Network Address Translation

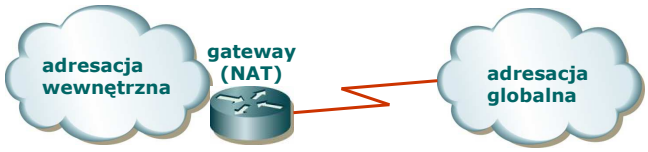
Adresy prywatne

- RFC 1918
- **Niepubliczne** czyli do użycia w sieci wewnętrznej, ale nie w Internecie
- Z reguły filtrowane przez ISP

Klasa A - 1	**10.0.0.0**	**10.0.0.0 /8**
Klasa B - 16	**172.16.0.0 – 172.31.255.255**	**172.16.0.0 /12**
Klasa C - 256	**192.168.0.0 – 192.168.255.255**	**192.168.0.0 /16**

Idea NAT
Network Address Translation

- Mechanizm NAT pozwala na podmianę adresów wewnętrznych (prywatne) na publiczne (rutowalne).
- Obsługuje go ruter graniczny (*border gateway router*) sieci końcowej (*stub*).



Cele zastosowania NAT

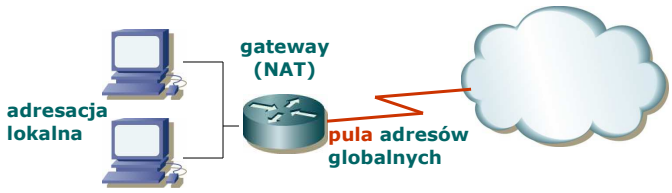
- Oszczędzenie zużycia adresów IP (multipleksacja PAT).
- Eliminacja konieczności readresacji (np.: przy zmianie dostawcy).
- Zwiększenie bezpieczeństwa sieci (ukrycie adresacji wewnętrznej).
- Przekierowanie ruchu TCP na inny port TCP.
- W czasie przebudowy sieci (po zmianie adresu serwera nie zrekonfigurowane maszyny nadal powinny go widzieć).
- Umożliwienie komunikacji sieciom o nakładającej się adresacji.

Terminologia NAT

Adresy:

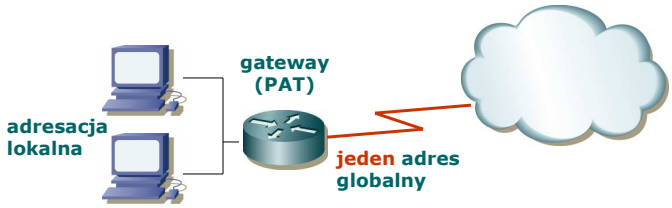
- **wewnętrzny lokalny** –
  adr. w sieci wewn., zazwyczaj prywatny a nie przypisany przez Network Information Center
- **wewnętrzny globalny** –
  adr. oficjalny reprezentujący hosta lub grupę hostów z sieci wewn. w sieci zewn.
- **zewnętrzny lokalny** –
  adr. maszyny zewn. tak, jak widzi go host wewn.
- **zewnętrzny globalny** –
  adr. maszyny zewn. przypisane jej przez właściciela

Rozwiązania NAT



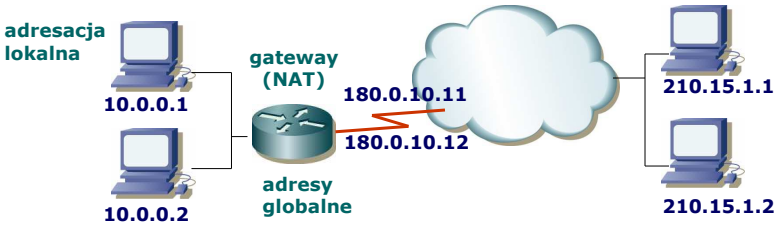
- **NAT statyczny**
  odwzoruje konkretne adresy wewn. lokalne na konkretne, ustalone przez konfiguracje adresy globalne;
  wpisy są niezmienne w czasie.
- **NAT dynamiczny**
  odwzoruje adresy wewn. na dowolne adresy z przyznanej puli adresów globalnych dynamicznie w trakcie komunikacji; wpisy mają określony czas ważności!

PAT – Port Address Translation



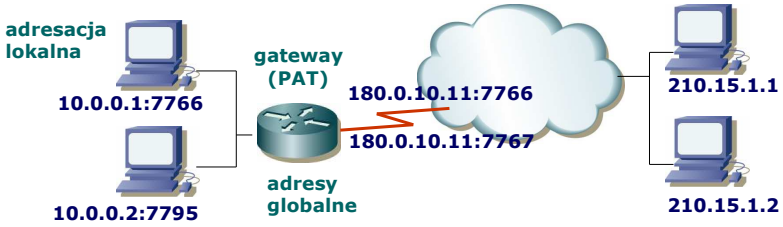
- NAT polega na odwzorowaniu jeden-do-jednego, zaś PAT – wiele-do-jednego. Nazywa się to przeładowaniem.
- Działanie PAT opiera się o użycie numerów portów do identyfikacji maszyn. Teoretycznie dostępnych jest 65536 portów, w rzeczywistości – ok. 4000.

