高度サイバーセキュリティ PBLI

08D19043
川原尚己

演習(1-1)

- 172.16.128.0/25 のネットワークアドレス，サブネットマスク，ブロードキャストアドレスを計算せよ．
  ネットワークアドレス：172.16.128.0
  サブネットマスク：255.255.255.128
  ブロードキャストアドレス：172.16.128.127
- 上記ネットワークに収容可能なホストアドレス数を計算せよ．
  7 ビット分をホストアドレスに割り当てることができるが, その中にはネットワークアドレスとブロードキャストアドレスが含まれているため，求める値は，

$$2^7 - 2 = 126$$

演習(1-2)

アドレスの上位 16 ビットはすべて一致しているから，下 16 ビット分だけを考える．
ルーティングテーブル上の各アドレスの下 16 ビットは以下のように表される：
192.168.128.0/20：10000000.00000000
192.168.129.0/24：10000001.00000000
192.168.2.0/24：00000010.00000000

- 192.168.137.10：10001001.00001010
  これは 192.168.128.0/20 のネットワークアドレス部分が完全一致しており，他の二つはしていないため，IF は eth0 が選ばれる．

- 192.168.201.67：11001001.00100011
  これはルーティングテーブルのアドレスのどのネットワークアドレスとも完全一致していないため，デフォルトルートが選択される．よって，IF は eth0 が選ばれる．

演習(1-3)

以下にルーティングテーブルを示す．

宛先 NW	IPv4 アドレス	IF
0.0.0.0/0	172.16.0.2	eth0
172.16.0.0/24		eth0

ノード A のルーティングテーブル

宛先 NW	IPv4 アドレス	IF
172.16.0.0/18		eth0
172.16.0.0/24		eth1
172.16.2.0/24		eth2

ノード B のルーティングテーブル

宛先 NW	IPv4 アドレス	IF
0.0.0.0/0	172.16.2.2	eth0
172.16.2.0/24		eth0

ノード C のルーティングテーブル

宛先 NW	IPv4 アドレス	IF
0.0.0.0/0	172.16.192.2	eth0
172.16.0.0/18		eth0

ノード D のルーティングテーブル

演習(1-4)

まず，ドメイン(www.osaka-u.ac.jp)を DNS のルートサーバに問い合わせ，Top Level Domain(.jp)の管理を行っているサーバの IP を取得する．次に，このサーバにドメインを問い合わせ，Second Level Domain(.ac)の管理サーバの IP アドレスを取得する．この操作を繰り返し，ドメイン全体に対応する IP アドレスを取得することができる．(アプリケーション層)

次に，通信相手とのコネクションの確率を行う．まず，クライアント側が通信先に SYN フラグをシーケンス番号と確認応答番号とともに送信する．次に，通信先側がクライアント側に SYN フラグと ACK フラグを送信する．最後に，クライアント側が通信先に ACK フラグを送信し，コネクションが確立される．(トランスポート層)

一般のネットワークにおいては，一つのグローバル IP アドレスに対し，複数のプライベート IP アドレスが存在する．

そこで，NAPT を用いて一つのプライベート IP アドレスとグローバル IP アドレスを一対一に対応付けるために，プライベート IP アドレスとグローバル IP アドレスの末尾にポート番号を付加し，それらの組をテーブルにキャッシュする．(ネットワーク層)

ARP によって，IP アドレスから MAC アドレスへの対応を取得する．IP での通信の際に毎回 ARP を用いて問い合わせていると非効率であるため，初めに通信を行い IP アドレスと MAC アドレスの対応をテーブルとしてキャッシュしておく．IP アドレス$addr_{IP}$を誰が持っているかをブロードキャストで問い合わせ，$addr_{IP}$を持つ人が問い合わせた人に MAC アドレス$addr_{MAC}$を返答する．最後に返答されてきた$addr_{IP}$と$addr_{MAC}$の組をキャッシュする．(データリンク層)

