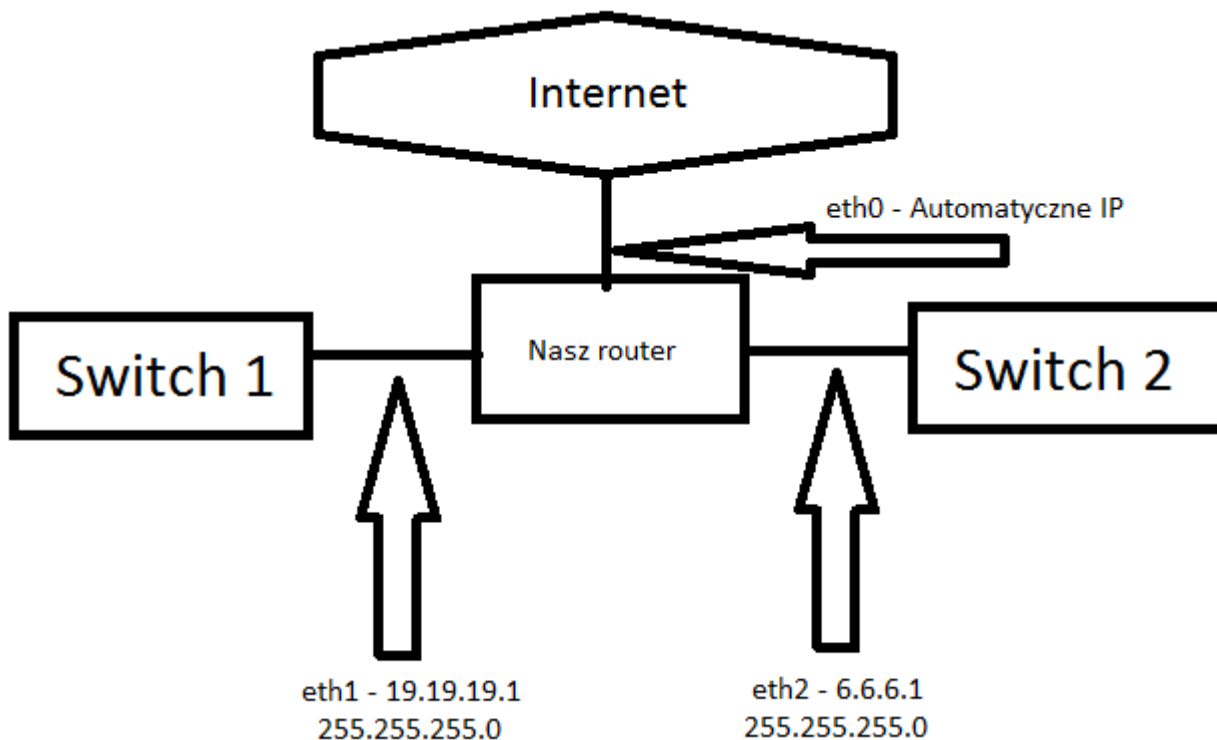


Dziś coś prostego dla ludzi, którzy już trochę czasu zajmują się zagadnieniami sieciowymi, czy ogólną IT, lecz nie dla początkujących.

Serwer DHCP służy do automatycznego przydzielania adresów IP w sieci. Często serwerami dhcp są domowe routery i to wystarcza. A co jeśli nie wystarcza? Wtedy albo kupujemy nowy router... Albo sami wykorzystujemy jakąś maszynę z Linuksem do skonfigurowania własnego serwera dhcp. Oczywiście samo DHCP to nie wszystko. Prócz DHCP skonfigurujemy sobie firewall, który będzie udostępniał połączenie z internetem.

Na początku musimy się zastanowić co będzie nam potrzebne. Jedną z kluczowych kwestii to kwestia jakiego Linuxa wybierzemy. Oczywiście, jeśli się uprzemy możemy postawić nasz DHCP na OS X czy na którymś z Windowsów. Ja osobiście preferuję Linuxy a dokładnie Debiana, który jest do tych celów najwygodniejszy i najprzyjemniej się z nim współpracuje. Drugą ważną kwestią jest ilość posiadanych interfejsów sieciowych. Minimalna ilość to dwa. pierwszy będzie interfejsem podłączonym do internetu. Maszyna na której przeprowadzam dzisiaj lab posiada trzy takie interfejsy (eth0, eth1 i eth2).

A oto schemat jak będzie wyglądała nasza sieć:





Domowy serwer DHCP w kilku prostych krokach.