Tablica NAT



wewnętrzny lokalny	wewnętrzny globalny	zewnętrzny lokalny	zewnętrzny globalny
10.0.0.1	**180.0.10.11**	**210.15.1.1**	**210.15.1.1**
10.0.0.2	**180.0.10.12**	**210.15.1.2**	**210.15.1.2**

Działanie PAT



wewnętrzny lokalny	wewnętrzny globalny	zewnętrzny lokalny/globalny
10.0.0.1:7766	**180.0.10.11:7766**	**210.15.1.1:8080**
10.0.0.2:7795	**180.0.10.11:7767**	**210.15.1.2:8080**

- PAT przydziela kolejnym adresom:portom wewnętrznym, kolejne porty adresu zewnęrznego;
- Gdy dostępne porty się skończą – bierze następny adres.
- PAT stara się odwzorowywać numery portów na identyczne.

Uwagi dodatkowe

- kierunek translacji adresów może być zarówno '*inside*' jak i '*outside*':
  - *ip nat inside* tłumaczy adresy źródłowe w pakietach z sieci wewn. oraz docelowe w pakietach z sieci zewn.
  - *ip nat outside* tłumaczy adresy docelowe w pakietach z sieci wewn. oraz źródłowe w pakietach z sieci zewn.
- kolejność czynności:
  - 'in → out' najpierw routing, potem translacja,
  - 'out → in' najpierw translacja, potem routing.

UWAGI

- NAT zwiększa opóźnienia:
  - CPU sprawdza każdy pakiet,
  - ewentualnie modyfikuje nagłówki (TCP, IP).
- Zastosowanie NAT powoduje utratę pewnej funkcjonalności (protokoły oparte o wysyłanie informacji o adresie IP)
- Tracimy zdolność śledzenia pakietów IP od nadawcy do odbiorcy (niektóre aplikacje mogą nie działać)

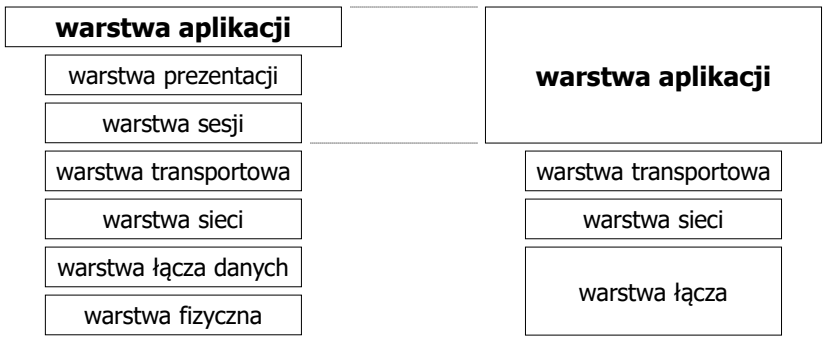
Warstwa aplikacji

Warstwa aplikacji

- Warstwa najwyższa
- Komunikuje się z użytkownikiem
- Obejmuje programy użytkowe:
  - Telnet – zdalny terminal
  - FTP – przesyłanie plików
  - SMTP – poczta elektroniczna
  - DNS – system nazw domen
  - HTTP – usługa WWW
  - i inne

Warstwa aplikacji

- Warstwa 7 modelu OSI/ISO
- Warstwa 4 modelu TCP/IP



Warstwa aplikacji

HyperText Transfer Protocol

HTTP

HyperText Transfer Protocol

- Przesyłanie zróżnicowanych rodzajów danych - zasobów (ang. resource):
  - strony HTML,
  - pliki graficzne, dane multimedialne,
  - aplikacje,
  - inne.
- Zasoby identyfikowane przez
  - URL Uniform Resource Locator

    http://www.cs.agh.edu.pl/dydaktyka/index.html

HTTP

- Protokół klient-serwer
  - serwer: serwer WWW, httpd
  - klient:
    - najczęściej przeglądarka HTML
    - specjalizowane aplikacje wykorzystujące HTTP do transferu danych
- Protokół bezstanowy i bezpołączeniowy
  - działa w oparciu o model żądanie/odpowiedź
  - po dostarczeniu danych połączenie najczęściej jest zamykane

HTTP

- Domyślnie używa *dobrze znanego* portu TCP 80
  - można używać inne numery portów, np. 8080
    - http://www.server.com:8080/
  - możliwe wykorzystanie innego niż TCP, ale niezawodnego protokołu transportowego

Pakiet HTTP

- Nie ma ścisłego podziału na pola
- Komendy oddzielone są końcem linii
- Postać:
  - typ operacji, jedna linia
  - zero lub więcej linii z parametrami postaci: **`nazwa: wartość`**
  - pusta linia
  - opcjonalne dane
    - zasób
    - treść formularza

