PODSTAWY SIECI KOMPUTEROWYCH

SPRAWOZDANIE

z realizacji zadania

Warstwa transportowa TCP, UDP (LR4)

Autor: posk_LR4_Michał_Gebel.odt

Wartość <I1> występująca w zadaniu: 79

UWAGA: W zadaniu posłużono się skróconymi określeniami typu „Otwórz port”, „Nawiąż połączenie z portem”. Przygotuj sobie odpowiedź na pytanie: jakich sformułowań należałoby użyć aby były one formalnie poprawne?

1. Na obu maszynach wirtualnych wyłącz interfejsy *enp0s3*, aby uniknąć ewentualnych konfliktów w adresach, oraz skonfiguruj interfejsy *enp0s8* nadając im adresy z podsieci 192.168.<I1>.0/24. W obu maszynach wyłącz też dla interfejsu *enp0s8* opcje transferu dużych segmentów przez interfejs (https://wiki.linuxfoundation.org/networking/tso), wykonując jako użytkownik root polecenia: *ethtool -K enp0s8 tso off*. Wszystkie dalsze eksperymenty, o ile treść zadania nie mówi inaczej, wykonuj używając nadanych w tym punkcie adresów (a co za tym idzie, interfejsów). Na jednej z maszyn wirtualnych uruchom program *wireshark*, włącz nasłuchiwanie na interfejsie *enp0s8* i nie ograniczaj prezentowanych wyników.

```
[root@229879one ~]# ip link set enp0s3 down
[root@229879one ~]# ip addr change 192.168.79.2/24 dev enp0s8
[root@229879one ~]# ethtool -K enp0s8 tso off
[root@229879one ~]# ip a show enp0s8
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 1000
    link/ether 08:00:27:96:66:3a brd ff:ff:ff:ff:ff:ff
    inet 192.168.79.2/24 scope global enp0s8
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe96:663a/64 scope link
       valid_lft forever preferred_lft forever
```

```
[root@229879two ~]# ip link set enp0s3 down
[root@229879two ~]# ip addr change 192.168.79.3/24 dev enp0s8
[root@229879two ~]# ethtool -K enp0s8 tso off
[root@229879two ~]# ip a show enp0s8
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 1000
    link/ether 08:00:27:04:ca:2e brd ff:ff:ff:ff:ff:ff
    inet 192.168.79.3/24 scope global enp0s8
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe04:ca2e/64 scope link
       valid_lft forever preferred_lft forever
```

2. Na jednej z maszyn **zaprezentuj listę gniazd TCP otwartych do nasłuchu (ss)**. **Jako użytkownik *lsk* spróbuj otworzyć do nasłuchu gniazdo TCP z portem 80 (ncat -l). Następnie powtórz tę próbę jako użytkownik *root.*** Jaka jest przyczyna niepowodzenia w obu przypadkach?

```
[root@229879one ~]# ss -lnt
State      Recv-Q Send-Q Local Address:Port               Peer Address:Port
LISTEN     0      9            *:21                        *:*
LISTEN     0      128          *:22                        *:*
LISTEN     0      128         :::80                       :::*
LISTEN     0      9           :::21                       :::*
LISTEN     0      128         :::22                       :::*
LISTEN     0      128         :::443                      :::*
LISTEN     0      64          :::7                        :::*
```

```
[lsk@229879one ~]$ ncat -l 80
Ncat: bind to :::80: Permission denied. QUITTING.
[lsk@229879one ~]$ su -
Hasło:
[root@229879one ~]# ncat -l 80
Ncat: bind to :::80: Address already in use. QUITTING.
```

3. Na jednej z maszyn uruchom program *wireshark* i włącz nasłuchiwanie *na interfejsie pętli zwrotnej*. Na tej samej maszynie **otwórz za pomocą programu *ncat* dowolny nieuprzywilejowany port TCP do nasłuchu**. W innym terminalu tej samej maszyny użyj **programu *ncat* aby przesłać krótki (kilkuznakowy) komunikat** do procesu *ncat* uruchomionego poprzednio (echo, |, ncat). **Zaprezentuj sekwencje segmentów prowadzących do nawiązania i zamknięcia połączenia**. **Powtórz eksperyment korzystając tym razem z protokołu UDP**. Jak oceniasz efektywność protokołu TCP w przypadku tak krótkich transmisji?