**Uwaga! Adresacje nie są zgodne z standardami RFC1918 ponieważ, artykuł powstał podczas konfiguracji sieci o specyficznej adresacji spowodowanej zaistniałą sytuacją. Prawidłowe zakresy to:**

10.0.0.0	-	10.255.255.255	(10/8 prefix)
172.16.0.0	-	172.31.255.255	(172.16/12 prefix)
192.168.0.0	-	192.168.255.255	(192.168/16 prefix)

**Zastosowanie adresacji niezgodnych z standardami powoduje, iż użytkownik nie dostanie się na adresacje poza siecią lan dublujące adresacje z sieci lokalnej.**

Na powyższym schemacie widzimy nasz router, który jest podłączony do internetu. Pobiera on adres IP od naszego ISP, więc można określić go jako router brzegowy. Jak wcześniej wspomniałem posiada on trzy interfejsy. Do eth1 oraz eth2 podłączymy switchy. Pierwszy interfejs (eth1) będzie odpowiedzialny za sieć o adresacji 19.19.19.0 z maską 255.255.255.0. Oznacza to, że w sieci dostępne będzie 253 adresy

Natomiast eth2 będzie odpowiedzialny za adresacje 6.6.6.0 z maską 255.255.255.0 więc jak w powyższym do dyspozycji mamy 253 adresy.

To tyle teorii na razie. Czas przejść do praktyki. Dziś skorzystamy z trybu graficznego, jednakże zalecam nie instalowanie tego trybu z względu na oszczędność miejsca na dysku i ramu.

Pierwszym to sprawdzenie ile faktycznie kart sieciowych (interfejsów) ma nasz router. Wykonamy to poleceniem:



## Domowy serwer DHCP w kilku prostych krokach.

```
root@debian:~# lspci | grep -i ether
00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controller (rev 02)
00:08.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controller (rev 02)
00:09.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controller (rev 02)
root@debian:~#
```

Jak widzimy powyżej zgodnie z tym co już pisałem mamy trzy karty sieciowe (interfejsy).

### Konfiguracja kart sieciowych

Przejdźmy teraz do ich konfiguracji. Zaczniemy od pierwszej sieciówki która będzie na styku połączenia z internetem czyli systemowo nazwanej eth0.

Jak widzimy interfejs eth0 dostaje adres IP automatycznie z sieci 192.168.2.1/24. Możemy ustawić statyczny adres dla tego interfejsu, ale nie jest to wymagane.



## Domowy serwer DHCP w kilku prostych krokach.

```
Programy Miejsca          śro 24 gru, 01:03
Przeglądanie i uruchamianie zainstalowanych programów
Terminal
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
root@debian:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 08:00:27:19:c6:21
          inet addr:192.168.2.103  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe19:c621/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3593 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1831 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4280522 (4.0 MiB)  TX bytes:189714 (185.2 KiB)

root@debian:~#
```

Następnie skonfigurujemy odpowiednio eth1 oraz eth2, aby to zrobić należy wyedytować plik `/etc/network/interfaces` w następujący sposób:

```
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
address 19.19.19.1
netmask 255.255.255.0

auto eth2
iface eth2 inet static
address 6.6.6.1
```



Domowy serwer DHCP w kilku prostych krokach.

```
netmask 255.255.255.0
```

Następnie zostaje nam zrestartowanie sieci

```
/etc/init.d/networking restart
```

## KONFIGURACJA SERWERA DHCP

Zacznijemy od zainstalowania odpowiednich paczek

```
apt-get install isc-dhcp-server
```

Zapewne po instalacji pojawił się taki widok:

```
Programy Miejsca          śro 24 gru, 01:39          root
Terminal
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
Konieczne pobranie 935 kB archiwów.
Po tej operacji zostanie dodatkowo użyte 2075 kB miejsca na dysku.
Pobieranie:1 http://ftp.pl.debian.org/debian/ wheezy/main isc-dhcp-server amd64 4
.2.2.dfsg.1-5+deb70u6 [935 kB]
Pobrano 935 kB w 1s (514 kB/s)
Prekonfiguracja pakietów ...
Wybieranie wcześniej niewybranego pakietu isc-dhcp-server.
(Odczytywanie bazy danych ... 145019 plików i katalogów obecnie zainstalowanych.)
Rozpakowywanie pakietu isc-dhcp-server (z ../isc-dhcp-server_4.2.2.dfsg.1-5+deb7
0u6_amd64.deb) ...
Przetwarzanie wyzwalaczy pakietu man-db...
Konfigurowanie pakietu isc-dhcp-server (4.2.2.dfsg.1-5+deb70u6) ...
Generating /etc/default/isc-dhcp-server...
[FAIL] Starting ISC DHCP server: dhcpd[...] check syslog for diagnostics. ... fa
iled!
failed!
invoke-rc.d: initscript isc-dhcp-server, action "start" failed.
root@debian:~#
```



## Domowy serwer DHCP w kilku prostych krokach.

Taki komunikat informuje nas o tym, że serwer DHCP nie został uruchomiony. To normalne, po prostu nie jest jeszcze skonfigurowany.

