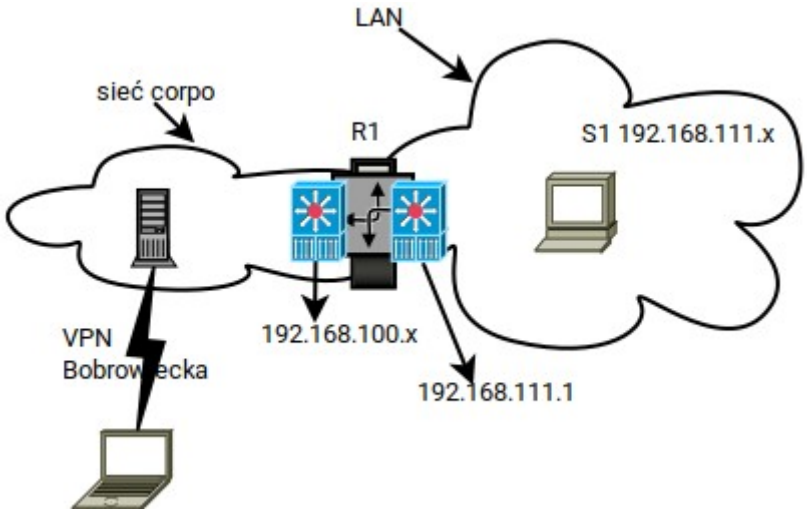
Ćwiczenia wykonujemy na następującym schemacie



Dostęp do sieci corpo jest poprzez VPN-a lub guacamole
W przykładach robimy to na virt-lab.wumed.edu.pl:10400/guacamole
przez guacamole proszę na R1 i S1 zmienić sobie hasło na root-a proponuję na novell

Ćwiczenie 1
Uzyskaj dostęp do internetu z maszyny S1
1. Na maszynie R1 mamy dostęp do internetu więc musimy zrobić maskaradę
Proszę sprawdzić na maszynie R1 bramę domyślną poleceniem ip route Powinna być ustawiona na 192.168.100.1
R1:/home/opensuse # ip r
default via 192.168.111.1 dev eth1 proto dhcp
192.168.100.0/24 dev eth0 proto kernel scope link src 192.168.100.93
192.168.111.0/24 dev eth1 proto kernel scope link src 192.168.111.74
R1:/home/opensuse # ip r del default
R1:/home/opensuse # ip r
192.168.100.0/24 dev eth0 proto kernel scope link src 192.168.100.93
192.168.111.0/24 dev eth1 proto kernel scope link src 192.168.111.74
R1:/home/opensuse # ip r add default via 192.168.100.1 dev eth0
R1:/home/opensuse # ip r
default via 192.168.100.1 dev eth0
192.168.100.0/24 dev eth0 proto kernel scope link src 192.168.100.93
192.168.111.0/24 dev eth1 proto kernel scope link src 192.168.111.74
R1:/home/opensuse #

Instalujemy pakiet firewalld
R1:~ # zypper in -y firewalld
udostępniamy usługę firewalld poleceniem
R1:/ # systemctl enable firewalld
Created symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service → /usr/lib/systemd/system/firewalld.service.
Created symlink /etc/systemd/system/multi-user.target.wants/firewalld.service → /usr/lib/systemd/system/firewalld.service.

```
R1:/ # systemctl start firewalld
R1:/ # systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2021-02-23 22:04:01 UTC; 7s ago
     Docs: man:firewalld(1)
 Main PID: 1743 (firewalld)
    Tasks: 2
   CGroup: /system.slice/firewalld.service
           └─1743 /usr/bin/python3 -Es /usr/sbin/firewalld --nofork --nopid

Feb 23 22:04:00 R1 systemd[1]: Starting firewalld - dynamic firewall daemon...
Feb 23 22:04:01 R1 systemd[1]: Started firewalld - dynamic firewall daemon.
R1:/ #
```

