



Sieciowe systemy operacyjne

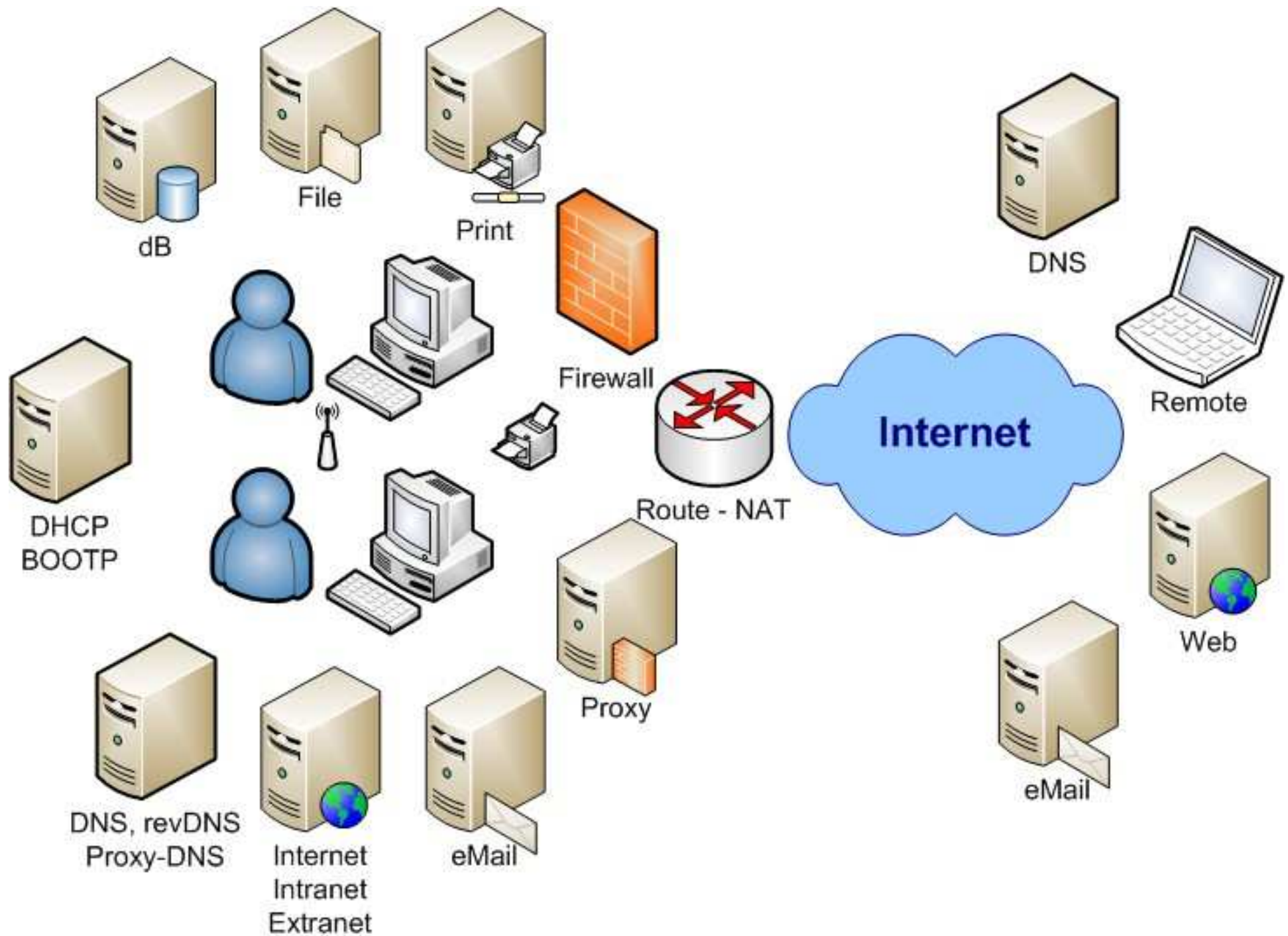
Część 3. Usługi sieciowe

Autor Wojciech Gumiński



### Ostrzeżenie:

Informacje zawarte w tym dokumencie są materiałami pomocniczymi do prowadzenia wykładu. Nie zastąpią ani podręcznika, ani tym bardziej obecności na wykładach. Niektóre wpisy w przykładowych plikach konfiguracyjnych mogą być wzajemnie sprzeczne, ale ilustrują możliwości uzyskania określonych właściwości usług.





## Metody konfiguracji urządzeń w sieciach IP

- Manualne
  - Interfejs CLI
  - Interfejs GUI
  - Interfejs Webowy
  - Lokalne (konsola RS, konsola CLI i GUI)
  - Zdalne (Telnet, SSH, WWW)
- Automatyczne
  - DHCPv4 (RFC2131, RFC2132 dawniej RFC1541, RFC1533)
  - SLAAC IPv6 (RFC2462)
  - DHCPv6 (RFC3315)



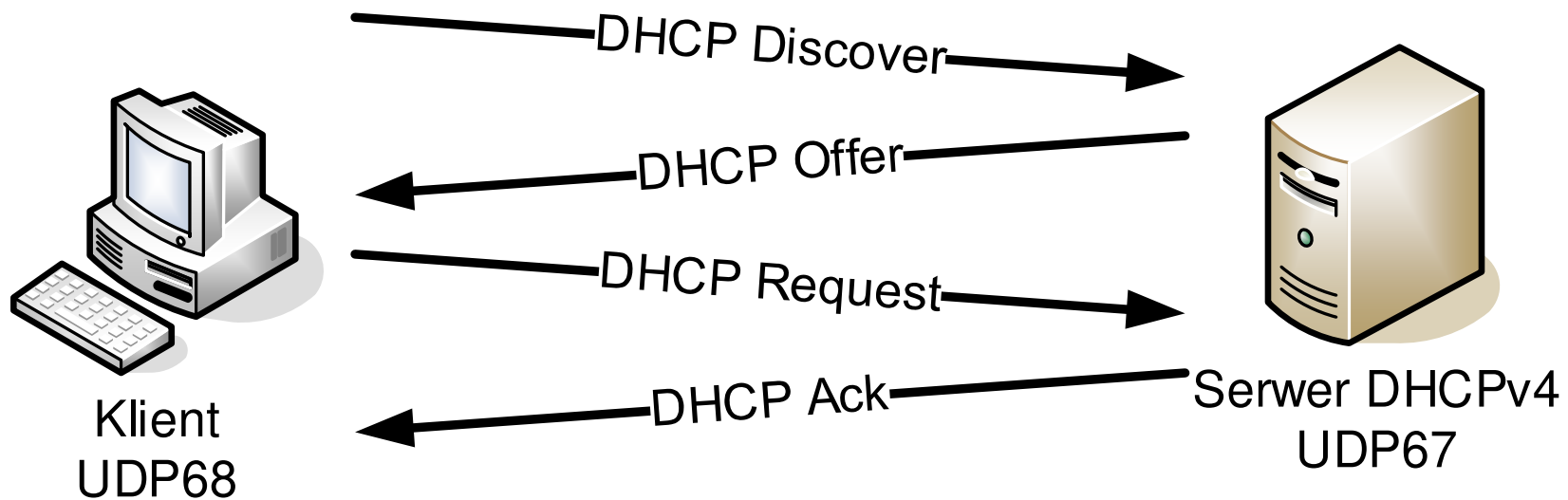
# **DHCPv4 (RFC2131)**

## **Dynamic Host Configuration Protocol**

- **Możliwość automatycznego konfigurowania ustawień sieci dla hostów w sieci lokalnej.**
- **Kilkadziesiąt standardowych opcji konfiguracji hosta (RFC2132).**
- **Możliwość przekazywania żądań i ofert DHCP poprzez przekaźniki (ang. *Relay*) poza lokalną sieć.**
- **Możliwość rezerwacji statycznych parametrów dla wybranych hostów.**
- **Wygodna, centralna administracja adresami IP i konfiguracjami hostów.**
- **Kłopotliwe działanie w sieciach o słabej jakości łączy np. WiFi.**

# DHCPv4

## Dynamic Host Configuration Protocol





## Przykładowa konfiguracja /etc/dhcp/dhcpd.conf

#Opcje globalne

# Serwer WINS

**option netbios-name-servers 10.8.0.254;**

# Serwer DNS podstawowy i zapasowe

**option domain-name-servers 10.8.0.254, 194.204.159.1;**

# Domena

**option domain-name "domena.pl";**

# Czas dzierżawy

**default-lease-time 3600;**

**max-lease-time 86400;**



## Przykładowa konfiguracja /etc/dhcp/dhcpd.conf

#Definicja podsieci

**subnet 10.8.0.0 netmask 255.255.0.0**

**{**

# Domyślna brama

**option routers 10.8.0.254;**

# Maska sieci

**option subnet-mask 255.255.0.0;**

# Broadcast

**option broadcast-address 10.8.255.255;**

# Zakres dynamicznie przydzielanych adresów IP

**range 10.8.1.194 10.8.1.199;**





## Przykładowa konfiguracja /etc/dhcp/dhcpd.conf

# Statyczne adresy IP przydzielane przez DHCP

# Switch D-Link DES3226

**host NET09 {**

**hardware ethernet 00:05:5d:75:be:d9;**

**fixed-address 10.8.0.109;**

**}**

# Print serwer HP2200DTN

**host HP2200DTN {**

**hardware ethernet 00:01:E6:50:CB:41;**

**fixed-address 10.8.0.102;**

**}**

# PC002 Jasio Fasola p. 213

**host PC002 {**

**hardware ethernet 00:04:75:C3:04:09;**

**fixed-address 10.8.1.102;**

**}**



## Przykładowa konfiguracja /etc/dhcp/dhcpd.conf

#definicja grupy hostów

**group terminale {**

# Boot image

**filename "Xncd19r"**

# TFTP Server

**next-server 10.8.0.254;**

# Terminal NCR

**host ncr1 {**

**hardware ethernet 00:00:A7:11:AF:4A;**

**fixed-address 10.8.1.181;**

**}**

**}**

**}**



## **Uruchamianie serwera DHCP**

```
/etc/init.d/dhcpd-server start
```

## **Przydatne polecenia klienta DHCP**

```
dhclient
```

```
ipconfig
```

## **Uruchomienie klienta DHCP – pobranie konfiguracji**

```
dhclient
```

```
ipconfig /renew
```

## **Sprawdzenie stanu klienta DHCP**

```
ipconfig /all
```

## **Wyłączenie klienta DHCP – zwolnienie konfiguracji**

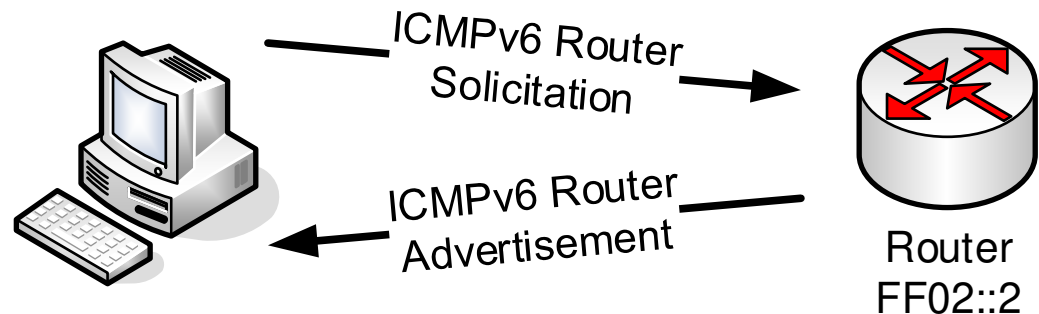
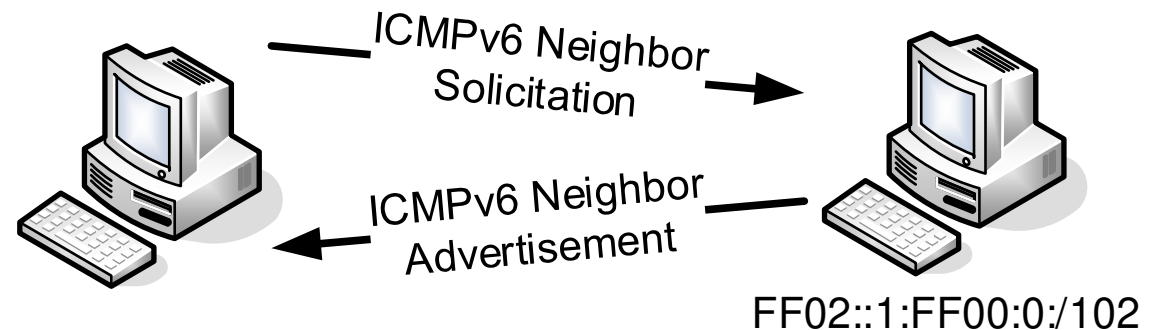
```
dhclient -r
```

```
ipconfig /release
```

