B2B Control

Genel Talimatlar

signature.txt ve açınca .vdi dosyasındaki signature txt aynı olması gerekiyor

normal terminalden goinfre dosyasına girip;

```
shasum Born2BeRoot.vdi
```

Yazarak signature.txt'le aynı olması gerekiyor çıkan kod

SHA nedir?

`shasum` komutu, bir dosyanın SHA (Secure Hash Algorithm) kontrol toplamını hesaplar. Bu kontrol toplamı, dosyanın bütünlüğünü doğrulamak için kullanılır. Eğer dosya üzerinde herhangi bir değişiklik yapılmışsa, SHA kontrol toplamı da değişir.

.VDI Uzantılı dosya?

".vdi" uzantılı dosya, Oracle VirtualBox sanal makine platformu tarafından kullanılan bir disk görüntü dosya formatıdır. Bu dosya formatı, sanal makinenin sabit diskini temsil eder. Sanal makine üzerinde çalışan işletim sistemi ve uygulamaların verilerini depolamak için kullanılır.

Proje Genel Bakış

Sanal Makine Nedir?

Sanal makineler, donanım kaynaklarını (işlemci, bellek, depolama vb.) sanallaştırarak, birden fazla işletim sistemi ve uygulamanın aynı fiziksel bilgisayar üzerinde çalışmasını sağlar. Her bir sanal makine, kendi sanal işlemcisine, belleğine ve diğer kaynaklara sahipmiş gibi davranır.

Sanal makinanın amacı nedir?

Aynı makinada, birden çok işletim sistemini aynı anda kullanma imkanı sağlar. virüslü olduğunu düşündüğümüz dosyayı burada test edebiliriz. Hazırladığımız web sitesinin farklı işletim sistemlerinde nasıl çalışacağını test edebiliriz.

En önemli avantajı büyük bir güvenlik sağlıyor olması. çünkü sanal makinan sağladığı kaynağı kullanan yazılım, içinde bulunduğu sanal ortamın dışına çıkamaz. host üzerinde bir değişikliği sebep olamaz.

Sanal Makine Nasıl Çalışır?

Sanal makine, fiziksel bir bilgisayar üzerinde yazılım tabanlı olarak oluşturulan ve çalıştırılan bir sanal bilgisayardır. Bu sanal makine, bir bilgisayarın donanım kaynaklarını sanal bir ortamda simüle eder. İşletim sistemi, uygulamalar ve diğer yazılımlar bu sanal makine üzerinde çalışır.

Sanal Makine Faydaları?

1. **İzolasyon ve Güvenlik:** Bağımsız çalışma, güvenlik sağlar.
2. **Kaynak Optimizasyonu:** Donanım kaynakları etkin paylaşılır.
3. **Hızlı Dağıtım ve Yedekleme:** Kolay yedekleme ve hızlı kurulum.
4. **Esneklik ve Taşınabilirlik:** Farklı platformlarda çalıştırılabilir.
5. **Raporlama ve İzleme:** Kaynak kullanımını takip eder.
6. Bir program açar gibi ikinci bir işletim sistemi çalıştırmak.
7. Birbirinden izole edilmiş, birden fazla işletim sistemi kurabilirsin.
8. Virüslü düşündüğünüz bir dosyayı burada açabilirsiniz.
9. Sanal ortamın dışına çıkamadığı için büyük bir güvenlik sağlıyor.

Debian Nedir?

İşletim sistemi dağıtımı (İngilizcesi: "Operating System Distribution" ya da kısaltmasıyla "OS Distribution"), bir işletim sisteminin temel bileşenleriyle bir araya getirilen ve genellikle özel yazılımlar, uygulamalar, sürücüler ve yapılandırmalarla paketlenmiş bir sürümüdür.

Linux ve diğer açık kaynaklı işletim sistemlerinde, birçok farklı bileşen ve uygulama mevcuttur. İşletim sistemi dağıtımları, bu bileşenleri bir araya getirerek bir kullanıcıya veya geliştiriciye hazır bir çözüm sunar. Bu, kullanıcıların işletim sistemini yüklemek, yapılandırmak ve kullanmak için gereken çoğu işi kolaylaştırır.

