کالج پارت

پروژه اختیاری

تَکتورَانداز ۴

تیم شبکه و زیرساخت ابری

گروه نرم افزاری یاس

بهار - تابستان ۱۴۰۲

کاملیار حیمی

::: شرح پروژه

پژمان به تازگی تو یه شرکت تامین‌کننده زیرساخت ابری مشغول به کار شده! این شرکت وظیفه تامین زیرساخت‌های ابری مشتری‌های سازمانی رو بر عهده گرفته و رویکرد سنتی راه‌اندازی شبکه که صرفا مبتنی بر سخت‌افزاره، جوابگوی نیازهای این شرکت نیست. این شرکت برای سرعت بخشیدن به پیاده‌سازی زیرساخت ابری نیاز داره تا در یکی از بخش‌های DMZ داخلی هر سازمان بخشی از اجزای شبکه رو به صورت نرم‌افزاری پیاده‌سازی کنه!

پژمان به عنوان طراح این سیستم باید بتونه شبکه‌ای رو طراحی کنه که در نهایت تحویل یک ادمین سازمانی بشه و این امکان رو براش فراهم کنه تا برخی از اجزاء شبکه رو با استفاده از یک رابط خط فرمان یا CLI کنترل کنه. هدف‌گذاری‌ای که پژمان برای طراحی این شبکه انجام داده اینه که این پروژه در ۲ فاز اولیه انجام و به نسخه پیش‌نمایش برسه! در ادامه خواسته‌هایی که از پژمان وجود داره به تفکیک شرح داده شده و میزان اهمیت هر بخش از سمت مالک هم در قالب نمره وزن‌دهی شده.

::: فاز اول (نمره: ۲۰٪)

تو این فاز باید خودکارسازی برخی از تنظیمات شبکه لینوکس پیاده بشه! در این فاز «موقت» به این معنیه که بعد از ریبوت کردن ماشین، تنظیمات ماشین به حالت اولیه برمی‌گردن و «دائمی» هم به این معنیه که بعد از ریبوت، تنظیمات مورد نظر در ماشین باقی می‌مونن!

حواست باشه: برای اینکه بتونی یه نمره خیلی خوب رو به دست بیاری، باید قابلیت‌های زیر رو تو کارت لحاظ کنی:

- تعویض DNS
- تغییر Hostname
- تعیین IP استاتیک Interface به صورت موقت و دائمی
- استفاده از DHCP برای تعین IP یک اینترفیس به صورت موقت و دائمی
- اضافه کردن روت موقت و دائمی به لینوکس
- امکان پاک کردن روت های دائمی و موقت از لینوکس

دقت کن که حالت‌های مختلفی برای تعیین/تعویض IP ممکنه رخ بده! برای مثال ممکنه یه ماشین به صورت موقت یا دائمی طوری تنظیم شده باشه که IP اینترفیس رو از DHCP دریافت و کاربر تصمیم بگیره که به صورت موقت یا دائمی IP استاتیک برای Interface تعیین کنه. واضحه که باید IP ای که از DHCP گرفته شده آزاد بشه و اگه نیازه تنظیمات مربوط به استفاده از DHCP به صورت دائم هم از ماشین پاک بشه. حواست باشه که نرم‌افزاری که طراحی می‌کنی باید حالت‌های مختلف رو تشخیص و اقدامات مناسب هر حالت رو اجرا کنه!

برای دائمی کردن تنظیمات/روت‌های مربوطه راه‌های مختلفی وجود داره و شما به عنوان یه طراح نرم‌افزار باید راهی که مناسب و استانداردده رو شناسایی کنی و اون رو پیاده‌سازی و در مستندات شرح بدی که چرا راهی که انتخاب کردی مناسب و/یا استانداردده!

