



Linux - II

II. kısımda Linux'da metinler üzerinde işlemler, fitreleme ve yönetimler ele alınmıştır.

Metin Düzenleme

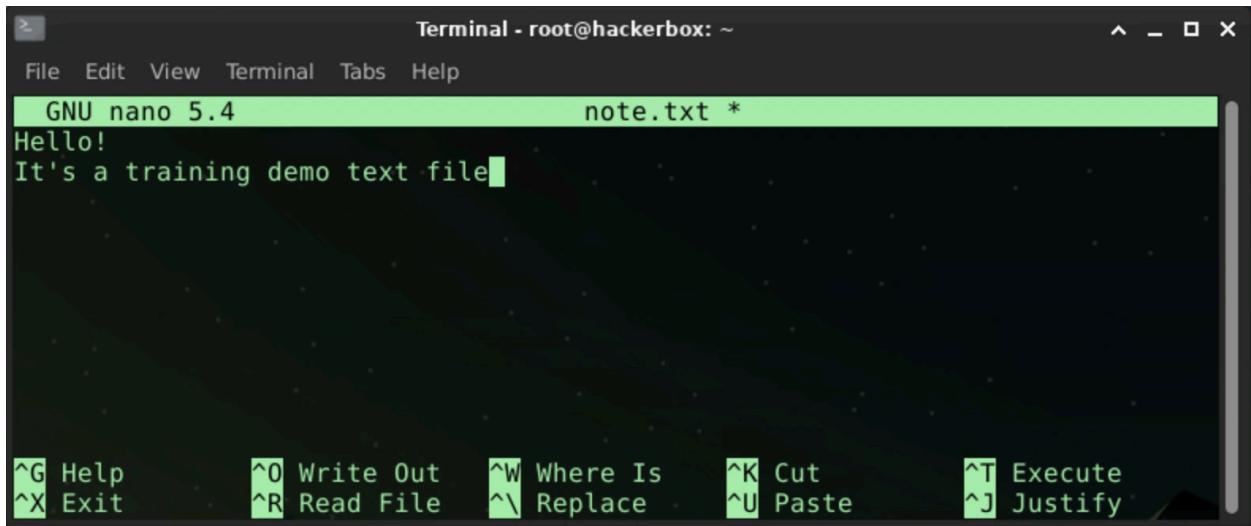
Nano Metin Editörü

Nano, Linux tabanlı sistemlerde en çok kullanılan editörlerden biridir. Basit ve etkili bir metin editörüdür, aynı zamanda linux dağıtımları ile önyüklü olarak gelmektedir. Kullanmadan önce nano editörü hakkında herhangi bir ön bilgiye sahip olmamız gerekmektedir. Nano'da dosya üzerinde işlem yapmak için komut kullanılmaz, tüm temel işlemler editörün alt kısmında görüntülenir. Bunları **CTRL** tuşu ile tetikleyebiliriz, örneğin dosyayı kaydetmek için **CTRL+O** tuşlarına, editörden çıkmak için **CTRL+X** tuşlarına basmak yeterlidir.

Bir dosyayı **nano** editör ile düzenlemek için aşağıdaki komutu çalıştırın:

```
root@hackerbox:~$nano note.txt
```

Yukarıdaki komut `note.txt` dosyasını `nano` editör ile açacaktır. Dosyayı düzenlemek için imleci hareket ettirip, istediğiniz metni girebilirsiniz.



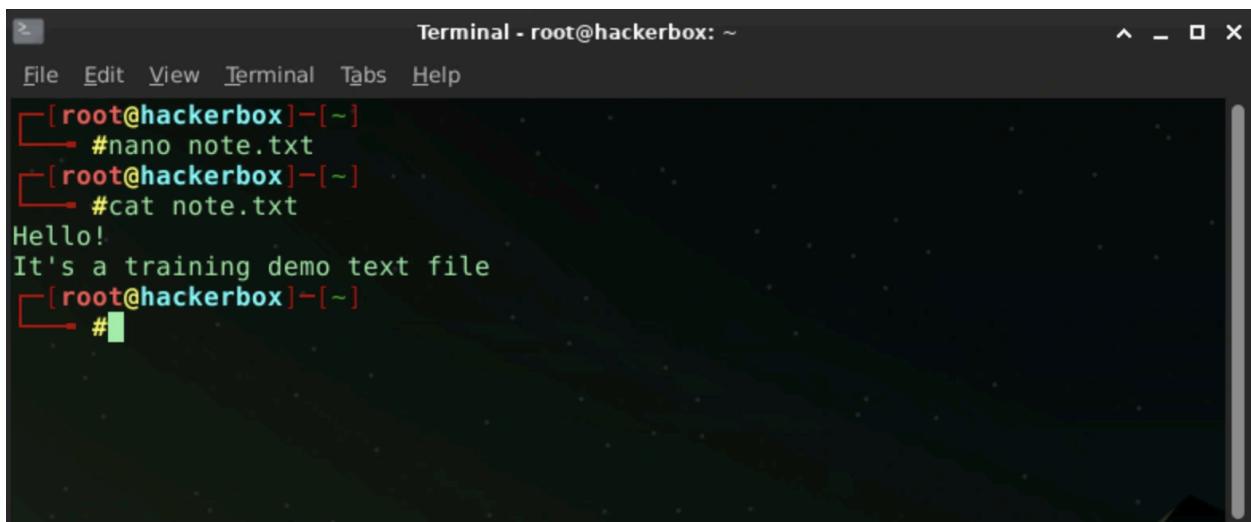
The screenshot shows a terminal window titled "Terminal - root@hackerbox: ~". The title bar also displays "GNU nano 5.4" and "note.txt *". The main area of the terminal shows the contents of the "note.txt" file:

```
Hello!
It's a training demo text file
```