Otwieram port 2137 do nasłuchu:

```
[root@229879one ~]# ncat -l 2137
```

Następnie w drugim terminalu wysyłam wiadomość:

```
[lsk@229879one ~]$ echo testujemy | ncat localhost 2137
```

W pierwszym terminalu otrzymujemy:

```
[root@229879one ~]# ncat -l 2137
```

testujemy

1	0.000000000	::1	::1	TCP	94 33494 → 2137 [SYN] Seq=0 Win=43690 Len=0 MSS=65476 SACK_PERM=1 TSval=717183243 TSecr=0 WS=64
2	0.000017451	::1	::1	TCP	94 2137 → 33494 [SYN, ACK] Seq=0 Ack=1 Win=43690 Len=0 MSS=65476 SACK_PERM=1 TSval=717183243 TSecr=717183243 WS=64
3	0.000027171	::1	::1	TCP	86 33494 → 2137 [ACK] Seq=1 Ack=1 Win=43712 Len=0 TSval=717183243 TSecr=717183243
4	0.000105632	::1	::1	TCP	96 33494 → 2137 [PSH, ACK] Seq=1 Ack=1 Win=43712 Len=10 TSval=717183243 TSecr=717183243
5	0.000111532	::1	::1	TCP	86 2137 → 33494 [ACK] Seq=1 Ack=11 Win=43712 Len=0 TSval=717183243 TSecr=717183243
6	0.000172152	::1	::1	TCP	86 33494 → 2137 [FIN, ACK] Seq=11 Ack=1 Win=43712 Len=0 TSval=717183243 TSecr=717183243
7	0.000186194	::1	::1	TCP	86 2137 → 33494 [FIN, ACK] Seq=1 Ack=12 Win=43712 Len=0 TSval=717183243 TSecr=717183243
8	0.000190648	::1	::1	TCP	86 33494 → 2137 [ACK] Seq=12 Ack=2 Win=43712 Len=0 TSval=717183243 TSecr=717183243

Otwieram port 2137 do nasłuchu korzystając z protokołu UDP:

```
[root@229879one ~]# ncat -ul 2137
```

Następnie w drugim terminalu wysyłam wiadomość:

```
[lsk@229879one ~]$ echo testujemyUDP | ncat -u localhost 2137
```

W pierwszym terminalu otrzymujemy:

```
[root@229879one ~]# ncat -ul 2137
testujemyUDP
```

9	340.9357310(	::1	::1	UDP	75 57654 → 2137 Len=13

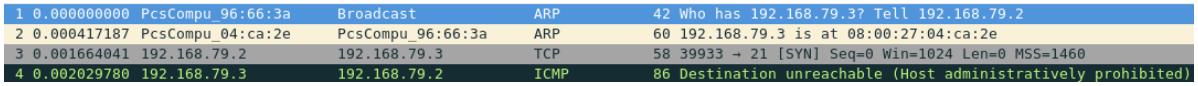
Protokół UDP jest szybszy poprzez mniejszą ilość wysyłanych komunikatów do portów w zestawieniu z TCP, u którego ta sama operacja zajmuje 8 komunikatów, jednak TCP zapewnia cały datagram.

