İşletim Sistemi Kurulumu sonrası istenilen ayarlar/ detaylandırmalar

-> Subject'te istenilen birim oluşturma işlemi doğru şekilde olduğunu kontrol etmek için "lsblk" komutunu kullanarak elde ettiğimiz çıktıyı inceliyoruz.

LVM kullanarak en az 2 tane şifrelenmiş bölüm oluşturmalısınız. Aşağıda sizden beklenen bölümlemeye bir örnek gösterilmiştir.

```
akaraca@akaraca42:/$ lsblk
NAME                    MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda                       8:0    0 30.8G  0 disk
├─sda1                    8:1    0  476M  0 part  /boot
├─sda2                    8:2    0    1K  0 part
└─sda5                    8:5    0 30.3G  0 part
  └─sda5_crypt          254:0    0 30.3G  0 crypt
    ├─LVMGroup-root     254:1    0  9.3G  0 lvm   /
    ├─LVMGroup-swap     254:2    0  2.1G  0 lvm   [SWAP]
    ├─LVMGroup-home     254:3    0  4.7G  0 lvm   /home
    ├─LVMGroup-var      254:4    0  2.8G  0 lvm   /var
    ├─LVMGroup-srv      254:5    0  2.8G  0 lvm   /srv
    ├─LVMGroup-tmp      254:6    0  2.8G  0 lvm   /tmp
    └─LVMGroup-var--log 254:7    0  3.7G  0 lvm   /var/log
sr0                      11:0    1 1024M  0 rom
```

-> Subject'te bahsedildiği gibi giriş bilgisi(örn: akaraca) ile oluşturulan kullanıcı yani yönetici olmayan kullanıcı hesabımızı kök(root) kullanıcı gibi düzenlememiz gerektiğini belirtmiş ve bu kullanıcımız sudo ve user42 grubunda olmalıdır.

- Kök kullanıcıya ek olarak, kullanıcı adı giriş bilgileriniz olan bir kullanıcı olması gerekmektedir.
- Bu kullanıcı **user42** ve **sudo** grupları altında olmalıdır.

-> Sunucumuzda yönetici olmayan kullanıcı hesabı üzerinden giriş yaparsak, sunucu üzerinde izin isteyen işlemleri yerine getiremeyiz. Bu yüzden root hesabına geçiş yaptıktan sonra "apt-get Install sudo" komutu ile sudo(super user do) yani başka bir kullanıcı olarak root kullanıcısı gibi işlemlerin yürütülmesine izin veren bir Linux komutunu sunucuyu kurmuş oluruz.

```
akaraca@akaraca42:/$ apt-get install sudo
E: Could not open lock file /var/lib/dpkg/lock-frontend - open (13: Permission denied)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontend), are you root?
akaraca@akaraca42:/$ sudo apt-get install sudo
-bash: sudo: command not found
```

-> root hesabına geçiş yapmak için komut satırına “su” veya “su -“ komutunu yazın. İstenilen password, sunucu oluştururken bizlerin root hesabı için belirlediğimiz şifredir.

```
akaraca@akaraca42:/$ su
Password:
root@akaraca42:/#
```

-> root hesabına geçiş yaptığımızdan dolayı artık izin almadan istenilen paketleri yükleyebiliriz. Şimdilik sudo paketini kuralım.

```
root@akaraca42:/# apt-get install sudo
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  sudo
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 1,059 kB of archives.
After this operation, 4,699 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian bullseye/main amd64 sudo amd64 1.9.5p2-3 [1,059 kB]
Fetched 1,059 kB in 0s (2,627 kB/s)
Selecting previously unselected package sudo.
(Reading database ... 22302 files and directories currently installed.)
Preparing to unpack .../sudo_1.9.5p2-3_amd64.deb ...
Unpacking sudo (1.9.5p2-3) ...
Setting up sudo (1.9.5p2-3) ...
```

-> Yönetici olmayan kullanıcı hesabımızdan devam etmek istiyorsak öncelikle yapılması gereken izin oluşturma ayarları bulunmaktadır.

getent nedir?; getent, kullanıcının veritabanları adı verilen bir dizi önemli metin dosyasına girdi almasına yardımcı olan bir Unix komutudur. Bu, kullanıcı bilgilerini depolayan passwd ve grup veritabanlarını içerir - bu nedenle getent, Unix'te kullanıcı ayrıntılarını aramanın yaygın bir yoludur.