At the bottom of the terminal window, there is a menu bar with "File", "Edit", "View", "Terminal", "Tabs", and "Help". Below the menu bar, there is a toolbar with various keyboard shortcut keys and their corresponding functions:

Key	Action
<code>^G</code>	Help
<code>^X</code>	Exit
<code>^O</code>	Write Out
<code>^R</code>	Read File
<code>^W</code>	Where Is
<code>^\\</code>	Replace
<code>^K</code>	Cut
<code>^U</code>	Paste
<code>^T</code>	Execute
<code>^J</code>	Justify

Metin editöründen çıktıktan sonra, yazdığımız metnin gerçekten kaydedildiğini doğrulamak için `cat` komutu ile dosya içeriğini yazdırabiliriz. `cat` komutu, dosyaların içeriğini terminale yazdırmak için kullanılır.



The screenshot shows a terminal session with the following history:

- Entered `#nano note.txt` to create the file.
- Entered `#cat note.txt` to display the file's content.

The terminal output shows the file's content:

```
Hello!
It's a training demo text file
```

Vim (Vi IMproved)

`vim` (veya klasik `vi`), Linux dünyasının en güçlü editörüdür. Öğrenme eğrisi diktir ancak bir kez öğrenildiğinde klavyeden elinizi kaldırmadan şimşek hızında düzenleme yapmanızı sağlar. Fare kullanmadan her şeyi klavyeyle yaparsınız.

Vim ile dosya açmak:

```
vim script.sh
```

Vim Modları

Vim açıldığında **Normal Modda** başlar. Bu modda yazı yazamazsınız, sadece komut verirsiniz.

1. **Normal Mod (Varsayılan):** Gezinme, kopyalama, silme işlemleri yapılır. ESC tuşu her zaman sizin bu moda döndürür.
2. **Insert Modu (Ekleme):** Yazı yazma modudur. Normal moddayken i tuşuna basarak geçilir. Sol altta -- INSERT -- yazar.
3. **Visual Mod (Seçim):** Metin seçmek (blok seçimi) için kullanılır. Normal moddayken v tuşuna basarak geçilir.

Temel Vim Komutları ve Senaryoları (Normal Modda)

Senaryo 1: Yazı yazmak

1. Vim ile dosya açın: vim test.txt
2. i tuşuna basarak Insert moduna geçin. (Sol altta -- INSERT -- yazar).
3. "Merhaba Dünya" yazın.
4. ESC tuşuna basarak Normal moda dönün.

Senaryo 2: Bir satırı silmek

1. Silmek istediğiniz satırın üzerine yön tuşlarıyla veya h,j,k,l ile gelin.
2. dd tuşlayın. Satır silinecektir.

Senaryo 3: Dosyayı Kaydedip Çıkmak

1. Normal modda olduğunuzdan emin olun (ESC).
2. :wq yazın (write and quit) ve Entera basın.

Kısayol Tablosu:

Tuş	Eylem
i	Yazma moduna geç (Insert).
ESC	Normal moda dön.
:w	Kaydet (Write).
:q	Çık (Quit).
:wq	Kaydet ve Çık.
:q!	Kaydetmeden zorla çıkış.
dd	Satırı sil (Kes).
yy	Satırı kopyala.
p	Yapıştır.
u	Geri al (Undo).
/kelime	"kelime"yi ara. (n ile sonraki).

Diğer Editörler

- **view:** Vim'i sadece okuma modunda açar. Dosyayı yanlışlıkla değiştirme riskini önler.

```
view onemli_dosya.txt
```

- **Grafiksel Editörler:** Eğer masaüstü ortamınız (GNOME, KDE vb.) varsa, gedit, kate, VS Code (code.) gibi grafiksel editörleri de kullanabilirsiniz.

Fitreleme Komutları

Cat

Cat komutunun temel amacı bir veya birden fazla metin dosyasının içeriğini terminalde göstermektir. Bu komutu kullanarak dosya içeriklerini hızlı bir şekilde görüntüleyebilmemekteyiz.

```
root@hackerbox:~$ cat /etc/ssh/sshd_config
# Port 22
```

```
# AddressFamily any
# ListenAddress 0.0.0.0
# ListenAddress ::

PermitRootLoginno
PasswordAuthenticationyes
PermitEmptyPasswordsno
ChallengeResponseAuthenticationno
UsePAMyes
```

Yukarıdaki örnekte, SSH servisinin /etc/ssh/sshd_config yolundaki ayar dosyasının içeriği cat komutu ile ekrana yazdırılmıştır.

Head

head komutu, belirtilen bir dosyanın en başından belirli sayıda satırı görüntülemek için kullanılır. Varsayılan olarak, head komutu dosyanın ilk 10 satırını gösterir, ancak bu sayı -n parametresi ile değiştirilebilir.

Bu komut, dosya içeriğinin tamamını değil, sadece başlangıç kısmını hızlıca gözden geçirmek istediğinizde kullanılabilir.

```
root@hackerbox:~$ head -n3 /var/log/apache2/access.log
192.168.1.1 - -[15/Mar/2024:10:00:00 +0000]"GET /index.html HTTP/1.1"20061
2""Mozilla/5.0 (Windows NT 10.0; Win64; x64)"
192.168.1.2 - -[15/Mar/2024:10:00:02 +0000]"POST /login.php HTTP/1.1"2004
52"http://example.com/login""Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/
MRA58N)"
192.168.1.3 - -[15/Mar/2024:10:00:03 +0000]"GET /wp-admin HTTP/1.1"40349
7""Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6)"
```

Yukarıdaki örnekte, Apache2 Web Server servisine ait /var/log/apache2/access.log yolundaki erişim kayıtlarını barındıran dosyanın içeriğinin **ilk 3 satırı** ekrana yazdırılmıştır.