Ustalamy że pracujemy na tym routerze w 2 strefach external i dmz
strefę external ustawiamy jako domyślną i przypisujemy jej interfejs eth0 do strefy dmz dodajemy interfejs eth1 na obu strefach ustawiamy maskaradę

Ustawiamy domyślną strefę na external

```
R1:~ # firewall-cmd --set-default-zone=external
success
R1:~ # firewall-cmd --get-default-zone
external
```

Dodajemy interfejs eth0 do strefy domyślnej

```
R1:~ # firewall-cmd --add-interface=eth0
success
R1:~ # firewall-cmd --list-all
external (active)
  target: default
  icmp-block-inversion: no
  interfaces: eth0
  sources:
  services: ssh
  ports:
  protocols:
  masquerade: yes
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Widzimy że w strefie external mamy włączoną maskaradę
dodajemy interfejs eth1 do strefy zdemilitaryzowanej dmz i uruchamiamy na tej strefie maskaradę

```
R1:~ # firewall-cmd --add-interface=eth1 --zone=dmz
success
R1:~ # firewall-cmd --list-all --zone=dmz
dmz (active)
  target: default
  icmp-block-inversion: no
  interfaces: eth1
  sources:
  services: ssh
  ports:
```

```
 protocols:
 masquerade: no
 forward-ports:
 source-ports:
 icmp-blocks:
 rich rules:

R1:~ # firewall-cmd --add-masquerade --zone=dmz
success
R1:~ # firewall-cmd --list-all --zone=dmz
dmz (active)
 target: default
 icmp-block-inversion: no
 interfaces: eth1
 sources:
 services: ssh
 ports:
 protocols:
 masquerade: yes
 forward-ports:
 source-ports:
 icmp-blocks:
 rich rules:

R1:~ #
```

2. Na maszynie S1 zmieniany bramę na R1 interfejs eth1 (odczytaj ip interfejsu eth1 z maszyny R1 w przykładzie 192.168.111.10)

```
S1:~ # ip r
192.168.111.0/24 dev eth0 proto kernel scope link src
192.168.111.11
S1:~ # ip r add default via 192.168.111.10
S1:~ # ip r
default via 192.168.111.10 dev eth0
192.168.111.0/24 dev eth0 proto kernel scope link src
192.168.111.11
S1:~ # ping wp.pl
PING wp.pl (212.77.98.9) 56(84) bytes of data.
64 bytes from www.wp.pl (212.77.98.9): icmp_seq=1 ttl=58 time=1.94
ms
64 bytes from www.wp.pl (212.77.98.9): icmp_seq=2 ttl=58 time=1.91
ms
^C
--- wp.pl ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.915/1.932/1.949/0.017 ms
S1:~ #
```

Uzyskaliśmy poprzez maskaradę na maszynieR1 dostęp do internetu z S1

Ćwiczenie 2

Na hoście S1 uruchom usługę WWW apache2 i spróbuj czy z R1 można się połączyć z stroną.
Instalujemy na S1 serwer www – apache2

```
S1:~ # zypper in -y apache2 apache2-example-pages
Odczytywanie zainstalowanych pakietów...
Rozwiązywanie zależności pakietu...

Następujące 11 NOWE pakiety zostaną zainstalowane:
 apache2 apache2-example-pages apache2-prefork apache2-utils
libapr1 libapr-util1 libbrotlicommon1 libbrotlienc1 libgc1 system-
user-wwwrun w3m

Następujący zalecany pakiet został automatycznie zaznaczony:
 w3m

11 nowych pakietów do zainstalowania.
Całkowity rozmiar pobieranego pliku: 3,2 MiB. Już zbuforowane: 0
B. Po wykonaniu operacji użyte zostanie dodatkowo 8,7 MiB.
Czy kontynuować? [t/n/w/...? wyświetla wszystkie opcje] (t): t
Pobieranie pakiet libapr1-1.6.3-lp152.4.4.x86_64
(1/11), 103,7 KiB (rozpakowano: 244,5 KiB)
```

Udostępniamy apache2 podczas startu systemu i startujemy usługę