# SLAAC (RFC2462)

## Stateless Address Auto Configuration

- Interface Autoconfiguration
  - EUI-64
  - Duplicate Address Detection
- Neighbor Discovery
  - Multicast FF02::1
  - Multicast FF02::1:FF00:0:78:9ABC
- Router Advertisement
  - Router is a neighbor
  - Multicast FF02::2



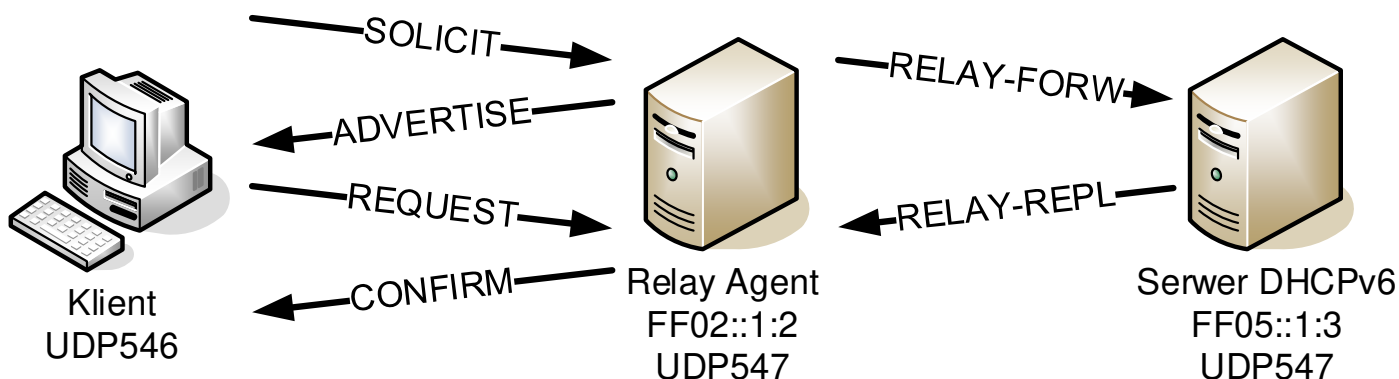
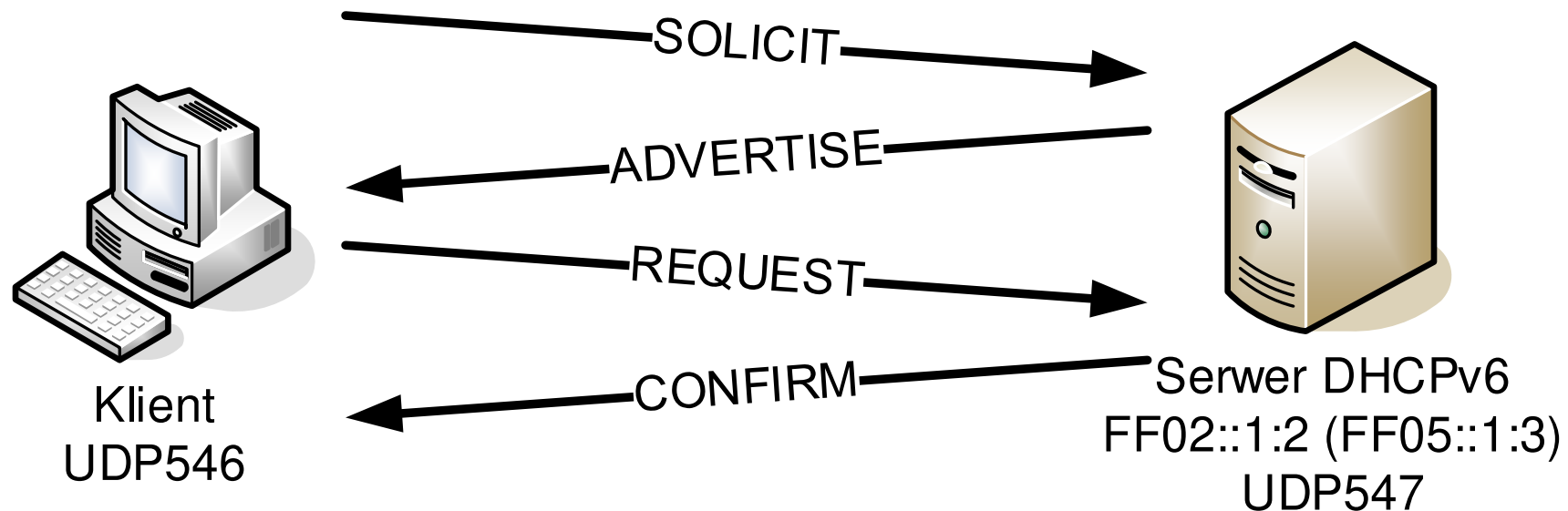


## Router Advertisement (radvd.conf) (+RFC6106)

```
interface eth0 {  
    AdvSendAdvert on;  
    AdvManagedFlag on;  
    AdvOtherConfigFlag off;  
    MinRtrAdvInterval 30;  
    MaxRtrAdvInterval 100;  
    prefix 2001:db8:1:0::/64 {  
        AdvOnLink on;  
        AdvAutonomous on;  
        AdvRouterAddr off;  
    };  
    RDNSS 2001:db8::1 2001:db8::2  
        { AdvRDNSSLifetime 30; };  
    DNSSL branch.example.com example.com  
        { AdvDNSSLLifetime 30; };  
};
```

## DHCPv6 (RFC3315)

### Dynamic Host Configuration Protocol for IPv6





DHCPv6 Dibbler (server.conf)

```
iface eth0 {  
    T1 600  
    T2 900  
    preferred-lifetime 1800-3600  
    valid-lifetime 3600-86400  
    class {  
        pool 2001:db8:1::1000 - 2001:db8:1::FFFF  
    }  
    option dns-server 2001:db8:1::1  
    option domain example.com, test.example.com  
    option lifetime 7200  
    client duid 0x000102030406 {  
        address 2001:db8:1::1234  
        option domain second.com  
        option dns-server 2001:db8:2::678  
    }  
}
```



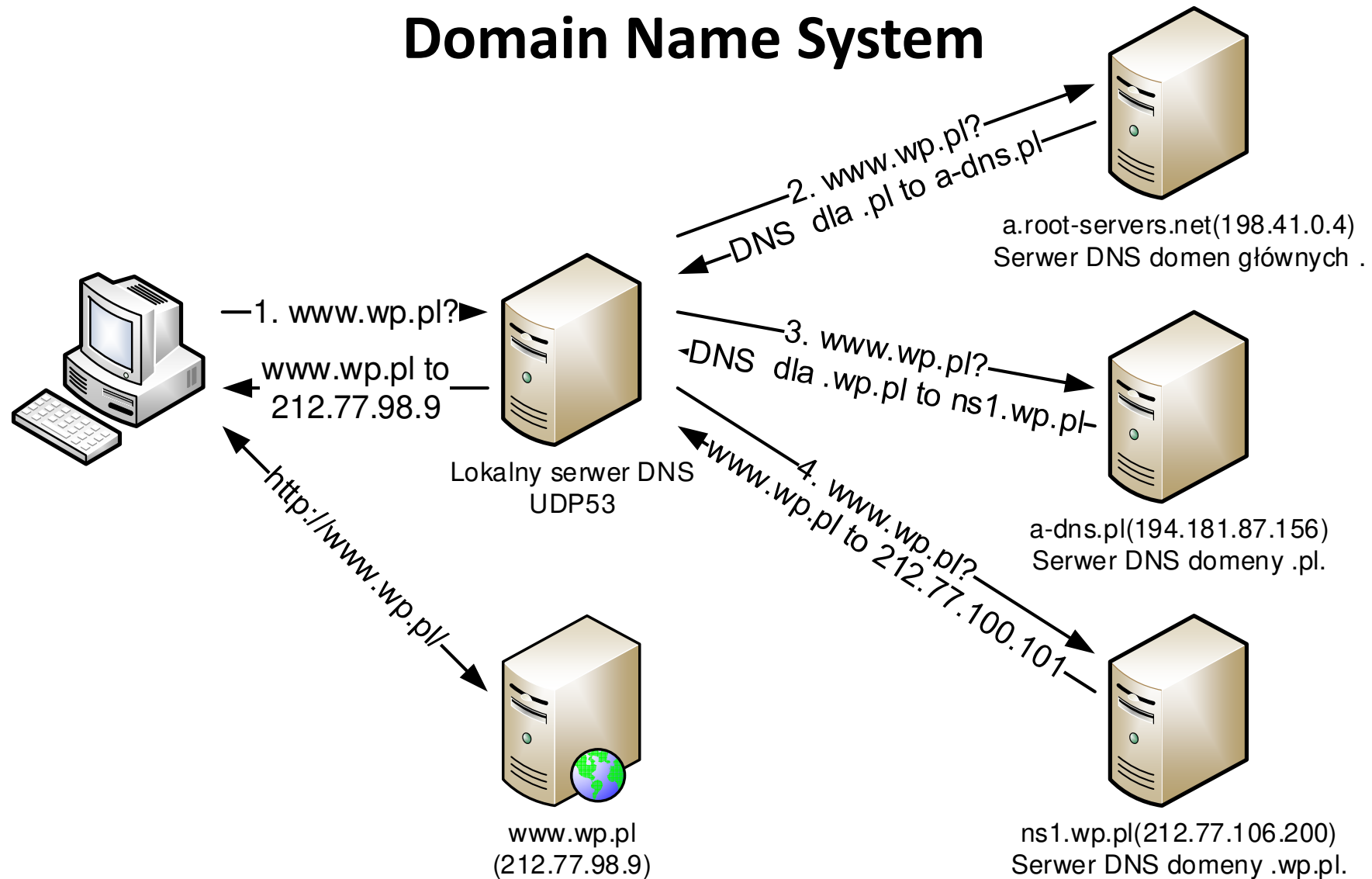
# DNS

# Domain Name System

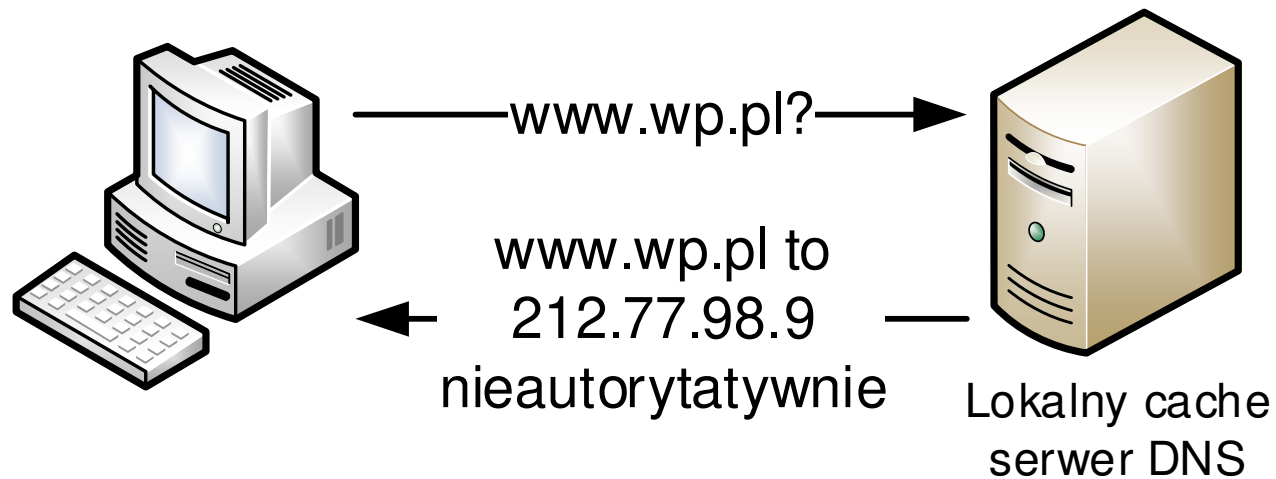


# DNS (RFC 1035)

## Domain Name System



## DNS - Lokalny serwer pośredniczący





## Odpytywanie serwera DNS

```
nslookup www.domena.edu  
nslookup -type=MX domena.edu  
nslookup host.domena.edu 8.8.8.8
```

```
host www.domena.edu
```

```
dig www.domena.edu  
dig domena.edu -t ANY  
dig host.domena.edu @8.8.8.8
```

## Popularne publiczne rekursywne serwery DNS google i cloudflare:

```
IPv4: 8.8.8.8, 8.8.4.4, 1.1.1.1, 1.0.0.1  
IPv6: 2001:4860:4860::8888, 2001:4860:4860::8844,  
      2606:4700:4700::1111, 2606:4700:4700::1001
```