Tail

tail komutu, belirtilen bir dosyanın sonundan itibaren belirli sayıda satırı görüntülemek için kullanılır. Varsayılan olarak, tail komutu dosyanın son 10 satırını gösterir, ancak bu sayı -n parametresi ile değiştirilebilir.

Bu komut, özellikle log dosyaları gibi sürekli büyüyen dosyaların en son eklenen içeriğini gözlemlemek için son derece yararlıdır.

```
root@hackerbox:~$ tail -n3 /var/log/auth.log
Mar15 12:00:00 servername sshd[23456]: Failed password for invalid user ad
min from 192.168.1.1 port 54321 ssh2
Mar15 12:01:00 servername sshd[23457]: Accepted password for user1 from 19
2.168.1.2 port 65432 ssh2
Mar15 12:02:00 servername sshd[23458]: Failed password for user2 from 192.1
68.1.3 port 76543 ssh2
```

Yukarıdaki örnekte, auth.log dosyasının son üç satırı ekrana yazdırılmıştır. auth.log dosyası, Linux sisteminde kullanıcı kimlik doğrulama işlemleri ile ilgili olayların kaydedildiği bir log dosyasıdır.

Bir log dosyasını sürekli izlemek (örneğin bir sunucuya gelen istekleri anlık görmek) için -f (follow) parametresi kullanılır.

```
user@hackerbox:~$ tail -f /var/log/syslog
Aug 11 00:01 server CRON[123]: (root) CMD (command)
Aug 11 05:01 server sshd[456]: Accepted password for user...
```

Sort

sort komutu, verilen dosyanın içeriğini alfabetik sıralar.

```
root@hackerbox:~$ cat names.txt
Bob
Charlie
Alice
```

```
root@hackerbox:~$ sort names.txt
Alice
```

```
Bob  
Charlie
```

Yukarıdaki örnekte, "names.txt" içerisinde yer alan isimleri "alfabetik" olarak ekrana yazdırıldı.

Uniq

uniq komutu, ardışık olarak tekrar eden satırları filtreleyerek dosya içerisindeki benzersiz satırları gösterir.

Genellikle sort komutu ile birlikte kullanılır çünkü tek başına kullanıldığında sadece ardışık olarak tekrar eden satırları tespit eder.

```
root@hackerbox:~$cat names.txt  
Alice  
Charlie  
Alice  
Bob
```

```
root@hackerbox:~$sort names.txt | uniq  
Alice  
Bob  
Charlie
```

- c: Tekrar sayısını gösterir (Count).

Örnek: Kimden kaç tane var sayalım

```
user@hackerbox:~$sort isimler.txt | uniq -c  
2 Ali Veli  
1 Ayşe Yılmaz  
1 Mehmet Öz
```

Grep

grep komutu, dosyalar içindeki belirli metin dizelerini aramak, satırları filtrelemek ve eşleşen sonuçları göstermek için kullanılır.

grep çok güçlü bir araçtır ve log dosyaları, konfigürasyon dosyaları veya herhangi bir metin dosyası üzerinde aramalar yapmak için sıkça kullanılır.

```
root@hackerbox:~$ grep '192.168.1.1' /var/log/apache2/access.log
192.168.1.1 - -[15/Mar/2024:10:00:00 +0000]"GET /index.html HTTP/1.1"20061
2""Mozilla/5.0 (Windows NT 10.0; Win64; x64)"
192.168.1.1 - -[15/Mar/2024:10:00:02 +0000]"POST /login.php HTTP/1.1"20045
2"http://example.com/login""Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/M
RA58N)"
192.168.1.1 - -[15/Mar/2024:10:00:03 +0000]"GET /wp-admin HTTP/1.1"40349
7""Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6)"
```

Bu komut, Apache 2 web sunucusuna ait erişim loglarının içerisinde yalnızca IP adresi 192.168.1.1 olan kayıtları ekrana yazdıracaktır.

İleri Düzey grep Parametreler

- i: Büyük/küçük harf duyarsız.
- v: Eşleşmeyenleri göster (hariç tut).
- r: Dizin içinde özyineli (recursive) arama.

Wc

wc (word count) komutu, dosyaların ne kadar büyük olduğunu veya ne kadar veri içerdığını hızlıca anlamanızı sağlar.

```
root@hackerbox:~$ wc /etc/passwd
46672544 /etc/passwd
```

Yukarıdaki örnekte wc komutu, Linux işletim sistemlerindeki kayıtlı kullanıcıların listesini içeren /etc/passwd dosyasının sırasıyla satır, kelime ve toplam karakter sayısını getirmiştir.

Sütun değeri	Açıklama
46	Satır sayısı
67	Kelime sayısı
2544	Karakter sayısı
/etc/passwd	Dosya yolu

Cut

Satırları böler ve istediğiniz sütunu alır.

/etc/passwd dosyası buna en iyi örnektir.

```
user@hackerbox:~$cut -d":" -f1 /etc/passwd | head -n3
root
daemon
bin
```

Awk

awk komutu metin ve veri işleme görevleri için tasarlanmıştır ve özellikle sütun bazlı verilerle çalışırken oldukça etkilidir. Dosyaları satır satır okuyup, her satırı alanlara (sütunlara) ayırır ve belirtilen koşullara göre işlem yapar. awk, karmaşık metin işlemleri için çok sayıda fonksiyon ve kontrol yapıları sunar.

```
root@hackerbox:~$cat names.txt
John Doe
Emily Clark
Alex Turner
```

```
root@hackerbox:~$awk'{print $1}' names.txt
John
Emily
Alex
```

Bu örnekte, names.txt adlı bir dosya içerisinde üç isim-soyisim çifti bulunmaktadır: John Doe, Emily Clark ve Alex Turner. awk '{print \$1}' names.txt komutu, awk programını kullanarak bu dosyanın içeriğini işler. awk, metin dosyalarını satır satır okuyarak her satırı boşluk veya tab karakterlerine göre alanlara ayırır. Bu örnekte {print \$1} ifadesi, her satırın ilk alanını (yani ismi) yazdırması talimatını verir.