Konfiguracja naszego serwera opiera się na edycji pliku `/etc/dhcp/dhcp.conf`. Gdy otworzymy ten plik w programie tekstowym takim jak `kwite`, `gedit` lub w środowisku tekstowym `vi` czy `nano` zobaczymy coś takiego:

```
GNU nano 2.2.6      Plik: /etc/dhcp/dhcpd.conf
##
# Sample configuration file for ISC dhcpd for Debian
#
#
# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
ddns-update-style none;

# option definitions common to all supported networks...
option domain-name "example.org";
option domain-name-servers ns1.example.org, ns2.example.org;

default-lease-time 600;
max-lease-time 7200;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
#authoritative;
```

Jest to pomoc od twórców, gotowe ustawienia, które trzeba jedynie uzupełnić. My jednak, usuniemy ten plik i stworzymy własny, by zyskać na czytelności ustawień. Po usunięciu pliku otwieramy ponownie edytor i dodajemy do niego następujące wpisy:

```
subnet 19.19.19.0 netmask 255.255.255.0 {
range 19.10.19.2 19.19.19.254;
default-lease-time 6100;
option domain-name „nasza.domena”;
option domain-name-servers pierwszy.dns, drugi.dns;
option routers 19.19.19.1;
```



## Domowy serwer DHCP w kilku prostych krokach.

```
option subnet-mask 255.255.255.0;
option broadcast-address 19.19.19.255;
}

subnet 6.6.6.0 netmask 255.255.255.0 {
range 6.6.6.2 6.6.6.254;
default-lease-time 6100;
option domain-name „nasza.domena”;
option domain-name-servers pierwszy.dns, drugi.dns;
option routers 6.6.6.1;
option subnet-mask 255.255.255.0;
option broadcast-address 6.6.6.255;
}
```

Następnie otwieramy do edycji plik `/etc/default/isc-dhcp-server` i odszukujemy linie zamierającą

```
INTERFACES=""
```

Pomiędzy „” należy wpisać interfejsy na których ma działać nasz serwer DHCP

```
INTERFACES="eth1 eth2"
```

Kolejnym krokiem jest zrestartowanie procesu serwera DHCP. Zrobimy to poleceniem:

```
/etc/init.d/isc-dhcp-server restart
```

Po restarcie serwisu do swichy podpinamy komputer i sprawdzamy czy otrzymał adres IP. Jeśli tak możemy przejść dalej.

### **Udostępnianie sieci**

Teraz zajmiemy się udostępnianiem internetu dla naszych sieci. Opisze standardową konfigurację, polecam zapoznać się z manuałem iptables, znajdziecie tam wiele innych możliwości konfiguracyjnych.

Zacniemy od utworzenia i edycji pliku w którym napiszemy regułki dla naszego firewalla. Plik, który edytujemy to `/etc/init.d/firewall`.



## Domowy serwer DHCP w kilku prostych krokach.

```
#linijka potrzebna do włączenia udostępniania internetu
echo 1 > /proc/sys/net/ipv4/ip_forward
#wyczyszczenie starych reguł z pamięci iptables
iptables -F
iptables -X
iptables -t nat -X
iptables -t nat -F
# ustawienie domyślnej polityki
iptables -P INPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -P OUTPUT ACCEPT
iptables -A INPUT -i lo -j ACCEPT
iptables -A FORWARD -o lo -j ACCEPT
# Najważniejsze - udostępnienie sieci dla wybranych podsieci
iptables -t nat -A POSTROUTING -s 19.19.19.0/24 -j MASQUERADE
iptables -t nat -A POSTROUTING -s 6.6.6.0/24 -j MASQUERADE
```

Plik zapisujemy, nadajemy mu prawa wykonywalności:

```
# chmod +x /etc/init.d/firewall
```

dopisujemy do standardowych runleveli:

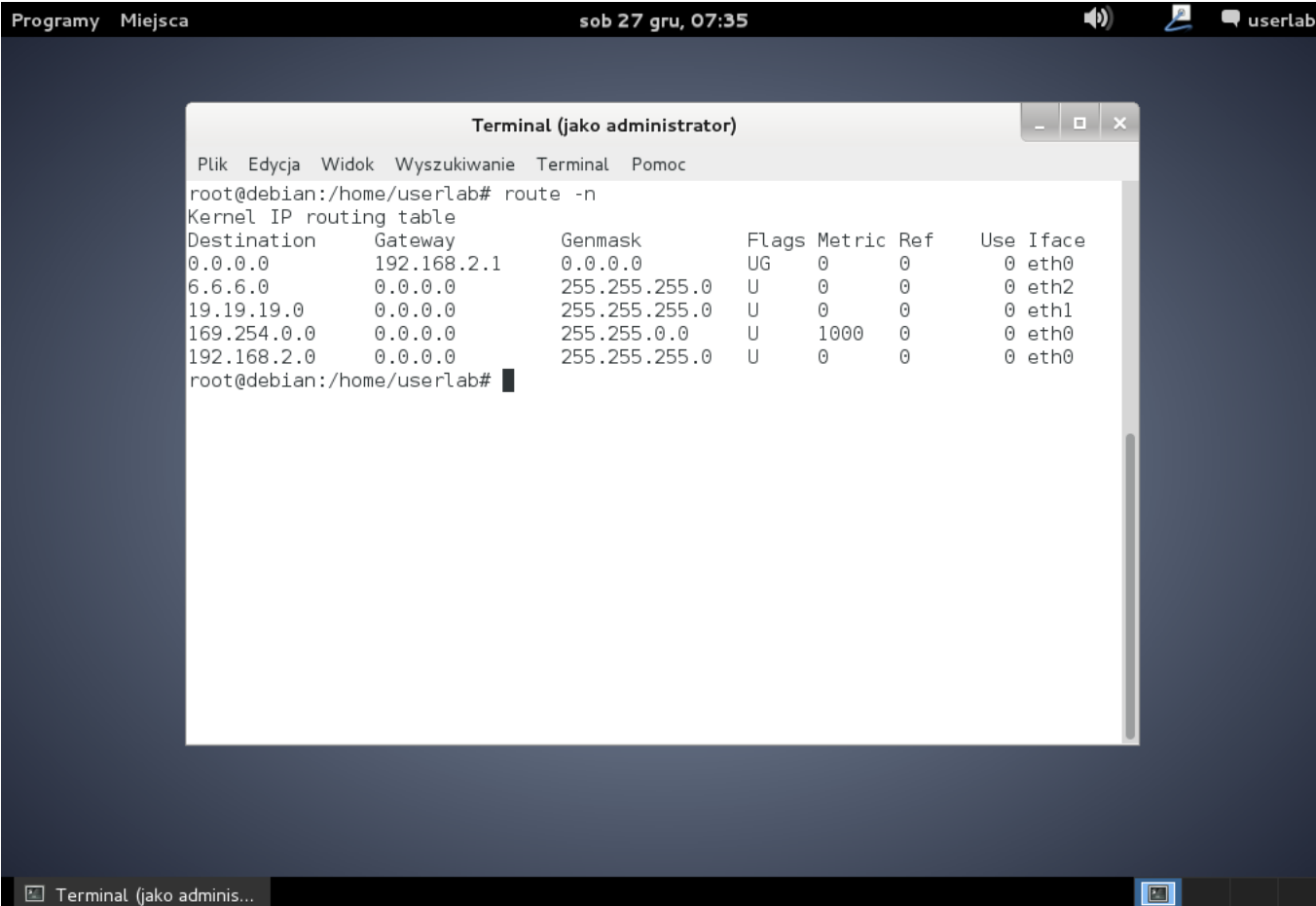
```
# update-rc.d firewall defaults 90
```

i odpalamy:

```
# /etc/init.d/firewall
```

W tym momencie mamy działający internet na wszystkich komputerach w naszych sieciach. Trzy ostatnie linijki w skrypcie powinny stworzyć nam odpowiednią tablicę routingu:





```
Terminal (jako administrator)
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
root@debian:/home/userlab# route -n
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0          192.168.2.1    0.0.0.0         UG    0      0      0 eth0
6.6.6.0          0.0.0.0        255.255.255.0   U    0      0      0 eth2
19.19.19.0       0.0.0.0        255.255.255.0   U    0      0      0 eth1
169.254.0.0     0.0.0.0        255.255.0.0     U   1000   0      0 eth0
192.168.2.0     0.0.0.0        255.255.255.0   U    0      0      0 eth0
root@debian:/home/userlab#
```

Teraz nasza konfiguracja powoduje, że komputery z obydwu podsieci się widzą i mogą się pingować. Z względów bezpieczeństwa powinno się unikać takiej praktyki jeżeli nie jest to wymagane. Oczywiście w naszym wypadku, już sama rozbieżność adresacji podsieci utrudni działania w sieci jeśli nie wiemy jak skonfigurowany jest nasz dhcp.

## Przekierowywanie portów

Opcjonalnie możemy do naszego skryptu firewall regułą przekazującą ruch. A o to ona:

```
iptables -t nat -A PREROUTING -p tcp -i eth0 --dport 800 -j DNAT --to 6.6.6.2:22
```

Czyli:

Cały ruch przychodzący z zewnątrz na port 800 przekazujemy do wewnętrznej sieci o masce 6.6.6.0/24 do komputera o ip 6.6.6.2 na port 22.



Domowy serwer DHCP w kilku prostych krokach.

Na dziś to już wszystko. W następnym artykule postaram się pokazać jak poprawić standardową konfigurację.