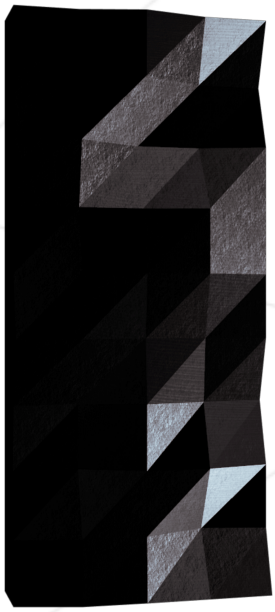


# Born2beRoot

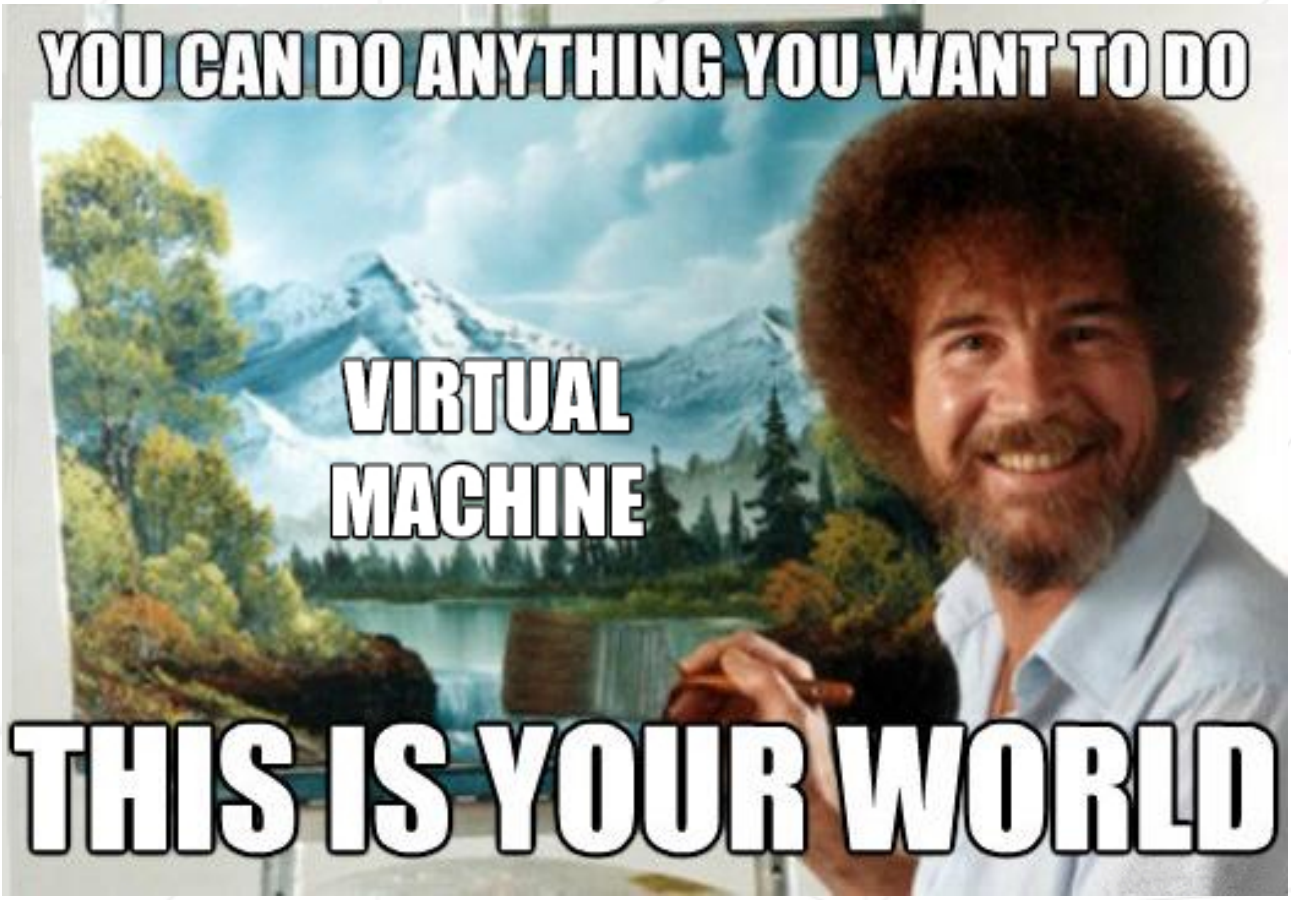
*Özet: Bu doküman sistem yönetimi ile ilgili bir alıştırmadır.*

*Versiyon: 1*

# İçindekiler

# Bölüm I

# Önsöz



# Bölüm II

# Giriş

Bu proje sanallaştırmanın harika dünyasını tanıtmayı amaçlamaktadır.