Gelişmiş Örnek: Süreç Listesi

ps aux çıktısından sadece Kullanıcı Adı (\$1) ve Komut (\$11) sütunlarını alalım.

```
user@hackerbox:~$ ps aux | awk '{print $1, $11}' | head -n3
USER COMMAND
root /sbin/init
root [kthreadd]
```

Sed

sed (stream editor) komutu metinleri işlemek, değiştirmek, eklemek, silmek veya dosyalar arasında yer değiştirmek gibi çeşitli metin düzenlemeleri yapabilen bir araçtır.

sed komutu, genellikle metinleri filtrelemek ve dönüştürmek için kullanılır.

```
root@hackerbox:~$cat names.txt
Alice
Charlie
Bob

root@hackerbox:~$sed's/Alice/George/' names.txt
George
Charlie
Bob
```

Yukarıdaki örnekte, names.txt içerisinde bulunan

Alice ismi, sed komutu yardımıyla **George** olarak değiştirilmiştir. Ancak, sed komutunun dosyada değişiklik yapmayı, yalnızca ekrana yeni yapılan değişikliği yazdırmıştır. Dosayı kalıcı değiştirmek için -i parametresi kullanılır.

Tee

Çıktıyı hem ekrana basar hem dosyaya yazar.

```
user@hackerbox:~$ echo "Log Kaydı" | tee log.txt  
Log Kaydı
```

Diff

İki dosya arasındaki farkları gösterir.

```
user@hackerbox:~$ diff dosya1.txt dosya2.txt  
< Eski satır  
> Yeni satır
```

Tr

Karakter değişimi yapar.

Örnek: Küçük harfleri büyük yapalım

```
user@hackerbox:~$ echo "merhaba" | tr "a-z""A-Z"  
MERHABA
```

İzinler

İzinleri Okuma

Linux dosya sisteminde her dosya ve dizinin, sistemdeki kullanıcılar tarafından nasıl erişilebileceğini belirleyen izinleri vardır. Bu izinler, dosyanın güvenliğini ve bütünlüğünü sağlamak için kritiktir. İzin yapısını anlamak için

ls -l komutunun çıktısını detaylıca incelememiz gereklidir.

```
root@hackerbox:~$ ls -l
```

```
-rwxr--r--2 john development 4096 Jul29 12:34 notes.txt
```

ls -l çıktısı örneği: -rwxr--r--

Bu ifade, aslında bir dizi karakterden oluşur ve her karakterin özel bir anlamı vardır.

- İ parametresi ile elde ettiğimiz çıktıdaki sütunların açıklamaları aşağıdaki gibidir:

Sütun içeriği	Açıklama
d	Dosya türü. Eğer dizin ise d, dosya ise - karakteri ile gösterilir. Bu örnekte d karakteri olduğu için bu bir dizin .
rwxr--r--	Dosya izinleri
2	Dosyaya/dizine verilen sabit bağlantı (hard link) sayısı
john	Dosya/dizinin sahibi olan kullanıcı
development	Dosya/dizinin sahibi olan grup
4096	Dosyanın boyutu veya dizin bilgilerini saklamak için kullanılan blok sayısı
Jul 29 12:34	Dosya/dizinin oluşturulma veya son düzenlenme tarihi
notex.txt	Dosya/dizinin ismi

r, w, x ve - Karakterleri

İzinler temel olarak üç farklı yetki türü ve bir de yetkisizlik durumu ile ifade edilir:

- r (Read)**: Dosyanın içeriğini **okuma** iznini ifade eder. Eğer bu izin bir dizin (dizin) için verilmişse, o dizinin içindeki dosyaları listeleme (ls) yetkisi verir.
- w (Write)**: Dosyanın içeriğine **yazma** veya dosyayı düzenleyebilme iznini ifade eder. Bir dosya üzerinde değişiklik yapmak, içeriğini silmek veya üzerine yazmak için bu izne ihtiyaç vardır. Dizinler için ise, o dizinin içine yeni dosya oluşturma veya silme yetkisi verir.
- x (Execute)**: Dosyayı **çalıştırabilme** iznini ifade eder. Bu izin, genellikle scriptler (komut dosyaları) veya derlenmiş programlar için verilir. Eğer bir dosya çalıştırılabilir bir programsa ancak x izni yoksa, sistem bu dosyanın çalışmasına izin vermez. Dizinler için ise, o dizinin içine erişim (cd ile girme) yetkisi verir.

- **(Tire):** Eğer r, w veya x karakterlerinden herhangi biri yerine - yazılmışsa, o pozisyondaki izin **verilmemiş** demektir.