## Typy wpisów DNS

<b>A</b>	- adres IPv4
<b>AAAA</b>	- adres IPv6
<b>CNAME</b>	- alias nazwy domenowej
<b>NS</b>	- adres serwera DNS utrzymującego daną domenę
<b>MX</b>	- adres serwera poczty elektronicznej dla domeny
<b>PTR</b>	- domena przypisana do adresu
<b>TXT</b>	- tekst przypisany do domeny
<b>SRV</b>	- serwery usług w domenie
<b>SOA</b>	- opcje domeny i opis odpowiedzialności za domenę

## Ogólny format wpisu w konfiguracji DNS

name ttl class type type-specific-data

Przykłady:

```
jeden 86400 IN AAAA 2001:db8::1
jeden      IN AAAA 2001:db8::1
jeden      AAAA 2001:db8::1
mail       MX    5  serwer7.domena.edu.
_http._tcp SRV    0 0 80 www.domena.edu.
_imap._tcp SRV    0 0 143 mail.domena.edu.
_imaps._tcp SRV    0 0 993 mail.domena.edu.
```



## **Pliki konfiguracyjne klienta i serwera usługi DNS**

/etc/resolv.conf

### **Przykładowa zawartość pliku /etc/resolv.conf**

```
search firma.edu.net  
nameserver 10.0.0.1  
nameserver 153.19.40.250  
nameserver 194.204.159.1
```

### **Przykładowa zawartość pliku /etc/hosts**

```
127.0.0.1      localhost  
192.168.1.1    ap        ap.local.net
```



## Tylko serwer pośredniczący - cache DNS

/etc/bind/named.conf  
/etc/bind/named.root

Aktualna wersja pliku z konfiguracją głównych root serwerów DNS (w tym przykładzie plik named.root) jest utrzymywana pod adresem:  
**<http://www.internic.net/domain/named.root>**

## Plik named.conf dla serwera pośredniczącego

```
zone "."  
{  
    type hint;  
    file "/etc/bind/named.root";  
};
```



## Serwer domeny podstawowej i odwrotnej

```
/etc/bind/named.conf  
/etc/bind/named.root  
/etc/bind/firma.conf  
/etc/bind/10.0.0.conf  
/etc/bind/2001.db8.1.conf
```

## Plik named.conf – główny plik konfiguracyjny

```
//opcje folder przechowywania kolejnych plików i obsługa IPv6  
options  
{  
    directory "/etc/bind";  
    listen-on-v6 { any; };  
};
```



zone "."

//cache DNS

```
{  
  type hint;  
  file "named.root";  
};
```

zone "firma.edu.net"

//DNS dla domeny firma.edu.net

```
{  
  type master;  
  file "firma.conf";  
  allow-transfer { 10.1.0.2; }; //10.1.0.2 to zapasowy DNS dla domeny  
};
```

zone "firma2.com.pl"

//Zapasowy DNS dla domeny firma2.com.pl

```
{  
  type slave;  
  file "firma2.conf";  
  masters { 10.1.0.1; }; //10.1.0.1 to podstawowy DNS dla tej domeny  
};
```





//revDNS dla sieci 10.0.0.0

zone "0.0.10.in-addr.arpa"

{

type master;

file "10.0.0.conf";

};

//revDNS dla sieci 2001:db8:1::

zone "0.0.0.0.1.0.0.0.8.d.b.0.1.0.0.2.ip6.arpa"

{

type master;

file "2001.db8.1.conf";

};



## Plik firma.conf dla serwera podstawowego domeny

```
$ORIGIN firma.edu.net.  
$TTL 86400 ; Cache TTL  
@ SOA ns.firma.edu.net. admin.firma.edu.net. (  
    ; odpowiedzialność i email  
    2014030301 ; Serial zwykle data i nr wersji  
    604800 ; Refresh czas odświeżania  
    86400 ; Retry czas powtórnego odświeżania  
    2419200 ; Expire czas wygaśnięcia wpisu  
    604800 ) ; minimalny czas wpisu  
;główny DNS domeny  
@ NS ns.firma.edu.net.  
;zapasowy DNS domeny  
@ NS ns.provider.net.  
;adres przypisany do domeny  
@ A 10.0.0.3  
@ AAAA 2001:db8::3  
;serwer pocztowy (najniższy numer priorytetu)  
@ MX 10 mail.firma.edu.net.  
;zapasowy serwer pocztowy (wyższy numer priorytetu)  
@ MX 20 mail2.firma.edu.net.
```



## Plik firma.conf dla serwera podstawowego domeny

*;adresy hostow*

```
mercury A      10.0.0.1
venus  A      10.0.0.2
earth  A      10.0.0.3
earth  AAAA    2001:db8:1::3
mars   A      10.0.0.4
jupiter A     10.0.0.5
saturn A      10.0.0.6
saturn AAAA    2001:db8:1::6
sun    A      10.0.0.10
sun    A      10.0.0.11
sun    A      10.0.0.12
*      A      10.0.0.3
ns     A      10.0.0.4
```

*;aliasy nazw*

```
www    CNAME    earth
mail   CNAME    venus
mail2  CNAME    mercury
ftp    CNAME    mars
```



## Plik 10.0.0.conf dla serwera podstawowego domeny odwrotnej

```
@      SOA      ns.firma.edu.net.  admin.firma.edu.net.  (
        2014030301 604800 86400 2419200 604800 )
;
@      NS       ns.firma.edu.net.
;
1      PTR      mercury.firma.edu.net.
2      PTR      venus.firma.edu.net.
3      PTR      earth.firma.edu.net.
4      PTR      mars.firma.edu.net.
5      PTR      jupiter.firma.edu.net.
6      PTR      saturn.firma.edu.net.
10     PTR      sun.firma.edu.net.
```

## DNSSEC – DNS Security Extension, RFC 4033, 4034, 4035

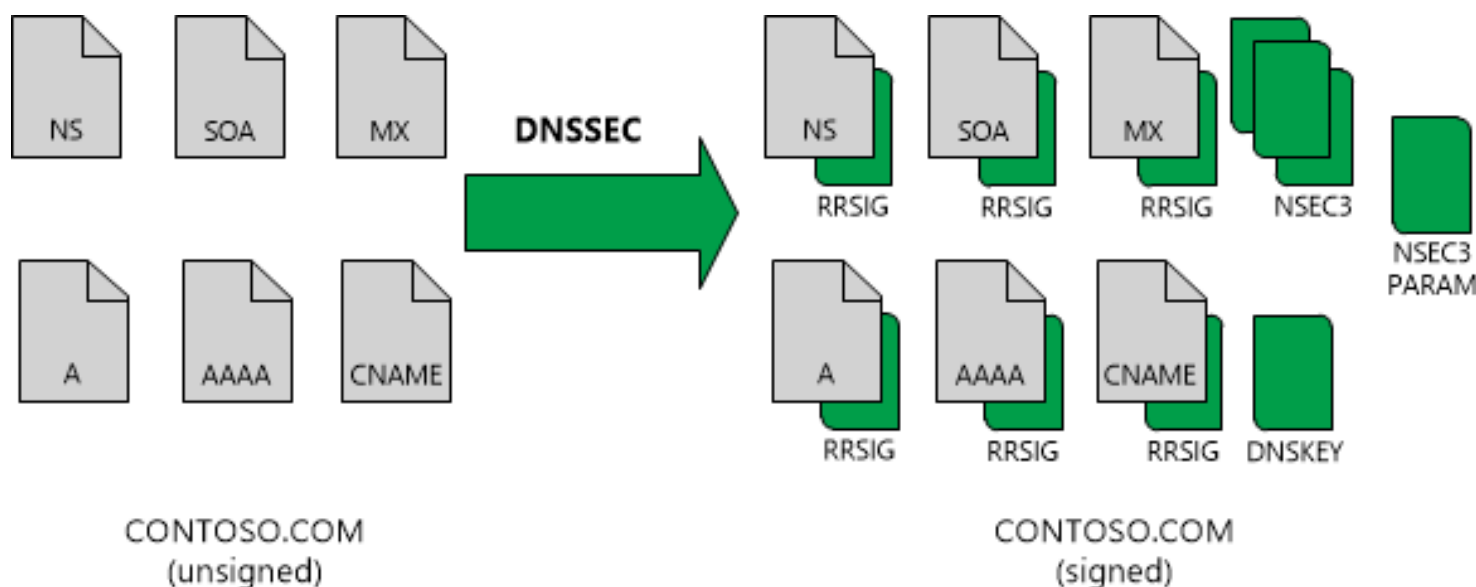
Kompletne typowe odpowiedzi serwera DNS, tzw. RRset, są cyfrowo podpisane z wykorzystaniem kryptografii asymetrycznej (SHA1, SHA256 i RSA)

Mocne strony:

- zapewnia integralność i wiarygodność odpowiedzi.

Słabe strony:

- brak szyfrowania, nie gwarantuje poufności,
- słabe wdrożenie 8% 2013, 15% 2016





## DoH – DNS over HTTPS, RFC 8484,

Mocne strony:

- wiarygodność źródła, poufność, tcp/443
- wdrożony standardowo w Firefox'ie

Słabe strony:

- kłopotliwe statystyki i filtrowanie ruchu korporacyjnego,
- brak kontroli rodzicielskiej,
- konieczność używania zcentralizowanych, publicznych DNS (Cloudflare, Google, NextDNS),
- ciągle prace w toku,
- dwie wersje binarny rekord DNS albo JSON,
- wydajność

## DoT – DNS over TLS, RFC 7858, 8310

- jak DNS tylko po szyfrowanym, wiarygodnym połączeniu TLS
- tcp/853
- mocne/słabe strony patrz DoH



# FTP

# File Transfer Protocol



## Konfiguracja serwera usługi FTP (RFC 959) na przykładzie vsftpd w systemie Knoppix Linux.

Instalacja z pakietów źródłowych

Pobranie ze strony: **<http://vsftpd.beasts.org>**

albo wprost z ftp: **<ftp://vsftpd.beasts.org/users/cevans/>**

np.:

**cd /home/knoppix**

# **wget ftp://vsftpd.beasts.org/users/cevans/vsftpd-2.0.5.tar.gz**

**wget http://guminski.net/vsftpd-2.0.5.tar.gz**

Rozpakowanie

**tar xzvf vsftpd-2.0.5.tar.gz**

Kompilacja

**cd vsftpd-2.0.5**

**make**





## Czynności instalacyjne

```
useradd -d /var/ftp -s /bin/false ftp
```

```
mkdir /var/ftp
```

```
chown root.root /var/ftp
```

```
chmod 755 /var/ftp
```

```
mkdir /var/ftp/pub
```

```
#poniższe polecenia tylko jeśli chcemy zezwolić
```

```
#na upload dla użytkowników anonymous
```

```
mkdir /var/ftp/pub/upload
```

```
chown ftp /var/ftp/pub/upload
```



## Instalacja

Kopiowanie pliku do docelowej lokalizacji

```
cp /home/knoppix/vsftpd-2.0.5/vsftpd /usr/sbin/
```

Kopiowanie plików pomocy

```
cp /home/knoppix/vsftpd-2.0.5/vsftpd.8 /usr/share/man/man8/
```

```
cp /home/knoppix/vsftpd-2.0.5/vsftpd.conf.5 /usr/share/man/man5/
```

Edycja konfiguracji - zawartość w dalszej części.

```
touch /etc/vsftpd.conf
```

```
vim /etc/vsftpd.conf
```

Uruchamianie

```
vsftpd &
```

Zatrzymywanie

```
killall vsftpd
```



## Konfiguracja /etc/vsftpd.conf

Uwaga: w pliku konfiguracyjnym vsftpd.conf nie mogą występować spacje i jest rozróżniana wielkość znaków.