4. Na jednej z maszyn za pomocą programu *nmap* **zbadaj dostępność portów w zakresie 1-100** (nmap -p z konta użytkownika *root*), za pierwszym razem dla tej samej maszyny (użyj adresu interfejsu pętli zwrotnej) zaś za drugim razem dla drugiej maszyny. UWAGA! Program nmap skraca wyświetlany wynik traktując te porty, które są w najczęściej występującym stanie, jako „pozostałe" (*not shown*). W obu przypadkach porty te mają inne stany i trzeba wziąć to pod uwagę analizując wynik.

```
[root@229879one ~]# nmap -p 1-100 localhost

Starting Nmap 7.40 ( https://nmap.org ) at 2020-12-20 17:28 CET
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled.
Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000080s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 97 closed ports
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds

[root@229879one ~]# nmap -p 1-100 192.168.79.3

Starting Nmap 7.40 ( https://nmap.org ) at 2020-12-20 17:38 CET
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled.
Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.79.3
Host is up (0.00058s latency).
Not shown: 96 filtered ports
PORT   STATE  SERVICE
7/tcp  closed echo
22/tcp open   ssh
23/tcp closed telnet
80/tcp open   http
MAC Address: 08:00:27:04:CA:2E (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 2.08 seconds
```

5. Porównując wyniki poprzedniego eksperymentu wybierz trzy numery portów o następujących stanach: gniazdo otwarte do nasłuchu lecz połączenia blokowane przez filtr pakietów (*firewall*); gniazdo otwarte do nasłuchu i połączenia nie blokowane; brak otwartego gniazda, ale połączenia nie blokowane. Dla tych trzech portów (dla każdego z osobna lub wymieniając je na liście opcji -p) **przeprowadź ponownie próby programem nmap**. **Przedstaw historię komunikacji w programie *wireshark***. Na jakiej podstawie program nmap odróżnił stany tych portów? Czy możliwa jest inna reakcja filtra pakietów na niechciane połączenia?

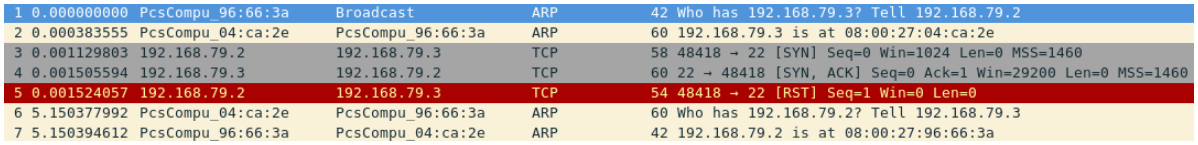
```
[root@229879one ~]# nmap -p 21 192.168.79.3

Starting Nmap 7.40 ( https://nmap.org ) at 2020-12-20 17:49 CET
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled.
Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.79.3
Host is up (0.00043s latency).
PORT   STATE    SERVICE
21/tcp filtered ftp
MAC Address: 08:00:27:04:CA:2E (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds
```



1 0.000000000	PcsCompu_96:66:3a	Broadcast	ARP	42 Who has 192.168.79.3? Tell 192.168.79.2
2 0.000417187	PcsCompu_04:ca:2e	PcsCompu_96:66:3a	ARP	60 192.168.79.3 is at 08:00:27:04:ca:2e
3 0.001664041	192.168.79.2	192.168.79.3	TCP	58 39933 → 21 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4 0.002029780	192.168.79.3	192.168.79.2	ICMP	86 Destination unreachable (Host administratively prohibited)

W przypadku portu 21 otrzymujemy komunikat ICMP „Host administratively prohibited”

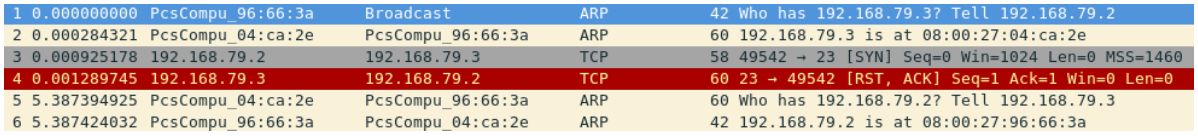
```
[root@229879one ~]# nmap -p 22 192.168.79.3

Starting Nmap 7.40 ( https://nmap.org ) at 2020-12-20 17:51 CET
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled.
Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.79.3
Host is up (0.00039s latency).
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 08:00:27:04:CA:2E (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds
```