İzinleri Değiştirme (chmod)

Örnek Senaryo: Bir scripti çalıştırılabilir yapma

script.sh için çalışma izni ekleme (755):

```
user@hackerbox:~$ ls -l script.sh
-rw-r--r--1 user user0 Aug0112:00 script.sh
user@hackerbox:~$ chmod755 script.sh
user@hackerbox:~$ ls -l script.sh
-rwxr-xr-x1 user user0 Aug0112:00 script.sh
```

Herkese tam yetki (777) — güvensizdir:

```
user@hackerbox:~$ chmod777 dosya.txt
user@hackerbox:~$ ls -l dosya.txt
-rwxrwxrwx1 user user0 Aug0112:00 dosya.txt
```

Sadece sahip okur/yazar (600):

```
user@hackerbox:~$ chmod600 ozel.txt
user@hackerbox:~$ ls -l ozel.txt
-rw-----1 user user0 Aug0112:00 ozel.txt
```

notes.txt dosyasında diğer kullanıcılarla (o) yazma izni ekleyelim:

```
root@hackerbox:~$ chmod o+w notes.txt
root@hackerbox:~$ ls -l notes.txt
-rw-rw-rw-1 john development4096 Jul2912:34 notes.txt
```

Tek seferde tüm gruplara tüm izinleri vermek isterseniz:

```
root@hackerbox:~$ chmod ugo+rwx notes.txt
root@hackerbox:~$ ls -l notes.txt
```

```
-rwxrwxrwx1 john development4096 Jul29 12:34 notes.txt
```

Dosya Özellikleri (chattr ve Isattr)

İzinlerden daha güçlü koruma sağlar. Bir dosyayı **değiştirilemez (immutable)** yapmak için kullanılır.

- chattr +i dosya: Dosyayı kilitler. Root bile silemez.
- Isattr dosya: Özellikleri gösterir.

```
user@hackerbox:~$ sudo chattr +i kritik_dosya.conf
user@hackerbox:~$ rm kritik_dosya.conf
rm: cannot remove 'kritik_dosya.conf': Operation not permitted
```

(Kilidi açmak için sudo chattr -i kullanılır).

Process Yönetimi

Process Nedir ve Nasıl Çalışır?

Linux işletim sisteminde, o an çalışan her programa veya komuta **Process (Süreç)** denir. Siz bir programa (örneğin Firefox'a) tıkladığınızda veya terminale bir komut (örneğin ls) yazdığında, işletim sistemi bu işlem için bellekte bir yer ayırır ve ona benzersiz bir kimlik numarası (PID) verir.

Process'lerin Temel Özellikleri:

- **PID (Process ID):** Her process'in TC Kimlik numarası gibi benzersiz bir numarası vardır.
- **Sahip (User):** Process'i başlatan kullanıcıdır.
- **Ebeveyn (Parent):** Bir process, başka bir process tarafından başlatılabilir. (Örneğin terminalden Firefox açarsanız, terminal "baba", Firefox "çocuk" process olur).

Process'leri yönetmek, sistemin performansını kontrol etmek ve kilitlenen programları kapatmak için hayatı önem taşır.

1. Süreç Türleri

Ön Plan (Foreground) Process'leri

Varsayılan olarak tüm process'ler ön planda yürütülür. Klavyeden girdi alırlar ve ekrana çıktı sağlarlar.

pwd komutunu çalıştırırmak buna iyi bir örnektir.

```
user@hackerbox:~$pwd  
/home/user
```

Örnekte, *pwd* komutu tarafından yürütülen process ön planda çalışarak çıktı sağladı ve görevini yerine getirdikten sonra çıkış yaptı. Ön planda çalışan process'ler görevini yerine getirene kadar terminali kilitleyip, başka bir işlem yapmaya izin vermezler.

Arka Plan (Background) Process'leri

Arka plan process'leri, çalıştırıldığında terminali kitlemez ve arka planda çalışmaya başlar. Arka plan process'leri çalıştırıldığında, paralel olarak bir çok process çalıştırılabilir.

Bir komutu arka planda çalıştırırmak istersek sonuna & karakteri ekleyebiliriz.

```
user@hackerbox:~$ping127.0.0.1 &  
[1]54017
```

[1] ifadesi, process'in iş (job) numarasının 1 olduğunu belirtiyor. 54017 ise process'in arka planındaki PID (Process ID) değeridir.

2. Mevcut Terminal Oturumunda Çalışan Process'lerin Yönetimi

Arka planda çalışan işleri listelemek için *jobs* kullanılır.

```
user@hackerbox:~$jobs  
[1] running ping127.0.0.1
```

Arka planda çalışan bu komutu tekrar ön plana almak için fg (foreground) komutunu kullanabiliriz.

```
user@hackerbox:~$fg %1
ping127.0.0.1
```

Durdurmak için CTRL+C kullanabiliriz veya arka plana atıp duraklatmak için CTRL+Z kullanabiliriz.

3. Sistem Genelinde Çalışan Process'lerin Yönetimi (ps, top)

Sadece mevcut terminal oturumumuzda değil, tüm sistemde çalışan process'ler de bulunmaktadır.

ps aux : Tüm process'lerin anlık fotoğrafını çeker.

```
user@hackerbox:~$ ps aux | head -n10
USER      PID %CPU %MEM   VSZ   RSS TTY      STAT START  TIME COMM
AND
root10.00.116843610240 ?    Ss08:000:02 /sbin/init
root20.00.000 ?    S08:000:00 [kthreadd]
root12340.00.512345623456 ?    Sl08:050:10 /usr/sbin/apache2
syslog8500.00.02201005000 ?    Ssl08:000:01 /usr/sbin/rsyslogd -n
message+8550.00.185004200 ?    Ss08:000:01 /usr/bin/dbus-daemon --sy
stem
root9500.00.1150006500 ?    Ss08:000:00 /usr/sbin/sshd -D
mehmet54321.53.256789065432 pts/0  R+10:300:05 python3 script.py
mehmet54330.00.190004000 pts/0  S+10:310:00 tail -f log.txt
root60000.12.040000050000 ?    S09:000:15 /usr/bin/containerd
```

Sütunların Anlamları:

Sütun	Açıklama
USER	Process'i başlatan kullanıcı.
PID	Process ID - process'in benzersiz kimlik numarası.