“getent group sudo” komutu ile kullanıcı bilgilerinde group veritabanındaki sudo grubunun içeriğini çeker. Şu anki çıktıya göre herhangi bir kullanıcı sudo grubu altında değil.

```
root@akaraca42:/# getent group sudo
sudo:x:27:
```

-> Sudo grubuna kullanıcı eklemek için; “sudo adduser akaraca sudo” komutunu

giriyoruz bu komut, sudo grubuna akaraca kullanıcısını ekle anlamına geliyor. Komut satırının başında yer alan sudo ise “adduser”’in bir komut olduğunu belirtmesi için kullanıyoruz.

```
root@akaraca42:/# sudo adduser akaraca sudo
Adding user `akaraca' to group `sudo' ...
Adding user akaraca to group sudo
Done.
```

```
root@akaraca42:/# getent group sudo
sudo:x:27:akaraca
```

-> Kullanıcı hesabımızın user42 grubunda da olması istendiğinden dolayı, “sudo groupadd user42” ile user42 grubu oluşturulur.

```
akaraca@akaraca42:/$ sudo groupadd user42_
```

->user42 grubuna akaraca kullanıcısını “sudo adduser akaraca user42” ile ekliyoruz.

```
akaraca@akaraca42:/$ sudo adduser akaraca user42
Adding user `akaraca' to group `user42' ...
Adding user akaraca to group user42
Done.
```

-> Kullanıcı hesabımızı ekledik lakin, güncelleme durumu olmadığından dolayı sisteme reboot atıyoruz. Reboot sonrası görüldüğü gibi akaraca kullanıcımız sudo ve user42 gruplarına eklenmiştir.

```
akaraca@akaraca42:~$ groups
akaraca cdrom floppy sudo audio dip video plugdev netdev user42
```

-> Kullanıcı hesabımızın “sudo” komutunu kullanma yetkisinin olup olmadığı bu komut ile görebiliriz.
-> Kullanıcı hesabımızı kök kullanıcı gibi düzenlemek için; “sudo visudo” komutu ile eriştiğimiz belgeye root satırı altına akaraca ALL=(ALL:ALL) ALL satırı eklenmelidir.

-> Neden sadece visudo komutu kullanılır? /etc/sudoers klasörü düzenlenmesi tavsiye edilmez.

Sadece visudo kullanmak daha güvenlidir. /etc/sudoers'ı doğrudan düzenleyebilirsiniz, ancak orada bir yazım hatası yaparsanız, artık sudo kullanamazsınız. Ve hatanızı düzeltemeyeceksiniz. visudo, sudoers dosyasını

birden çok eşzamanlı düzenlemeye karşı kilitler, temel akıl kontrolleri sağlar ve ayrıştırma hatalarını kontrol eder.

```
root@akaraca42:/# sudo visudo_
```

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
akaraca ALL=(ALL:ALL) ALL
```

-> Sunucuda kök kullanıcı harici olan bir kullanıcıda “sudo” komutunu ilk kez kullandığımızda bizlerden bir şifre isteyecektir. Bu şifre kullanıcı hesabımızın oturum şifresidir. Subject bizlerden bu şifre için düzenlemeler istemektedir.

```
akaraca@akaraca42:/$ sudo visudo

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for akaraca: _
```

`sudo` grubunuz için güçlü bir yapılandırma ayarlamak için aşağıdaki gereksinimlere uymanız gerekir:

- `sudo` ile yetkilendirme 3 yanlış parola denemesi ile sınırlandırılmalıdır.
- `sudo` kullanırken yanlış şifre sebebiyle bir hata meydana gelirse seçtiğiniz özel bir mesaj gösterilmelidir.
- `sudo` kullanırken yapılan her işlem (tüm girdi ve çıktılar) kayıt altında tutulmalıdır. Kayıtların tutulduğu log dosyası `/var/log/sudo/` klasörüne kaydedilmelidir.
- Güvenlik sebepleriyle `TTY` modu aktif hale getirilmelidir.
- Yine güvenlik sebebiyle, `sudo` tarafından kullanılan dizinler sınırlandırılmalıdır. Örnek olarak: `/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin`

-> Subject’te istenilen ayarlamalar aşağıdaki gibidir. Bunların detayları;
env_reset : Tüm potansiyel olarak tehlikeli ortam değişkenlerini kara listeye almak mümkün olmadığından, varsayılan env_reset davranışının kullanılması teşvik edilir. İşletim sisteminin kurulumunda standart olarak bulunan bir ayardır.

mail_badpass: Standart bir flag’dir.

Secure_path=“ “: Secure_path değeri ayarlandıysa, sudo kullanarak çalıştırdığınız komutlar için PATH ortam değişkeni olarak kullanılacaktır. Bunun anlamı, örneğin çalıştırdığınızda sudo apt update , sistem apt komutunu Secure_path içinde belirtilen dizinlerde belirtilen sırayla aramaya çalışacaktır.

badpass_message=“ “: Sudo komutu dahilinde girilen şifrenin yanlış olması durumunda ekrana çıktısı olacak olan mesajdır.

passwd_tries=3: Sudo komutu dahilinde girilen şifrenin 3 kerelik hatalı girme durumunun olduğunu belirtmektedir. Çıplak hali ile herhangi bir işlevi yoktur.

logfile=“ “; log_input ve log_output olarak verilen çıktıların ~/sudo/sudo.log dosyası içerisine kaydedileceğini belirtir.

requiretty: TTY gereksinimi, SSH'nin, parola isterken diğer birçok Unix programının yaptığı gibi, standart girdi yoluyla parola istemine yanıt olarak yönlendirme girişimlerini reddetmesine olanak tanır.