```
S1:~ # systemctl enable apache2
Created symlink /etc/systemd/system/httpd.service →
/usr/lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/apache.service →
/usr/lib/systemd/system/apache2.service.
Created symlink
/etc/systemd/system/multi-user.target.wants/apache2.service →
/usr/lib/systemd/system/apache2.service.
S1:~ # systemctl start apache2
S1:~ # w3m localhost
```

Widzimy że strona domyślna się otwiera
Na maszynie R1 wydając polecenie

```
R1:~ # w3m http://192.168.111.11
```

otwiera się strona. Jeżeli to samo polecenie wydamy z maszyny isi_x to otwiera się strona na adresie 192.168.111.11 Jest to wynik maskarady na strefie external

```
isi0:~ # w3m https://192.168.111.11
```

Wywołując stronę na R1 mamy odmowę dostępu

```
isi0:~ # w3m https://192.168.100.55
w3m: Can't load https://192.168.100.55.
isi0:~ #
```

Ćwiczenie 3
Zrób na maszynie R1 takie przekierowanie aby z zewnątrz wywołując stronę z maszyny R1 otwierało stronę na S1
Dodajemy usługę http do domyślnej strefy external

```
R1:~ # firewall-cmd --add-service=http --permanent
success
```

Dodajemy wstrefie domyślnej przekierowanie portu 80 na port 80 na maszynie S1 192.168.111.11 przeładowujemy firewall i patrzymy co mamy w strefie external

```
R1:~ # firewall-cmd --add-forward-
port=port=80:proto=tcp:toport=80:toaddr=192.168.111.11 --permanent
success
```

```
R1:~ # firewall-cmd --reload
success
R1:~ # firewall-cmd --list-all
external (active)
  target: default
  icmp-block-inversion: no
  interfaces: eth0
  sources:
  services: ssh http
  ports:
  protocols:
  masquerade: yes
  forward-ports: port=80:proto=tcp:toport=80:toaddr=192.168.111.11
  source-ports:
  icmp-blocks:
  rich rules:

R1:~ #
```

Z maszyny isi_x wywołujemy stronę na maszynie R1

```
isi0:~ # w3m http://192.168.100.55
```

Otwiera się strona na S1

Ćwiczenie 4

Połączenie ssh na porcie 1111 maszyny R1 przekaż na maszynę S1 na port 22.

Na maszynie R1 dokonujemy wpisu w strefie external pisząc w jednej linijce

```
R1:~ # firewall-cmd --add-forward-
port=port=1111:proto=tcp:toport=22:toaddr=192.168.111.11 --permanent
```

zobaczmy jakie reguły mamy w strefie external

```
R1:~ # firewall-cmd --list-all
external (active)
  target: default
  icmp-block-inversion: no
  interfaces: eth0
  sources:
  services: ssh http
  ports:
  protocols:
  masquerade: yes
  forward-ports: port=80:proto=tcp:toport=80:toaddr=192.168.111.11
        port=1111:proto=tcp:toport=22:toaddr=192.168.111.11
  source-ports:
  icmp-blocks:
  rich rules:

R1:~ #
```

Połączmy się z maszyny isi_x poprzez ssh do maszyny R1 na porcie 1111

```
isi0:~ # ssh root@192.168.100.55 -p 1111
The authenticity of host '[192.168.100.55]:1111 ([192.168.100.55]:1111)' can't be established.
ECDSA key fingerprint is SHA256:Ul/elR89oSre/+eMC7v0EnmqG+5HilCUfczzn3d3LTs.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[192.168.100.55]:1111' (ECDSA) to the list of known hosts.
root@192.168.100.55's password:
```

Last login: Thu Feb 25 17:17:15 2021 from 192.168.111.1
openSUSE Leap 15.2 x86-64

As "root" use the:
- zypper command for package management
- yast command for configuration management

Have a lot of fun...
S1:~ #
R1 przełączył nas na maszynę S1