Żądania i odpowiedzi HTTP

- Żądania HTTP
  - GET
  - POST
  - HEAD
  - inne - PUT, DELETE
- Odpowiedzi
  - kod odpowiedzi + tekst
- Po uzyskaniu odpowiedzi połączenie TCP między klientem a serwerem najczęściej jest zamykane

Żądania HTTP - GET

- Używane najczęściej
- Ciąg znaków identyfikujący zasób na serwerze
  - najprościej: statyczny zasób serwera

```
GET /dydaktyka/wyniki.html HTTP/1.0
```

  - parametry do skryptu lub bazy danych

```
GET /dane/script.cgi?field1=value1
  &field2=value2 HTTP/1.0
```

- Stosowane do przesyłania małych ilości informacji

Żądania HTTP - POST

- Żądanie nie jest zawarte w URL lecz w samym ciele informacji

```
POST /dane/script.cgi HTTP/1.0
Content-Type: application/x-www-form-urlencoded
Content-Length: 76
                    Dane (z wypełnionego formularza; będą
                    przetworzone przez powyższy skrypt)
home=Cosby&favorite+flavor=flies
```

- Często używane przy pobieraniu informacji dla stron generowanych dynamicznie lub do wysyłania formularzy

Żądania HTTP - HEAD

- Analogicznie jak GET; zwraca jedynie nagłówek strony, nie sam zasób
- Przydatne do zorientowania się w zawartości strony przed jej pobraniem (lub zamiast pobrania)

Inne żądania HTTP

- PUT
  - zapisuje dołączony zasób pod podanym URL
- DELETE
  - usuwa zasób podany w URL
  - nie ma gwarancji wykonania akcji

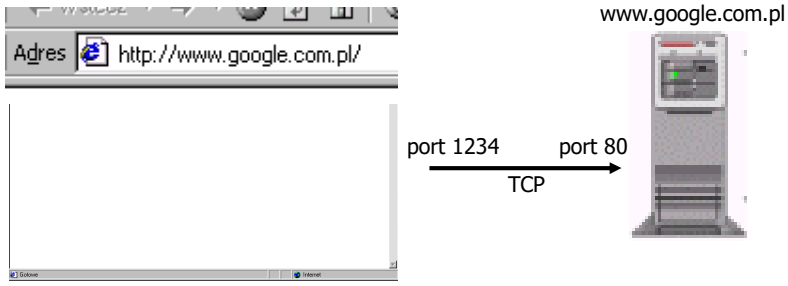
Odpowiedzi HTTP

- **1xx** - informacja
- **2xx** - powodzenie, żądanie zrozumiane i zaakceptowane
  - np. 200 OK
- **3xx** – musi zostać podjęta dalsza akcja
  - np. 301 Moved Permanently, 304 Not Modified
- **4xx** - błąd po stronie klienta
  - najczęściej 404 Not Found, także 403 Forbidden, 401 Unauthorized
- **5xx** - błąd po stronie serwera
  - np. 500 Internal Server Error, 501 Not Implemented

Negocjowalne opcje

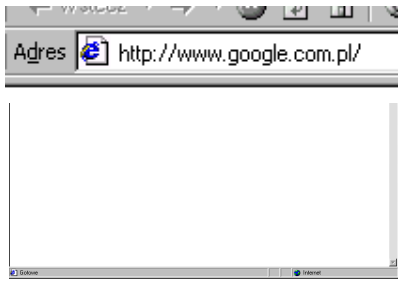
- Żądania od klienta do serwera
  - podanych może być kilka akceptowalnych wartości
  - parametr `q=`*`wartość`* określa preferencje klienta
  - możliwa odpowiedź serwera: `406 Not Acceptable`
- Przykłady
  - standard kodowania znaków (`Accept-Charset`)
    `Accept-Charset: iso-8859-1,*,utf-8`
  - standard kompresji (`Accept-Encoding`)
    `Accept-Encoding: compress, gzip`
  - język naturalny (`Accept-Language`)
    `Accept-Language: pl, en-gb;q=0.8, en;q=0.7`

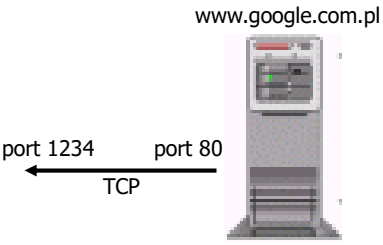
Przykład połączenia HTTP




```
GET / HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Windows 98;
   DigExt)
Host: www.google.com.pl
Accept-Language: pl      # możliwość negocjacji wersji językowej
Connection: Keep-Alive
Cookie: ....
```