```
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
Defaults        badpass_message="Password is wrong, try again"
Defaults        passwd_tries=3
Defaults        logfile="/var/log/sudo/sudo.log"
Defaults        log_input, log_output
Defaults        requiretty
```

-> Sudo kurallarının çalışıp çalışmadığını denemek için; sunucuda henüz sudo komutu ile işlem giriş yapmadığımız durumunda deniyoruz. Görüldüğü gibi 3 kere şifre denedikten sonra işlem sonlandırılmış ve hata mesajı olarak, belirtilen çıktı oluşmuş. Ancak log dosyalarının kayıt edilebileceği bir dosya henüz oluşturulmadığından dolayı, log dizini içerisinde “sudo mkdir sudo” komutu ile sudo dosyasını oluşturuyoruz.

```
akaraca@akaraca42:/$ sudo visudo
[sudo] password for akaraca:
Password is wrong, try again
[sudo] password for akaraca:
Password is wrong, try again
[sudo] password for akaraca:
sudo: unable to open log file: /var/log/sudo/sudo.log: No such file or directory
sudo: 3 incorrect password attempts
akaraca@akaraca42:/$ cd var
akaraca@akaraca42:/var$ cd log
akaraca@akaraca42:/var/log$ cd sudo
-bash: cd: sudo: No such file or directory
```

-> Sudo dizini içerisinde sudo.log dosyası henüz oluşturulmamış. Bunun sebebi henüz yazdırılacak bir bilgi olmadığından dolayı.

```
akaraca@akaraca42:/var/log/sudo$ ls
akaraca@akaraca42:/var/log/sudo$ sudo visudo
[sudo] password for akaraca:
Password is wrong, try again
[sudo] password for akaraca:
Password is wrong, try again
[sudo] password for akaraca:
sudo: 3 incorrect password attempts
akaraca@akaraca42:/var/log/sudo$ ls
sudo.log
```

-> Sudo için istenilen şifreleme politikaları düzenlendi şimdi sırada kullanıcı için istenilen şifrelemelerin düzenlenmesinde. Dosya düzenlemesinde kolaylık olması adına vim'i kullanmayı tercih ettiğimden dolayı;

```
akaraca@akaraca42:/$ sudo apt-get install vim
```

-> Şifre kalite kontrol kitaplığı yüklenmesi tavsiye edilir, kolaylık olması adına.

```
akaraca@akaraca42:/$ sudo apt-get install libpam-pwquality
```

-> Artık şifre politikalarını belirleyebiliriz, bunun için common-password dosyası düzenlenir.

```
akaraca@akaraca42:/$ sudo vim /etc/pam.d/common-password
```

-> Eklenecek olan komutlar;
minlen=10, en az 10 karakter uzunluğu istenmektedir.
ucredit=-1, en az 1 büyük karakter istenmektedir. (Pozitif durumda Max içeriği belirler.)
lcredit=-1, en az 1 küçük karakter istenmektedir. (Pozitif durumda Max içeriği belirler.)
maxrepeat=3, 3'ten fazla art arda karakter içermemelidir.
user_check=0, şifre, kullanıcı adını içermemlidir.
difok=7, eski şifre kullanılmak isteniyorsa, eski şifreden farklı olarak en az 7 karakter içermelidir.
enforce_for_root, belirtilen kuralları root yani kök kullanıcı şifresi içinde uygula anlamına geliyor.
retry=3, standart kuraldır. Art arda 3 defa şifre giriş işlemi gerçekleştirilebilir.

- Şifreniz en az 10 karakter uzunluğunda olmalıdır. Şifre büyük ve küçük karakter içermelidir. Ayrıca 3'ten fazla art arda karakter içermemelidir.
- Şifreniz kullanıcı adını içermemelidir.

- Şifre eski şifrenin içermediği en az 7 karakter içermelidir (Bu kural kök kullanıcı için geçerli değildir.).
- Kök kullanıcı şifresi de yukarıdaki kurallara uymalıdır.