以上によって TCP セッションが開始される．

演習(2-1)

h1:

sudo ip address add 172.16.0.10/24 dev ens4

sudo ip link set up dev ens4

h2:

sudo ip address add 172.16.0.20/24 dev ens4

sudo ip address add 172.20.0.20/24 dev ens4

sudo ip link set up dev ens4

sudo ip link set up dev ens5

h3:

sudo ip address add 172.16.0.30/24 dev ens4

sudo ip link set up dev ens4

以上のコマンドで h1-h2 間，h2-h3 間の通信が可能となる．h1-h3 間の通信にはルーティングテーブルを与える必要があり，以下のコマンドで得られる．

h1:

sudo ip route add 172.20.0.0/24 via 172.16.0.20

h3:

sudo ip route add172.16.0.0/24 via 172.20.0.20

ip address show, ip route show で得られる出力は以下の通り．

h1:

ip address show dev ens4:

ip route show :

```
enpit@h1:~$ ip route show
10.10.0.0/24 dev ens3 proto kernel scope link src 10.10.0.1 metric 100
172.16.0.0/24 dev ens4 proto kernel scope link src 172.16.0.10
172.20.0.0/24 via 172.16.0.20 dev ens4
```

h2:

ip address show:

```
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:50:00:00:01:00 brd ff:ff:ff:ff:ff:ff
    altname enp0s3
    inet 10.10.0.1/24 metric 100 brd 10.10.0.255 scope global dynamic ens3
       valid_lft 443sec preferred_lft 443sec
    inet 192.168.1.10/24 scope global ens3
       valid_lft forever preferred_lft forever
    inet 171.16.0.10/24 scope global ens3
       valid_lft forever preferred_lft forever
    inet6 fe80::250:ff:fe00:100/64 scope link
       valid_lft forever preferred_lft forever
3: ens4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:50:00:00:01:01 brd ff:ff:ff:ff:ff:ff
    altname enp0s4
    inet 172.16.0.10/24 scope global ens4
       valid_lft forever preferred_lft forever
    inet6 fe80::250:ff:fe00:101/64 scope link
       valid_lft forever preferred_lft forever
```

ip route show:

```
10.10.0.0/24 dev ens3 proto kernel scope link src 10.10.0.1 metric 100
172.16.0.0/24 dev ens4 proto kernel scope link src 172.16.0.10
172.20.0.0/24 via 172.16.0.20 dev ens4
```

h3:

ip address show dev ens4:

```
enpit@h3:~$ ip address show dev ens4
3: ens4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:50:00:00:03:01 brd ff:ff:ff:ff:ff:ff
    altname enp0s4
    inet 172.20.0.30/24 scope global ens4
       valid_lft forever preferred_lft forever
    inet6 fe80::250:ff:fe00:301/64 scope link
       valid_lft forever preferred_lft forever
```

ip route show:

```
10.10.0.0/24 dev ens3 proto kernel scope link src 10.10.0.3 metric 100
172.16.0.0/24 via 172.20.0.20 dev ens4
172.20.0.0/24 dev ens4 proto kernel scope link src 172.20.0.30
```

演習(2-2)

h1:

sudo ip address add 172.16.0.10/24 dev ens4

sudo ip link set up dev ens4

h2:

sudo ip address add 172.16.0.20/24 dev ens4

sudo ip address add 172.16.1.20/24 dev ens5
sudo ip address add 172.20.0.20/24 dev ens6
sudo ip link set up dev ens4
sudo ip link set up dev ens5
sudo ip link set up dev ens6

h3:
sudo ip address add 172.20.0.30/24 dev ens4
sudo ip link set up dev ens4

h4:
sudo ip address add 172.16.1.10/24 dev ens4
sudo ip link set up dev ens4

以上のコマンドで隣り合ったノード間の通信が可能となる．その他のノード同士の通信にはルーティングテーブルを与える必要があり，以下のコマンドで得られる．

h1:
sudo ip route add 172.20.0.0/24 via 172.16.0.20
sudo ip route add 172.16.1.0/24 via 172.16.0.20

h3:
sudo ip route add 172.16.0.0/24 via 172.20.0.20
sudo ip route add 172.16.1.0/24 via 172.20.0.20

h4:
sudo ip route add 172.16.0.0/24 via 172.16.1.20
sudo ip route add 172.20.0.0/24 via 172.16.1.20

ip address show, ip route show で得られる出力は以下の通り．

h1:
ip address show dev ens4:

```
3: ens4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:50:00:00:01:01 brd ff:ff:ff:ff:ff:ff
    altname enp0s4
    inet 172.16.0.10/24 scope global ens4
       valid_lft forever preferred_lft forever
    inet6 fe80::250:ff:fe00:101/64 scope link
       valid_lft forever preferred_lft forever
```

ip route show :

```
enpit@h1:~$ ip route show
10.10.0.0/24 dev ens3 proto kernel scope link src 10.10.0.1 metric 100
172.16.0.0/24 dev ens4 proto kernel scope link src 172.16.0.10
172.16.1.0/24 via 172.16.0.20 dev ens4
172.20.0.0/24 via 172.16.0.20 dev ens4
```

