/mysecrets
[linuxlab@asus ~]$ getfacl ~/mysecrets
getfacl: Removing leading '/' from absolute path names
# file: home/linuxlab/mysecrets
# owner: linuxlab
# group: linuxlab
user::rw-
user:student:r--
group:---
mask:r--
other:---
```

Bu, "student" kullanıcılarına dosyaya okuma erişimi verir.

stat programının(komutu) genellikle ACL'nin farkında olmadığını, bu nedenle olağan dışı hiçbir şeyi rapor etmeyeceğini unutmayın:

```
stat ~/mysecrets
```

ls programı Dosya ACL'lerini algılamak için kullanılabilir:

```
ls -la ~/mysecrets
```

```
-rw-r-----+ 1 cliff e kullanıcı 22 Şub 28 11:47 mysecrets
```

Çıktının bir "+" içerdiğini unutmayın. Bu, ACL'nin yerinde olduğunu gösterir.

İzinleri görüntülemek için getfacl kullanın:

```
getfacl ~/mysecrets
```

Bir veya daha fazla belirli kullanıcıya (diğer sınıf üyeleri) mysecrets dosyanıza okuma erişimi sağlamak için Linux Dosya ACL'lerini kullanın.

ACL'leri kullanarak, herhangi bir gruba (sizin seçeceğiniz) mygroupshare dosyanıza okuma-yazma erişimi verin.

Yeni eklediğiniz grup iznini kaldırın.

Örnek: setfacl -x g:staff file

10.Sonuç

Bu noktada çıkarılan sonuçlar:

1. Dosya izinleri, sabit bağlantılar ve inode'lar hakkında bilgi edinildi
2. İzinlerin sekizlik(octal) temsilleri hakkında bilgi edinildi
3. chmod kullanarak belirli kullanıcılara ve gruplara erişim sağlamak için Unix dosya izinleri değiştirildi
4. Yeni dosyalara uygulanan izinleri değiştirmek için umask kullanıldı
5. Set UID (SUID) hakkında bilgi edinildi, C'yi daha yakından tanıdı ve bir SUID C programı derlendi

6. Kendiniz de biraz daha programlama yapmış olabilirsiniz
 7. Daha gelişmiş güvenlik politikalarını yapılandırmak için Linux Genişletilmiş ACL'leri kullandı.
- Tebrikler!

Referanslar

1. Z. Cliffe Schreuders. Access controls and Linux/Unix file permissions, z.cliffe.schreuders.org/edu/ADS/Access%20Controls.pdf