1 0.000000000	PcsCompu_96:66:3a	Broadcast	ARP	42 Who has 192.168.79.3? Tell 192.168.79.2
2 0.000383555	PcsCompu_04:ca:2e	PcsCompu_96:66:3a	ARP	60 192.168.79.3 is at 08:00:27:04:ca:2e
3 0.001129803	192.168.79.2	192.168.79.3	TCP	58 48418 → 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4 0.001505594	192.168.79.3	192.168.79.2	TCP	60 22 → 48418 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460
5 0.001524057	192.168.79.2	192.168.79.3	TCP	54 48418 → 22 [RST] Seq=1 Win=0 Len=0
6 5.150377992	PcsCompu_04:ca:2e	PcsCompu_96:66:3a	ARP	60 Who has 192.168.79.2? Tell 192.168.79.3
7 5.150394612	PcsCompu_96:66:3a	PcsCompu_04:ca:2e	ARP	42 192.168.79.2 is at 08:00:27:96:66:3a

W przypadku portu 22 otrzymujemy segment ACK, dzięki któremu wiemy, że jest otwarty.

```
[root@229879one ~]# nmap -p 23 192.168.79.3

Starting Nmap 7.40 ( https://nmap.org ) at 2020-12-20 17:57 CET
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled.
Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.79.3
Host is up (0.00030s latency).
PORT   STATE  SERVICE
23/tcp closed telnet
MAC Address: 08:00:27:04:CA:2E (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds
```



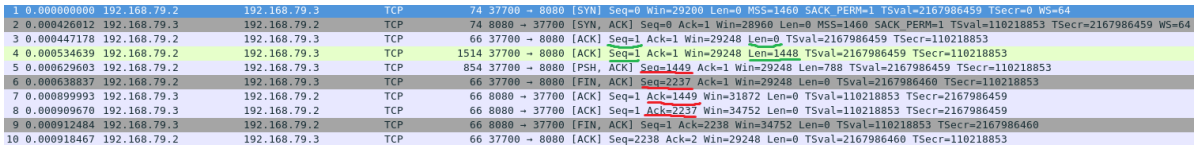
1 0.000000000	PcsCompu_96:66:3a	Broadcast	ARP	42 Who has 192.168.79.3? Tell 192.168.79.2
2 0.000284321	PcsCompu_04:ca:2e	PcsCompu_96:66:3a	ARP	60 192.168.79.3 is at 08:00:27:04:ca:2e
3 0.000925178	192.168.79.2	192.168.79.3	TCP	58 49542 → 23 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4 0.001289745	192.168.79.3	192.168.79.2	TCP	60 23 → 49542 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
5 5.387394925	PcsCompu_04:ca:2e	PcsCompu_96:66:3a	ARP	60 Who has 192.168.79.2? Tell 192.168.79.3
6 5.387424032	PcsCompu_96:66:3a	PcsCompu_04:ca:2e	ARP	42 192.168.79.2 is at 08:00:27:96:66:3a

W przypadku portu 23 otrzymujemy segment ACK, dzięki któremu wiemy, że jest zamknięty.

6. Na jednej z maszyn **otwórz za pomocą programu *ncat* port 8080 TCP do nasłuchu**. Z poziomu drugiej maszyny **prześlij do tego procesu zawartość pliku */etc/passwd***. **Zaprezentuj w programie *wireshark* historię wymiany segmentów** tak, aby widoczna była relacja pomiędzy długością wysyłanych segmentów a zmianą numeru sekwencyjnego, a także relacja pomiędzy numerem sekwencji a numerem potwierdzenia dla jednego z kierunków transmisji.