h2:

ip address show:

```
3: ens4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:50:00:00:02:01 brd ff:ff:ff:ff:ff:ff
    altname enp0s4
    inet 172.16.0.20/24 scope global ens4
       valid_lft forever preferred_lft forever
    inet6 fe80::250:ff:fe00:201/64 scope link
       valid_lft forever preferred_lft forever
4: ens5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:50:00:00:02:02 brd ff:ff:ff:ff:ff:ff
    altname enp0s5
    inet 172.16.1.20/24 scope global ens5
       valid_lft forever preferred_lft forever
    inet6 fe80::250:ff:fe00:202/64 scope link
       valid_lft forever preferred_lft forever
5: ens6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:50:00:00:02:03 brd ff:ff:ff:ff:ff:ff
    altname enp0s6
    inet 172.20.0.20/24 scope global ens6
       valid_lft forever preferred_lft forever
    inet6 fe80::250:ff:fe00:203/64 scope link
       valid_lft forever preferred_lft forever
```

ip route show:

```
enpit@h2:~$ ip route show
10.10.0.0/24 dev ens3 proto kernel scope link src 10.10.0.2 metric 100
172.16.0.0/24 dev ens4 proto kernel scope link src 172.16.0.20
172.16.1.0/24 dev ens5 proto kernel scope link src 172.16.1.20
172.20.0.0/24 dev ens6 proto kernel scope link src 172.20.0.20
```

h3:

ip address show dev ens4:

```
enpit@h3:~$ ip address show dev ens4
3: ens4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:50:00:00:03:01 brd ff:ff:ff:ff:ff:ff
    altname enp0s4
    inet 172.20.0.30/24 scope global ens4
       valid_lft forever preferred_lft forever
    inet6 fe80::250:ff:fe00:301/64 scope link
       valid_lft forever preferred_lft forever
```

ip route show:

```
enpit@h3:~$ ip route show
10.10.0.0/24 dev ens3 proto kernel scope link src 10.10.0.3 metric 100
172.16.0.0/24 via 172.20.0.20 dev ens4
172.16.1.0/24 via 172.20.0.20 dev ens4
172.20.0.0/24 dev ens4 proto kernel scope link src 172.20.0.30
```

h4:

ip address show dev ens4:

```
enpit@h4:~$ ip route show
10.10.0.0/24 dev ens3 proto kernel scope link src 10.10.0.4 metric 100
172.16.0.0/24 via 172.16.1.20 dev ens4
172.16.1.0/24 dev ens4 proto kernel scope link src 172.16.1.10
172.20.0.0/24 via 172.16.1.20 dev ens4
```

ip route show:

```
enpit@h4:~$ ip route show
10.10.0.0/24 dev ens3 proto kernel scope link src 10.10.0.4 metric 100
172.16.0.0/24 via 172.16.1.20 dev ens4
172.16.1.0/24 dev ens4 proto kernel scope link src 172.16.1.10
172.20.0.0/24 via 172.16.1.20 dev ens4
```