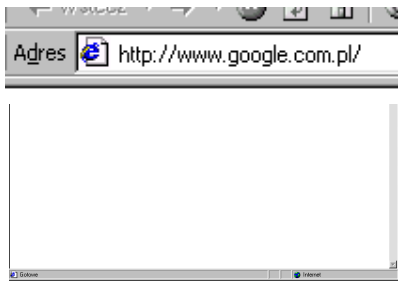
Przykład połączenia HTTP

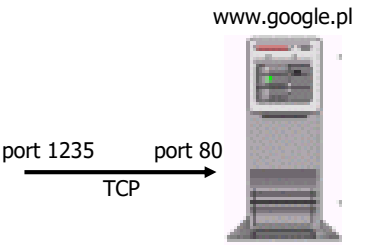





```
HTTP/1.0 302 Moved Temporarily
Content-Length: 151
Date: Tue, 07 Jan 2003 18:13:42 GMT
Content-Type: text/html
Location: http://www.google.pl
Set-Cookie: ...
# poniżej dane (151 B)
<HTML><HEAD><TITLE>302 Moved</TITLE></HEAD><BODY><H1>The document has moved <A
   HREF=„http://www.google.pl/”>here</A></BODY></HTML>
```

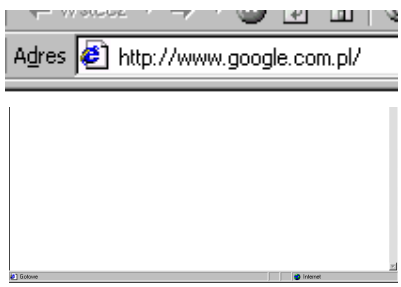
Przykład połączenia HTTP

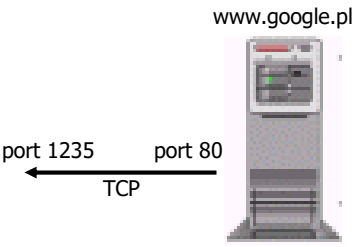





```
GET / HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)
Host: www.google.pl
Accept-Language: pl
Connection: Keep-Alive
Cookie: ....
```

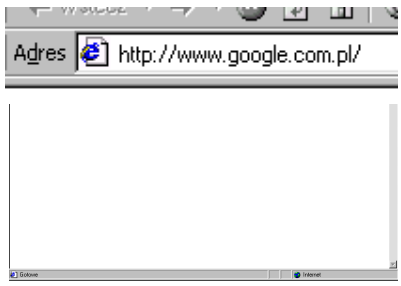
Przykład połączenia HTTP

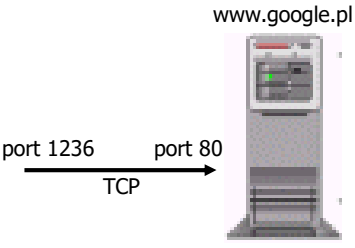





```
HTTP/1.0 200 OK
Content-Length: 1420
Date: Tue, 07 Jan 2003 18:13:43 GMT
Content-Type: text/html
Dane (1228 B)
-------- # kontynuacja w następnym pakiecie
Dane (28 B)
-------- # kontynuacja w następnym pakiecie
Dane (164 B)
```

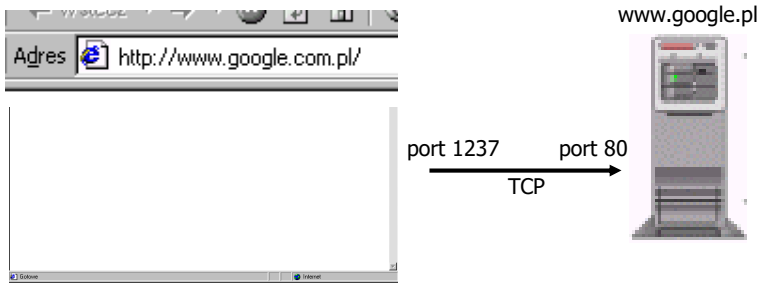
Przykład połączenia HTTP





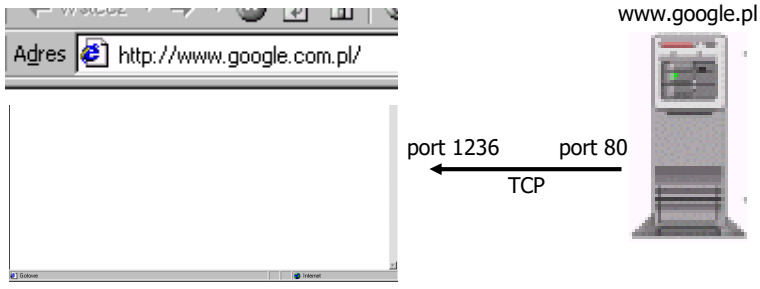

```
GET /images/hp0.gif HTTP/1.1
Accept-Language: pl
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)
Host: www.google.pl
Connection: Keep-Alive
Cookie: ....
```

