Warsztaty z Sieci komputerowych
Lista 1

Przed zajęciami

Otwórz znajdujący się na stronie wykładu dokument *Opis maszyny wirtualnej Virbian* i przeczytaj go uważnie.

Konfiguracja początkowa

Pobierz obraz maszyny ze strony wykładu, stwórz konfigurację dla maszyn *Virbian0* i *Virbian1*. Następnie uruchom maszynę *Virbian0*.

Znak `Vi$>` oznacza wykonanie danego polecenia w konsoli maszyny *Virbiani* z uprawnieniami zwykłego użytkownika. Natomiast znak `Vi#>` oznacza konieczność wykonania polecenia z prawami administratora. W tym celu należy uprzednio zalogować się na konto użytkownika `root` albo poprzedzić takie polecenie komendą `sudo`.

Tutorial #1

Poniższe zadanie należy wykonać w uruchomionej maszynie wirtualnej *Virbian0*. Na początku włącz interfejs graficzny poleceniem `startx`.

- Poleceniem

  ```
  V0$> ip addr
  ```

  wyświetl wszystkie dostępne interfejsy sieciowe. Powinny być dostępne dwa interfejsy: `lo` i `enp0s3`. Jeśli nazwa drugiego interfejsu jest trochę inna, należy odpowiednio zmodyfikować poniższe polecenia.

- Uzyskaj konfigurację sieciową dla maszyny wirtualnej poleceniem

  ```
  V0#> dhclient -v enp0s3
  ```

  Ponownie wykonaj polecenie

  ```
  V0$> ip addr
  ```

  i sprawdź, że wyświetlana informacja zmieniła się i karta maszyny wirtualnej ma teraz przypisany adres IP równy `10.0.2.15` (lub podobny).

- Uruchom przeglądarkę Firefox. Po lewej stronie powinien być widoczny pasek z rozszerzeniem *HTTP Header Live* wyświetlającym wysyłane i odbierane nagłówki HTTP. Wejdź przeglądarką na stronę `http://example.org/` i obejrzyj przesyłane nagłówki protokołu HTTP. Ile żądań HTTP jest wysyłanych? Do jakich serwerów są one skierowane?

- Sprawdź jaki jest adres IP związany z adresem `example.org` poleceniem

  ```
  V0$> host -t a example.org
  ```

  Niech *w.x.y.z* będzie tym adresem IP. Uruchom program Wireshark i włącz w nim obserwację interfejsu `enp3s0` klikając dwukrotnie na jego nazwie. Aby odfiltrować wyświetlanie zbędnych pakietów w polu *Apply a display filter ...* wpisz `ip.addr == w.x.y.z` i kliknij przycisk *Apply* (niebieska strzałka po prawej stronie tego pola). W razie potrzeby możesz również kliknąć ikonę *Restart current capture* (jedna z pierwszych ikon od lewej na górze okna programu).

- Odśwież oglądaną stronę w przeglądarce naciskając `Shift` + `Ctrl` + `R`. W Wiresharku wśród wysyłanych pakietów znajdź ten zawierający żądanie HTTP pobierające stronę HTML. Obejrzyj w tym pakiecie nagłówki warstwy sieciowej (IP) i transportowej (TCP). Klikając poszczególne pola opisu, podświetlasz w widoku szesnastkowym pakietu (na dole okna) odpowiadające im bajty. Jaki jest źródłowy i docelowy adres IP tego pakietu? Jaki jest jego źródłowy i docelowy port? W których nagłówkach znajdują się te informacje?

  Powtórz powyższe operacje dla pakietu zawierającego odpowiedź HTTP (powinien zawierać kod odpowiedzi `200 OK` wraz ze stroną w HTML lub kod odpowiedzi `304 Not Modified`). Czy dane identyfikujące połączenie (źródłowy/docelowy adres/port) zmieniły się czy są takie same? Dlaczego?