Belirli talimatlara uyarak `VirtualBox` (eğer `VirtualBox` kullanamazsanız `UTM`) ile ilk makinenizi oluşturacaksınız. Ardından, projenin sonunda katı kuralları yerine getirerek kendi işletim sisteminizi kurma yeteneğine sahip olacaksınız.

# Bölüm III

# Genel Yönergeler

- `VirtualBox` (ya da `UTM` ) kullanımı zorunludur.
- Yalnızca kök dizininde bulunan `signature.txt` dosyasını teslim etmelisiniz. Bu dosya içine makinenizin sanal diskinin imzasını yapıştırmalısınız. Daha fazla bilgi için Teslim ve Ön Değerlendirme Bölümünü inceleyin.

# Bölüm IV

# Zorunlu kısım

Bu proje belirli adımları izleyerek ilk sunucunuzu kurmanızı içermektedir.

Sunucunuzu kurmanızla ilgilenildiği için en az sayıda servis kurmalısınız. Bu sebeple, grafiksel arayüz kullanılmayacaktır. Yani, X.org ya da buna denk başka bir grafik sunucusu kurulumu yasaklanmıştır. Aksi taktirde 0 alacaksınız.

Debian ya da CentOS'in son stabil sürümünden birini işletim sistemi olarak seçmelisiniz. Eğer sistem yönetiminde yeniyseniz Debian şiddetle tavsiye edilmektedir.

CentOS kurulumu biraz karmaşıktır. Bu sebeple KDump kurmak zorunda değilsiniz. Fakat, SELinux başlangıçta çalıştırılmalı ve ayarlamaları proje gereksinimlerine adapte edilmelidir. Debian için de AppArmor başlangıta çalıştırılmalıdır.

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

```
wil@wil:~$ lsblk
NAME                  MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda                     8:0    0    8G  0 disk
├─sda1                  8:1    0  487M  0 part  /boot
├─sda2                  8:2    0    1K  0 part
└─sda5                  8:5    0  7.5G  0 part
  └─sda5_crypt        254:0    0  7.5G  0 crypt
    ├─wil--vg-root    254:1    0  2.8G  0 lvm   /
    ├─wil--vg-swap_1  254:2    0  976M  0 lvm   [SWAP]
    └─wil--vg-home    254:3    0  3.8G  0 lvm   /home
sr0                    11:0    1 1024M  0 rom
wil@wil:~$ _
```

Savunma sırasında, size seçtiğiniz işletim sistemi ile ilgili birkaç soru sorulacak. Bu sebeple, aptitude ile apt arasındaki farkı ya da SELinux ve AppArmor'un ne olduğunu bilmelisiniz. Kısaca ne kullandığınızı anlayın!

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

SSH kullanımı yeni hesap oluşturarak savunma kısmında test edilecektir. Bu amaçla nasıl çalıştığını anlamalısınız.

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

Güvenlik duvarınız sanal makineyi çalıştırdığınızda aktif olmalıdır. CentOS için varsayılan güvenlik duvarı yerine UFW kullanmalısınız. Bunu kurmak için muhtemelen DNF'e ihtiyaç duyacaksınız.

- Sanal makinenizin hostname'i giriş bilginizin sonuna 42 eklenmiş hali olmalıdır (örnek olarak kadir42). Hostname'i değerlendirmeniz sırasında değiştireceksiniz.
- Güçlü bir şifreleme politikası kullanmalısınız.
- Sudo yu katı kurallara uyarak kurmalı ve konfigüre etmelisiniz.
- Kök kullanıcıya ek olarak, kullanıcı adı giriş bilgileriniz olan bir kullanıcı olması gerekmektedir.
- Bu kullanıcı user42 ve sudo grupları altında olmalıdır.

Savunma sırasında yeni bir kullanıcı oluşturacaksınız ve bu kullanıcıyı ilgili gruplara atayacaksınız.

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.
- 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.

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

`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`

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.

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.

