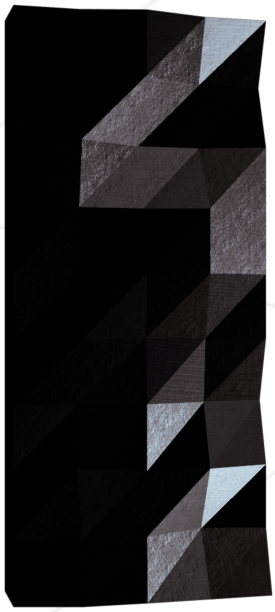


Born2beRoot

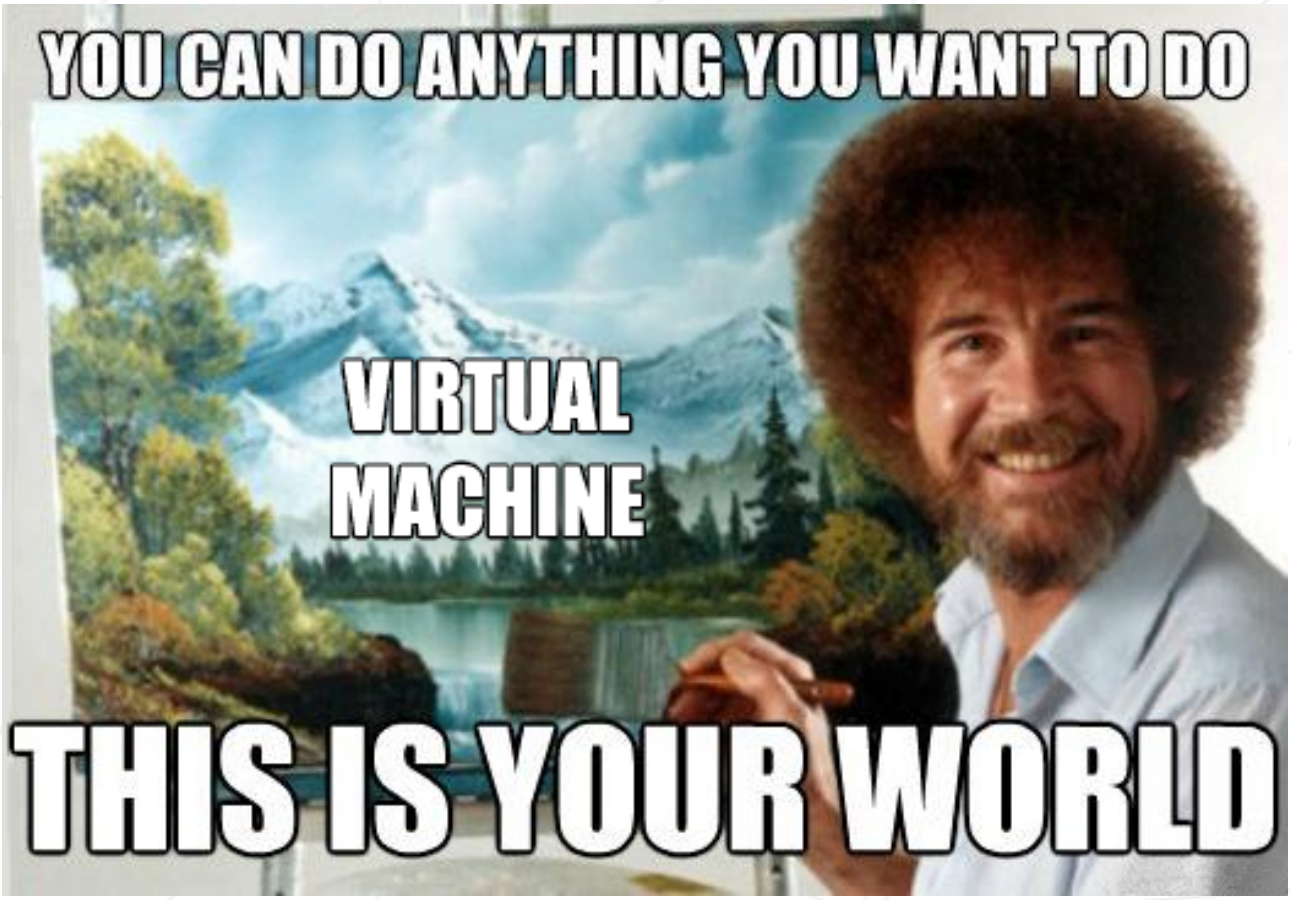
Özet: Bu proje Sistem Yönetimi ile ilgili bir egzersizdir.