Sütun	Açıklama
%CPU	İşlemci kullanım oranı.
%MEM	Bellek (RAM) kullanım oranı.
STAT	Durum (R: Çalışıyor, S: Uyuyor, Z: Zombi).
START	Başlangıç zamanı.
COMMAND	Çalıştırılan komut.

top : Sistemdeki process'leri canlı olarak izlemek için kullanılır. (Çıkmak için q).

```
top -14:30:00 up10 days,4:20,1 user, load average:0.05,0.10,0.05
Tasks:123 total,1 running,122 sleeping,0 stopped,0 zombie
%Cpu(s):1.0 us,0.5 sy,0.0 ni,98.5 id,0.0 wa,0.0 hi,0.0 si,0.0 st
MiB Mem :3900.0 total,1500.0 free,1200.0 used,1200.0 buff/cache
```

4. Çalışan Process'i Durdurmak (kill)

Çalışan process'leri durdurmak için

kill komutu kullanılır. PID numarasına ihtiyaç duyar.

```
user@hackerbox:~$kill5432
```

Süreç Sinyalleri (Process Signals): Bir process'i durdurmak için ona sinyal göndeririz.

kill varsayılan olarak 15 numaralı sinyali gönderir.

Sinyal No	Ad	Anlamı
15	SIGTERM	Nazikçe kapan. (Varsayılan). Dosyaları kaydetmeye izin verir.
9	SIGKILL	Derhal ölü. Kaydetmeye izin vermez. Zorla kapatır.
1	SIGHUP	Yeniden başla (Config dosyasını tekrar oku).
2	SIGINT	Kesme sinyali. Klavyeden CTRL+C tuşuna basıldığında gönderilir.
19	SIGSTOP	Process'i duraklatır (Pause). CTRL+Z ile gönderilir.

Sinyal No	Ad	Anlamı
18	SIGCONT	Duraklatılmış process'i devam ettirir (Continue).

Eğer process normal kill ile kapanmıyorsa, zorla kapatmak için -9 kullanılır:

```
user@hackerbox:~$kill -9 5432
```

İsmine Göre Öldürme (pkill, killall): PID aramadan ismiyle kapatmak için:

```
user@hackerbox:~$pkill python
user@hackerbox:~$killall firefox
```

5. Öncelik Ayarı (nice, renice)

Linux'ta process'lerin önceliği vardır (-20 en yüksek, 19 en düşük). Varsayılan 0'dır.

- **nice:** Programı başlatırken öncelik verir.

```
user@hackerbox:~$nice -n10 tar -czf yedek.tar.gz /home
```

(Yedekleme işlemi düşük öncelikle çalışın, bilgisayarı yormasın).

- **renice:** Çalışan process'in önceliğini değiştirir.

```
user@hackerbox:~$renice -n-5 -p5432
```

6. Servis Yönetimi (systemd)

Modern Linux sistemlerinde (Ubuntu, CentOS vb.) arka plan servislerini systemd yönetir. Komut aracı systemctl'dir.

Örnek: Nginx Servisi

Durumunu gör:

```
user@hackerbox:~$ systemctl status nginx
●nginx.service - A high performance web server
```

```
Active: active (running) since Mon2023-10-01 ...
```

Başlat / Durdur:

```
sudo systemctl start nginx  
sudo systemctl stop nginx
```

Otomatik Başlat (Bilgisayar açılıncı):

```
sudo systemctl enable nginx
```

Ağ Yönetimi

1. Ağ Arayüzü Konfigürasyonu (ifconfig ve ip)

Ağ Nedir ve Temel Kavramlar

Ağ yönetimine geçmeden önce temel kavramları anlamak önemlidir:

- **IP Adresi (Internet Protocol Address):** Bilgisayarların ağ üzerinde birbirini bulmasını sağlayan kimlik numarasıdır (Örn: 192.168.1.50).
- **MAC Adresi (Media Access Control):** Ağ kartının donanımsal, değişmeyen fiziksel adresidir (Örn: 00:1A:2B:3C:4D:5E).
- **Ağ Arayüzü (Network Interface):** Bilgisayarın ağa bağlı olduğu donanım veya yazılım kapısıdır (Örn: eth0 kablolu, wlan0 kablosuz).
- **Gateway (Ağ Geçidi):** Yerel ağdan çıkıştıktan sonra internete gitmek için kullanılan çıkış kapısıdır (Genellikle modem IP'si).

Mevcut Cihazları Listelemek

ifconfig komutu parametresiz çağrııldığında mevcut ağ cihazlarını (NIC) ve detaylı istatistiklerini listeler.

```
root@hackerbox:~$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu1500
```

```
inet192.168.1.50 netmask255.255.255.0 broadcast192.168.1.255
inet6 fe80::a00:27ff:fe4e:66a1 prefixlen64 scopeid0x20<link>
ether08:00:27:4e:66:a1 txqueuelen1000 (Ethernet)
RX packets10542 bytes1252144 (1.2 MiB)
RX errors0 dropped0 overruns0 frame0
TX packets8475 bytes8213607 (7.8 MiB)
TX errors0 dropped0 overruns0 carrier0 collisions0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu65536
inet127.0.0.1 netmask255.0.0.0
inet6 ::1 prefixlen128 scopeid0x10<host>
loop txqueuelen1000 (Local Loopback)
RX packets152 bytes11200 (10.9 KiB)
RX errors0 dropped0 overruns0 frame0
TX packets152 bytes11200 (10.9 KiB)
TX errors0 dropped0 overruns0 carrier0 collisions0
```

Yukarıdaki çıktıda *eth0* ethernet kartımızı, *lo* ise yerel (loopback) arayüzümüzü gösterir. Çıktıdaki *inet* IP adresimizi, *ether* MAC adresimizi, RX alınan paketleri, TX gönderilen paketleri gösterir.