Przykład połączenia HTTP



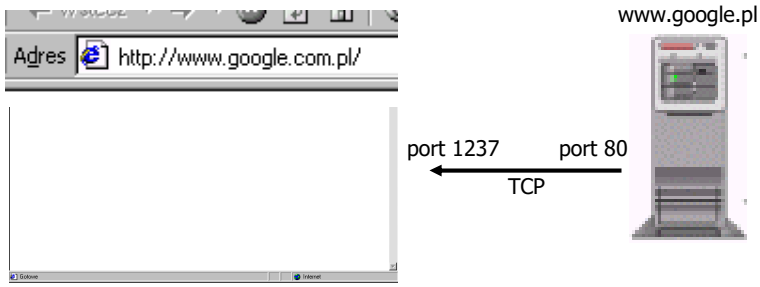

```
GET /images/hp1.gif HTTP/1.1
If-Modified-Since: Wed, 28 Nov 2001 19:25:06 GMT # jest w buforze
Accept-Language: pl
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)
Host: www.google.pl
Connection: Keep-Alive
Cookie: ....
```

Przykład połączenia HTTP




```
HTTP/1.0 200 OK.
Date: Wed, 07 Jan 2003 18:13:44 GMT
Last-Modified: Wed, 28 Nov 2001 19:25:06
Content-Type: image/gif                 # typ zasobu
Content-Length: 4277                    # jego długość
Expires: Sun, 17 Jan 2038 19:14:07 GMT
Dane (1198 B)
--------- # kontynuacje w następnych pakietach, stosownie do MSS
Dane (1460 B)
```

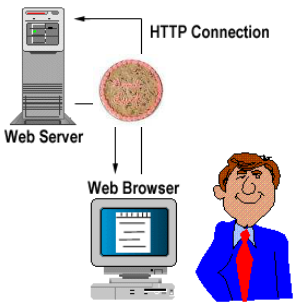
Przykład połączenia HTTP




```
HTTP/1.0 304 Not Modified
Content-Length: 0                    # nie ma potrzeby wysyłać
Date: Thu, 07 Jan 2003 18:13:45 GMT
Content-Type: image/gif
Age: 893
```

Ciasteczko - *cookie*

- Bezstanowość HTTP - każde odwołanie do serwera HTTP jest takie samo
- Ciasteczka to sposób na:
  - utrzymanie stanu sesji,
  - zapewnienia personalizacji stron WWW,
  - prowadzenie statystyk przez administratorów serwera.



Ciasteczka, cd.

- Ciąg znaków przechowywany w pamięci przeglądarki. Jeśli czas ważności jest dłuższy - może być zapisany w pliku
- Przekazywane w nagłówku HTTP
- Wady:
  - związane z pojedynczym komputerem (a nie z użytkownikiem),
  - mogą być łatwo usunięte z komputera użytkownika.
- Wymagania/ograniczenia:
  - 300 ciasteczek,
  - 4 kB na ciasteczko,
  - 20 ciasteczek na domenę.

Ciasteczka, cd.

- Zawartość: sekwencje par nazwa-wartość
- Przekazywanie od serwera do klienta:

Set-Cookie: *nazwa*=*wartość*; **expires=***data*;
path=*ścieżka*; **domain=**nazwa-domeny;
secure=true/false

secure=true/false - wskazuje, czy ciasteczko ma być używane jedynie w stronach zabezpieczonych kryptograficznie - domyślnie wyłączone

- Przekazywanie od klienta do serwera:

Cookie: *nazwa1*=*wartość1*; *nazwa2*=*wartość2* ...

- Przeglądarka na podstawie czasu i parametrów pobieranego URL podejmuje decyzję, czy i które ciasteczko przesłać

Ciasteczka - przykład

1. Pierwsze odwołanie do serwera www.pajacyk.pl


```
GET /cgi-bin/nzlicz.cgi HTTP/1.0
Referer: http://www.pajacyk.pl
Host: www.pajacyk.pl
Cookie: cc=cc
```

2. Odpowiedź serwera


```
HTTP/1.0 200 OK.
Date: Tue, 07 Jan 2003  19:34:46 GMT
Set-Cookie: Mazurek=Mazurek;
  expires=Tue,7-Jan-2003 21:59:59 GMT
Content-type: text/html
```

zapis w pliku przeglądarki WWW:
www.pajacyk.pl FALSE /cgi-bin FALSE 1041976799 Mazurek Mazurek
(1041976799 : ilość sekund od 1.1.1970 = 7 stycznia 2003, 21:59:59)

Ciasteczka - przykład

3. Kolejne odwołanie do serwera www.pajacyk.pl (po chwili)


```
GET /cgi-bin/nzlicz.cgi HTTP/1.0
Referer: http://www.pajacyk.pl
Host: www.pajacyk.pl
Cookie: Mazurek=Mazurek; cc=cc
```

4. Odpowiedź serwera


```
HTTP/1.0 200 OK.
Date: Tue, 07 Jan 2003  19:44:11 GMT
Set-Cookie: Mazurek=Mazurek; expires=True,
  7-Jan-2003 21:59:59 GMT
Content-type: text/html
```