Versiyon: 3.2

İçindekiler

Bölüm I

Önsöz



Bölüm II

Giriş

Bu proje size sanallaştırmanın harika dünyasını tanıtmayı amaçlamaktadır.

Belirli talimatlara uyarak `VirtualBox` (eğer `VirtualBox` kullanamıyorsanız `UTM`) ile ilk makinenizi oluşturacaksınız. Ardından, projenin sonunda, katı kurallar ekleyerek kendi işletim sisteminizi kurabilme becerisine sahip olacaksınız.

Bölüm III

Genel Yönergeler

- `VirtualBox` (eğer `VirtualBox` kullanamıyorsanız `UTM`) kullanımı zorunludur.
- Yalnızca root/kök dizininde bulunan `signature.txt` dosyasını teslim etmelisiniz. Bu dosya içine makinenizin sanal diskinin imzasını yapıştırmalısınız. Daha fazla bilgi için Proje Teslimi ve Akran Değerlendirmesi bölümünü inceleyin.

Bölüm IV

Zorunlu Bölüm

Bu proje belirli adımları izleyerek ilk sunucunuzu kurmanızdan oluşmaktadır.

Konu bir server kurmak olduğundan, olabilecek en az sayıda servis kurmalısınız. Bu nedenle, grafiksel arayüz kullanılmayacaktır. Yani, X.org ya da buna denk başka bir grafik sunucusu kurulumu yasaklanmıştır. Aksi taktirde final notunuz 0 olacaktır.

Debian ya da Rocky'nin son stabil sürümünden birini (stabil olmayan veya test aşamasında olan sürümler kullanılmamalıdır) işletim sistemi olarak seçmelisiniz. Eğer sistem yönetiminde yeniyseniz Debian şiddetle tavsiye edilmektedir.

Rocky kurulumu biraz karmaşıktır. Bu nedenle KDump kurmak zorunda değilsiniz. Fakat, SELinux başlangıçta çalıştırılmalı ve ayarlamaları proje gereksinimlerine adapte edilmelidir. Debian için de AppArmor başlangıçta çalıştırılıyor olmalıdır.

LVM kullanarak en az 2 tane şifrelenmiş partition oluşturmalısınız. Aşağıda sizden beklenen bölümlendirmeye 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. Örnek olarak, aptitude ile apt arasındaki farkı ya da SELinux ve AppArmor'un ne olduğunu bilmelisiniz. Kısaca ne kullandığınızı anlayın!

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

SSH kullanımı yeni bir hesap oluşturarak savunma sürecinde test edilecektir. Bu nedenle nasıl çalıştığını anlamalısınız.

İşletim sisteminizi UFW (ya da Rocky için firewalld) güvenlik duvarıyla ve sanal makenide sadece 4242 portunu açık bırakarak konfigüre etmelisiniz.

Güvenlik duvarınız sanal makineyi çalıştırdığınızda aktif olmalıdır. Rocky için UFW yerine firewalld kullanmalısınız.

- Sanal makinenizin hostname'i logininizin sonuna 42 eklenmiş hali olmalıdır (örneğin nkahrima42). Hostname'i değerlendirmeniz sırasında değiştirebiliyor olmanız gerekiyor.
- Güçlü bir şifreleme politikası kullanmalısınız.
- sudo 'yu katı kurallara uyarak kurmalı ve konfigüre etmelisiniz.
- Root kullanıcıya ek olarak, kullanıcı adı giriş bilgileriniz olan bir kullanıcı olması gerekmektedir.
- Bu kullanıcı user42 ve sudo gruplarına ait olmalıdır.

Savunma sırasında yeni bir kullanıcı oluşturabilmeli ve bu kullanıcıyı ilgili gruplara atayabilmelisiniz.

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.
- Şifre en az 10 karakter uzunluğunda olmalıdır. Şifre en az bir büyük karakter, bir küçük karakter ve bir sayı içermelidir. Ayrıca 3'ten fazla art arda aynı karakteri içermemelidir.
- Şifre kullanıcının adını içermemelidir.

- Şifre, eski şifrenin içermediği en az 7 karakter içermelidir (bu kural root kullanıcı için geçerli değildir).
- Root kullanıcı şifresi de yukarıdaki kurallara uymalıdır.