```
# here are the per-package modules (the "Primary" block)
password        requisite                       pam_pwquality.so minlen=10 ucredit=-1 lcredit=-1 max
repeat=3 user_check=0 difok=7 enforce_for_root retry=3
password        [success=1 default=ignore]      pam_unix.so obscure use_authtok try_first_pass yescr
ypt
```

-> Dahada güçlü bir şifreleme politikası için, login.defs dosyası düzenlenir.

Güçlü bir şifreleme politikası kurmak için aşağıdaki gereksinimleri sağlamalısınız:

- Şifrenin süresi her 30 günde bir dolmalıdır.
- Şifre değiştirildikten en az 2 gün sonra tekrar değiştirilebilir olmalıdır.
- Kullanıcı şifresinin süresinin dolmasına 7 gün kala bir uyarı mesajı almalıdır.

```
akaraca@akaraca42:/$ sudo vim /etc/login.defs
```

-> Şifre eskime kontrolü ayarlardı değiştirilir. Standart olarak verilen değerler istenilen değerler ile değiştirilir.

PASS_MAX_DAYS : Koyulan şifrenin 30 günlük bir süresi olduğunu belirtir.
PASS_MIN_DAYS : Koyulan şifrenin min 2 Gün sonra değiştirilebilir olduğunu belirtir.
PASS_WARN_AGE : Şifrenin dolma süresine 7 gün kala uyarı mesajı verir.

```
# Password aging controls:
#
#       PASS_MAX_DAYS   Maximum number of days a password may be used.
#       PASS_MIN_DAYS   Minimum number of days allowed between password changes.
#       PASS_WARN_AGE   Number of days warning given before a password expires.
#
PASS_MAX_DAYS   99999
PASS_MIN_DAYS   0
PASS_WARN_AGE   7
```

```
PASS_MAX_DAYS   30
PASS_MIN_DAYS   2
PASS_WARN_AGE   7
```

-> Tüm şifreleme politikaları düzenlendiğinden dolayı, bu şifreleme politikaları olmadan kurulan hesapların şifrelerini güncellememiz gerekmektedir. Güncelleme için “sudo passwd akaraca” komutunu kullanıyorum.

Konfigürasyon dosyanızı ayarladıktan sonra, kök kullanıcı dahil sanal makinedeki tüm kullanıcıların şifresini değiştirmelisiniz.

-> Şifre değişmesine rağmen, parola eskime politikaları güncellenmemiş.

```
akaraca@akaraca42:/$ chage -l akaraca
Last password change                                    : Mar 16, 2022
Password expires                                        : never
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 99999
Number of days of warning before password expires       : 7
akaraca@akaraca42:/$ sudo passwd akaraca
New password:
Retype new password:
Sorry, passwords do not match.
New password:
Retype new password:
passwd: password updated successfully
akaraca@akaraca42:/$ chage -l akaraca
Last password change                                    : Mar 17, 2022
Password expires                                        : never
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 99999
Number of days of warning before password expires       : 7
```

-> Parola eskime politikalarını güncellemek için “charge -m 2 -M 30 akaraca” komutunu kullanıyorum. Bu kurallar kök kullanıcı içinde geçerli olduğundan, kök kullanıcı içinde bu düzenleme işlemi yapılmalıdır.

```
akaraca@akaraca42:/$ sudo chage -m 2 -M 30 akaraca
akaraca@akaraca42:/$ chage -l akaraca
Last password change                                    : Mar 17, 2022
Password expires                                        : Apr 16, 2022
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 2
Maximum number of days between password change          : 30
Number of days of warning before password expires       : 7
```

```
akaraca@akaraca42:/$ sudo chage -l root
Last password change                                    : Mar 16, 2022
Password expires                                        : never
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 99999
Number of days of warning before password expires       : 7
akaraca@akaraca42:/$ sudo chage -m 2 -M 30 root
akaraca@akaraca42:/$ sudo chage -l root
Last password change                                    : Mar 16, 2022
Password expires                                        : Apr 15, 2022
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 2
Maximum number of days between password change          : 30
Number of days of warning before password expires       : 7
```

-> Şifreleme politikaları ile ilgili ayarlamalar bittiğinden dolayı artık diğer işlemlere geçebiliriz.

-> SSH veya Secure Shell, iki bilgisayarın iletişim kurmasını (http veya web sayfaları gibi köprü metni aktarmak için kullanılan protokol olan köprü metni aktarım protokolü) ve verileri paylaşmasını sağlayan bir ağ iletişim protokolüdür.
-> sunucumuzda henüz ssh kurulu olmadığından dolayı kurmak için; “apt-get Install openssh-server” komutu kullanılır.

SSH servisi sadece 4242 portu üzerinde çalışmaktadır. Güvenlik sebebiyle SSH ’a kök (root) olarak bağlanmak mümkün değildir.