Debian ve Rocky Arasındaki temel farklar

- **Köken**: Rocky Linux, CentOS'in yerine geçmek amacıyla geliştirilen bir proje. Debian ise 1993 yılında başlatılan kendi köklerine sahip bir Linux dağıtımı.
- **Topluluk ve Destek**: Rocky Linux, özellikle kurumsal ve sunucu ortamları için tasarlanmış bir topluluk destekli projedir. Debian ise geniş ve aktif bir gönüllü topluluğuna dayanır.
- **Paket Yönetimi**: Rocky Linux, Red Hat tabanlı olduğundan `dnf` veya `yum` gibi Red Hat paket yöneticilerini kullanır. Debian ise kendi paket yöneticisi olan `apt`'i kullanır.
- **Yazılım Ekosistemi**: Rocky Linux, Red Hat tabanlı olduğundan bu ekosisteme erişim sağlar. Debian kendi büyük ve çeşitli yazılım deposuna sahiptir.
- **Kararlılık ve Güncellik**: Rocky Linux, CentOS'in kararlılık odaklı bir devamıdır. Debian'ın "Stable" sürümü kararlılık odaklıdır. Bunun yanı sıra "Testing" ve "Unstable" sürümleri de bulunur.
- **Hedef Kitle**: Rocky Linux, genellikle iş dünyasında tercih edilirken, Debian daha geniş bir kullanıcı kitlesine hitap eder. Hem kişisel hem de kurumsal kullanım için uygundur.

Debianın avantajları?

Debian, yüksek kararlılık seviyeleriyle bilinir ve uzun süreli destek sunarak güvenilir bir işletim sistemi sağlar. Aynı zamanda geniş bir yazılım deposuna sahip olmasıyla kullanıcıların ihtiyaçlarına uygun çeşitli uygulamaları kolayca erişmelerini sağlar. Debian'ın açık kaynak ilkeleri, özgür yazılımın teşvik edilmesi anlamında önemli bir katkı sağlar, bu da kullanıcıların özgür ve açık kaynaklı yazılımlarla çalışmalarını kolaylaştırır.

1. **Kararlı ve Güvenilir**
2. **Geniş Yazılım Seçeneği**
3. **Açık Kaynak İlkeleri**

4. **Donanım Uyumluluğu**
5. **Etkili Paket Yönetimi (APT)**
6. **Uzun Süreli Güvenlik Desteği**
7. **Güçlü Güvenlik Politikaları**
8. **Esnek ve Özelleştirilebilir**

Aptitude ve APT arasındaki farklar?

`apt`, Debian tabanlı Linux dağıtımlarında (örneğin Debian, Ubuntu) paket yönetim sistemidir.

`aptitude`, Debian tabanlı Linux dağıtımlarında yaygın olarak kullanılan bir paket yönetim aracıdır. "aptitude" komutu, Debian, Ubuntu ve diğer türevlerinde mevcuttur.

---

1. Apt “Advanced Packaging Tool” (Gelişmiş Paket Aracı).

2. Aptitude işlevsellik açısından daha geniştir. Aptitude get, mark ve cache de dahil olmak üzere
apt’nin işlevlerini bünyesinde barındırır.

3. Aptitude arayüze sahipken, apt sahip değildir.

4. Aptitude sisteme yüklendiğinde paketleri otomatik olarak izler. APT bu konuda yetersizdir.

5. Aptitude paketlerin ismi, tanimları, bağımlıkları vb. Gibi bir çok bilgiye erişebilir. Güçlü bir filtreleme
ve arama yapısına sahiptir.

6. Aptitude eski paketleri takip eder. APT bu tur paketleri bünyesinde bulundurur.

7. **Aptitude yaptığınız işlemlerin kaydını tutar(/var/log/aptitude)**

APPArmor Nedir?