#użytkownik z prawami którego pracuje vsftpd

**nopriv\_user=ftp**

#tryb standalone - serwer samodzielny YES tryb inetd NO

**listen=YES**

#w trybie standalone uruchamianie od razu w tle

**background=YES**

#port

**listen\_port=21**

#zgoda na użytkowników anonimowych

**anonymous\_enable=YES**

#zgoda na użytkowników systemowych

**local\_enable=YES**

#lista zakazanych użytkowników lokalnych

**userlist\_enable=YES**

**userlist\_file=/etc/vsftpd\_users.conf**



```
#tekst przywitania serwera
ftpd_banner="Serwer vsFTPd wita"
#albo plik z tekstem przywitania banner_file=/var/ftp/welcome
#logowanie transferów w wersji standardowej
xferlog_enable=YES
xferlog_file=/var/log/xferlog.log
xferlog_std_format=YES
#zakaz wychodzenia poza katalog domowy
chroot_local_user=YES
#katalog dla użytkownika anonimowego
secure_chroot_dir=/var/ftp
#brak pytania o hasło dla użytkownika anonimowego
no_anon_password=YES
#zgoda na zapis
write_enable=YES
#zgoda na zapis dla użytkownika anonimowego
anon_upload_enable=YES
#zgoda na tworzenie katalogów
anon_mkdir_write_enable=NO
```



#ukrycie identyfikatorów właścicieli plików

**hide\_ids=YES**

#maski zakazanych atrybutów plików

**local\_umask=022**

**anon\_umask=022**

#limity liczby równoczesnych połączeń

**max\_clients=5**

**max\_per\_ip=2**

#limity transferu w bajtach na sekundę

**anon\_max\_rate=10000**

**local\_max\_rate=50000**

#czasy wylogowania

**idle\_session\_timeout=120**

**data\_connection\_timeout=900**

#zakazane nazwy plików i folderów

**deny\_file={\*.mp3,files/,\*.avi}**

#ukryte nazwy plików i folderów

**hide\_file={\*.doc,\*.xsl}**



### Inne możliwości vsftpd

- Wirtualne serwery
- Wirtualni użytkownicy
- Indywidualne konfiguracja dla użytkowników
- Indywidualne konfiguracje dla adresów IP użytkowników
- Współpraca z inetd
- Szyfrowane połączenia FTPS
- Obsługa IPv6
- i inne



Uruchamianie przez skrypt w folderze **/etc/init.d/**:

**/etc/init.d/vsftpd start**

Zatrzymywanie:

**/etc/init.d/vsftpd stop**

Utworzenie pliku **/etc/init.d/vsftpd**

**touch /etc/init.d/vsftpd**

**chmod +x /etc/init.d/vsftpd**

Prosta zawartość pliku:

**#!/bin/sh**

**case \$1 in**

**start) vsftpd &**

**echo "Uruchamianie vsftpd"**

**;;**

**stop) killall vsftpd**

**echo "Zatrzymanie vsftpd"**

**;;**

**\*) echo "Usage: \$0 {start|stop}"**

**exit 1**

**esac**



Uruchamianie przez skrypt w folderze **/etc/init.d/**:

**/etc/init.d/vsftpd start**

Zatrzymywanie:

**/etc/init.d/vsftpd stop**

Utworzenie pliku **/etc/init.d/vsftpd**

**touch /etc/init.d/vsftpd**

**chmod +x /etc/init.d/vsftpd**

Prosta zawartość pliku:

**#!/bin/sh**

**case \$1 in**

**start) vsftpd &**

**echo "Uruchamianie vsftpd"**

**;;**

**stop) killall vsftpd**

**echo "Zatrzymanie vsftpd"**

**;;**

**\*) echo "Usage: \$0 {start|stop}"**

**exit 1**

**esac**



## **FTPS bezpieczne FTP z TLS, RFC-4217**

Dwa tryby pracy

1. Najpierw TLS, a następnie FTP
  - Control channel TCP port 990
  - Data channel TCP port 989
  - Pełna niekompatybilność z NAT
2. Normalne FTP z rozszerzeniem TLS na żądanie FEAT, ~~AUTH SSL~~, AUTH TLS, CDC
  - Czasami wykorzystywane wyłącznie w procesie uwierzytelniania



# SFTP

# SCP



## SFTP Secure File Transfer Protocol, SSH File Transfer Protocol

**SCP Secure Copy Protocol** to protokół na bazie RCP (Remote Commands Protocol) umożliwiający jedynie kopiowanie plików między lokalny a zdalnym systemem.

**SFTP** jest rozszerzeniem protokołu SSH-2 obejmującym pełną obsługę zdalnego systemu plików. Do kopiowania plików można użyć scp2. W wielu systemach scp zostało zastąpione scp2.

```
scp plik user@remote:/folder/nazwa  
scp -P port user@remote:/folder/nazwa plik  
scp -r ~/* user@remote:/backup
```

Uwierzytelnianie parą kluczy prywatny/publiczny  
~/.ssh/id\_rsa  
~/.ssh/authorized\_keys



# HTTP

# HyperText

# Transfer Protocol



# KONFIGURACJA HTTPD (RFC 2616, RFC 7230-7235)

Główny plik konfiguracyjny usługi http to: **/etc/apache2/apache2.conf** (dawniej **/etc/apache/httpd.conf** czasami **/etc/httpd/httpd.conf**)

**ServerRoot /etc/apache**

Określa położenie bazowe plików konfiguracyjnych, logów itp.

**Port 80**

Określa port na którym nasłuchuje serwer usługi httpd

**ServerName Knoppix**

Określa nazwę domenową serwera. Konieczna protokół http nie posługuje się adresami IP.

**User apache**

**Group apache**

Określa nazwę użytkownika i grupy z uprawnieniami których pracuje httpd. Ważne dla określenia uprawnień do katalogów zawierających aplikacje internetowe np. prawa zapisu.



## **DocumentRoot /var/www**

Określa folder główny serwera http, w którym przechowywane są dokumenty html.

## **<IfModule mod\_dir.c>**

**DirectoryIndex index.html index.php index.cgi**

## **</IfModule>**

Określa nazwy i kolejność poszukiwania pliku domyślnego dla folderu.

## **<IfModule mod\_userdir.c>**

**UserDir public\_html**

## **</IfModule>**

Zezwala na tworzenie stron użytkowników w podkatalogach folderów domowych o nazwie **public\_html**.



```
<Directory /home/*/public_html>
```

```
    AllowOverride All
```

```
    Options Indexes FollowSymLinks ExecCGI
```

```
</Directory>
```

Sekcja **Directory** określa opcje dla katalogów, tu: dla wszystkich podkatalogów **public\_html** w katalogach domowych użytkowników.

**Indexes** - automatyczne wyświetlanie listy plików w folderze, gdy brak pliku indeksowego **index.html**

**FollowSymLinks** - zgoda na używanie dowiązań symbolicznych do innych plików lub katalogów

**ExecCGI** - zgoda na uruchamianie skryptów, programów CGI

**Includes** - zgoda na dołączanie dodatkowych plików



### **Alias /icons/ /usr/hsare/apache/icons/**

Definiuje alias **icons** prowadzący do określonego folderu. Zwykle umieszcza się też tu sekcję **<Directory>** dla tego folderu.

### **AddHandler cgi-script .cgi .sh .pl**

Pozwala na uruchamianie skryptów CGI poza folderem **cgi-bin**

### **LoadModule php5\_module libphp5.so**

Dodaje obsługę php jako modułu serwera httpd.

### **AddType application/x-httpd-php .php .php5 .php4 .php3**

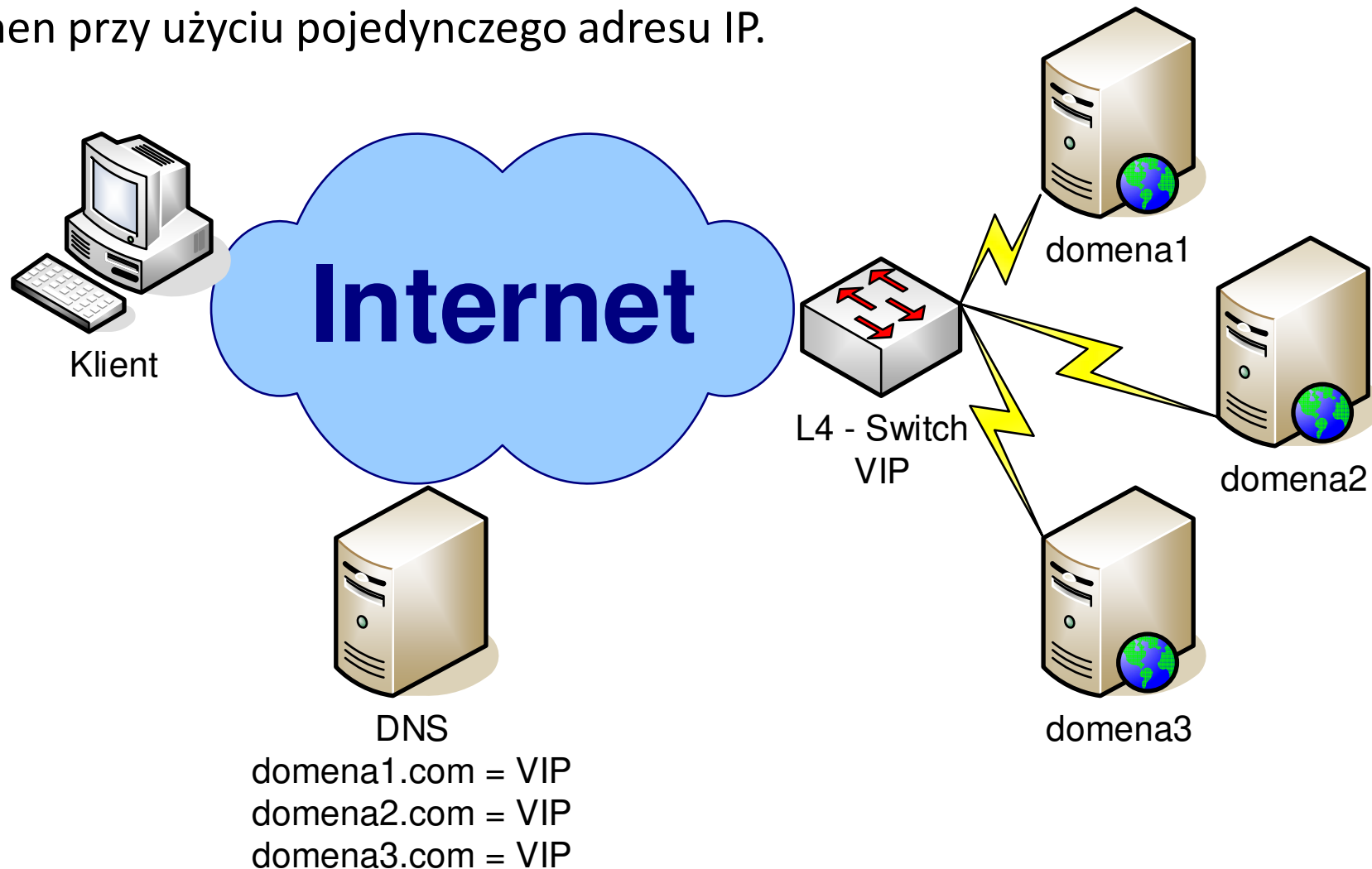
### **AddType application/x-httpd-php-source .phps**

Określa pliki **.php** jako skrypty modułu php.



## Serwery wirtualne

Serwer Apache umożliwia obsługę wielu serwerów wirtualnych dla różnych domen przy użyciu pojedynczego adresu IP.





```
NameVirtualHost *:80
<VirtualHost *:80>
    ServerName domena1.edu.pl
    ServerAlias www.domena1.edu.pl
    DocumentRoot /home/domena1
</VirtualHost>
<VirtualHost *:80>
    ServerName domena2.org.pl
    ServerAlias www.domena2.org.pl
    DocumentRoot /home/domena2
</VirtualHost>
```

\* oznacza nasłuchiwanie odwołań na wszystkich adresach IP serwera. Można ją zastąpić konkretnym adresem IP serweta np. 192.168.10.1. Z powyższą konfiguracją serwer będzie wystawiał strony z katalogu **/home/domena1** dla odwołań do **domena1.edu.pl** i z katalogu **/home/domena2** dla odwołań do **domena2.org.pl**.

Sekcja **VirtualHost** umożliwia dla każdego serwera wirtualnego zdefiniowanie oddzielnie zmienionych opcji konfiguracyjnych w stosunku do głównej domeny. Oczywiście należy zapewnić prawidłowe rozwiązywanie nazw dla domena1.edu.pl i domena2.org.pl na adres IP serwera. (Dla potrzeb ćwiczenia wystarczy przypisać adresy domenowe do adresu IP w pliku **/etc/hosts**).