تعویض DNS و راهنمای استفاده کاربر:

```python
import subprocess

def change_dns(server):
    subprocess.run(['sudo', 'bash', '-c', 'echo "nameserver {0}" > /etc/resolv.conf'.format(server)])

#example change dns to 8.8.8.8
change_dns('8.8.8.8')
```

تغییر Hostname و راهنمای استفاده کاربر:

```python
#change hostname
import subprocess
def change_hostname(hostname):
    subprocess.run(['sudo', 'hostnamectl', 'set-hostname', hostname])

#change hostname to example
change_hostname('example')
```

تعیین IP استاتیک اینترفیس به صورت دائمی و راهنمای استفاده کاربر :

```python
#determine the static ip interface
import subprocess

def set_static_ip(interface, ip, netmask):
    subprocess.run(['sudo', 'ifconfig', interface, ip, 'netmask', netmask])

#determine the static ip interface example ith0 , ip :192.168.1.10 and netmask :255.255.255.0
set_static_ip('eth0', '192.168.1.10', '255.255.255.0')
```

استفاده از DHCP برای تعیین آیپی یک اینترفیس به صورت دائمی و موقت و راهنمای استفاده کاربر :

```
#use dhcp to determine the ip of an interface
import subprocess

def set_dhcp(interface):
    subprocess.run(['sudo', 'dhclient', interface])

#use dhcp to determine the ip of an interface 0
set_dhcp('eth0')
```

اضافه کردن روت موقت و دائمی به لینوکس و راهنمای استفاده کاربر :

```
#add permanent root
import subprocess

def add_permanent_route(destination, gateway):
    subprocess.run(['sudo', 'ip', 'route', 'add', destination, 'via', gateway])

#add permanent root example ip 192.168.2.0 and gateway : 192.168.1.1
add_permanent_route('192.168.2.0/24', '192.168.1.1')


#remove permanent root
import subprocess

def delete_permanent_route(destination):
    subprocess.run(['sudo', 'ip', 'route', 'del', destination])

#remove permanent root fo example 192.168.2.0/24
delete_permanent_route('192.168.2.0/24')
```

امکان پاک کردن روت های دائمی و موقت از لینوکس و راهنمای استفاده کاربر :

```python
#add permanent root
import subprocess

def add_permanent_route(destination, gateway):
    subprocess.run(['sudo', 'ip', 'route', 'add', destination, 'via', gateway])

#add permanent root example ip 192.168.2.0 and gateway : 192.168.1.1
add_permanent_route('192.168.2.0/24', '192.168.1.1')


#remove permanent root
import subprocess

def delete_permanent_route(destination):
    subprocess.run(['sudo', 'ip', 'route', 'del', destination])

#remove permanent root fo example 192.168.2.0/24
delete_permanent_route('192.168.2.0/24')


#remove temporary root
import subprocess

def delete_temp_routes():
    subprocess.run(['sudo', 'ip', 'route', 'flush', 'cache'])

#remove all temporaryroot
delete_temp_routes()
```

پایان فاز 1

::: فاز دوم – گام اول (نمره: ۲۰٪)

در این فاز با استفاده از یک فایروال نرم‌افزاری لینوکسی به نام NFTables قراره اقداماتی رو انجام بدیم تا سرورمون رو امن‌تر کنیم. به طور خلاصه می‌شه گفت که هدف از این بخش خودکارسازی اولیه پیکربندی NFTables هستش! در این فاز فرض کن که پروتکل‌ها و سرویس‌ها از پورت‌های پیش‌فرض و استاندارد خودشون استفاده می‌کنن! مثلا HTTP روی پورت ۸۰ در حال گوش دادنه! دقت کن که در هر جایی که برنامه‌ای که طراحی کردی امکان محدود کردن چیزی رو فراهم می‌کنه، باید امکان رفع دقیقا همون محدودیت فراهم بشه!

حواست باشه: برای اینکه بتونی یه نمره خیلی خوب رو به دست بیاری، باید قابلیت‌های زیر رو تو کارت لحاظ کنی:

- محدود کردن IP هایی که می‌تونن به ماشین SSH بزنن به یک IP یا رنج خاص
- پاک کردن همه قوانین فایروال
- فرستادن همه بسته‌هایی که مقصدشون 4.2.2.4:53 هست به 1.1.1.1:53
- قطع کردن اینترنت ماشین با حفظ دسترسی به شبکه داخلی
- بستن همه ارتباطات یک یوزر خاص
- دائمی کردن وضعیت فعلی فایروال به طوری که بعد از ریبوت، وضعیت فایروال ریست نشه!