AppArmor, kötü amaçlı yazılımların ve yetkisiz erişim girişimlerinin önlenmesine yardımcı olabilir. AppArmor bir güvenlik özelliğidir, **Arka planda sessizce çalışır ve sisteme zarar verebilecek uygulamarı kontrol edip, sınırlandırır.**

```
dpkg -l | grep apparmor //yüklü olup olmadığına bakıyor.

sudo apparmor_status //Çalışıp çalışmadığını kontrol ediyoruz
```

CentOS ve Debian arasındaki temel farklar?

CentOS yeni sürümleri genellikle uzun bir aradan sonra yayınlanır ve bu nedenle bu sistemler çok kararlıdır. Debian CentOSa göre daha fazla güncelleme alıyor. CentOSun arayüzü karışıktır. Debianın daha kolay.

Basit Kurulum

```
sudo systemctl status ssh // running

sudo systemctl status ufw // Çalışıp çalışmadığını gösteriyor (exited)
```

Seçilen İşletim Dağıtım Sisteminin ne olduğunu öğrenmek için;

```
cat /etc/os-release
//YA DA
uname -a //(Tüm Bilgileri İçerir)
//YA DA
uname -v //(Karnel Sürümünü ve yayınlandığı tarihle birlikte gösterir)
```

User

Kullanıcı Oluşturma

```
adduser user42 //user42 oluşturur ve user42 grubu oluşturur Yüksek yetki

useradd 42user //Düşük yetki
```

```
sudo userdel -r user42 // user42 kullanıcısını sildik
```

Daha sonra şifre politikasına uygun şifre yazmamızı istiyor yazıyoruz

```
adduser user42 sudo //user42 ' yi sudo grubuna ekledik.
//YADA
add user42 sudo //user 42 böyle de eklenebilir.

sudo deluser user42 sudo //user42'yi sudo grubundan kaldırdık.

groups user42 // user42 hangi gruplarda tek tek gösterir

getent group sudo // sudo grubunun içinde kimler var gösterir
```

Atamaları yaptıktan sonra şifre politikası kontrol edilmeli

```
sudo passwd user42 //Yazarak user42'nin şifresini değişebiliyoruz

sudo chage -l user42 //Şifre Politikasına uygun olup olmadığını öğreniyoruz
```

Grup oluşturmak için

```
sudo groupadd hepsi1 //hepsi1 adında grup oluşturduk

sudo groupdel heps1 //hepsi1 grubunu sildik
```

tüm grupları göstermek için;

```
getent group //tüm grupları gösterir

groups // sadece genel grupları gösterir
```

Şifre Politikası nasıl ayarlanıyor?

```
sudo vim /etc/login.defs // Şifre Değişme süresi ayarlanan yer

sudo vim /etc/pam.d/common-password /* lipbam paketiyle yüklediğimiz şifre
politikası burada ayarlanıyor */
```

```
sudp chage -l user42 // User42'nin bu kurallara uyup uymadığını sorguluyoruz

passwd user42 // user42'nin şifresini böyle değişiyoruz
```

1. İnsanlar genelde basit şifreler seçmeye meyilli olduğundan. Şifre oluşturmadan önce bir takım
ön koşullar getirilir. Böylelikle basit şifre olasılıklarının önüne geçiliyor.

2. Şifre politikaları çok fazla koşul istediğinden genelde şifre unutmaları, hesap kilitlenmesi ve
şifre oluştururken yine tekrar aynı kurallar içerisinde seçme gibi vakit kayıpları olabiliyor.

3. Hackerlar tarafından şifre tahmin riskini azaltıyor.

4. Saldırganın iş yükü ve harcadığı zaman artığı için hedef olmaktan çıkıyoruz.

```
Şifrenin en az bir büyük harf içermesini zorunlu kılmak için:
ucredit=-1

Şifrenin en az bir küçük harf içermesi zorunlu kılmak için:
lcredit=-1

Şifrenin en az bir sayısal karakter içermesini zorunlu kılmak için:
dcredit=-1

En fazla 3 ardışık aynı karakter ayarlamak için:
maxrepeat=3

Bir biçimde <kullanıcı adı> içeriyorsa parolayı reddetmek için:
usercheck=1

Şifreyi 7 kere değiştikten sonra eski şifresini tekrar kullanabilir:
difok=7

Tüm bu şifre politikasını root kullanıcısı üzerinde uygulamak için :
enforce_for_root

Şifre minimum uzunluğunu 10 karakter olarak ayarlamak için:
minlen=10
```

