Krzysztof Dąbrowski gr. 3

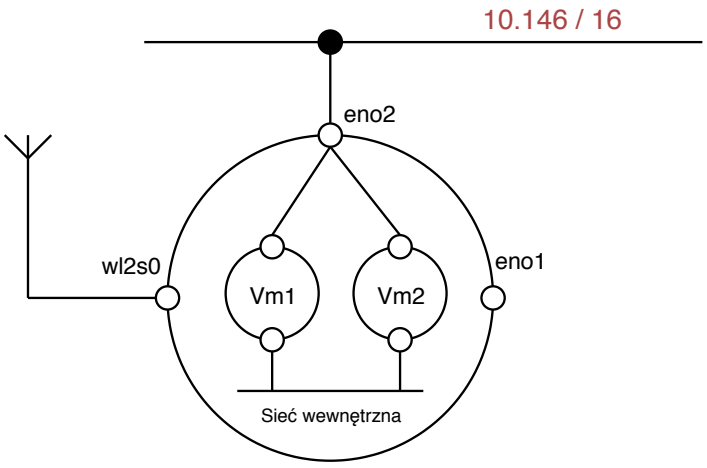
Laboratorium sieci komputerowych - c2
Konfiguracja interfejsów sieciowych

6 kwietnia 2019

Spis treści

1. Pierwsze zajęcia

Celem pierwszej części zajęć c2 było odwzorowanie schematu sieci na stacji roboczej. By to osiągnąć trzeba było wykreować maszyny wirtualne oraz odpowiednie interfejsy sieciowe.



Rysunek 1. Schemat sieci do wykreowania

1.1. Zbadanie lokalnych interfejsów

Dzięki poleceniu `ifconfig` można uzyskać informacje o interfejsach gospodarza. Uzyskanie w tens sposób nazwy interfejsów zostały naniesione na schemat

1.2. Kreacja wirtualnych maszyn

W celu automatyzacji tworzenia maszyn wirtualnych napisałem prosty skrypt konfiguracyjny.

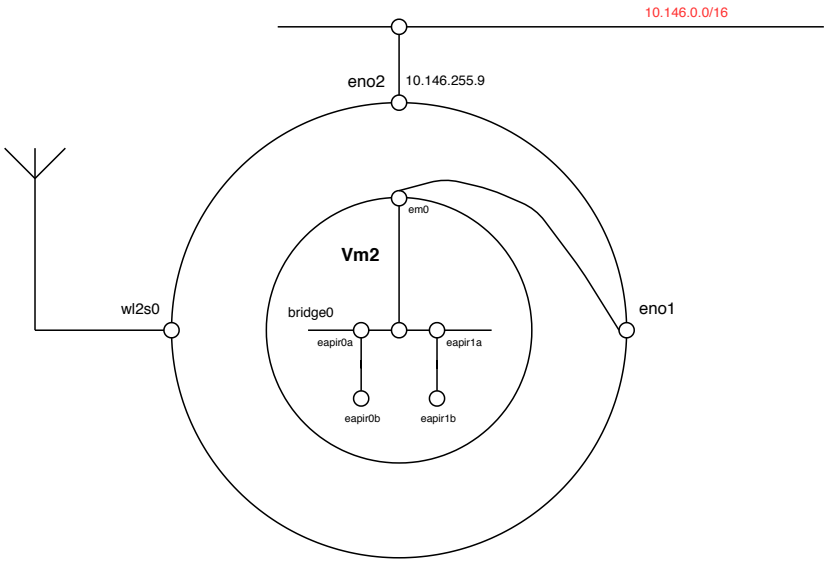
```
#!/bin/sh

#Lista maszyn do utworzenia
machinesToCreate="Vm1 Vm2"

for VM in $machinesToCreate; do
    VBoxManage createvm --name $VM --ostype "Debian_64" --register

    #Ustawienei kolejnosci bootowania
    VBoxManage modifyvm $VM --boot1 net --boot2 disk --boot3 dvd --boot4 no

    #Ustawnienie ramu i rdzeni procesora
    VBoxManage modifyvm $VM --memory 1024 --vram 128 --cpus 2

    #Ustawienie kontrolera sieci
    VBoxManage modifyvm $VM --nic1 bridged --bridgeadapter1 eno2

    #Dodanie do sieci wewnetrznej
    VBoxManage modifyvm $VM --nic2 intnet
    #Nadanie nazwy sieci wewnetrznej
    #VBoxManage modifyvm $VM --intnet2 "siec1"

done
```