## Plik .htaccess

- umożliwia zmianę konfiguracji pracy serwera webowego dla wybranego katalogu o ile ustawiono opcję AllowOverride

*#lista plików w katalogu z wyjątkiem wybranych*

**Options +Indexes**

**IndexIgnore .?\* \*.php**

*#zmiana domyślnego kodowania znaków*

**AddDefaultCharset UTF-8**

*#download plików PDF bez otwierania w przeglądarce*

**AddType application/octet-stream .pdf**



## Plik .htaccess

- ogranicza dostęp do katalogu z określonych podsieci  
**order allow, deny**

**deny from 153.19.48.**

**deny from 10.1.1.**

**allow from all**

- pozwala na uwierzytelnianie dostępu do plików w danym katalogu

**AuthName "Podaj hasło"**

**AuthType Basic**

**AuthUserFile "/var/www/.htpasswd"**

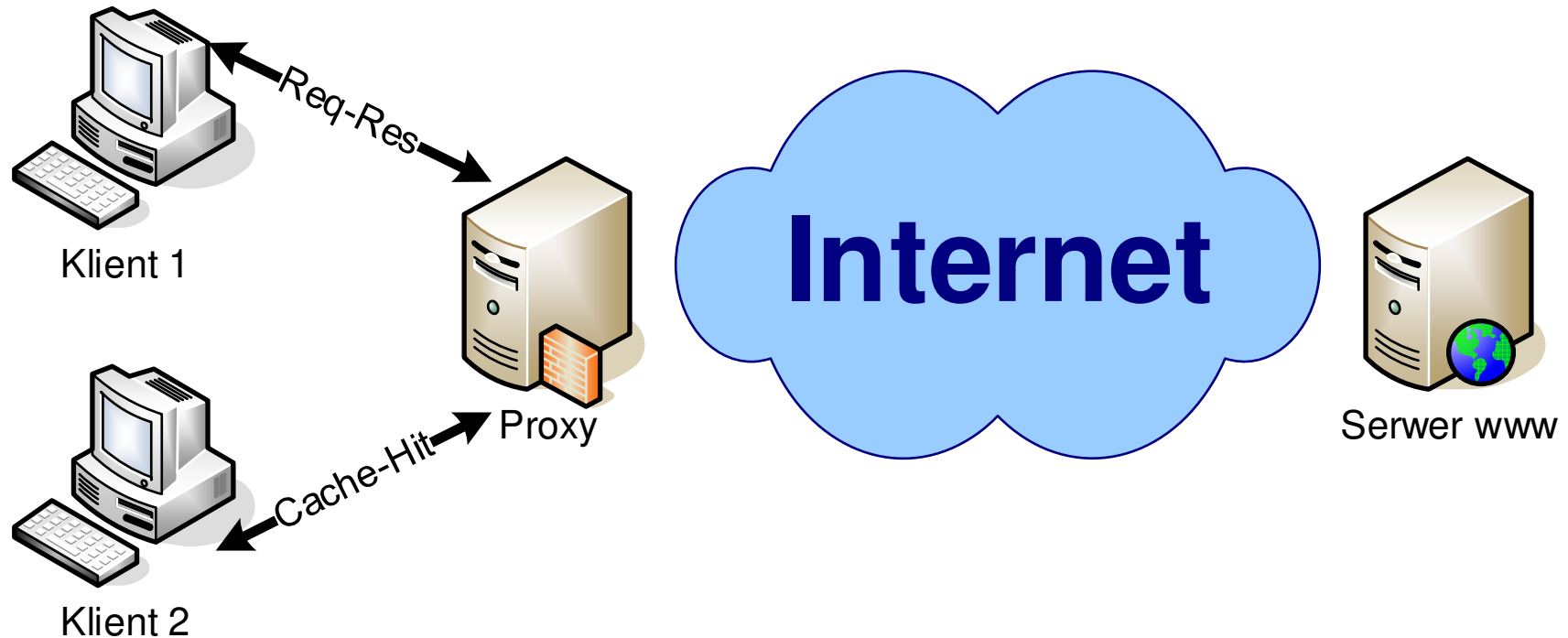
**AuthGroupFile /dev/null**

**require valid-user**

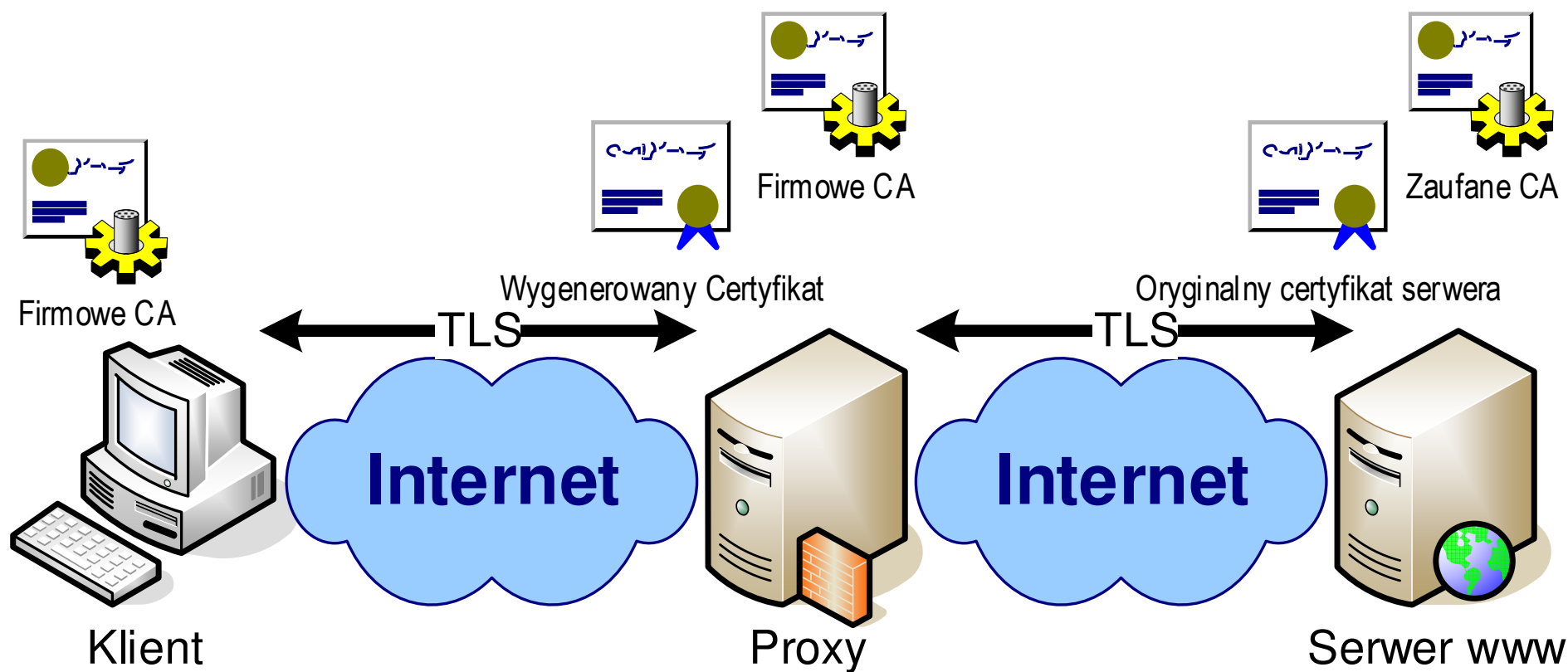


# PROXY HTTP

## HTTP Proxy jako zapora aplikacyjna



# Inspekcja ruchu szyfrowanego HTTPS





## Zmienne HTTP ustawiane dodatkowo przez serwer proxy:

REMOTE\_ADDR = 194.85.1.1

HTTP\_ACCEPT\_LANGUAGE = pl

HTTP\_USER\_AGENT = Mozilla/4.0 (compatible; MSIE 6.0; Windows 5.1)

HTTP\_HOST = www.wp.pl

**HTTP\_VIA = 194.85.1.1 (Squid/2.4.STABLE7)**

**HTTP\_X\_FORWARDED\_FOR = 192.168.5.5**





## Plik konfiguracyjny `/etc/squid/squid.conf`

`# http_port 3128`

**`http_port 8080`**

`# maximum_object_size 4096 KB`

**`maximum_object_size 512 MB`**

`# cache_dir ufs /var/spool/squid 100 16 256`

**`cache_dir ufs /var/spool/squid 2048 256 256`**



# Wpisy dotyczące sieci lokalnej

**acl SIEC src 10.8.0.0/255.255.0.0**

**acl PC045 src 10.8.1.145/255.255.255.255**

**acl PC048 src 10.8.1.148/255.255.255.255**

**acl PC052 src 10.8.1.152/255.255.255.255**

**acl PC060 src 10.8.1.160/255.255.255.255**

**acl serwis.usb src 10.8.1.200/255.255.255.255**

# Wpisy dotyczące całych serwisów

**acl POCZTA url\_regex poczta**

**acl POCZTA2 dstdomain profil.wp.pl gmail.com**

**acl WIN4UPDATE dstdomain download.windowsupdate.com**

**v4.windowsupdate.microsoft.com windowsupdate.microsoft.com**

**wustat.microsoft.com**



# Reguły dotyczące LAN

**http\_access allow localhost**

**http\_access allow WIN4UPDATE**

**http\_access deny PC045**

**http\_access allow PC048**

**http\_access deny PC052**

**http\_access allow PC060**

**http\_access allow serwis.usb**

**http\_access deny POCZTA**

**http\_access deny POCZTA2**

**http\_access allow SIEC**

# Na koniec zakaz wszystkich innych dostępów

**http\_access deny all**



## Podstawowe typy wpisów acl:

src

dst

srcdomain

dstdomain

srcdom\_regex

dstdom\_regex

url\_regex

urlpath\_regex

time

browser

**itd...**

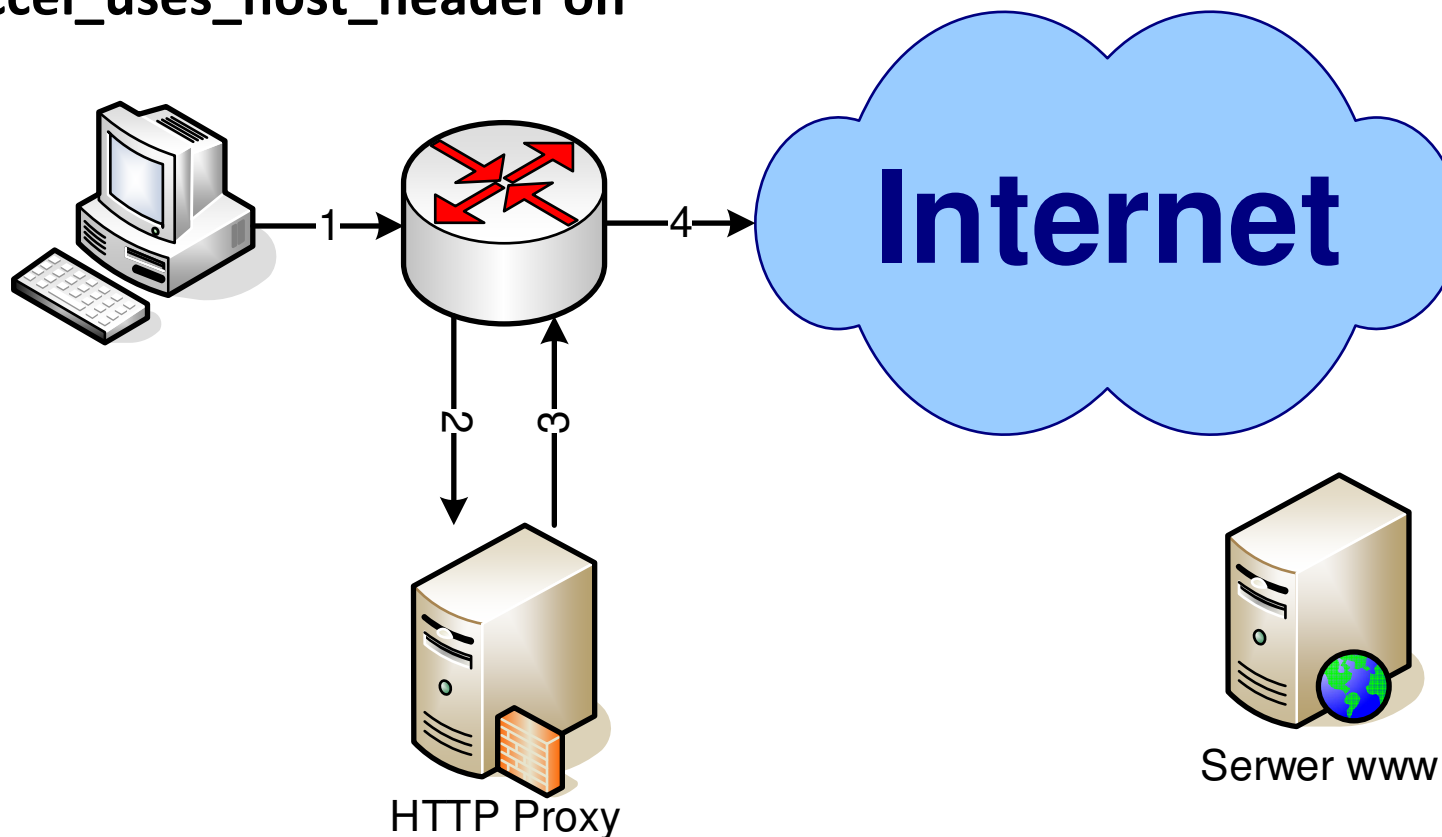
#Dodatkowo dla **transparentnego proxy**

**httpd\_accel\_host virtual**

**httpd\_accel\_port 80**

**httpd\_accel\_with\_proxy on**

**httpd\_accel\_uses\_host\_header on**





## Automatyczna konfiguracja proxy w przeglądarkach przez DNS wpad

```
function FindProxyForURL(url, host)
{
  if (url.substring(0, 6) == "https:" || url.substring(0, 6) == "snews:")
    return "DIRECT";
  if (isPlainHostName(host)) return "DIRECT";
  if (dnsDomainIs(host, "127.0.0.1")) return "DIRECT";
  if (dnsDomainIs(host, "localhost")) return "DIRECT";
  if (dnsDomainIs(host, "moja.domena.pl")) return "DIRECT";
  return "PROXY 192.168.2.1:8080";
}
```

Jeśli adres *wpad.moja.domena/wpad.dat* jest aktywny to działa automatyczna konfiguracja ustawień proxy dla przeglądarek. Wymaga to wpisu w lokalnym DNS-ie dla nazwy *wpad* i umieszczenia pliku *wpad.dat* (*proxy.pac*) w głównym folderze serwera www na hoście o nazwie *wpad*.