Libpam kullandık şifre kalitesi için bunu indirdiğimizin kontrolünü

```
dpkg -l | grep libpam-pwquality
```

Ana Bilgisayar Adı Ve Bölümleri

Makinenin ana bilgisayar adını öğrenmemiz gerek

```
hostname //sadece ana bilgisayar adını öğreniriz

hostnamectl //detaylı bir şekilde hostname alırız
```

Hostname adını değişmek lazım

```
sudo hostnamectl set-hostname makarna42 //hostname adını makarna42 yaptık

sudo vim /etc/hosts /* burdan uuygun42 yazan hostnameyi makarna42 yapıp
reboot etmemiz lazım */

sudo reboot // böyle yapıp yeni hostname adını öğren
```

Sanal Makine Bölümleri

bölümleri göstermek

```
lslbk // bölümleri gösterir

NAME                    MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINTS
sda                       8:0    0 30.8G  0 disk
├─sda1                    8:1    0  500M  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   10G  0 lvm   /
    ├─LVMGroup-swap     254:2    0  2.3G  0 lvm   [SWAP]
    ├─LVMGroup-home     254:3    0    5G  0 lvm   /home
    ├─LVMGroup-var      254:4    0    3G  0 lvm   /var
    ├─LVMGroup-srv      254:5    0    3G  0 lvm   /srv
    ├─LVMGroup-tmp      254:6    0    3G  0 lvm   /tmp
    └─LVMGroup-var--log 254:7    0    4G  0 lvm   /var/log
sr0                      11:0    1 1024M  0 rom
```

sda(1-4) arası öncelikli cihazları temsil ederken Sda4 sonrası logical birimler olduklarını gösterir.

[SWAP]: Eğer RAM kapasitesi yetersizse ve bir uygulama çok fazla bellek tüketirse, işletim sistemi swap alanını kullanarak bu fazla veriyi geçici olarak sabit diske kaydeder.

/boot: dizini, işletim sisteminin başlatılması için gerekli olan temel dosyaların ve yapılandırmaların bulunduğu kritik bir dizindir.

/srv: sistemdeki servislerin veya uygulamaların verilerini depolamak için kullanılan bir dizindir

/tmp: geçici dosyaların saklandığı bir dizindir. Bu dizin, işletim sistemi ve uygulamalar tarafından geçici olarak oluşturulan verileri depolamak için kullanılır. Bu dosyalar, genellikle işlem tamamlandıktan sonra hemen silinir.

/var: değişken verilerin saklandığı bir dizindir. Bu dizin, çalışma sırasında sürekli olarak değişen veya güncellenen verileri içerir

/var/log: bir Linux sistemdeki log dosyalarının saklandığı dizindir. Log dosyaları, sistem ve uygulamaların çalışma durumu hakkında bilgi sağlar. Bu dosyalar, hata mesajları, uyarılar, olay kayıtları ve diğer önemli bilgileri içerir.

sr0: CD/DVD

LVM Nedir?

1. Büyük disk alanı ihtiyacı olan sistemlerde LVM ile disk veri kümeleri oluşturularak yada sisteme yeni bir disk daha eklenerek toplam disk boyutu arttırılabilir.
2. LVM(logical volume manager) ile birden fazla diski tek bir disk bölümü olarak kullanabilir ve disk yönetimi işlemlerinde büyük kolaylık sağlar.

SUDO

Sudo, sıradan kullanıcıların sisteme yönetici olarak bağlanmaları gerekmeden yönetici yetkisi gerektiren işlemleriyapabilmesini sağlayan bir programdır.

Sudo ile belirli yönetici yetkilerini kullanacak kullanıcılara root parolasının paylaşılması gibi güvenlik açısından sıkıntıÇıkartabilecek durumlar engellenmiş olur.

Sudo yetkisiyle yapılan işlemlerde kimin hangi işlemi yaptığının takibi daha kolaydır sudo Log dosyasında gözüküyor kiminhangi işlemi yaptığı.