演習(2-3)

```
enpit@h2:~$ sudo iptables -L --line-num
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination

Chain FORWARD (policy DROP)
num  target     prot opt source               destination
1    FIREWALL   all  --  172.16.0.10          172.20.0.0/24
2    FIREWALL   all  --  172.16.1.10          172.20.0.0/24
3    ACCEPT     tcp  --  172.16.0.0/24        172.16.1.0/24        state NEW,ESTABLISHED
4    ACCEPT     udp  --  172.16.0.0/24        172.16.1.0/24        state NEW,ESTABLISHED
5    ACCEPT     icmp --  172.16.0.0/24        172.16.1.0/24        state NEW,ESTABLISHED
6    ACCEPT     tcp  --  172.20.0.0/24        172.16.0.0/24        state ESTABLISHED
7    ACCEPT     udp  --  172.20.0.0/24        172.16.0.0/24        state ESTABLISHED
8    ACCEPT     icmp --  172.20.0.0/24        172.16.0.0/24        state ESTABLISHED
9    ACCEPT     tcp  --  172.16.1.0/24        172.16.0.0/24        state ESTABLISHED
10   ACCEPT     udp  --  172.16.1.0/24        172.16.0.0/24        state ESTABLISHED
11   ACCEPT     icmp --  172.16.1.0/24        172.16.0.0/24        state ESTABLISHED
12   ACCEPT     tcp  --  172.20.0.0/24        172.16.1.0/24        state ESTABLISHED
13   ACCEPT     udp  --  172.20.0.0/24        172.16.1.0/24        state ESTABLISHED
14   ACCEPT     icmp --  172.20.0.0/24        172.16.1.0/24        state ESTABLISHED
15   ACCEPT     tcp  --  172.20.0.0/24        172.16.1.0/24        tcp dpt:http state NEW,ESTABLISHED
16   ACCEPT     udp  --  172.20.0.0/24        172.16.1.0/24        udp dpt:80 state NEW,ESTABLISHED
17   ACCEPT     tcp  --  172.20.0.0/24        172.16.1.0/24        tcp dpt:https state NEW,ESTABLISHED
18   ACCEPT     udp  --  172.20.0.0/24        172.16.1.0/24        udp dpt:https state NEW,ESTABLISHED
19   ACCEPT     tcp  --  172.16.1.0/24        172.20.0.0/24        tcp spt:http state ESTABLISHED
20   ACCEPT     udp  --  172.16.1.0/24        172.20.0.0/24        udp spt:80 state ESTABLISHED
21   ACCEPT     tcp  --  172.16.1.0/24        172.20.0.0/24        tcp spt:https state ESTABLISHED
22   ACCEPT     udp  --  172.16.1.0/24        172.20.0.0/24        udp spt:https state ESTABLISHED

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination

Chain FIREWALL (2 references)
num  target     prot opt source               destination
1    ACCEPT     tcp  --  172.16.0.0/24        172.20.0.0/24        state NEW,ESTABLISHED
2    ACCEPT     udp  --  172.16.0.0/24        172.20.0.0/24        state NEW,ESTABLISHED
3    ACCEPT     icmp --  172.16.0.0/24        172.20.0.0/24        state NEW,ESTABLISHED
4    ACCEPT     tcp  --  172.16.1.0/24        172.20.0.0/24        state NEW,ESTABLISHED
5    ACCEPT     udp  --  172.16.1.0/24        172.20.0.0/24        state NEW,ESTABLISHED
6    ACCEPT     icmp --  172.16.1.0/24        172.20.0.0/24        state NEW,ESTABLISHED
```

課題の条件をすべて満たす iptable は上の図で得られる.

まず，FORWARD チェインの policy を DROP にすることで，チェイン内で許可している通信以外はできないようにし，条件 6 及び 9 を達成している.

FIREWALL チェインの 1-3 行目と FORWARD チェインの 6-8 行目で条件 5「社内ネットワークから対外ネットワークの TCP, UDP, ICMP 接続が可能」を表している. FIREWALL チェインの 1-3 行目で通信を開始する際の接続を許可しており, FORWARD チェインの 6-8 行目で返答の際の接続を許可している. 同様に，FORWARD チェインの 3-5 及び 9-11 行目で条件後 5 の後半部分である「社内ネットワークから DMZ ネットワークの TCP, UDP, ICMP 接続が可能」を表している.

また，FIREWALL チェインの 4-6 行目と FORWARD チェインの 12-14 行目が条件 7 を満たしている.

FORWARD チェインの 15-22 行目が条件 8 を表している．15-18 行目が対外ネットワークから DMZ ネットワークへ接続するための許可で，19-22 行目が DMZ ネットワークから対外ネットワークへの返答のための許可である.

最後に，FORWARD チェインの 1,2 行目にて，FIREWALL チェインの 1-6 行目を参照し, h1 と h4 の持つ ip アドレスからの通信のみを許可することでソース IP アドレス詐称を

禁止している．

nc コマンド，mtr コマンド及び ping コマンドを用いて TCP，UDP，ICMP 接続を試している．
(送信先にて“sudo nc -u -l -p <ポート番号(80:http,443:https)>”及び送信元にて”sudo nc -u <送信先 IP アドレス> <ポート番号>”で UDP 接続, ”mtr -T <通信先 IP アドレス>”で TCP 接続，”ping <通信先 IP アドレス>”で ICMP 接続の通信ができる．)
以下のコマンド実行結果によって各条件を満たしていることの確認を行った．