بدیهی هست که ممکنه در طول اجرای برنامه، ترکیبی از موارد بالا ازت خواسته بشه، بنابراین زمانی که قراره یه Rule اضافه کنی، اون رو در بالاترین نقطه Chain مورد نظر قرار بده. راستی تو این فاز نگهداری، ایجاد و حذف Tableها، Chainها و Ruleها در اختیار خودته و محدودیتی در شیوه انجام خواسته‌های ذکر شده نداری!

پاک کردن همه قوانین فایروال ها و راهنمای استفاده کاربر:

```
#clear firewall rules
import subprocess

def clear_firewall_rules():
    subprocess.run(['sudo', 'iptables', '-F'])

#clear firewall rules
clear_firewall_rules()
```

قطع کردن اینترنت ماشین با حفظ دسترسی به شبکه داخلی و راهنمای استفاده کاربر :

```
#disconnecting the internet while maintaining access to the internal network
import subprocess

def disable_internet():
    subprocess.run(['sudo', 'iptables', '-A', 'OUTPUT', '-p', 'tcp', '--dport', '80', '-j', 'REJECT'])

#disconnecting the internet while maintaining access to the internal network
disable_internet()
```

بستن همه ارتباط های یک یوزر خاص و راهنمای استفاده کاربر :

```python
#close the communication of a specific user
import subprocess

def block_user(user):
    subprocess.run(['sudo', 'iptables', '-A', 'INPUT', '-p', 'tcp', '-m', 'owner', '--uid-owner', user, '-j', 'REJECT'])

#close the communication of a user with example uid 1000
block_user('1000')
```

پایان گام اول فاز 2

::: فاز دوم – گام دوم (نمره: ۵۰٪)

بعد از انجام تنظیمات اولیه که توی گام قبلی انجامشون دادی، باید به برنامت قابلیتی رو اضافه کنی که برای کاربر این امکان رو فراهم کنه تا بتونه بدون نیاز به دانش کار با NFTables پیکربندی دلخواهش رو روی NFTables پیاده‌سازی کنه! هدف از این بخش اینه که برنامه شما راه‌حلی آسون‌تر و قابل فهم‌تر برای پیکربندی NFTables فراهم کنه.

دقت کن که پیاده‌سازی یک Wrapper برای NFTables که تمامی ویژگی‌ها و قابلیت‌های NFTables رو پشتیبانی کنه امری بسیار دشوار و به تبع، در فازهای اولیه قابل پیاده‌سازی توسط پژمان و تیم کوچیکش نیست! اما تصمیم‌گیری اینکه برنامه‌ای که نوشتی تا چه حد امکانات NFTables رو پشتیبانی می‌کنه بر عهده خودته. سعی کن اگه قابلیتی را برای کاربر فراهم می‌کنی تا جای ممکن استفاده ازش راحت باشه و جلوی خطای کاربر هم گرفته بشه.

حواست باشه: برای اینکه بتونی یه نمره خیلی خوب رو به دست بیاری، باید قابلیت‌های زیر رو تو کارت لحاظ کنی:

- ایجاد Table
- ایجاد Chain
- ایجاد رول‌های محدود کننده دسترسی
- ایجاد رول‌های مربوط به NAT

ایجاد Table

```python
#create table
import subprocess

def create_table(table):
    subprocess.run(['sudo', 'nftables', '-t', table, '-N', table])

#create new table
create_table('mytable')
```

ایجاد Chain :

```
#create chain
import subprocess

def create_chain(table, chain):
    subprocess.run(['sudo', 'nftables', '-t', table, '-N', chain])

#create a new chain
create_chain('mytable', 'mychain')
```

ایجاد رول های محدودکننده دسترسی :

```
#create access-restricting roles
import subprocess

def create_rule(table, chain, rule):
    subprocess.run(['sudo', 'iptables', '-t', table, '-A', chain] + rule)

#create access-restricting roles
create_rule('mytable', 'mychain', ['-p', 'tcp', '--dport', '22', '-j', 'ACCEPT'])
```

پایان گام دوم فاز 2