SON-GAZ-BORN2

`1- Sanal makine nasıl çalışır ?`

Sanal Makineler, donanımdan ayrılmış katmanda bir bilgisayarın sanal örneğini çalıştırırlar.

`2- Seçtiğin işletim sistemi nedir`

Debian

`3- Debian ile CentOS arasındaki basit farklar`

- CentOS rpm paket yöneticisine sahiptir Debian ise apt-get paket yöneticisine sahiptir.
- CentOS repoları daha az sıklıkla güncellenirken Debian repoları daha sık bir şekilde güncellenir

`4- Sanal makinelerin amacı nedir`

tek bir fiziksel bilgisayarda çalışarak çoklu işletim sistemi kullanma olanağı sağlar.

`-6 Debianı seçtiysen aptitude ve apt arasındaki farklar ve APParmor nedir`

- Aptitude, apt-get'den daha iyi bir paket yönetimine sahiptir
- Apt-get kullanıcı arayüzünden yoksun olsa da, Aptitude'un sade metin ve etkileşimli bir kullanıcı arayüzü vardır
- `apparmor` = sisteme verilebilecek zararı sınırlandırır veya yapılan bu işlemi tamamen durduruan bir uygulamadır

`--------- SIMPLE SETUP ---------`

`-1 UFW service başlatılmış mı diye kontrol et`

sudo ufw status

`-2 SSH service başlatılmış mı diye kontrol et`

sudo systemctl status ssh

`-3 Sistemin Debian mı CentOS'mu olup olmadığını kontrol et`

hostnamectl

`--------- USER ---------`

`-1 user42 sudoya ekli mi diye kontrol et`

getent group sudo

`-2 Oncelikle yeni bir kullanıcı oluştur, şifreyi kurallara uygun olarak koy, bu kuralları nasıl oluşturduğunu açıkla.`

sudo adduser kullanıcı

sudo vim /etc/pam.d/common-password

`minlen=10` şifrenin minimum uzunluğu

`retry=3` maximum 3 deneme hakki

`lcredit=-1` en az 1 küçük harf

`ucredit=-1` en az 1 büyük harf

`dcredit=-1` en az 1 sayı

`maxrepeat=3` en fazla 3 arka arka aynı karakter

`usercheck=0` şifre kullanıcı adını içeriyor mu diye kontrol

`difok=7` yeni şifre eski şifreden minimum 7 karakter farklı olmalı

`enforce_for_root` root kullanıcıları için de aynı şeylerin geçerli olduğunu belirtir

`-3 evaluating adında bir grup oluştur, bu gruba oluşturduğun yeni kullanıcı ata ve kontrol et.`

sudo groupadd evo = grup oluştu

sudo usermod -aG evo kullanıcı = kullanıcı evo grubuna atandı

`--------- HOSTNAME AND PARTITIONS ---------`

`-2 hostnameyi degistir ve reboot at, isim degismezse -42 kardsim.`

`hostnamectl` == Mevcut ana bilgisayar adını kontrol edin

`hostnamectl set-hostname new_hostname` == ana bilgisayar adını değiştir

`sudo reboot` == yeniden başlat ve kontrol et.

`-4 partitonları listele`

lsblk

`--------- SUDO ---------`

`-1 "sudo" nun yüklü olup olmadığını kontrol et.`

sudo —version

`-2 sudo'nun amacını örneklendirerek anlat.`

`sudo` sıradan kullanıcıların, sisteme yönetici olarak bağlanmaları gerekmeden yönetici yetkisi gerektiren işlemleri yapabilmesini sağlayan bir komuttur.

`--------- UFW ---------`

`-3 UFW nin ne olduğunu açıkla`

UFW – Karmaşık olmayan Güvenlik Duvarı

`-4 UFW'nin aktif kurallarını listele, 4242 portu listede olmalı`

sudo ufw status

`-5 8080 portu için yeni bir kural ekle, kontrol et.`

sudo ufw allow 8080

sudo ufw status numbered

sudo ufw delete numberfufw

`--------- SSH ---------`

`-3 basitçe SSH nedir anlat`

kullanıcılara sunucularını internet üzerinden kontrol etmesini ve düzenlemesini sağlayan uzak yönetim protokolüdür

`-4 SSH servisinin sadece 4242 portunu kullandığından emin ol`

sudo grep Port /etc/ssh/sshd_config

`-5 Yeni oluşturulan kullanıcı ile SSH kullanarak bağlantı kurunuz.`

ssh your_username@127.0.0.1 -p 4242

`--------- SCRIPT MONITORING ---------`

`-1 Script nasıl çalışıyor kod olarak göster`

`-2 cron nedir`

`crontab` == belirlediğiniz bir zaman yada zaman diliminde belirlediğiniz komut, script yada uygulamanın çalışmasını sağlarsınız.

`3 bu script her 10 dk'da bir nasıl çalışıyor`

sudo crontab -u root -e

`**********************************************************************`

- `arc=$(uname -a)`

sistem mimarisini gösterir.

- `pcpu=$(grep "physical id" /proc/cpuinfo | sort | uniq | wc -l)`

grep ile o dosyadaki "..." bu satırı buluyoruz sort ile sıralıyoruz uniq ile tekrar eden satırları geçiyoruz wc -l ile çıktıyı alıyoruz.

- `fram=$(free -m | awk '$1 == "Mem:" {print $2}')`

free ile sistemdeki bellek miktarını görüntülüyoruz (-m mb cinsi) awk ile burdaki satırla eşleşen ilgili eylemi gerçekleştiriyoruz 1. argüman "Mem" e eşitse 2.argümanı ekrana yazdırıyor.

- `pram=$(free | awk '$1 == "Mem:" {printf("%.2f"), $3/$2*100}')`

1. argüman mem e eşitse float olarak 3. argüman ile 2.argümanın % sini alacak

- `fdisk=$(df -Bg | grep '^/dev/' | grep -v '/boot$' | awk '{ft += $2} END {print ft}')`

df disk alanı kullanımını rapor et -B yazdırmadanz< önce ölçeklendir -g gigabyte grep ^dev satır başındaysa al grep -v sadece sonunda boot olanları al anlamına gelir

- `cpul=$(top -bn1 | grep '^%Cpu' | cut -c 9- | xargs | awk '{printf("%.1f%%"), $1 + $3}')`

top linux işlemlerini göster. -b düzenli mod
-n yineleme veya kare sayısını belirtir.

Xargs: standart girdiden komut satırları oluşturur ve yürütür
cut -c 9- == 9. sıradakini almamızı sağlıyor

- `lb=$(who -b | awk '$1 == "system" {print $3 " " $4}')`

who -b kimin oturum açtığını gösteriyor

- `lvmu=$(if [ $lvmt -eq 0 ]; then echo no; else echo yes; fi)`

koşul doğruysa then ve fi arasında değerler yürütülür

- `ctcp=$(cat /proc/net/sockstat{,6} | awk '$1 == "TCP:" {print $3}')`

TCP6 ile iliskili bu yere bak

- `ulog=$(users | wc -w)`

wc her dosya için yeni satır, kelime ve bayt sayılarını yazdır -w kelime sayısını yazdırır

- `cmds=$(journalctl _COMM=sudo | grep COMMAND | wc -l)`

journalctl - Systemd günlüğünü sorgula _comm sudo için özel bir parametre