```
akaraca@akaraca42:~$ sudo apt-get install openssh-server
```

-> UFW veya Uncomplicated Firewall(karmaşık olmayan güvenlik duvarı), Arch Linux, Debian veya Ubuntu'da güvenlik duvarı kurallarını yönetmek için bir ön uçtur. UFW, komut satırı aracılığıyla kullanılır ve güvenlik duvarı yapılandırmasını kolaylaştırmayı (veya karmaşıklaştırmayı) amaçlar.
-> Sunucumuza kurmak için; “apt-get Install ufw” komutu kullanılır.

İşletim sisteminizi UFW güvenlik duvarıyla ve sadece 4242 portunu açık bırakarak konfigüre etmelisiniz.

```
akaraca@akaraca42:~$ sudo apt-get install ufw_
```

-> ufw ve ssh ile güvenli bağlantı gerçekleştirmeyi amaçlıyoruz. Bu bağlantının ayarlanması gerekmektedir. Ssh'ın durumuna göz attığımızda aktif bir şekilde port 22 slotunda çalıştığını görüyoruz.

```
akaraca@akaraca42:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2022-03-17 09:43:34 EDT; 4min 12s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 1089 (sshd)
      Tasks: 1 (limit: 2336)
     Memory: 1.2M
        CPU: 12ms
     CGroup: /system.slice/ssh.service
             └─1089 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Mar 17 09:43:34 akaraca42 systemd[1]: Starting OpenBSD Secure Shell server...
Mar 17 09:43:34 akaraca42 sshd[1089]: Server listening on 0.0.0.0 port 22.
Mar 17 09:43:34 akaraca42 sshd[1089]: Server listening on :: port 22.
Mar 17 09:43:34 akaraca42 systemd[1]: Started OpenBSD Secure Shell server.
```

-> Lakin Subject bizden bunu 42 portunda yapmamızı istiyor. Bu durumda ssh'ın bağlantılarını düzenleriz.

```
akaraca@akaraca42:~$ sudo vim /etc/ssh/sshd_config_
```

```
#Port 22
```

```
Port 4242
```

Güvenlik nedeniyle ssh'a "ssh root@127.0.0.1 -p 4242" şeklinde bağlantı kurduğumuzda erişememizi istiyor bu yüzden sshd_config dosyası içerisine "PermitRootlogin no" satırı eklenmelidir.

SSH servisi sadece 4242 portu üzerinde çalışmaktadır. Güvenlik sebebiyle SSH 'a kök (root) olarak bağlanmak mümkün değildir.

```
PermitRootlogin no
#PermitRootLogin prohibit-password
```

-> Açtığımız bu portları UFW için ayarlamamız gerekmektedir.

```
akaraca@akaraca42:/etc/ssh$ sudo ufw status
Status: inactive
```

```
akaraca@akaraca42:/etc/ssh$ sudo ufw enable
Firewall is active and enabled on system startup
```