Proxy HTTP

- Program pośredniczący między klientem a serwerem HTTP:
  - przechwytuje żądania klienta, przekazuje do serwera, a odpowiedź kieruje do klienta,
  - możliwość buforowania danych,
  - funkcjonalność klienta i serwera równocześnie.
- Niezbędne przy ograniczeniach dostępu
- GET musi zawierać kompletny URL
  - nie może być tak:
    `GET index.html`
  - musi być tak:
    `GET http://www.agh.edu.pl/index.html`

Utrzymywanie połączenia TCP

- HTTP działa na zasadzie żądanie-odpowiedź - po zakończeniu tej transakcji połączenie jest zamykane
  - proste,
  - nieefektywne (wolny start TCP, kosztowne fazy nawiązania i zakończenia).
- HTTP 1.0
  - **`Connection: Keep-Alive`** przesyłane w żądaniu i odpowiedzi,
  - domyślne lub jawne **`Connection: Close`** wygenerowane przez klienta lub serwer zamyka połączenie.

Utrzymywanie połączenia TCP

- HTTP 1.1
  - domyślne utrzymywanie połączenia (tzw. *persistent connection*)
  - wysłanie `Connection:close` w ostatnim żądaniu
  - czasami używa go serwer - klient powinien zaprzestać używania tego połączenia
  - zamknięcie po czasie nieaktywności

HTTP, wersja 1.1

- Wiele poprawek w stosunku do wersji 1.0
  - rozbudowany mechanizm buforowania
  - lepsze wykorzystanie pasma
  - obsługa wielu domen z jednego adresu IP
  - …
- Serwer HTTP 1.1 musi obsługiwać żądania wersji 1.0

HTTP, wersja 2.0

- Opracowany przez IETF w oparciu o protokół SPDY (stworzony dla przeglądarki Google Chrome w 2009 roku)
  - Umożliwia jednoczesną obsługę wielu zapytań do serwera przez przeglądarki WWW.
- Bezpieczeństwo danych – brak automatycznego szyfrowania ale przeglądarki Mozilla Firefox i Google Chrome nie obsłużą HTTP 2.0 w wersji bez szyfrowania danych
- Specyfikacja RFC 7540 (14.05.2015)
- Trwają prace nad HTTP/3 (HTTP/3.0)

Protokół HTTPS

- Wykorzystuje SSL (Secure Socket Layer)
  - otwarty standard opracowany przez Netscape
  - rozwiązanie oparte na kluczu publicznym
  - klucz o długości 128 bitów
  - szyfrowanie danych na czas transmisji kluczem sesji
  - uwierzytelnianie klienta i serwera
- Domyślnie używa portu 443 TCP
- Zastosowania
  - bankowe, biznesowe
  - medyczne
  - inne

Kiedy HTTPS?

- Zalety:
  - wprowadzenie poufności danych i uwierzytelniania obu stron
- Wady:
  - wymaga więcej mocy obliczeniowej
    - nie wszystko musi być szyfrowane (np. grafika)
  - nie każda przeglądarka go obsługuje

Dokumenty dotyczące HTTP

- RFC 1945 -> HTTP/1.0
- RFC 2616 -> HTTP/1.1
- RFC 7540 -> HTTP/2.0 (HTTP/2)
- RFC 2109 -> Cookie

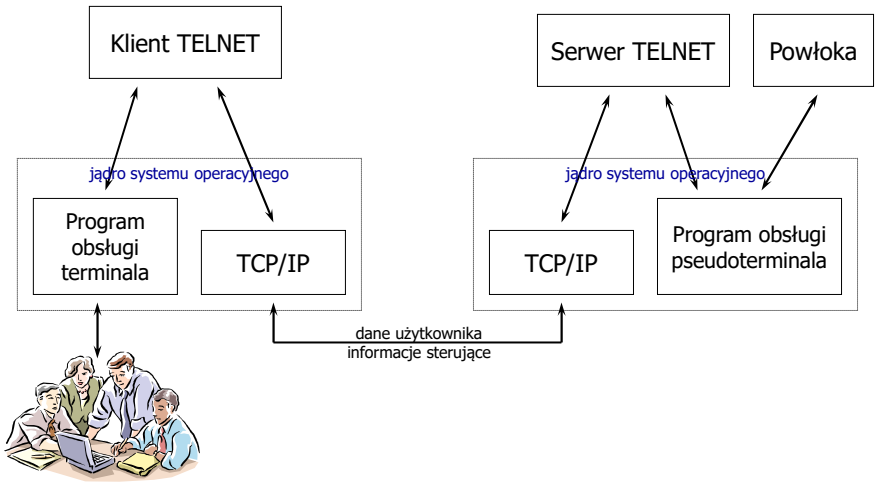
TELecommunications
NETwork protocol

TELNET

TELNET

- TELecommunications NETwork protocol
- Jedna z najstarszych aplikacji Internetu
  - Powstał w 1969 dla sieci ARPANET
- Standardowa aplikacja dostarczana wraz z implementacją TCP/IP (RFC 854)
- Łączy komputery z różnymi systemami operacyjnymi
- Pracuje w modelu klient-serwer