Sudo neden inactive dead

Sudo log

```
sudo cat /var/log/sudo/sudo.log /* tüm yazdığımız komutların geçmişi
ve nerden yapıldığı */
```

sudo’ya PDF’de eklenmiş katı kuralları kontrol etmemiz lazım

```
sudo visudo // Katı kurallar bu dosyada mevcuttur.
```

dosyanın içindeki kurallar bunlar

```
Defaults        log_input,log_output
Defaults        logfile="/var/log/sudo/sudo.log"
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:>
Defaults        requiretty
Defaults        passwd_tries=3
Defaults        badpass_message="Wrong password, Try again!"
```

1. `Defaults log_input,log_output`: Bu ayar, sudo komutunun girdi ve çıktılarını günlüğe kaydetmesini sağlar. Yani, kullanıcıların sudo komutunu ne zaman kullandığını ve bu komutların ne tür verilerle çalıştığını izler.
2. `Defaults logfile="/var/log/sudo/sudo.log"`: Bu satır, sudo komutunun günlük dosyasının yolu ve adını belirtir. Bu durumda, sudo günlükleri `/var/log/sudo/sudo.log` dosyasına kaydedilir.
3. `Defaults env_reset`: Bu ayar, sudo komutunun çevresel değişkenlerin sıfırlanmasını (reset) sağlar. Bu, sudo komutunun çalıştırılmasıyla ilgili çevresel değişkenlerin temizlenmesini sağlar.
4. `Defaults mail_badpass`: Bu ayar, yanlış bir parola girildiğinde, sistemin yönetici veya sistem yöneticisine bir e-posta göndermesini sağlar. Bu, güvenlik önlemi olarak kullanılabilir.
5. `Defaults secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"`: Bu

ayar, sudo komutunun çalıştırılabilir dosyaları araması için geçerli yolları belirtir. Bu, güvenlik ve çevresel bağımsızlık sağlamak için önemlidir.

6. `Defaults requiretty`: Bu ayar, sudo komutunun bir terminal gerektirip gerektirmediğini belirtir. Bu durumda, sudo komutunun bir terminalde çalıştırılmasını gerektirir.
7. `Defaults passwd_tries=3`: Bu ayar, bir kullanıcının sudo komutunu kullanırken hatalı bir parola girmesi durumunda kaç denemeye izin verileceğini belirtir. Bu durumda, kullanıcıya üç deneme hakkı verilir.
8. `Defaults badpass_message="Wrong password, Try again!"`: Bu ayar, yanlış bir parola girildiğinde ekranda görüntülenecek özel bir hata mesajını belirtir.

```
sudo systemctl status sudo /* ssh masked bir şekilde çalışır */
```

UFW (Uncomplicated Firewall)

düzgün şekilde kurulmuş mu kontrol edelim

```
dpkg -l | grep ufw // Yüklenip yüklenmediğini kontrol et

sudo systemctl status ufw // düzgün çalışıyor mu kontrol et (exited güzükmeli)
```

UFW’nin önemi:

1. Güvenlik işlerini yapmamıza yarayan güvenlik duvarı.

2. İptables güvenlik duvarı yapılandırmasını kolaylaştırmak için UFW geliştirildi.

3. Firewall, zararlı yazılımlara karşı bir duvar örür ve bunların ağ yolu ile bilgisayara sızmasının önüne geçer.

4. güvenlik duvarı dediğimiz yapı temelde, bilgisayarımızın ya da sunucumuzun internet dünyasında güvenli hale gelmesini sağlayan kurallar setidir. Belirli portların açılması, kapatılması, sınırlanması, ip bazlı engelleme vs pek çok spesifik kural tanımlanabiliriz.

UFW'deki aktif kuralları listeleyin. 4242 numaralı bağlantı noktası için bir kural bulunmalıdır.