```
akaraca@akaraca42:/etc/ssh$ sudo ufw status
Status: active
```

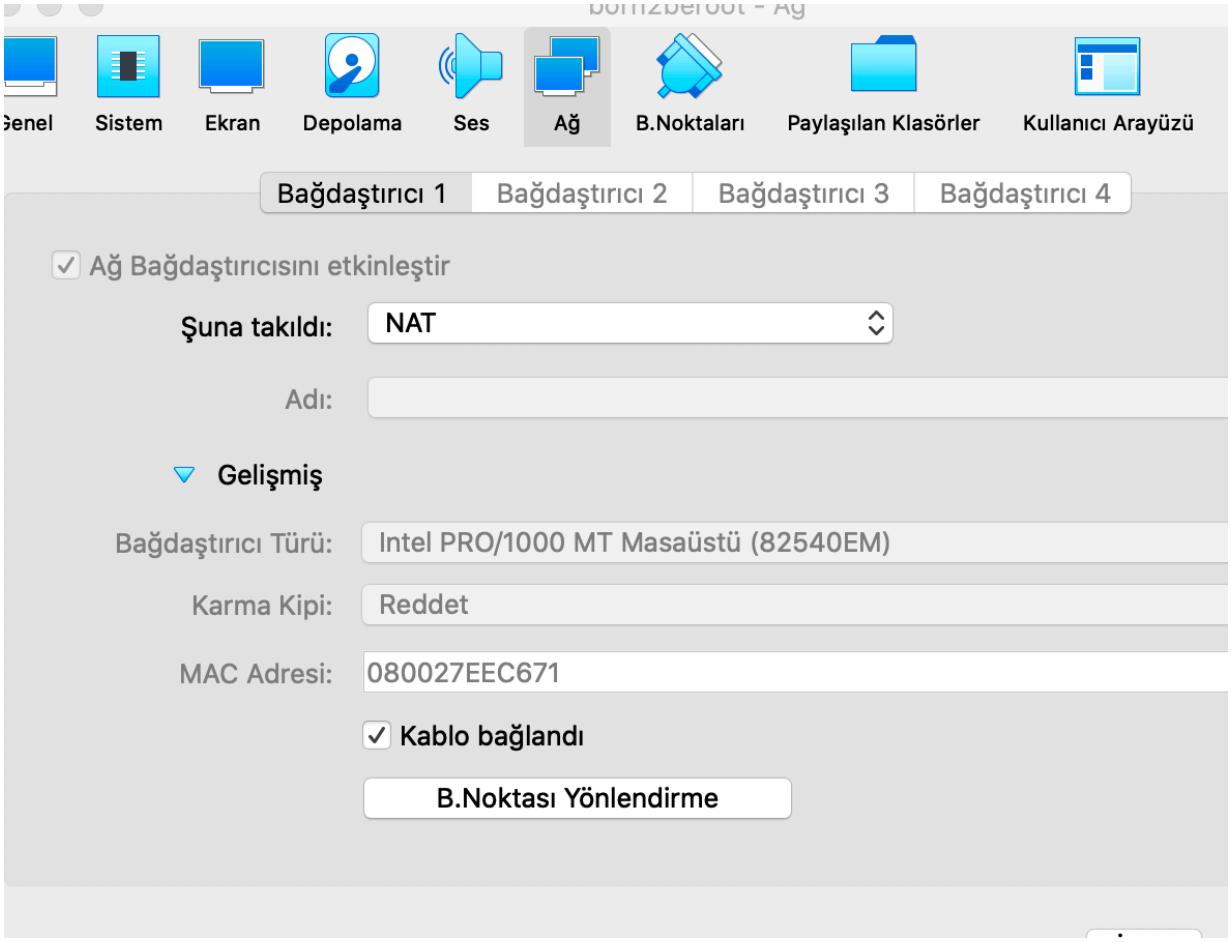
-> ufw'yi aktif etmeden öncede portlar eklenebilir. "Sudo ufw allow 4242" komutu ile güvenlik duvarına 4242 portunun izinli olduğunu belirtiyoruz ve kural olarak ekleniyor.

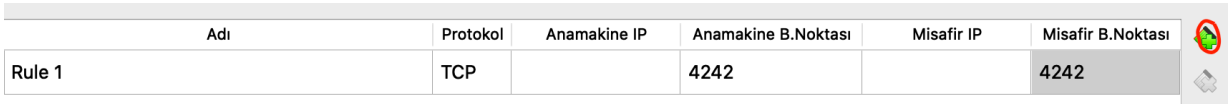
```
akaraca@akaraca42:/etc/ssh$ sudo ufw allow 4242
Rule added
Rule added (v6)
```

```
akaraca@akaraca42:/etc/ssh$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
4242                       ALLOW       Anywhere
4242 (v6)                  ALLOW       Anywhere (v6)
```

-> Ana makinamızdan sunucumuza bağlanabilmek için; Bağlantı noktası yönlendirmesinden 4242 portlarını ekliyoruz. IP belirlememize gerek yok, yerel ip üzerinden bağlantı sağlayacağız. Bu işlem sonrasında portları güncellemek için sunucumuza reboot atıyoruz.





Adı	Protokol	Anamakine IP	Anamakine B.Noktası	Misafir IP	Misafir B.Noktası
Rule 1	TCP		4242		4242

-> Artık ana bilgisayarın terminali üzerinden sunucuya bağlantı gerçekleştirebiliriz, bu bağlantıyı “ssh akaraca@127.0.0.1 -p 4242” komutu aracılığı ile yapmaktayım. Sunucuya ilk kez bağlanırken host bağlantı dosyayı oluştururken onay soracaktır “yes” olarak cevaplanmalıdır.
-> Giriş için kullanacağımız şifre oluşturduğumuz kullanıcı hesabının şifresidir.