TELNET



TELNET

- Praca pomiędzy dowolnymi systemami operacyjnymi
- Terminal klienta → NVT → terminal serwera
- Konieczność negocjacji parametrów, m.in.:
  - echo,
  - typ terminala,
  - rozmiar okna.
- Wykorzystanie zestawu znaków NVT ASCII
- Wprowadzenie sygnalizacji (in-band) z użyciem bajta IAC = 0xff

Polecenia TELNET

- Każde polecenie poprzedzone bajtem 0xff – IAC (Interpret As Command)
- Wybrane polecenia:

EOF	236	Koniec pliku
SE	240	Koniec podopcji
SB	250	Początek podopcji
WILL	251	Negocjowanie opcji
WONT	252	
DO	253	
DONT	254	
IAC	255	Bajt danych 255

File Transfer Protocol

FTP

File Transfer Protocol

- FTP – protokół transmisji plików (RFC 959)
- Pozwala na kopiowanie pliku z jednego systemu na drugi
- Przeznaczony do pracy z różnymi systemami operacyjnymi
- Pracuje w modelu klient-serwer
- Pozwala przesyłać pliki, ale nie udostępnia ich
  - FTP to nie to samo co NFS lub „Udostępnianie Plików Windows”

FTP

Wykorzystuje dwa rodzaje połączeń:

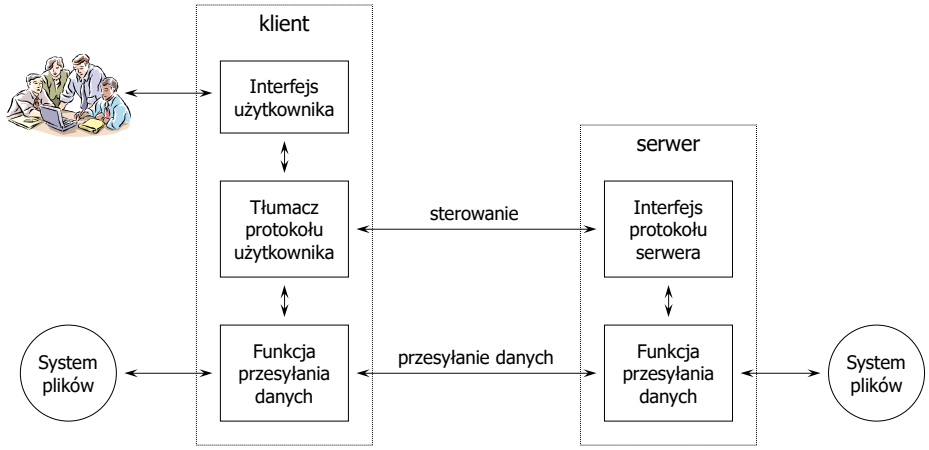
1. Połączenie sterujące
   - służy do przesyłania poleceń do/od klienta od/do serwera
   - typowe połączenie: pasywne otwarcie serwera (port TCP 21), aktywne otwarcie klienta
   - aktywne przez cały czas trwania sesji
   - ToS – „minimalizacja opóźnień”

FTP

Wykorzystuje dwa rodzaje połączeń:

2. Połączenie danych
   - służy do przesyłania danych (nie tylko plików) do/od klienta od/do serwera
   - tworzone za każdym razem, gdy potrzeba przesyłać dane (port TCP 20)
   - ToS – „maksymalizacja przepustowości”

Model klienta i serwera



Sposoby przesyłania

- Typ pliku
  - **ASCII**, EBCDIC, binarny, lokalny
- Format
  - **Niedrukowalny**, format Telnet, format Fortran
- Struktura
  - **Plik**, rekord, strona
- Typ przesyłania
  - **Strumieniowy**, blokowy, z kompresją

Sposoby przesyłania

- W praktyce stosuje się:
  - Typ: ASCII lub binarny
  - Format: tylko Niedrukowalny
  - Struktura: tylko Plik
  - Tryb przesyłania: tylko Strumieniowy
- Stąd tylko dwa sposoby przesyłania:
  - ASCII lub binarny

Polecenia FTP

- Format NVT ASCII
  - każde polecenie zakończone <CR><LF>
- Polecenia specjalne (Telnet)
  - przerwanie procesu <IAC, IP>
    - przerwanie wysyłania pliku
  - synchronizacja <IAC, DM>
    - wysłanie zapytania do serwera w trakcie przesyłania pliku
- Polecenia FTP
  - ciągi 3 lub 4 dużych liter
  - część z poleceń posiada argumenty
  - istnieje ponad 30 różnych poleceń

Polecenia FTP

- Najczęściej wykorzystywane polecenia

USER	Nazwa użytkownika na serwerze
PASS	Hasło użytkownika na serwerze
LIST	Wyświetla listę plików i katalogów
RETR	Pobranie pliku z serwera
STOR	Umieszczenie pliku na serwerze
TYPE	Typ przesyłanego pliku
ABOR	Przerywa polecenie FTP i transmisję danych
SYST	Odczytanie rodzaju systemu serwera
QUIT	Wylogowanie z serwera