- h3 にて，”sudo nc -u -l -p 80”，h1 にて，”sudo nc -u 172.20.0.30 80”，”mtr -T 172.20.0.30”，”ping 172.20.0.30”の各通信が可能であること(条件 5 の前半)
- h4 にて，”sudo nc -u -l -p 80”，h1 にて，” sudo nc -u 172.16.1.10 80”，”mtr -T 172.16.1.10”，”ping 172.16.1.10” の各通信が可能であること(条件 5 の後半)
- h1 にて，”sudo nc -u -l -p 80”，h4 にて，” sudo nc -u 172.16.0.10 80”，”mtr -T 172.16.0.10”，”ping 172.16.0.10”の各通信が不可能であること（条件 6）
- h3 にて，”sudo nc -u -l -p 80”，h4 にて，” sudo nc -u 172.20.0.30 80”，”mtr -T 172.20.0.30”，”ping 172.20.0.30”の各通信が可能であること（条件 7）
- h3 にて，”mtr -T 172.16.1.10 -P 80”，”mtr -T 172.16.1.10 -P 443”の各通信が可能であるが，他のポート番号では不可能であること（条件 8-TCP）
- h4 にて，”sudo nc -u -l -p 80”，h3 にて，” sudo nc -u 172.16.1.10 80”の通信及び，h4 にて，”sudo nc -u -l -p 443”，h3 にて，” sudo nc -u 172.16.1.10 443”の通信が可能であるが，他のポート番号では不可能であること．（条件 8-UDP）
- h1 にて，”sudo nc -u -l -p 80”，h3 にて，” sudo nc -u 172.16.0.10 80”，”mtr -T 172.16.0.10”，”ping 172.16.0.10”の各通信が不可能であること（条件 9）
- h1 にて，通常の IP アドレスでは h3 と通信を行えるが，”sudo iptables -t nat -A POSTROUTING -j SNAT –to-source 172.16.0.15”を入力し IP アドレスを詐称した場合には通信が不可能となること．（条件 10）（下図参照）
- h4 にて，通常の IP アドレスでは h3 と通信を行えるが，”sudo iptables -t nat -A POSTROUTING -j SNAT –to-source 172.16.1.15” を入力し IP アドレスを詐称した場合には通信が不可能となること．（条件 10）

```
enpit@h1:~$ ping 172.20.0.30
PING 172.20.0.30 (172.20.0.30) 56(84) bytes of data.
64 bytes from 172.20.0.30: icmp_seq=1 ttl=63 time=3.43 ms
64 bytes from 172.20.0.30: icmp_seq=2 ttl=63 time=1.65 ms
^C
--- 172.20.0.30 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 1.647/2.539/3.432/0.892 ms
enpit@h1:~$ sudo iptables -t nat -A POSTROUTING -j SNAT --to-source 172.16.0.15
enpit@h1:~$ ping 172.20.0.30
PING 172.20.0.30 (172.20.0.30) 56(84) bytes of data.
^C
--- 172.20.0.30 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 5115ms
```

```
enpit@h4:~$ ping 172.20.0.30
PING 172.20.0.30 (172.20.0.30) 56(84) bytes of data.
64 bytes from 172.20.0.30: icmp_seq=1 ttl=63 time=2.48 ms
64 bytes from 172.20.0.30: icmp_seq=2 ttl=63 time=1.38 ms
64 bytes from 172.20.0.30: icmp_seq=3 ttl=63 time=1.81 ms
^C
--- 172.20.0.30 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 1.377/1.889/2.484/0.455 ms
enpit@h4:~$ sudo iptables -t nat -A POSTROUTING -j SNAT --to-source 172.16.1.15
enpit@h4:~$ ping 172.20.0.30
PING 172.20.0.30 (172.20.0.30) 56(84) bytes of data.
^C
--- 172.20.0.30 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 5128ms
```

演習(2-7)

py-radix をインストールする過程で以下のようなエラーメッセージが出現し，インストールできなかった．

```
(base) PS C:¥Users¥owner> conda install -c activisiongamescience py-radix
Collecting package metadata (current_repodata.json): done
Solving environment: unsuccessful initial attempt using frozen solve. Retrying with flexible solve.
Collecting package metadata (repodata.json): done
Solving environment: unsuccessful initial attempt using frozen solve. Retrying with flexible solve.

PackagesNotFoundError: The following packages are not available from current channels:

  - py-radix
```

Conda update conda や conda update –all などのコマンドを実行したが結果が変わらず，実行できなかった．

Radix tree の実行としては，以下のサイト(https://sonickun.hatenablog.com/entry/2014/07/26/150612)を参考とすることで実装できるかと思われる．