UFW kurallarını göster:

```
sudo ufw status numbered /* etkin olan 2 tane 4242 portu gözükcek */

sudo ufw allow 8080 /* 8080 portunu ufw'de etkin kıldık */

sudo ufw delete 2 /* 8080 portunun ufw etkinliğini kaldırdık */
```

SSH (Secure Shell)

SSH düzgün kurulmuş mu ve çalışıyor mu:

```
dpkg -l | grep ssh //kurulup kurulmadığına bakar

systemctl status ssh // çalışmasını kontrol eder (running)

sudo vim /etc/ssh/sshd_config /* port 4242 ve root'u da bu dosyada ayarladık */
```

Rootun ssh üzerinden bağlanma yetkisini kaldırmamız gerekiyor. Root" hesabı, sistem üzerindeki tüm işlemleri yapma yetkisine sahiptir. Bu, yanlışlıkla veya dikkatsizlikle yapılan bir komutun ciddi sonuçlara yol açabileceği anlamına gelir.

```
sudo vim /etc/ssh/sshd_config /*burdan SSH Configuration dosyasına girdik
ve SSH'ın kullanacağı portu 4242 olarak belirledik */

PermitRootLogin no // bunu yazarak root'un ssh bağlantısını engelliyoruz
```

SSH Nedir ve önemi

1. Linux sunuculara erişim sağlamak için SSH protokolü kullanıyoruz. Yani uzaktaki bir sunucuya bağlanmak, ona komutlar ve dosyalar göndermek üzere kullanılan şifrelenmiş bir uzaktan sağlayıcı protokolüdür. Çoğu kullanıcı SSH bağlantısını varsayılan ayarlar ile kullanıyor. Ancak bu şekilde bir kullanım

güvenlik risklerini de beraberinde getiriyor.
SSH erişimi dışarı açık bir sunucunun root parolasının kırılması sunucu açıldıktan sonra dakikalar içinde gerçekleşebilir. (Biz de projede ssh erişimini root kullanıcısına kapatarak güvenli bir ssh bağlantısı oluşturmaya çalışıyoruz. Etc/ssh/sshd_config klasöründe permitrootlogin no diyerek ssh erişimini root kullanıcısına yasaklıyoruz...)

2. Diğer önemli değişiklik port değişikliğidir. SSH bağlantısının portu varsayılan olarak 22'dir. Portu değiştirerek
saldırganların 22 portundan sunucuya erişimini engelleyeceğiz. (Biz de 4242 portundan bağlanarak güvenli bir SSH bağlantısı oluşturmaya çalışıyoruz)

3. Sadece belirlediğimiz adreslerden SSH erişimi sağlamak istiyorsak güvenlik duvarı(UFW) burada çok işe yarar

4. UFWyi ilk olarak aktif hale getiriyoruz. Ufw enable, ufw allow 4242 gibi komutlar sadece belirlenen SSH adreslerinden erişim yapabilmemizi sağlar ve SSH ile belirttiğimiz 4242 portu ÖNLEMİNE EK BİR ÖNLEM OLARAK GÖRÜLEBİLİR....

5. SSH hizmetinin yalnızca 4242 numaralı bağlantı noktasını kullandığını doğrulayın.Değerlendirilen öğrenci, yeni oluşturulan kullanıcı ile giriş yapabilmeniz için SSH kullanmanıza yardımcı olmalıdır. Bunu yapmak için bir anahtar veya basit bir şifre kullanabilirsiniz. Değerlendirilen öğrenciye bağlı olacaktır. Tabii ki konuda belirtildiği gibi "root" kullanıcısı ile SSH kullanamayacağınızdan emin olmalısınız.

SSH bağlantılarını kontrol et!

```
ssh root42@localhost -p 4242 /*root olarak dene ve
kabul edilmediğini göster)*/

ssh <username>@localhost -p 22 /*22 portundan dene ve kabul
edilmediğini göster*/

ssh aoner42@localhost -p 4242 //giriş sağla son olarak
```

Monitoring

Netstat araçlarını yükledik ve monitoring.sh dosyasını oluşturduk

```
sudo vim /usr/local/bin/monitoring.sh //monitoring.sh dosyasına girdik

sudo bash /usr/local/bin/monitoring.sh /* crontab'ı kullanmadan monitoring.sh
'daki çıktıyı bu şekilde alabiliyoruz
```

