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，

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](http://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アドレスを誰が持っているかをブロードキャストで問い合わせ，を持つ人が問い合わせた人にMACアドレスを返答する．最後に返答されてきたと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

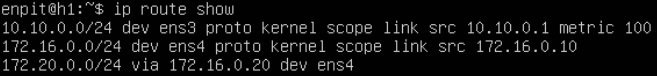
ip address show, ip route show で得られる出力は以下の通り．

h1:

ip address show dev ens4:

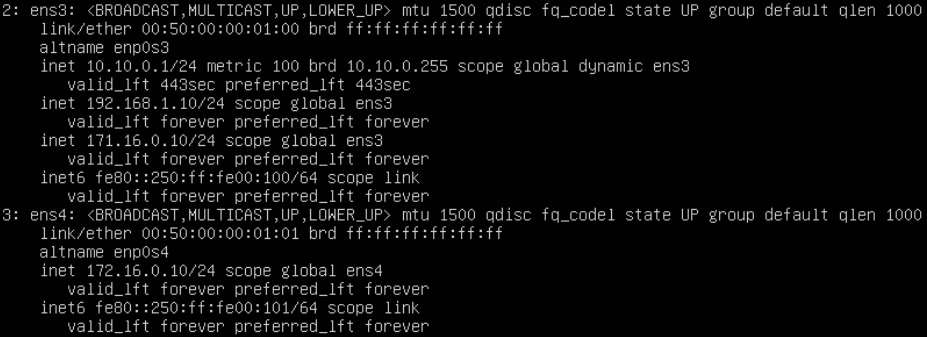


ip route show :



h2:

ip address show:



ip route show:



h3:

ip address show dev ens4:



ip route show:



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:



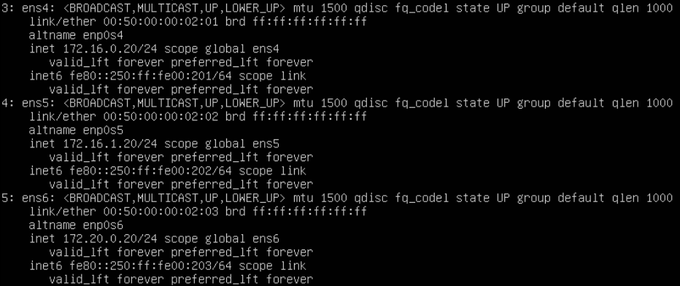
ip route show :

テキスト

自動的に生成された説明

h2:

ip address show:



ip route show:

テキスト

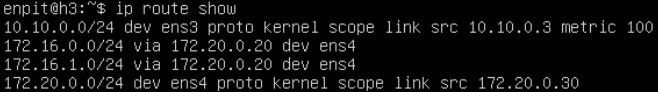
自動的に生成された説明

h3:

ip address show dev ens4:

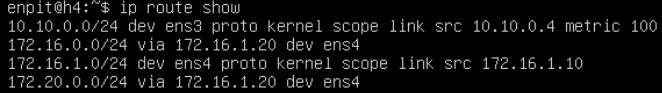


ip route show:

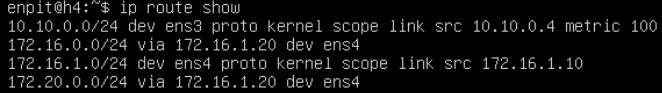


h4:

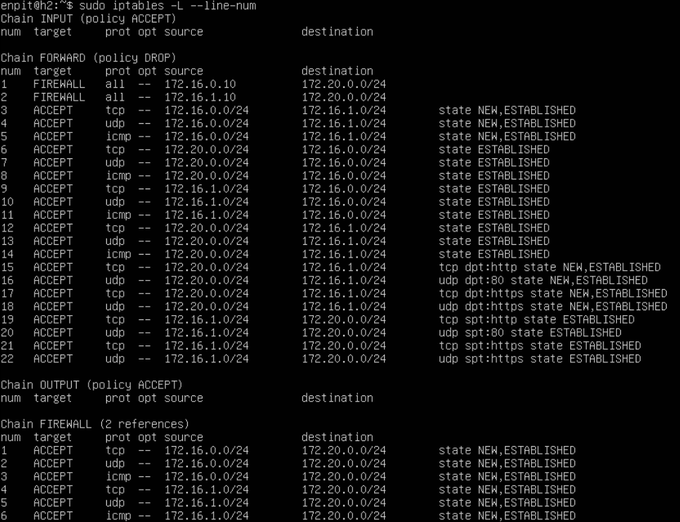
ip address show dev ens4:



ip route show:



演習(2-3)



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．

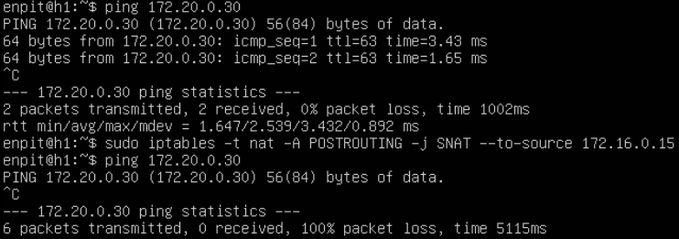
　最後に，FORWARDチェインの1,2行目にて，FIREWALLチェインの1-6行目を参照し，h1とh4の持つipアドレスからの通信のみを許可することでソースIPアドレス詐称を禁止している．

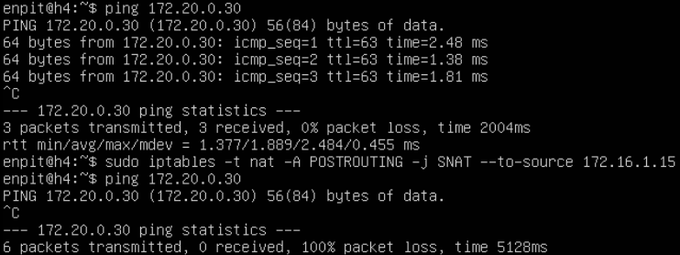
ncコマンド，mtrコマンド及びpingコマンドを用いてTCP，UDP，ICMP接続を試している．

(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）





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．

テキスト

自動的に生成された説明

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．