```
akaraca@k2m15s09 ~ % ssh akaraca@127.0.0.1 -p 4242
The authenticity of host '[127.0.0.1]:4242 ([127.0.0.1]:4242)' can't be estab
lished.
ECDSA key fingerprint is SHA256:w0HV5ErmfKsx6QdFoGGvlbkUj9rJUn5NiZfG5+dD0Gk.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[127.0.0.1]:4242' (ECDSA) to the list of known ho
sts.
akaraca@127.0.0.1's password:
```

-> Subject’te bizden sistem bilgisi çıktısı veren bir dosya düzenlemizi istiyor.

Sonunda, `monitoring.sh` adında basit bir kod oluşturmalısınız.Bu kod `bash`'te geliştirilmelidir.

Kod, sunucu çalıştığında tüm terminallere her 10 dakikada bir aşağıdaki listedeki bilgileri yazdırmalıdır(`wall` komutlarına göz atın). Başlık tercihe bırakılmıştır. Ayrıca herhangi bir hata gösterilmemelidir.

-> monitoring.sh dosyasını elle oluşturup içerisine istenilen çıktı komutlarını giriyoruz.

```
akaraca@akaraca42:/usr/local/bin$ sudo touch monitoring.sh
[sudo] password for akaraca:
akaraca@akaraca42:/usr/local/bin$ ls
monitoring.sh
```

Kodunuz aşağıdaki bilgileri terminallere yazdırabilmelidir:

- İşletim sisteminizin mimarisi ve kernel versiyonu.
- Fiziksel işlemci sayısı.
- Sanal işlemci sayısı.
- Sunucunun erişilebilir RAM'i ve yüzde olarak RAM'in kullanım oranı.
- Sunucunun erişilebilir depolama alanı ve yüzde olarak depolama alanı kullanım oranı.
- Yüzde olarak işlemcinin kullanım oranı.
- Son yeniden başlatmanın tarihi ve saati.
- LVM'nin aktif olup olmadığı bilgisi.
- Aktif bağlantı sayısı.
- Sunucuyu kullanan kullanıcı sayısı.
- Sunucunun IPv4 ve MAC (Media Access Control) adresleri.
- `sudo` ile çalıştırılmış komut sayısı.

```
#!/bin/bash
arc=$(uname -a)
pcpu=$(grep "physical id" /proc/cpuinfo | sort | uniq | wc -l)
vcpu=$(grep "^processor" /proc/cpuinfo | wc -l)
fram=$(free -m | awk '$1 == "Mem:" {print $2}')
uram=$(free -m | awk '$1 == "Mem:" {print $3}')
pram=$(free | awk '$1 == "Mem:" {printf("%.2f"), $3/$2*100}')
fdisk=$(df -Bg | grep '^/dev/' | grep -v '/boot$' | awk '{ft += $2} END {print ft}')
udisk=$(df -Bm | grep '^/dev/' | grep -v '/boot$' | awk '{ut += $3} END {print ut}')
pdisk=$(df -Bm | grep '^/dev/' | grep -v '/boot$' | awk '{ut += $3} {ft+= $2} END {printf("%d"), ut/ft*100}')
cpul=$(top -bn1 | grep '^%Cpu' | cut -c 9- | xargs | awk '{printf("%.1f%%"), $1 + $3}')
lb=$(who -b | awk '$1 == "system" {print $3 " " $4}')
lvmt=$(lsblk | grep "lvm" | wc -l)
lvmu=$(if [ $lvmt -eq 0 ]; then echo no; else echo yes; fi)
#You need to install net tools for the next step [$ sudo apt install net-tools]
ctcp=$(cat /proc/net/sockstat{,6} | awk '$1 == "TCP:" {print $3}')
ulog=$(users | wc -w)
ip=$(hostname -I)
mac=$(ip link show | awk '$1 == "link/ether" {print $2}')
cmds=$(journalctl _COMM=sudo | grep COMMAND | wc -l) # journalctl should be running as sudo but our script is running as root so we don't need in sudo here
wall "  #Architecture: $arc
        #CPU physical: $pcpu
        #vCPU: $vcpu
        #Memory Usage: $uram/${fram}MB ($pram%)
        #Disk Usage: $udisk/${fdisk}Gb ($pdisk%)
        #CPU load: $cpul
        #Last boot: $lb
        #LVM use: $lvmu
        #Connexions TCP: $ctcp ESTABLISHED
        #User log: $ulog
        #Network: IP $ip ($mac)
        #Sudo: $cmds cmd" # broadcast our system information on all terminals
```

-> bu çıktıyı ekrana alabilmek için crontab işlevini kullanıyoruz.