- W rozszerzeniu HTTP Header Live obejrzyj jeszcze raz żądanie HTTP wysyłane w momencie pobierania strony `http://example.org/`. Po kliknięciu przycisku *File Save* zawartość okna zapisze się w pliku `Downloads/HTTPHeaderLive.txt`. Zmień zawartość tego pliku, tak żeby zawierał tylko nagłówek żądania HTTP pobierającego stronę HTML i następujący po nich pusty wiersz. Następnie zmień w pierwszym wierszu napis

  ```
  http://example.org/
  ```

  na

  ```
  GET / HTTP/1.1
  ```

  Otrzymany plik powinien zawierać zapytanie HTTP podobne do:

  ```
  GET / HTTP/1.1
  Host: example.org
  User-Agent: ...
  ...
  <wiersz-odstępu>
  ```

  Wyślij to zapytanie do serwera WWW (tj. do portu `80` adresu IP związanego z nazwą `example.org`) poleceniem

```
V0$> nc -q 3 example.org 80 < HTTPHeaderLive.txt
```

Opcja `-q 3` czeka 3 sekundy przed zamknięciem połączenia. Na ekranie wyświetli się odpowiedź serwera WWW, ale będzie ona nieczytelna dla człowieka. Problematyczny okazuje się wiersz `Accept-Encoding: gzip, deflate` proszący serwer WWW o kompresję przesyłanych danych. Usuń ten wiersz z pliku `HTTPHeaderLive.txt` i spróbuj ponownie. Obejrzyj przesyłane pakiety w Wiresharku.

- Sprawdź, czy uzyskasz odpowiedź, jeśli w pliku `HTTPHeaderLive.txt` pozostawisz jedynie dwa pierwsze wiersze (zaczynające się od `GET` i `Host:`) i następujący po nich pusty wiersz. Ponownie obejrzyj pakiety w Wiresharku. Co stanie się, jeśli zostawisz tylko pierwszy wiersz i wiersz odstępu?

- Poleceniem

```
V0$> telnet example.org 80
```

otwórz strumień danych do serwera WWW na komputerze `example.org`. Wpisz tam zapytanie HTTP, czyli wiersze

```
GET / HTTP/1.1
Host: example.org
```

a następnie pusty wiersz. W odpowiedzi otrzymasz kolejny raz powyższą stronę WWW.

- Poleceniami

```
V0$> netstat -l46n
V0$> netstat -l46
```

wyświetl uruchomione na Twoim komputerze usługi „przypięte" do konkretnych portów warstwy transportowej. Pierwsze polecenie wyświetla wartości numeryczne, drugie zaś stara się je interpretować wykorzystując plik `/etc/services` (obejrzyj ten plik).

Uruchom serwer SSH poleceniem

```
V0#> systemctl start ssh
```

i ponownie wyświetl listę usług poleceniami `netstat`.

- Wybierz kilka lokalnych usług wykorzystujących protokół TCP, w tym usługę SSH (port 22), serwer echa (port 7) i serwer czasu (port 13). Za pomocą programu `telnet` połącz się z nimi w interaktywny sposób i wyślij do tych usług jakieś dane. Przykładowo z portem 7 połączysz się poleceniem

```
V0$> telnet localhost 7
```

Nazwa `localhost` zostanie zamieniona na adres IP maszyny wirtualnej, w której aktualnie pracujesz, tzn. powyższe polecenie utworzy połączenie z działającą lokalnie usługą (serwerem echa) „przypiętą" do portu 7. Aby rozłączyć się, naciśnij kombinację `Ctrl` + `]` i następnie wpisz polecenie `quit`.

Na końcu zamknij maszynę wirtualną *Virbian0*.

Tutorial #2

Zmień konfigurację maszyn *Virbian0* i *Virbian1*, tak żeby ich pierwsze (i jedyna) karty sieciowe były podłączone do wirtualnej sieci `local0`. Następnie uruchom obie maszyny.

- Na obu maszynach wyświetl dostępne interfejsy sieciowe poleceniami

  ```
  Vi$> ip link
  Vi$> ip addr
  ```

  Aktywne interfejsy oznaczone są napisem `UP`, nieaktywne — `DOWN`. Drugie z tych poleceń wyświetla dodatkowo przypisane do interfejsów adresy IP. Podobną informację można również uzyskać za pomocą starszego polecenia

  ```
  Vi$> ifconfig -a
  ```

- Interfejsy `enp0s3` obu maszyn są połączone ze sobą wirtualną siecią, ale obecnie nie mają one przypisanych adresów IP. Poleceniem

  ```
  Vi$> ethtool enp0s3
  ```

  sprawdź status warstwy fizycznej karty `enp0s3`. Zwróć uwagę na pola `Speed` i `Duplex`. Deklarowana szybkość połączenia powinna wynosić 1 Gbit/s.[1]