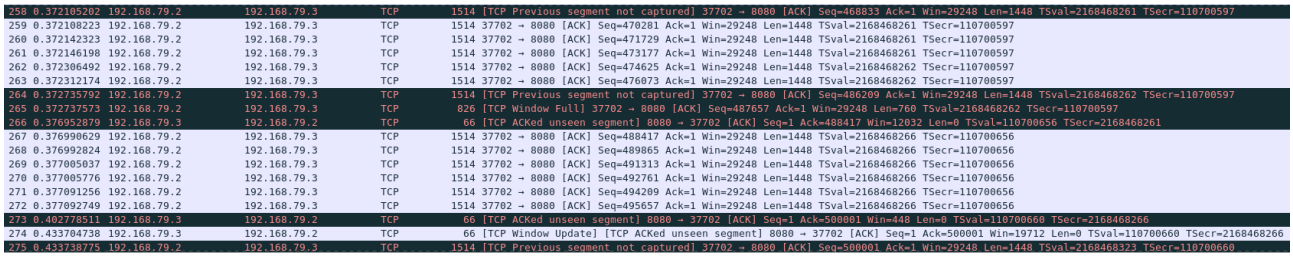
```
[root@229879two ~]# ncat -l 8080
[root@229879one ~]# cat /etc/passwd | ncat 192.168.79.3 8080
[root@229879two ~]# ncat -l 8080
```

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
systemd-timesync:x:999:998:systemd Time Synchronization:/:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:998:997:User for polkitd:/:/sbin/nologin
rtkit:x:172:172:RealtimeKit:/proc:/sbin/nologin
pulse:x:171:171:PulseAudio System Daemon:/var/run/pulse:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
geoclue:x:997:993:User for geoclue:/var/lib/geoclue:/sbin/nologin
chrony:x:996:992::/var/lib/chrony:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
setroubleshoot:x:995:989::/var/lib/setroubleshoot:/sbin/nologin
colord:x:994:988:User for colord:/var/lib/colord:/sbin/nologin
abrt:x:173:173::/etc/abrt:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
tss:x:59:59:Account used by the trousers package to sandbox the tcsd
daemon:/dev/null:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
tcpdump:x:72:72::/:/sbin/nologin
lsk:x:1000:1000:Laboratorium Sieci Komputerowych:/home/lsk:/bin/bash
systemd-coredump:x:987:987:systemd Core Dumper:/:/sbin/nologin
joe:x:1001:1001::/home/joe:/bin/bash
jane:x:1002:1002::/home/jane:/bin/bash
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
squid:x:23:23::/var/spool/squid:/sbin/nologin
vboxadd:x:986:1::/var/run/vboxadd:/bin/false
nginx:x:985:984:Nginx web server:/var/lib/nginx:/sbin/nologin
openvpn:x:984:983:OpenVPN:/etc/openvpn:/sbin/nologin
nm-openvpn:x:983:982:Default user for running openvpn spawned by
NetworkManager:/:/sbin/nologin
```



Kolorem zielonym zaznaczono relację pomiędzy długością wysłanego segmentu, a zmianą numeru sekwencyjnego, a kolorem czerwonym relacja pomiędzy numerem sekwencji, a numerem potwierdzenia.

7. **Powtórz poprzedni eksperyment przesyłając plik */etc/services***. UWAGA! Ważne jest, aby program *ncat* odbierający zawartość pliku wyprowadzał odebrane dane na swoje wyjście standardowe. **W historii wymiany segmentów zaprezentuj incydent polegający na wypełnieniu bufora odbiorcy**. W jaki sposób odbiorca przekazał nadawcy informacje wypełnieniu bufora, a następnie o wolnym miejscu w buforze?



W momencie wystąpienie przepełnienia bufora zostanie wysłany komunikat „TCP Window Full” do nadawcy. Po opróżnieniu bufora zostanie wysłany komunikat „TCP Windows Update” i przessył zostanie wznowiony.