Czasami konieczne jest dodanie linii do konfiguracji mime.types serwera www:

**application/x-ns-proxy-autoconfig dat pac**



## **Automatyczna konfiguracja proxy w systemach Linux**

W systemach Unix/Linux istnieje możliwość globalnego zdefiniowania serwera proxy przez zmienną środowiskową `http_proxy`.

```
http_proxy=http://user:password@192.2.0.1:8080  
export http_proxy
```

Z tak zdefiniowanego proxy korzystać potrafi większość aplikacji w systemie. Nie tylko przeglądarki ale również menadżery aktualizacji, akceleratorzy pobierania, aplikacje typu `wget`, `lynx`, `elinks` itp.



## Uruchamianie squid

Przed pierwszym uruchomieniem należy utworzyć strukturę katalogów dla pamięci cache

**squid -z**

Uruchomienie squid

**service squid start**

Zatrzymanie squid

**service squid stop**





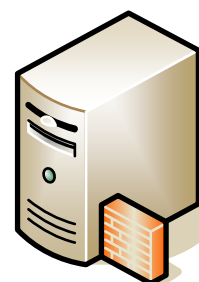
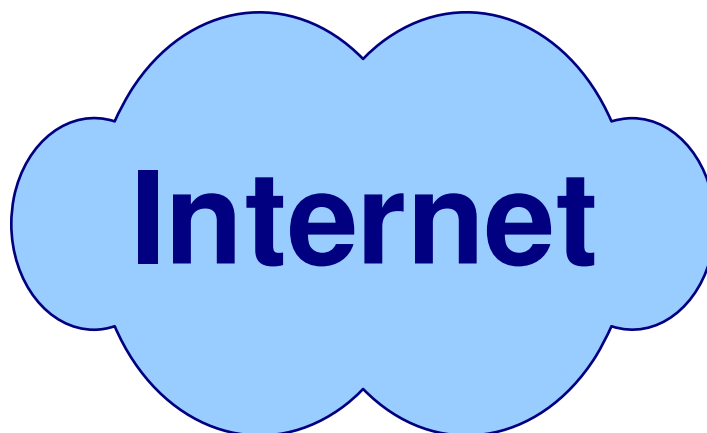
## Web caching – odciążanie serwera webowego



Klient 1



Klient 2



Proxy



Serwer www



# Zapora sieciowa

# Firewall



## ZAPORA SECIOWA – FIREWALL

Encyklopedyczna definicja zapory sieciowej zwykle definiuje zaporę sieciową jako **rozwiązanie sprzętowe lub programowe, które przeciwdziała nieautoryzowanemu dostępowi z sieci albo do sieci.**

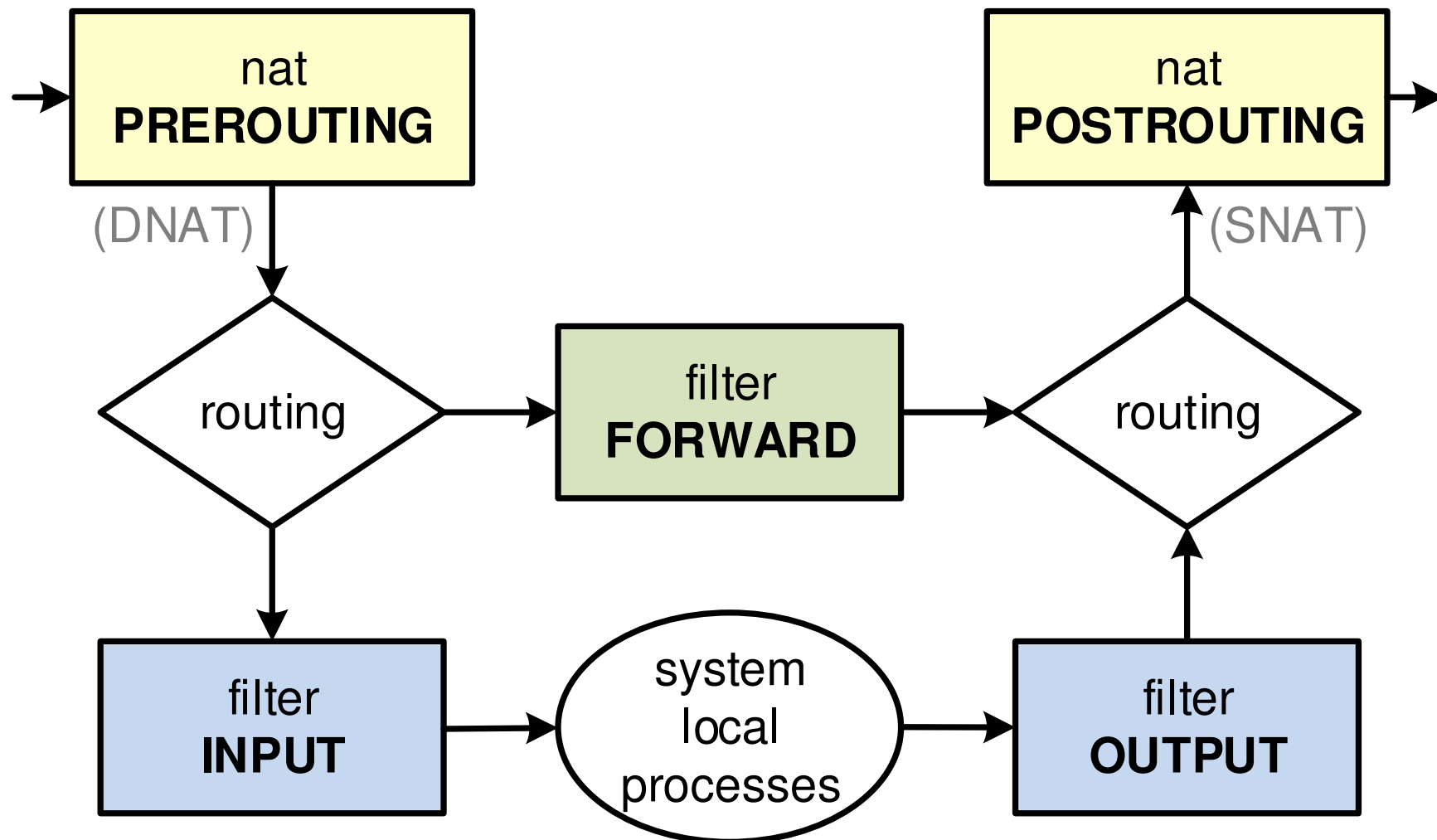
Lepszą definicją jest **zapewnienie wszelkich technik i narzędzi niezbędnych dla bezpiecznej i skutecznej wymiany informacji między urządzeniami i sieciami.**

Możemy wyróżnić 3 generacje zapór:

1. **Zapory** typu *stateless*, w których filtracja datagramów bazuje wyłącznie na klasyfikatorach testujących zawartość nagłówków datagramów L2, L3 i L4.
2. **Zapory** typu *stateful*, które podejmują decyzje śledząc stan wszystkich nawiązywanych połączeń.
3. **Zapory aplikacyjne**, w których inspekcji podlegają specyficzne dla danej usługi lub aplikacji informacje również z warstw L5-L7.

## KONFIGURACJA FILTRACJI DATAGRAMÓW – iptables, ip6tables

- trasy datagramów i standardowe filtry





## KONFIGURACJA FILTRACJI DATAGRAMÓW – iptables, ip6tables

- Zawartość nagłówków datagramów IPv6 i IPv4

Bity	0-3	4-7	8-11	12-15	16-19	20-23	24-27	28-31
0	Wersja	Klasa ruchu	Etykieta przepływu					
32	Długość danych			Następny nagłówek			Limit przeskoków	
64	Adres źródłowy (128 bitów)							
96								
128								
160								
192	Adres docelowy (128 bitów)							
224								
256								
288								

Bity 0-3	4-7	8-15	16-18	19-23	24-31
Wersja	IHL	Typ usługi / DS (6b) + ECN (2b)	Długość całkowita		
Identyfikator			Flagi	Przemieszczenie fragmentacji	
TTL		Protokół	Suma kontrolna nagłówka		
Adres źródłowy					
Adres docelowy					
Opcje					Dopełnienie



## KONFIGURACJA FILTRACJI DATAGRAMÓW – iptables (ip6tables)

- Zawartość nagłówków datagramów protokołów UDP i TCP

+	Bity 0 - 15	16 - 31
0	Port nadawcy	Port odbiorcy
32	Długość	Suma kontrolna
64	Dane	

TCP Header																																		
Offset Oktet		0								1								2								3								
Oktet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
0	0	Port nadawcy																Port odbiorcy																
4	32	Numer sekwencyjny																																
8	64	Numer potwierdzenia (jeżeli flaga ACK jest ustawiona)																																
12	96	Długość nagłówka				Zarezerwowane				N S	C W R	E C R	U R C	A C K	P S H	R S T	S S Y	F I N	Szerokość okna															
16	128	Suma kontrolna																Wskaźnik priorytetu (jeżeli flaga URG jest ustawiona)																
20	160	Opcje (jeżeli długość nagłówka > 5, to pole jest uzupełniane "0")																																
...	...	...																																



## **Parametry polecenia iptables, ip6tables**

Operacje na całych łańcuchach:

- N** Utworzenie nowego filtru
- X** Skasowanie pustych filtrów
- P** Zmiana polityki dla wbudowanego filtru
- L** Wypisanie reguł w filtrach
- F** Wyczyszczenie filtrów z reguł
- Z** Wyzerowanie liczników filtrów

Manipulowanie regułami w obrębie filtru:

- A** Dodanie nowej reguły do filtra
- I** Wstawienie nowej reguły na pewnej pozycji w filtrze
- R** Zamiana reguły na określonej pozycji w filtrze
- D** Skasowanie reguły na określonej pozycji w filtrze, albo pierwszej, która pasuje



### Opcje definiowania reguł:

- i** Interfejs wejściowy np. eth0
- o** Interfejs wyjściowy np. ppp0
- s** Adres źródłowy np. 192.168.1.1, 192.168.1.0/24., pc01.domena.pl
- d** Adres docelowy
- p** Protokół np. tcp, udp, icmp
- sport** Port źródłowy np. 22, ssh
- dport** Port docelowy

### Akcje filtrowania:

- j ACCEPT** Zaakceptuj - przepuść
- j REJECT** Odrzuć datagram z komunikatem o błędzie (ICMP)
- j DROP** Odrzuć datagram jakby nigdy nie dotarł
- j nowy** Przenieś datagram do filtru o nazwie **nowy**





## Przykładowa konfiguracja prostego firewalla dla routera z translacją adresów NAT

```
#!/bin/bash
```

```
# PRZYKŁADOWY SKRYPT FIREWALLA
```

```
# Wyczyszczenie wszystkich reguł iptables
```

```
iptables -F
```

```
iptables -X
```

```
iptables -t nat -F
```

```
iptables -t nat -X
```

```
# Ustalenie polityk, czyli domyślnych reguł, filtracji
```

```
iptables -P INPUT DROP
```

```
iptables -P OUTPUT ACCEPT
```

```
iptables -P FORWARD DROP
```



*# Zezwolenie na datagramy przychodzące z zaufanego interfejsu lokalnego lo*

**iptables -A INPUT -j ACCEPT -i lo**

*# Zezwolenie na odbieranie datagramów TCP z ustawioną flagą ACK*

*# oznacza to zgodę na przepuszczanie odpowiedzi na zestawione przez lokalny system połączenia TCP (STATELESS)*