Możliwe było również wyszukanie **gotowego skryptu** i modyfikacja go do własnych potrzeb. W tym celu można było skorzystać z programu `grep`.
Przykład wyszukania skryptów związanych z Virtual Box: `grep -iRl "VboxManage" /usr/local/zetis/bin`

2. Drugie zajęcia

Celem drugich zajęć było wygenerowanie pojedynczej wirtualnej maszyny oraz konfiguracja wirtualnych interfejsów sieciowych wewnątrz tej maszyny.

2.1. Schemat struktury sieci



Rysunek 2. Schemat sieci do wykreowania

2.2. Generacja maszyny

Do utworzenia maszyny zmodyfikowałem wcześniejszy skrypt. Na potrzeby aktualnego zadania skrypt tworzy teraz tylko jedną maszynę typu FreeBSD_64 z jednym interesem mostkowym.

2.3. Konfiguracja sieci

Dalszą część zadań laboratoryjnych kontynuowałem na w domu. Przy pomocy skryptu z zajęć utworzyłem maszynę oraz pobrałem obraz systemu FreeBSD i zainstalowałem go.

2.4. Logowanie zdalne ssh

Chciałem dostać się do maszyny przez terminal gospodarza. W tym celu chciałem wykorzystać protokół ssh.

Przy pomocy polecenia `ifconfig` zobaczyłem, że interfejs maszyny nie dostał adresu ip automatycznie.

```
root@:~ # ifconfig
em0: flags=8802<BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=81009b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HWCSUM,VLAN_HW
FILTER>
        ether 08:00:27:ad:1e:a0
        media: Ethernet autoselect (1000baseT <full-duplex>)
        status: active
        nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
```

Rysunek 3. Interfejs bez adresu IPv4

By pobrać adres od serwera DHCP wywołałem komendę `dhclient em0`.

Następnie wygenerowałem klucze maszyny poleceniem `ssh-keygen -A` oraz zmieniłem ustawienia demona ssh dodając do pliku */etc/rc.conf* linię „ sshd_enable="YES" ". Po zakończeniu konfiguracji uruchomiłem serwis poleceniem `service sshd restart`.

Aby przetestować efekt zalogowałem się z systemu gospodarza poleceniem `ssh root@192.168.1.126`.

2.5. Kreacja interfejsów

Poleceniem `ifconfig bridge create` utworzyłem wirtualny mostek oraz podłączyłem go do interfejsu em0 poleceniem `ifconfig bridge0 addm em0`.

Następnie utworzyłem dwie pary interfejsów *e-pair* wywołując dwukrotnie `ifconfig epair create`. Każdą z tak utworzonych par podłączyłem do mostka wykonując `ifconfig bridge0 addm epair0a` oraz `ifconfig bridge0 addm epair1a`.

Efekt wykonanych poleceń wyświetliłem wywołując `ifconfig` bez argumentów.

```
bridge0: flags=8802<BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 1500
        ether 8e:6e:4c:29:d1:7c
        id 00:00:00:00:00:00 priority 32768 hellotime 2 fwddelay 15
        maxage 20 holdcnt 6 proto rstp maxaddr 2000 timeout 1200
        root id 00:00:00:00:00:00 priority 0 ifcost 0 port 0
        member: epair1a flags=143<LEARNING,DISCOVER,AUTOEDGE,AUTOPTP>
                ifmaxaddr 0 port 6 priority 128 path cost 2000
        member: epair0a flags=143<LEARNING,DISCOVER,AUTOEDGE,AUTOPTP>
                ifmaxaddr 0 port 4 priority 128 path cost 2000
        member: em0 flags=143<LEARNING,DISCOVER,AUTOEDGE,AUTOPTP>
                ifmaxaddr 0 port 1 priority 128 path cost 20000
        groups: bridge
        nd6 options=9<PERFORMNUD,IFDISABLED>
```

Rysunek 4. Gotowa konfiguracja