DOWN durumda olan, yani aktif olmayan arayüzleri de görüntülemek için *a* parametresini kullanabiliriz.

```
root@hackerbox:~$ ifconfig -a

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu1500
inet172.20.1.109 netmask255.255.255.0 broadcast172.20.1.255
inet6 fe80::5054:ff:fe10:72c3 prefixlen64 scopeid0x20<link>
ether52:54:00:10:72:c3 txqueuelen1000 (Ethernet)
RX packets4542 bytes352144 (343.8 KiB)
RX errors2 dropped0 overruns0 frame2
TX packets1475 bytes6213607 (5.9 MiB)
TX errors0 dropped0 overruns0 carrier0 collisions0
device interrupt11 memory0xfc840000-fc860000
```

```
lo: flags=73<UP,LOOPBACK,RUNNING> mtu65536
inet127.0.0.1 netmask255.0.0.0
inet6 ::1 prefixlen128 scopeid0x10<host>
    loop txqueuelen1000 (Local Loopback)
        RX packets16 bytes1888 (1.8 KiB)
        RX errors0 dropped0 overruns0 frame0
        TX packets16 bytes1888 (1.8 KiB)
        TX errors0 dropped0 overruns0 carrier0 collisions0

eth1: flags=4099<UP,BROADCAST,MULTICAST> mtu1500
inet0.0.0.0 netmask0.0.0.0 broadcast0.0.0.0
ether52:54:00:ab:cd:ef txqueuelen1000 (Ethernet)
    RX packets0 bytes0 (0.0 B)
    RX errors0 dropped0 overruns0 frame0
    TX packets0 bytes0 (0.0 B)
    TX errors0 dropped0 overruns0 carrier0 collisions0
```

a sayesinde, aşağıdaki gibi DOWN durumda (IP almamış) arayüzler de listelenir. Örnekte eth1 aktif değilken yine de çıktıda görünüyor.

Modern Yöntem (ip):

```
user@hackerbox:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu65536 qdisc noqueue state UNKNOWN
    N group default qlen1000
    link/loopback00:00:00:00:00:00 brd00:00:00:00:00:00
    inet127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu1500 qdisc fq_codel s
    tate UP group default qlen1000
    link/ether08:00:27:4e:66:a1 brd ff:ff:ff:ff:ff:ff
    inet192.168.1.50/24 brd192.168.1.255 scope global dynamic eth0
        valid_lft86245sec preferred_lft86245sec
```

```
inet6 fe80::a00:27ff:fe4e:66a1/64 scope link  
  valid_lft forever preferred_lft forever
```

Arayüz Açıp Kapatmak

Bir ağ kartını (örn:eth0) devre dışı bırakmak veya açmak için:

Eski (ifconfig):

```
root@hackerbox:~$ifconfig eth0 down  
root@hackerbox:~$ifconfig eth0 up
```

Modern (ip):

```
user@hackerbox:~$sudo ip link set eth0 down  
user@hackerbox:~$sudo ip link set eth0 up
```

IP Adresi Atamak

Geçici olarak IP adresi atamak için:

Eski (ifconfig):

```
root@hackerbox:~$ ifconfig eth0 172.20.1.110 netmask 255.255.255.0
```

Modern (ip):

```
user@hackerbox:~$ sudo ip addr add 172.20.1.110/24 dev eth0
```

Promiscuous Mode (Trafiği İzleme Modu)

Eğer ethernet kartınız destekliyorsa ağ arayüzüne gelen ancak sizi ilgilendirmeyen paketleri de CPU'ya gönderip işlemenize olanak sağlayabilirsiniz. Böylece ağınzıdaki sizinle alakalı olmayan trafiği de izleyebilirsiniz.

Ağdaki tüm paketleri dinlemek için (Sniffing):

- **Eski:** ifconfig eth0 promisc
- **Modern:** ip link set eth0 promisc on

MAC Adresini Değiştirmek

Cihazınızın MAC adresini değiştirebilirsiniz. Ağdaki ARP tablolarının karışmasına sebep olabilir, bu yüzden dikkatli kullanmanızda fayda var.

```
root@hackerbox:~$ ifconfig eth0 hw etherAA:BB:CC:DD:EE:FF
```

2. Port ve Bağlantı Kontrolü (netstat ve ss)

Sistemde hangi portların açık olduğunu ve kimin dinlediğini görmek güvenlik için kritiktir.

Eski Yöntem (netstat):

```
root@hackerbox:~$ netstat -tulpn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State      PID/
Program name
tcp000.0.0.0:220.0.0.0:*      LISTEN950/sshd
tcp00127.0.0.1:54320.0.0.0:*    LISTEN800/postgres
udp000.0.0.0:680.0.0.0:*700/dhclient
```

Burada sshdservisinin 22. portu dinlediğini (LISTEN) görüyoruz.

Modern Yöntem (ss):

```
user@hackerbox:~$ sudo ss -tulpn
Netid State  Recv-Q Send-Q Local Address:Port  Peer Address:Port Process
tcp   LISTEN01280.0.0.0:220.0.0.0:*    users:(("sshd",pid=950,fd=3))
tcp   LISTEN0128127.0.0.1:54320.0.0.0:*  users:(("postgres",pid=800,fd=
3))
```

- t: TCP
- u: UDP
- l: Listening (Dinleyenler)
- p: Process (Hangi program)

- n: Numeric (İsim yerine sayı)

3. Yönlendirme Tablosu (Routing Table)

İnternete çıkış kapısını (Gateway) görmek için:

Eski (route):

```
root@hackerbox:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref  Use Iface
0.0.0.0         192.168.1.10   0.0.0.0       UG10000 eth0
          192.168.1.00.0.0.0255 255.255.0   U10000 eth0
```