- Aktywuj interfejsy `enp0s3` i nadaj im odpowiednie adresy IP poleceniami:

  ```
  V0#> ip link set up dev enp0s3
  V0#> ip addr add 192.168.0.1/24 dev enp0s3
  V1#> ip link set up dev enp0s3
  V1#> ip addr add 192.168.0.2/24 dev enp0s3
  ```

  Wartość `/24` jest tzw. maską podsieci i jej znaczenie zostanie wyjaśnione na przyszłych zajęciach. Sprawdź, jak zmieniła się informacja wyświetlana przez polecenia `ip link` i `ip addr`. Jeśli przypadkowo nadasz karcie `enp0s3` błędny adres IP, możesz usunąć wszystkie przypisane do tej karty adresy poleceniem `ip addr flush dev enp0s3`.

- Polecenie `ping` służy do testowania warstwy sieciowej. W polu danych pakietów IP wysyłane są wtedy specjalne komunikaty protokołu ICMP. Wykonaj polecenie

  ```
  V0$> ping 192.168.0.2
  ```

  Jaki jest wyświetlany RTT (*round trip time*)? Uruchom program Wireshark (na dowolnej z maszyn) i włącz w nim obserwację wszystkich interfejsów (wybierając sztuczny interfejs *any*). Obejrzyj pakiety wysyłane i odbierane przez program `ping`. Czy znaczniki czasowe (pole *timestamp*) w wysyłanym zapytaniu i odpowiedzi różnią się, czy są takie same?

[1] Niestety jak się okaże później w przypadku kart wirtualnych informacje te nie są do końca prawdziwe. Dodatkowo pole `Link detected` w przypadku fizycznej karty określa, czy z drugiej strony jest aktywna karta sieciowa. Tutaj natomiast będzie równe `yes`, jeśli tylko aktywujemy interfejs `enp0s3` maszyny wirtualnej.

- Na maszynie *Virbian0* zmodyfikuj plik `/etc/hosts`, tak aby zawierał następujący wiersz

  `192.168.0.2` ***`jakaś_nazwa`***

  Sprawdź, że polecenie `ping` działa też z wpisaną tutaj nazwą. Uwaga: takie przypisanie działa tylko lokalnie, na maszynie na której zostało skonfigurowane.

- Na maszynie *Virbian1* uruchom polecenie

  ```
  V1$> iperf3 -s
  ```

  zaś na maszynie *Virbian0* polecenie

  ```
  V0$> iperf3 -c 192.168.0.2
  ```

  Jaką prędkość przesyłania udaje Ci się uzyskać?

- Na końcu na obu maszynach usuń adres IP z interfejsu `enp0s3` i dezaktywuj ten interfejs poleceniami

  ```
  Vi#> ip addr flush dev enp0s3
  Vi#> ip link set down dev enp0s3
  ```

Wyłącz obie wirtualne maszyny.

Wyzwanie #1

- Utwórz dodatkową maszynę wirtualną *Virbian2*. Podłącz karty sieciowe *Adapter1* maszyn *Virbian1* i *Virbian2* do wirtualnej sieci `local1` i następnie uruchom obie maszyny.
- Aktywuj karty sieciowe w obu urządzeniach poleceniem `ip` i sprawdź stan warstwy fizycznej kart poleceniem `ethtool`.
- Karcie sieciowej maszyny *Virbian1* przypisz adres IP równy `192.168.100.1`, zaś karcie maszyny *Virbian2* adres `192.168.100.2`. Pamiętaj o masce podsieci `/24`.
- Poleceniem `ping` sprawdź, czy jedna maszyna jest osiągalna z drugiej. Jaki jest RTT? Obejrzyj przesyłane pakiety Wiresharkiem. Wskaż w pakiecie miejsce w którym przechowywany jest źródłowy i docelowy adres IP.
- Wykorzystaj program `iperf3`, żeby zbadać przepustowość połączenia między maszynami.
- Z maszyny *Virbian1* połącz się z serwerem echa maszyny *Virbian2*. Zaobserwuj przesyłane pakiety w Wiresharkach uruchomionych jednocześnie na obu maszynach.
- Zdekonfiguruj karty sieciowe obu maszyn i wyłącz wirtualne maszyny.

Materiały do kursu znajdują się w systemie SKOS: https://skos.ii.uni.wroc.pl/.

Marcin Bieńkowski