```
akaraca@akaraca42:/usr/local/bin$ sudo crontab -u root -e
```

-> her 10 dakikada bir monitoring.sh dosyasını çalıştıracak bir komut oluşturuyoruz. * */10 * * * şeklinde de olabilirdi bu durumda 10 saatte bir yazdır anlamına gelirdi.

```
*/10 * * * * bash /usr/local/bin/monitoring.sh
```

Subject root kullanıcısı için eklenmesini istediğinden “sudo crontab -u root -e” kullanılır.

Aşağıda kodun beklenen çıktısı gösterilmiştir:

```
Broadcast message from root@wil (tty1) (Sun Apr 25 15:45:00 2021):

        #Architecture: Linux wil 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64 GNU/Linux
        #CPU physical : 1
        #vCPU : 1
        #Memory Usage: 74/987MB (7.50%)
        #Disk Usage: 1009/2Gb (39%)
        #CPU load: 6.7%
        #Last boot: 2021-04-25 14:45
        #LVM use: yes
        #Connexions TCP : 1 ESTABLISHED
        #User log: 1
        #Network: IP 10.0.2.15 (08:00:27:51:9b:a5)
        #Sudo : 42 cmd
```

->Eğerki crontab için makro oluşturma yönetici olmayan bir kullanıcıda kullanmak istersek bu durumda şu yapılmalıdır; verilen komutun izinsiz bir şekilde ekrana çıktı verebilmesi için “sudo” içerisinde kullanıcımıza dosya için izin veriyoruz.

```
akaraca@akaraca42:/usr/local/bin$ bash monitoring.sh
Hint: You are currently not seeing messages from other users and the system.
      Users in groups 'adm', 'systemd-journal' can see all messages.
      Pass -q to turn off this notice.

Broadcast message from akaraca@akaraca42 (pts/0) (Thu Mar 17 10:23:45 2022):

        #Architecture: Linux akaraca42 5.10.0-12-amd64 #1 SMP Debian 5.10.103-1 (2022-03-07) x86_64 GNU/Linux
        #CPU physical: 1
        #vCPU: 1
        #Memory Usage: 71/1982MB (3.60%)
        #Disk Usage: 1266/28Gb (4%)
        #CPU load: 6.2%
        #Last boot: Mar 17
        #LVM use: yes
        #Connexions TCP: 2 ESTABLISHED
        #User log: 2
        #Network: IP 10.0.2.15  (08:00:27:ee:c6:71)
        #Sudo: 53 cmd
```

```
akaraca@akaraca42:/usr/local/bin$ sudo visudo
```

```
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
akaraca ALL=(ALL) NOPASSWD: /usr/local/bin/monitoring.sh
```

Eğerki root içerisinde düzenleme gerçekleştirirsek çıktı şu şekilde olur; (root@akaraca42)

```
Broadcast message from root@akaraca42 (somewhere) (Tue Mar 22 11:40:02 2022):

        #Architecture: Linux akaraca42 5.10.0-12-amd64 #1 SMP Debian 5.10.103-1 (2022-03-07) x86_64 GNU/Linux
        #CPU physical: 1
        #vCPU: 1
        #Memory Usage: 72/1982MB (3.68%)
        #Disk Usage: 1445/28Gb (5%)
        #CPU load: 11.8%
        #Last boot: 2022-03-22 11:16
        #LVM use: yes
        #Connexions TCP: 2 ESTABLISHED
        #User log: 1
        #Network: IP 10.0.2.15  (08:00:27:ee:c6:71)
        #Sudo: 137 cmd
```

Eğerki akaraca içerisinde düzenleme gerçekleştirirsek çıktı şu şekilde olur; (akaraca@akaraca42)

```
Broadcast message from akaraca@akaraca42 (somewhere) (Tue Mar 22 12:10:02 2022)

        #Architecture: Linux akaraca42 5.10.0-12-amd64 #1 SMP Debian 5.10.103-1 (2022-03-07) x86_64 GNU/Linux
        #CPU physical: 1
        #vCPU: 1
        #Memory Usage: 72/1982MB (3.67%)
        #Disk Usage: 1445/28Gb (5%)
        #CPU load: 11.8%
        #Last boot: 2022-03-22 11:16
        #LVM use: yes
        #Connexions TCP: 2 ESTABLISHED
        #User log: 1
        #Network: IP 10.0.2.15  (08:00:27:ee:c6:71)
        #Sudo: 137 cmd
```

-> Subject’in bonus kısmına kadar kurulum işlemleri buraya kadardı.