Modern (ip route):

```
user@hackerbox:~$ ip route
default via192.168.1.1 dev eth0 proto dhcp src192.168.1.50 metric100
192.168.1.0/24 dev eth0 proto kernel scope link src192.168.1.50
```

4. Bağlantı Testi (ping, nc, tracepath)

- **ping**: Sunucuya erişimi test eder.

```
user@hackerbox:~$ ping -c3 google.com
PING google.com (142.250.187.206)56(84) bytes of data.
64bytes from ...: icmp_seq=1 ttl=116 time=14.2 ms
```

- **nc (Netcat)**: Spesifik bir portun açık olup olmadığını test eder (Telnet yerine).

```
user@hackerbox:~$ nc -zv192.168.1.5022
Connection to192.168.1.5022 port [tcp/ssh] succeeded!
```

- **tracepath**: Paketin hedefe giderken hangi routelardan geçtiğini gösterir.

```
user@hackerbox:~$ tracepath google.com
1?: [LOCALHOST]                                pmtu 1500
1: gateway (192.168.1.1)                      2.112ms
2: 10.0.0.1                                     10.200ms
```

5. DNS Sorgulama (dig, nslookup)

Alan adının IP adresini bulur.

dig Kullanımı (Detaylı):

```
user@hackerbox:~$ dig google.com +short
142.250.187.206
```

nslookup

Kullanımı (Basit):

```
user@hackerbox:~$ nslookup google.com
Server:127.0.0.53
Non-authoritativeanswer:
Name: google.com
Address:142.250.187.206
```

6. DNS Ayarları

Linux'ta DNS ayarları /etc/resolv.conf dosyası içerisinde yer almaktadır. nano gibi bir metin editörü ile bu dosya içerisindeki DNS ayarlarını güncelleyebiliriz.

```
root@hackerbox:~$nano /etc/resolv.conf
```

Dosya içeriği aşağıdaki gibi olacaktır:

```
nameserver172.20.1.1
```

Kullanmak istediğimiz DNS sunucularını satır satır bu formatta dosya içeresine yazabiliris. Örneğin, Cloudflare'in sağladığı DNS sunucularını tüm sistemimizde kullanmak için dosyayı aşağıdaki şekilde güncellemeliyiz:

```
nameserver1.1.1  
nameserver1.0.0.1
```

7. Dosya Transferi

- **wget**: Dosya indirmek için.

```
user@hackerbox:~$wgethttps://example.com/dosya.zip
```

- **curl**: Web isteği atmak ve başlıklarını görmek için.

```
user@hackerbox:~$curl -Ihttps://google.com  
HTTP/1.1200 OK
```

- **scp**: SSH üzerinden güvenli dosya kopyalamak için.

```
user@hackerbox:~$scp rapor.pdf user@192.168.1.50:/home/user/ Belgeler
```

8. SSH (Secure Shell)

SSH, ağ üzerinden başka bir bilgisayara güvenli bir şekilde bağlanmak ve komutlar çalıştırılmak için kullanılan bir protokoldür. SSH, özellikle uzaktaki bilgisayarlara erişim ve yönetim için yaygın olarak kullanılır. SSH bağlantısı yaparken, 'ssh' komutu kullanılır.

SSH Servisini Kurmak ve Başlatmak

Öncelikle, SSH servisini kurmanız gerekebilir. Debian tabanlı bir sistemde, openssh-server paketini aşağıdaki komut ile kurabilirsiniz:

```
sudo apt-getupdate
```

```
sudo apt-get install openssh-server
```

Kurulum tamamlandıktan sonra servisi başlatabilirsiniz:

```
sudo systemctl start ssh
```

SSH servisinin sistem açıldığında otomatik olarak başlamasını sağlamak için:

```
sudo systemctl enable ssh
```

SSH İle Uzaktaki Bir Sunucuya Bağlanmak

Uzaktaki bir sunucuya bağlanmak için ssh komutunu şu şekilde kullanabilirsiniz:

```
ssh user@ip_address
```

Örneğin, kullanıcı adınız root ve sunucu adresiniz 192.168.1.100 ise:

```
ssh root@192.168.1.100
```

Bu komutu çalıştırdıktan sonra, uzaktaki sunucunun parolasını girmeniz istenecektir.

SSH Anahtar Çifti Oluşturmak

Parola tabanlı oturum açma yöntemine ek olarak, SSH anahtar çifti kullanarak parolasız (ve daha güvenli) bir şekilde bağlanabilirsiniz. SSH anahtarı oluşturmak için

ssh-keygen komutunu kullanabilirsiniz:

```
ssh-keygen
```

Bu komutu çalıştırdıktan sonra, oluşturulan açık anahtarı (public key) uzaktaki sunucuya kopyalamanız gerekecektir:

```
ssh-copy-id user@ip_address
```

Örneğin:

```
ssh-copy-id root@192.168.1.100
```

Bu işlem tamamlandıktan sonra, parola girmeden SSH ile bağlanabilirisiniz.

SSH Yapılandırma Dosyası

SSH yapılandırma ayarları genelde /etc/ssh/sshd_config dosyasında bulunur. Bu dosya üzerinde çeşitli SSH ayarlarını yapabilirsiniz. Örneğin, SSH portunu değiştirmek, root oturumlarını kapatmak gibi yapılandırmalar bu dosya üzerinde yapılabilir:

```
sudo nano/etc/ssh/sshd_config
```

Dosya içerisinde, Port ayarını bulup düzenleyerek port numarasını değiştirebilirisiniz:

```
Port2222
```

Değişiklikleri yaptıktan sonra SSH servisini yeniden başlatmanız gerekecektir:

```
sudo systemctl restart ssh
```

 **Ömer Faruk BAYSAL**