`Konfigürasyon dosyalarınızı ayarladıktan sonra, root kullanıcı da dahil olmak üzere, sanal makinedeki tüm kullanıcıların şifresini değiştirmelisiniz.`

`sudo` grubunuza güçlü bir konfigürasyon yapabilmek 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/` dizinine kaydedilmelidir.
- Güvenlik sebepleriyle `TTY` modu aktif hale getirilmelidir.
- Yine güvenlik sebepleriyle, `sudo` tarafından kullanılabilen dizinler sınırlandırılmalıdır. Örnek olarak:
  `/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin`

Son olarak, `monitoring.sh` adında basit bir script oluşturmalısınız. Bu kod `bash`'te geliştirilmelidir.

Script, sunucu çalıştığında tüm terminallere her 10 dakikada bir aşağıdaki listedeki bilgileri yazdırmalıdır(`wall` komutlarına göz atın). Banner kişisel 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 bu scriptin nasıl çalıştığı sorulacaktır. Ayrıca değişiklik yapmadan işleyişini kesebilmeniz (interrupt etmeniz) gerekmektedir. cron komutlarına göz atın.`

Aşağıda kodun beklenen çıktısı gösterilmiştir:

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

    #Architecture: Linux nkahrima 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 proje gereksinimlerini kontrol edebileceğiniz iki komut gösterilmiştir:

For `Rocky`:

```
[root@wil wil]# head -n 2 /etc/os-release
NAME="Rocky Linux"
VERSION="8.7 (Green Obsidian)"
[root@wil 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:      33
[root@wil wil]# ss -tunlp
Netid State  Recv-Q Send-Q  Local Address:Port   Peer Address:Port Process
tcp   LISTEN 0      128           0.0.0.0:4242       0.0.0.0:*     users:(("sshd",pid=28429,fd=6))
tcp   LISTEN 0      128              [::]:4242          [::]:*     users:(("sshd",pid=28429,fd=4))
[root@wil wil]# firewall-cmd --list-service
ssh
[root@wil wil]# firewall-cmd --list-port
4242/tcp
[root@wil wil]# firewall-cmd --state
running
[root@wil 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 bölüm

Bonus Listesi:

- Bölümlemeyi doğru ayarlayarak aşağıdakine benzer bir yapı elde edin:

```
# 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
```

- Şu servisleri kullanarak işlevsel bir WordPress sayfası kurun: lighttpd, MariaDB ve PHP.
- Faydalı olduğunu düşündüğünüz bir servis kurun (NGINX ve Apache 2 hariç). Savunma sırasında seçim nedeninizi açıklamak zorundasınız.

Bonus bölümü tamamlamak için ekstra servis kurmak sizin sorumluluğunuzda. Bu durumda ihtiyaçlarınız doğrultusunda daha fazla port açabilirsiniz. Tabii ki, UFW kuralları uygun şekilde adapte edilmelidir.

Bonus bölüm, yalnızca zorunlu bölüm KUSURSUZ ise değerlendirilecektir. Kusursuz, zorunlu bölümün tamamen yapıldığı ve sorunsuz çalıştığı anlamına gelir. TÜM zorunlu gereksinimleri tamamlamadıysanız, bonus bölüm hiçbir şekilde değerlendirilmeyecektir.

Bölüm VI

Proje Teslimi ve Akran Değerlendirmesi

`Git` reponuzun root/kök dizinine yalnızca `signature.txt` adlı belgeyi yüklemelisiniz. Dosyanın içinde makinenizin sanal diskinin imzası bulunmalıdır. Bu imzayı alabilmek 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 `rocky_serv.vdi` dosyası için 4 örnek komut gösterilmiştir:

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

Almanız gereken çıktı aşağıdaki gibidir:

- `6e657c4619944be17df3c31faa030c25e43e40af`

`Unutmayın ki ilk değerlendirmenizden sonra sanal makinenizin imzası değişebilir. Bu sorunu çözmek için, sanal makinenizi kopyalayarak çoğaltabilir ya da durumu kaydet/save state seçeneğini kullanabilirsiniz.`

Tabii ki, Git reponuza sanal makinenin kendisini yüklemeniz YASAKLANMIŞTIR. Savunma sırasında, signature.txt içindeki imza ile sanal makinenizin imzaları karşılaştırılacaktır. Eğer ikisi birbirinin aynısı değil ise notunuz 0 olacaktır.

0010 01 11 111 001 000 11 01 10 1 0000 01 1 1010 111 11 0 000
011 00 1 0000 1 0000 0 01 0100 1 0 010 10 01 1 0 0001 0 010 000
00 111 10 111 0010 001100 001100 001100