**iptables -A INPUT -j ACCEPT -p tcp --tcp-flags ACK ACK**

*# Zezwolenie na odpowiedzi z serwerów DNS (dla STATELESS)*

**iptables -A INPUT -j ACCEPT -p udp --sport 53 --dport 1024:65535**

*# albo dokładniej testując stan połączeń TCP oraz UDP i ICMP (STATEFUL)*

**iptables -A INPUT -j ACCEPT -m conntrack --ctstate ESTABLISHED,RELATED**

*# Uwaga! Brak spacji przed RELATED*



*# Zakaz dla datagramów z niepoprawnie ustawionymi flagami*

**iptables -A INPUT -j DROP -m conntrack --ctstate INVALID**

*# Limit liczby nowych połączeń w dodatkowym filtrze syn\_flood*

**iptables -N syn\_flood**

**iptables -A INPUT -p tcp -m conntrack --ctstate NEW -j syn\_flood**

**iptables -A syn\_flood -j RETURN -m limit --limit 10/s --limit-burst 5**

**iptables -A syn\_flood -j DROP**

*# Limit liczby obsługiwanych datagramów ICMP*

*# Uwaga: kolejne reguły dla ICMP nie mają wówczas sensu*

**iptables -A INPUT -j ACCEPT -p icmp -m limit --limit 2/s --limit-burst 4**

**iptables -A INPUT -j DROP -p icmp**

*# Zezwolenie na wybrane typy datagramów ICMP - ping i typowe błędy*

**for type in echo-request echo-reply destination-unreachable source-quench  
time-exceeded parameter-problem; do**

**iptables -A INPUT -j ACCEPT -p icmp --icmp-type \$type**

**done**



*# WYJĄTKI FIREWALLA - zezwolenia na dostęp do usług*

*# Zdalna praca poprzez SSH*

**iptables -A INPUT -j ACCEPT -p tcp --dport 22 -m conntrack --ctstate NEW**

*# Czytanie poczty POP3S*

**iptables -A INPUT -j ACCEPT -p tcp --dport 995 -m conntrack --ctstate NEW**

*# Wysyłanie poczty SMTP*

**iptables -A INPUT -j ACCEPT -p tcp --dport 587 -m conntrack --ctstate NEW**

*# Serwer WWW HTTP, HTTPS*

**iptables -A INPUT -j ACCEPT -p tcp --dport 80 -m conntrack --ctstate NEW**

**iptables -A INPUT -j ACCEPT -p tcp --dport 443 -m conntrack --ctstate NEW**

*# FTP ale tylko w trybie pasywnym*

**iptables -A INPUT -j ACCEPT -p tcp --dport 21 -m conntrack --ctstate NEW**

*# Ograniczenie ruchu wychodzącego*

**iptables -A OUTPUT -j REJECT -p tcp --dport 80 -d portal.pl**



*# NAT Network Address Translation*

*# Udostępnienie Internetu dla sieci lokalnej przez SNAT*

**iptables -t nat -A POSTROUTING -o eth0 -s 10.8.0.0/16 -j SNAT --to-source 80.50.200.1**

*# albo przez MASQUERADE*

**iptables -t nat -A POSTROUTING -o ppp0 -s 10.8.0.0/16 -j MASQUERADE**

*# Włączenie routingu datagramów IPv4*

**sysctl -w net.ipv4.ip\_forward=1**

*# Wymuszenie na użytkownikach lokalnego serwera PROXY tzw. TRANSPARENT PROXY*

**iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 80 -j REDIRECT --to-port 8080**

**iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 3128 -j REDIRECT --to-port 8080**

**iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 8080 -j REDIRECT --to-port 8080**

*# Przekierowanie ruchu do wybranych hostów w sieci lokalnej DNAT - PAT*

**iptables -A PREROUTING -t nat -p tcp -d host.edu --dport 3389 -j DNAT --to 10.8.1.170**

**iptables -A PREROUTING -t nat -p tcp -d host.edu --dport 5900 -j DNAT --to 10.8.1.169**

**iptables -A PREROUTING -t nat -p tcp -d host.edu --dport 1022 -j DNAT --to 10.8.1.1:22**



*# Firewall typu Stateful dla wszystkich klientów routera (eth0 WAN, eth1 LAN)*

*# zezwolenie na ruch wyjściowy LAN -> WAN*

**iptables -A FORWARD -j ACCEPT -i eth1 -o eth0 -s 10.8.0.0/16**

*# zezwolenie tylko na odpowiedzi w kierunku WAN -> LAN*

**iptables -A FORWARD -j ACCEPT -i eth0 -o eth1 -d 10.8.0.0/16 \  
-m conntrack --ctstate ESTABLISHED,RELATED**

*# zezwolenie na przekierowanie ruchu wybranych portów do wybranych hostów*

**iptables -A FORWARD -j ACCEPT -d 10.8.1.170 -p tcp --dport 3389 \  
-m conntrack --ctstate NEW**

**iptables -A FORWARD -j ACCEPT -d 10.8.1.169 -p tcp --dport 5900 \  
-m conntrack --ctstate NEW**

**iptables -A FORWARD -j ACCEPT -d 10.8.1.1 -p tcp --dport 22 \  
-m conntrack --ctstate NEW**



## Przykładowa konfiguracja prostego firewalla IPv6

```
#!/bin/bash
```

```
# PRZYKŁADOWY SKRYPT FIREWALLA IPv6
```

```
# Czyścimy wszystkie reguły filtracji
```

```
ip6tables -F
```

```
ip6tables -X
```

```
# Ustalamy polityki (domyślne reguły) filtracji
```

```
ip6tables -P INPUT DROP
```

```
ip6tables -P OUTPUT ACCEPT
```

```
ip6tables -P FORWARD DROP
```

```
# Firewall typu STATEFUL (tylko nawiązane połączenia)
```

```
ip6tables -A INPUT -j ACCEPT -m conntrack --ctstate ESTABLISHED,RELATED
```

```
ip6tables -A INPUT -j DROP -m conntrack --ctstate INVALID
```

```
# Zezwolenie na datagramy z zaufanego interfejsu lo
```

```
ip6tables -A INPUT -j ACCEPT -i lo
```



## Przykładowa konfiguracja prostego firewalla IPv6

*# Neighbour Discovery – mechanizm autokonfiguracji IPv6 i odkrywania sąsiedztwa*

**ip6tables -A INPUT -j ACCEPT -p icmpv6 --icmpv6-type neighbor-advertisement**

**ip6tables -A INPUT -j ACCEPT -p icmpv6 --icmpv6-type neighbor-solicitation**

**ip6tables -A INPUT -j ACCEPT -p icmpv6 --icmpv6-type router-advertisement**

**ip6tables -A INPUT -j ACCEPT -p icmpv6 --icmpv6-type router-solicitation**

*# ostatni wpis tylko dla routera, dla routera potrzebne również...*

*# uruchomienie routingu IPv6*

**sysctl -w net.ipv6.conf.all.forwarding=1**

*# albo trwale*

**echo "net.ipv6.conf.all.forwarding = 1" >> /etc/sysctl.d/ipv6\_forwarding.conf**

**sysctl -p**



## Przykład zawansowanej konfiguracji: algorytm Knok-knok dla SSH

- Algorytm Knok-knok (puk-puk) polega na przepuszczaniu określonych datagramów tylko jeśli bezpośrednio przed nimi zaszło określone zdarzenie. Technikę można stosować dla IPv4 i IPv6 oraz dla różnych zastosowań i różnych metod wyzwalana np. wyzwalaczem mogą być kolejne pingu o określonych rozmiarach.
- W przykładzie pokazano jak zezwolić na nowe połączenie SSH tylko jeśli wcześniej nastąpiły próby połączenia TCP kolejno na porty 12345 a następnie 1234.
- Moduł *recent* pozwala na zapisywanie adresów IP i ich testowanie w różnych aspektach.
- Listy zapisanych adresów są dla przykładu dostępne przez pliki:
  - `/proc/net/xt_recent/KNOK1`
  - `/proc/net/xt_recent/KNOK2`



## Przykład zawansowany konfiguracji: algorytm Knok-knok dla SSH

*#nowe filtry KNOK1 i KNOK2*

**ip6tables -N KNOK1**

**ip6tables -N KNOK2**

*#knok 1 - usunięcie wpisów starszych niż 15s, a następnie dodanie adresu źródła*

**ip6tables -A KNOK1 -m recent --rcheck --seconds 15 --reap --name KNOK1**

**ip6tables -A KNOK1 -j DROP -m recent --set --name KNOK1**

*#knok2 - usunięcie wpisów starszych niż 30s, a następnie dodanie adresu źródła*

**ip6tables -A KNOK2 -m recent --rcheck --seconds 30 --reap --name KNOK2**

**ip6tables -A KNOK2 -j DROP -m recent --set --name KNOK2**

*#akceptacja SSH jeśli ostatnio (<30s) był stan KNOK2*

**ip6tables -A INPUT -j ACCEPT -p tcp --dport 22 -m conntrack --ctstate NEW \**  
**-m recent --rcheck --seconds 30 --reap --name KNOK2**

*#klasyfikator KNOK2 tylko po KNOK1 w czasie <15s*

**ip6tables -A INPUT -j KNOK2 -p tcp --dport 1234 -m conntrack --ctstate NEW \**  
**-m recent --rcheck --seconds 15 --reap --name KNOK1**

*#klasyfikator KNOK1*

**ip6tables -A INPUT -j KNOK1 -p tcp --dport 12345 -m conntrack --ctstate NEW**



# ufw - Uncomplicated Firewall

ufw reset

ufw default deny incoming

ufw default allow outgoing

**ufw allow http,https**

**ufw allow 22/tcp**

**ufw allow 123/udp**

**ufw allow from 192.2.0.0/16 to any port 514 proto udp**

ufw enable

ufw status verbose



# Firewalld

systemctl enable firewalld

systemctl restart firewalld

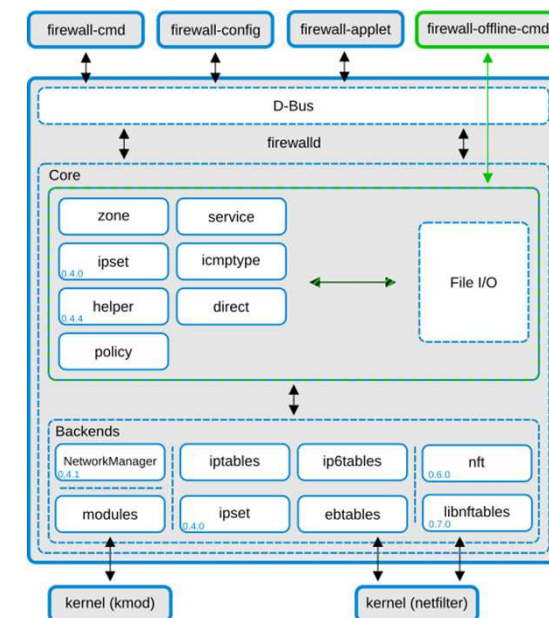
**firewall-cmd --permanent --zone=public --add-service=http**

**firewall-cmd --permanent --zone=public --add-port=22/tcp**

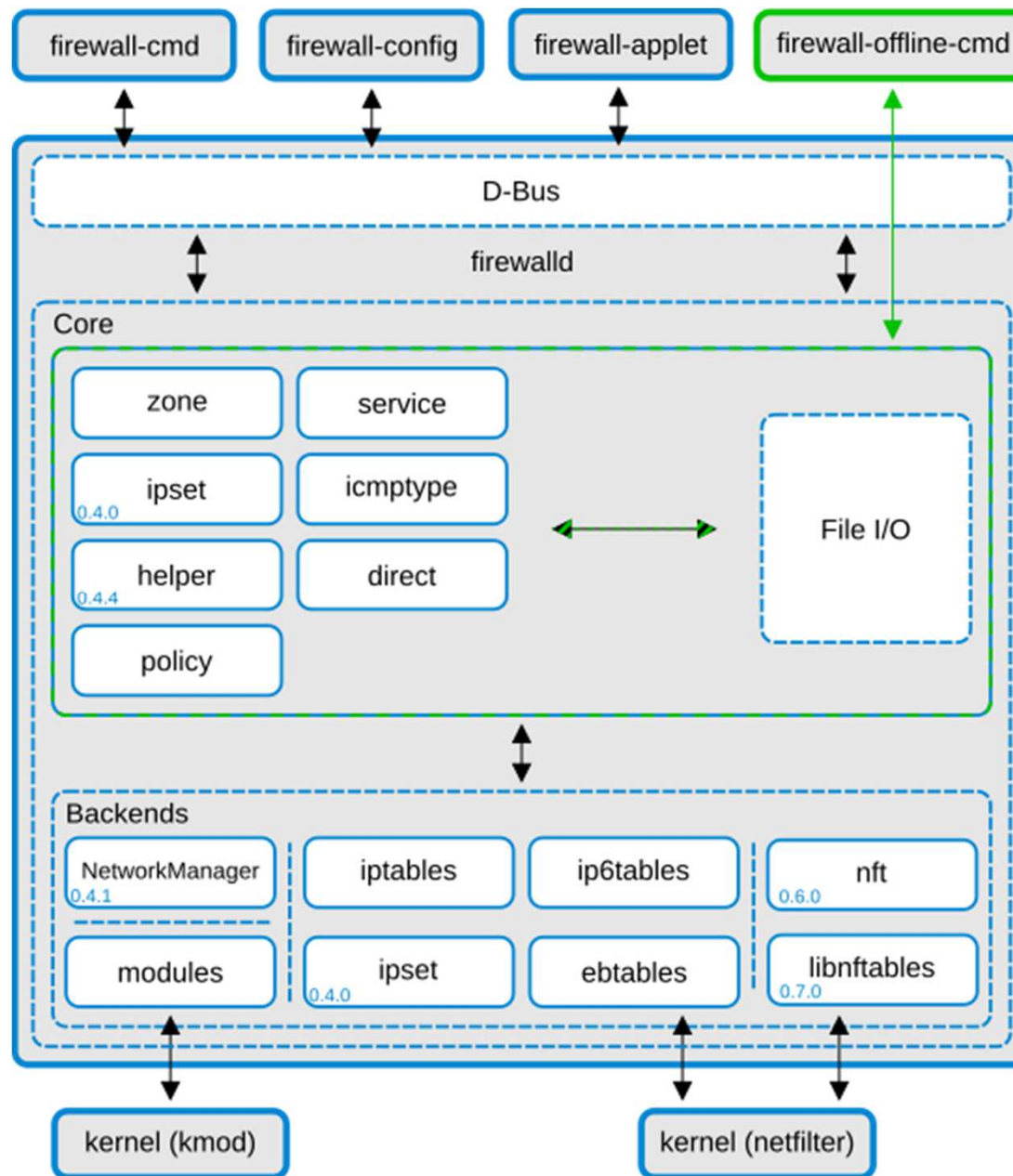
**firewall-cmd --permanent --zone=public --add-port=123/udp**

firewall-cmd --state

firewall-cmd --list-all-zones



# Firewalld





# Netfilter Tables



# nftables

## Krótką historia narzędzi do konfiguracji filtracji datagramów

1. **ipchains, ipfilter** (kernel 2.2 1998)
2. **iptables** (kernel 2.4 2001)

rozszerzone o:

- **ip6tables**
- **arptables**
- **ebtables**

rozbudowane o wygodne narzędzia

- **ufw** (Ubuntu, Debian)
  - **firewalld** (Red Hat)
3. **nftables** (kernel 3.13 2014)



# nftables

## Mocne strony:

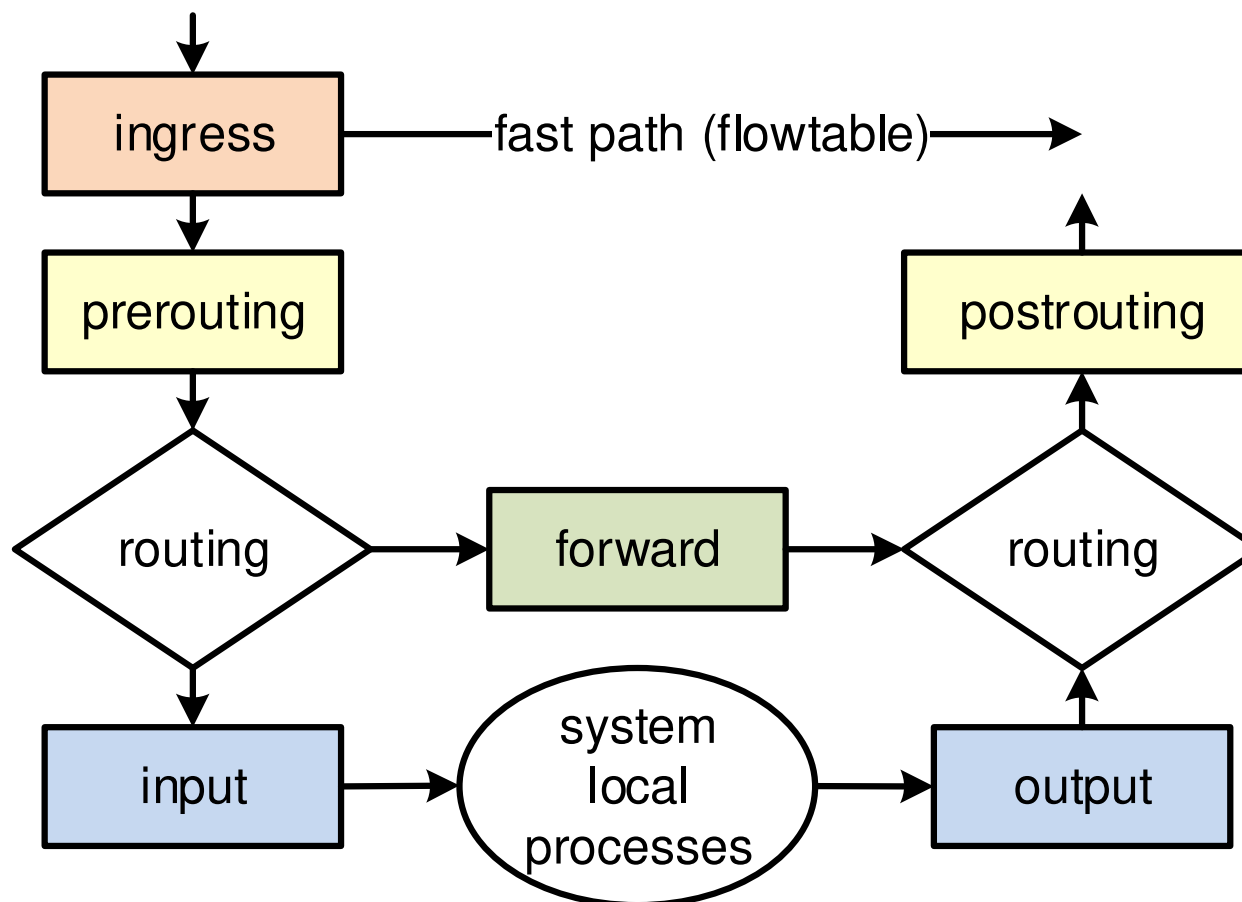
- Natywna obsługa w jądrze Linuksa
- Zachowuje znaną architekturę przepływu datagramów przy braku standardowych tabel i filtrów takich jak INPUT, FORWARD, OUTPUT
- Zastępuje iptables, ip6tables, arptables, ebtables, tylko jednym nft
- Umożliwia równoczesną obsługę IPv4 i IPv6
- Umożliwia konstruowanie złożonych akcji
- Obsługuje złożone struktury danych takie jak zbiory i słowniki
- Upraszcza liczbę reguł do sprawdzenia
- „Drastycznie” (cytat z twórców) zwiększa wydajność
- Wykorzystuje wieloprosesorowość SMP systemów
- Umożliwia optymalizację kodu i wprowadzanie nowych rozwiązań w *user space* bez konieczności modyfikowania jądra systemu

## Słabe strony

- Nowa składnia (w odniesieniu do iptables)
- Konieczność ponownej implementacji narzędzi takich jak ufw i firewalld
  - firewalld już zrobione



## nft Architektura przepływu i miejsc przechwytywania ruchu (ang. *hook*)



Hook **ingres** umożliwia przetwarzanie datagramów Layer2 jeszcze przed interpretacją Layer3 oraz realizację offloadingu fast path.



## nft – składnia CLI / skrypt bash

```
#!/bin/bash
```

```
nft flush ruleset
```

```
nft add table inet firewall
```

```
nft add chain inet firewall incoming \  
    { type filter hook input priority 0 \; policy drop \; }
```

```
nft add rule inet firewall incoming ct state established,related accept
```

```
nft add rule inet firewall incoming ct state invalid drop
```

```
nft add rule inet firewall incoming tcp dport { ssh, http, https } accept
```

```
nft add rule inet firewall incoming icmp type echo-request accept
```

```
nft add rule inet firewall incoming icmpv6 type \  
    { echo-request, nd-neighbor-solicit, nd-router-advert } accept
```

```
nft add rule inet firewall incoming iif lo accept
```



## nft – składnia skrypt nft

```
#!/usr/sbin/nft -f
```

```
define services = { ssh, http, https }  
define icmpv6ok = { echo-request, nd-neighbor-solicit, nd-router-advert }
```

```
flush ruleset
```

```
add table inet firewall  
add chain inet firewall incoming \  
    { type filter hook input priority 0; policy drop; }
```

```
add rule inet firewall incoming ct state established,related accept  
add rule inet firewall incoming ct state invalid drop  
add rule inet firewall incoming tcp dport $services accept  
add rule inet firewall incoming icmp type echo-request accept  
add rule inet firewall incoming icmpv6 type $icmpv6ok accept  
add rule inet firewall incoming iif lo accept
```



# nft – składnia konfiguracja nft

```
#!/usr/sbin/nft -f
```

```
define services = { ssh, http, https }  
define icmpv6ok = { echo-request, nd-neighbor-solicit, nd-router-advert }
```

```
flush ruleset
```

```
table inet firewall {  
    chain incoming {  
        type filter hook input priority 0; policy drop;  
  
        ct state established,related accept  
        ct state invalid drop  
        tcp dport $services accept  
        icmp type echo-request accept  
        icmpv6 type $icmpv6ok accept  
        iif lo accept  
    }  
}
```



# nft – <https://wiki.nftables.org>

## Protokoły:

ip, ip6, inet

## Operacje:

list, add, insert, replace, delete, flush

## Tablice:

nft list tables

nft add table inet TABLE

nft list table inet TABLE

nft flush table inet TABLE

## Łańcuchy reguł (filtry):

nft add chain inet TABLE INPUT { type filter hook input priority 0 \; }

nft flush chain inet TABLE INPUT

## Reguły:

nft add rule inet TABLE INPUT tcp dport ssh counter accept

nft add rule ip FILTER INCOME icmp type echo-request ip length lt 1052 \

ip saddr 10.1.0.0/16 counter accept

nft list table ip FILTER INCOME -n -a



# nft – Source Network Address Translation

SNAT:

```
nft add table nat
```

```
nft add chain nat postrouting { type nat hook postrouting priority 100 \; }
```

```
nft add rule nat postrouting ip saddr 192.2.0.0/16 oif eth0 snat 192.2.0.2
```

```
nft add rule nat postrouting oif eth0 masquerade
```

IPv6 NPTv6 (ang. IPv6-to-IPv6 Network Prefix Translation):

```
nft add table ip6 nat
```

```
nft add chain ip6 nat postrouting { type nat hook postrouting priority 100 \; }
```

```
nft add rule ip6 nat postrouting oif eth0 masquerade
```



# **nft – Destination Network Address Translation**

DNAT:

```
nft add table nat
```

```
nft add chain nat prerouting { type nat hook prerouting priority \-100 \; }
```

```
nft add rule nat prerouting iif eth0 tcp dport { 80, 443 } dnat 192.2.0.2
```

```
nft add rule nat prerouting iif eth0 tcp dport 2222 redirect to 22
```

```
nft add rule nat prerouting iif eth0 tcp dport map \  
  { 22 : 192.2.0.1, 80 : 192.2.0.2, 443 : 192.2.0.3 }
```



# **SAMBA**

## **SMB**

## **CIFS**





# **Samba - serwer plików i drukarek zgodny z usługami udostępniania zasobów w Windows**

**SMB** – Server Message Block

**CIFS** – Common Internet File System

Atuty:

- Darmowy
- Kompatybilny z MS
- Dobrze skalowalny
- Prosty w konfiguracji

Konfiguracja:

- `/etc/samba/smb.conf`



*#Bardzo prosta kompletna konfiguracja*

**[global]**

**workgroup = GRUPA**

**netbios name = SERWER**

*#Udostępnianie zasoby*

**[wspolny]**

**path = /home/wspolny**

**read only = no**

**guest ok = yes**

**force user = nobody**

**[prywatny]**

**path = /home/prywatny**

**read only = no**

**guest ok = no**

**valid users = jasio, kasia**



## Konto administratora domeny

- niezbędne do dodawania hostów do domeny.

```
smbpasswd -a root
```

New samba password: . . . .

## Konta zwykłych użytkowników domeny

- tu bez prawa logowania się do systemu.

```
useradd -g users -d /home/jasio -s /bin/false jasio
```

```
smbpasswd -a jasio
```

New samba password: . . .

Uwaga: Hasła samby nie muszą i zwykle nie powinny być identyczne z hasłami systemowymi, szczególnie gdy mogą występować trudności z synchronizacją.



## Obsługa sieci pod systemem Windows

- Przeglądanie sieci

**NET VIEW**

- Przeglądanie zasobów udostępnionych przez konkretny serwer

**NET VIEW \\SERWER**

- Synchronizacja czasu z serwerem o nazwie SERWER

**NET TIME \\SERWER /SET /YES**

- Przyłączenie zasobu sieciowego jako dysku H:

**NET USE H: \\SERWER\WSPOLNY**

- Przyłączenie zasobu sieciowego jako pierwszy wolny dysk

**NET USE \* \\SERWER\PUB**

- Przyłączenie drukarki sieciowej

**NET USE LPT1: \\SERWER\HP1200**



## Obsługa sieci pod systemem Linux

- Przeglądanie zasobów udostępnionych przez konkretny serwer

**`smbclient -N -L SERWER`**

- Podłączenie dysku sieciowego

**`mount -t cifs -o guest //serwer/wspolny /mnt/public`**

- Podłączanie z uwierzytelnianiem

**`mount -t cifs -o username=jasio -o password=tajne \  
//serwer/wspolny /mnt/public`**