Monitoring.sh'daki bilgiler kısaca

1. `arc=$(uname -a)`: Sistem mimarisini ve diğer bilgileri alır.
2. `pcpu=$(grep "physical id" /proc/cpuinfo | sort | uniq | wc -l)`: Fiziksel CPU sayısını alır.
3. `vcpu=$(grep "^processor" /proc/cpuinfo | wc -l)`: Sanal CPU (çekirdek) sayısını alır.
4. `fram=$(free -m | grep Mem: | awk '{print $2}')`: Toplam RAM miktarını alır.
5. `uram=$(free -m | grep Mem: | awk '{print $3}')`: Kullanılan RAM miktarını alır.
6. `pram=$(free | grep Mem: | awk '{printf("%.2f"), $3/$2*100}')`: RAM kullanım yüzdesini hesaplar.
7. `fdisk=$(df -Bg | grep '^/dev/' | grep -v '/boot$' | awk '{ft += $2} END {print ft}')`: Toplam disk kapasitesini alır.
8. `udisk=$(df -Bm | grep '^/dev/' | grep -v '/boot$' | awk '{ut += $3} END {print ut}')`: Kullanılan disk alanını alır.
9. `pdisk=$(df -Bm | grep '^/dev/' | grep -v '/boot$' | awk '{ut += $3} {ft+= $2} END {printf("%d"), ut/ft*100}')`: Disk kullanım yüzdesini hesaplar.
10. `cpul=$(top -bn1 | grep '^%Cpu' | cut -c 9- | xargs | awk '{printf("%.1f%%"), $1 + $3}')`: CPU kullanım yüzdesini alır.
11. `lb=$(who -b | awk '$1 == "system" {print $3 " " $4}')`: Son sistem başlatma tarihini alır.
12. `lvmt=$(lsblk -o TYPE | grep "lvm" | wc -l)`: LVM (Logical Volume Manager) kullanılıp kullanılmadığını kontrol eder.
13. `lvmu=$(if [ $lvmt -eq 0 ]; then echo no; else echo yes; fi)`: Eğer LVM kullanılıyorsa "yes", kullanılmıyorsa "no" değerini alır.

14. `ctcp=$(cat /proc/net/tcp | wc -l | awk '{print $1-1}' | tr '' ' ')`: TCP bağlantı sayısını alır.
15. `ulog=$(users | wc -w)`: Oturum açmış kullanıcı sayısını alır.
16. `ip=$(hostname -I)`: Sistem IP adresini alır.
17. `mac=$(ip link show | awk '$1 == "link/ether" {print $2}')`: MAC adresini alır.
18. `cmds=$(journalctl _COMM=sudo | grep COMMAND | wc -l)`: Sudo komutlarının sayısını alır.

Son olarak, `wall` komutuyla bu bilgileri ekrana yazar.

Cron nedir?

Cron, Unix benzeri işletim sistemlerinde zamanlanmış görevlerin otomasyonu için kullanılan bir zamanlama programıdır. "Cron" terimi, "chronograph" kelimesinden türetilmiştir ve zamanla ilgili işlerin programlanmasına dayanır.

Cron, belirli bir zaman diliminde (örneğin, her gün saat 02:00) veya belirli aralıklarla (örneğin, her saat) çalıştırılması gereken görevlerin planlanması için kullanılır. Bu görevler genellikle betikler, komutlar veya programlar olabilir

Bir **crontab** komutu, tek bir satırla temsil edilir. **** Bir komutu birden çok satıra yaymak için kullanamazsınız.

```
sudo crontab -u root -e //crontabı burdan açıyoruz
```

Crontab Düzenlemesi

```
*/10 * * * * bash /usr/local/bin/monitoring.sh /* her 10 dakikada çalıştırmak
için bu şekilde yazılmıştır VM açıldığından itibaren 10 dk değil saat'in 10
dklık diliminde çalışır
*/
```

- `/10`: Bu alan dakika kısmını temsil eder. `/10` ifadesi "her 10 dakikada bir" anlamına gelir. Yani, bu betik her saatteki her 10 dakika aralığında çalıştırılacaktır.
- `*`: Saat kısmını temsil eder. Asterisk (*) kullanıldığında, herhangi bir saatte çalıştırılmasını belirtir.