`Savunma esnasında, size kodun nasıl çalıştığı sorulacaktır. Ayrıca değişiklik yapmadan kodu kesmeniz (interrupt) gerekmektedir. cron komutlarına göz atın.`

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
```

Aşağıda işletim sisteminizin gereksinimlerini kontrol edebileceğiniz iki komut gösterilmiştir:

For `CentOS`:

```
[root@wil ~]# head -n 2  /etc/os-release
NAME="CentOS Linux"
VERSION="8"
[root@wil ~]# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      32
[root@wil ~]# ss -tunlp
Netid State  Recv-Q  Send-Q   Local Address:Port   Peer Address:Port
tcp   LISTEN 0       128           0.0.0.0:4242        0.0.0.0:*     users:(("sshd",pid=822,fd=5))
tcp   LISTEN 0       128              [::]:4242           [::]:*     users:(("sshd",pid=822,fd=7))
[root@wil ~]# ufw status
Status: active

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

[root@wil ~]# _
```

For `Debian`:

```
root@wil:~# head -n 2 /etc/os-release
PRETTY_NAME="Debian GNU/Linux 10 (buster)"
NAME="Debian GNU/Linux"
root@wil:/home/wil# /usr/sbin/aa-status
apparmor module is loaded.
root@wil:/home/wil# ss -tunlp
Netid State  Recv-Q Send-Q Local Address:Port  Peer Address:Port

tcp   LISTEN 0      128          0.0.0.0:4242       0.0.0.0:*     users:(("sshd",pid=523,fd=3))
tcp   LISTEN 0      128             [::]:4242          [::]:*     users:(("sshd",pid=523,fd=4))
root@wil:/home/wil# /usr/sbin/ufw status
Status: active

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

# Bölüm V

# Bonus kısım

Bonus Listesi:

- Bölümlemeyi doğru ayarlayın. Aşağıdakine benzer bir yapı elde edeceksiniz :

```
# lsblk
NAME                    MAJ:MIN RM   SIZE RO TYPE  MOUNTPOINT
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
```

- Lighttpd, MariaDB ve PHP kullanarak işlevsel bir WordPress sayfası kurun.
- Faydalı olduğunu düşündüğünüz bir servis kurun (NGINX ve Apache 2 harici). Savunma sırasında seçiminizi savunmak zorundasınız.

Bonus kısmı tamamlamak için ekstra servis kurmanız durumunda ihtiyaçlarınızı karşılamak için daha fazla port açabilirsiniz. Tabii ki, UFW kuralları uygun şekilde adapte edilmelidir.

Bonus kısım yalnızca zorunlu kısımlar ‘’mükemmel" ise değerlendirilecektir. Mükemmel ile kastedilen gerekli kısmın tamamen bitmiş ve hatasız çalışmasıdır. Eğer zorunlu kısımlardaki gerekliliklerin hepsini karşılayamazsanız, bonus kısım değerlendirilmeyecektir.

# Bölüm VI

# Teslim ve Ön Değerlendirme

`Git` reponuzun kök dizinine yalnızca `signature.txt` adlı belgeyi yükleyeceksiniz. Dosya içine makinenizin sanal diskinin imzasını kopyalayacaksınız. Bu imzayı almak için öncelikle varsayılan kurulum klasörünü açmalısınız (Sanal makinelerinizin kaydedildiği klasör).

- Windows: `%HOMEDRIVE%%HOMEPATH%\VirtualBox VMs\`
- Linux: `~/VirtualBox VMs/`
- MacM1: `~/Library/Containers/com.utmapp.UTM/Data/Documents/`
- MacOS: `~/VirtualBox VMs/`

Ardından sanal makinenizin imzasını `".vdi"` (`UTM` kullanıcıları için `".qcow2"`) dosyasından sha1 formatında alın. Aşağıda `centos_serv.vdi` dosyası için 4 örnek komut gösterilmiştir:

- Windows: `certUtil -hashfile centos_serv.vdi sha1`
- Linux: `sha1sum centos_serv.vdi`
- For Mac M1: `shasum Centos.utm/Images/disk-0.qcow2`
- MacOS: `shasum centos_serv.vdi`

Örnek olarak alacağınız çıktı:

- `6e657c4619944be17df3c31faa030c25e43e40af`

Unutmayın ki ilk değerlendirmeden sonra sanal makinenizin imzası değişebilir. Bu sorunu çözmek için, sanal makinenizi kopyalayabilir ya da durumu kaydet seçeneğini kullanabilirsiniz.

Tabii ki, Git reposuna sanal makinenin kendisini yüklemeniz YASAKLANMIŞTIR . Savunma sırasında, signature.txt içindeki imza ile sanal makinenizin imzaları karşılaştırılacak. Eğer ikisi birbirinin aynısı değilse notunuz 0 olacak.