- : Ayın hangi gününde çalıştırılacağını temsil eder. Yine asterisk (*) kullanıldığında, herhangi bir gün olabilir.
- : Ay kısmını temsil eder. Yine herhangi bir ay olabilir.
- : Haftanın hangi gününde çalıştırılacağını temsil eder. Yine asterisk (*) kullanıldığında, herhangi bir gün olabilir.
- `bash /usr/local/bin/monitoring.sh`: Bu komut, belirtilen yolu kullanarak `monitoring.sh` adlı betiği çalıştıracaktır.

Sonuç olarak, bu crontab satırı, her saatteki her 10 dakika aralığında `/usr/local/bin/monitoring.sh` betiğini çalıştıracaktır. Bu, belirli bir süre boyunca betiğin düzenli aralıklarla otomasyonunu sağlamak için kullanışlı olabilir.

```
* * * * * komut
- - - - -
| | | | |
| | | | ----- Günün haftada kaçıncı günü (0 - 7) (Pazar = 0 veya 7)
| | | ------- Ayın kaçıncı günü (1 - 31)
| | --------- Ay (1 - 12)
| ----------- Haftanın günü (0 - 6) (Pazar = 0)
------------- Saat (0 - 23)

0 2 * * * /usr/bin/betik.sh /* Örneğin, aşağıdaki crontab satırı, her gün
saat 02:00'de /usr/bin/betik.sh betiğini çalıştırır */
```

Crontab’ı durdurmak ve çalıştırmak

```
sudo service cron stop //reboot sonrası tekrar çalışır çünkü enable

sudo systemctl disable cron /* Reboot sonrası çalışmaz ama cron stop yapmazsan
reboot edilene kadar cron çalışır */

sudo systemctl enable cron /* Bu da enable ediyor boot sonrası çalışmasını
sağlıyor */

sudo service cron start
```

!!!!/etc dosyası:etc dosyası ve alt dizinlerinde sistemle ilgili bütün konfigürasyon dosyaları bulunur.

```
#!/bin/bash

# Sistem Bilgileri Toplanıyor
architecture=$(uname -a)
physical_cpus=$(grep "physical id" /proc/cpuinfo | sort | uniq | wc -l)
virtual_cpus=$(grep "^processor" /proc/cpuinfo | wc -l)
total_memory=$(free -m | awk '/Mem:/ {print $2}')
used_memory=$(free -m | awk '/Mem:/ {print $3}')
memory_percentage=$(free | awk '/Mem:/ {printf("%.2f"), $3/$2*100}')
total_disk=$(df -h --total | awk '/total/ {print $2}')
used_disk=$(df -h --total | awk '/total/ {print $3}')
disk_percentage=$(df -h --total | awk '/total/ {print $5}')
cpu_load=$(top -bn1 | grep '^%Cpu' | awk '{printf("%.1f%%"), $2 + $4}')
last_boot=$(who -b | awk '$1 == "system" {print $3 " " $4}')
lvm_usage=$(lsblk -o TYPE | grep "lvm" | wc -l)
lvm_used=$(if [ $lvm_usage -eq 0 ]; then echo no; else echo yes; fi)
tcp_connections=$(cat /proc/net/tcp | wc -l)
logged_in_users=$(who | wc -l)
ip_address=$(hostname -I)
mac_address=$(ip link show | awk '$1 == "link/ether" {print $2}')
sudo_commands=$(journalctl _COMM=sudo | grep COMMAND | wc -l)

# Duyuru Gönderiliyor
wall "Architecture: $architecture
CPU physical: $physical_cpus
vCPU: $virtual_cpus
Memory Usage: $used_memory/$total_memory MB ($memory_percentage%)
Disk Usage: $used_disk/$total_disk ($disk_percentage)
CPU load: $cpu_load
Last boot: $last_boot
LVM use: $lvm_used
Connexions TCP: $tcp_connections ESTABLISHED
User log: $logged_in_users
Network: IP $ip_address ($mac_address)
Sudo: $sudo_commands cmd"
```