Odpowiedzi FTP

- 3-cyfrowe numery, po których może występować opcjonalny komunikat
- Każda cyfra ma inne znaczenie

1xx	Wstępna odpowiedź pozytywna
2xx	Końcowa odpowiedź pozytywna
3xx	Pośrednia odpowiedź pozytywna
4xx	Wstępna odpowiedź negatywna
5xx	Stała odpowiedź negatywna

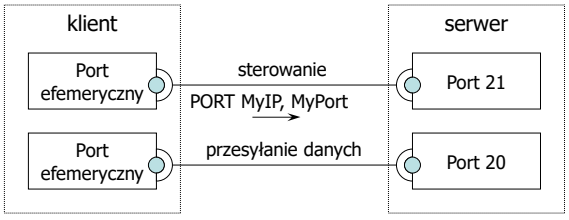
Odpowiedzi FTP

- Przykładowe odpowiedzi
  - 125 – połączenie danych otwarte; rozpoczęcie przesyłania
  - 200 – polecenie OK
  - 221 – zakończenie połączenia
  - 331 – nazwa użytkownika OK, wymagane hasło
  - 452 – błąd przy zapisie pliku
  - 500 – błąd składni; nierozpoznane polecenie
  - 501 – błąd składni; niewłaściwe argumenty

Połączenie dla danych

- Każde przesłanie pliku lub zawartości katalogu w osobnym połączeniu
- Zamknięcie połączenia oznacza koniec pliku
- Wykorzystanie polecenia PORT
- Aktywne otwarcie serwera na port efemeryczny klienta
- Aktywne zamknięcie serwera

Połączenie dla danych



Otwarcie pasywne

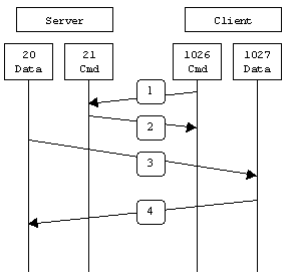
- Serwer nie może otworzyć połączenia do klienta, jeśli ten posiada adres prywatny (lub znajduje się za firewall'em)
- Rozwiązanie: tryb pasywny

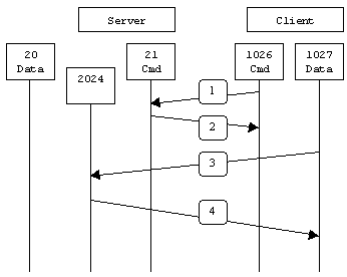
KLIENT **SERWER**

PASV →

← 227 Entering Passive Mode. 149,156,97,38,5,49

Otwarcie aktywne a pasywne





Zasoby FTP

- Korzystanie z FTP wymaga posiadania konta na serwerze
- Anonimowy FTP to sposób rozpowszechniania oprogramowania w Internecie
  - USER: anonymous
  - PASS: <adres e-mail>

Simple Mail Transfer Protocol

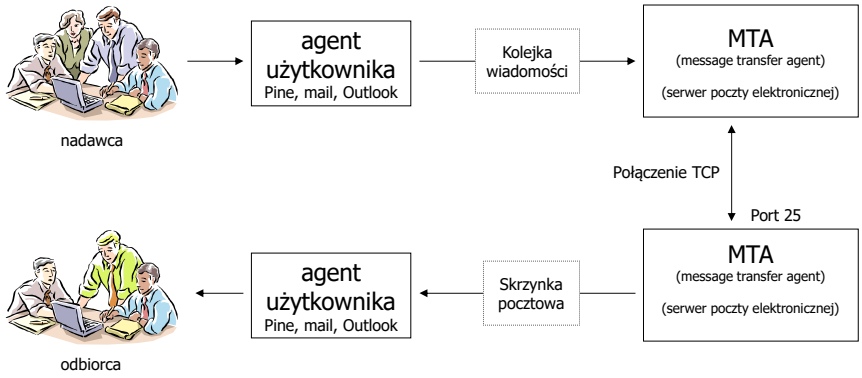
SMTP

Simple Mail Transfer Protocol

- Protokół do przesyłania wiadomości poczty elektronicznej
- Jedna z najbardziej popularnych aplikacji Internetu
- Protokół klient-serwer
- Korzysta z *dobrze znanego* portu 25 TCP

SMTP

- Przesyłanie poczty



SMTP

- Definiuje 8 podstawowych poleceń:
  - HELO – identyfikacja klienta (domena)
  - MAIL – identyfikacja nadawcy
  - RCPT – identyfikacja odbiorcy
  - DATA – zawartość wiadomości
  - QUIT – zamknięcie połączenia
  - RSET – przerwanie bieżącej transakcji

  często używane
  - VRFY – weryfikacja adresu odbiorcy
  - NOOP – polecenie testowe (odp. 200 OK)

  rzadko używane

MIME

- Multipurpose Internet Mail Extensions
- Dodanie nowych nagłówków do wiadomości:

```
Mime-Version:
Content-Type:
Content-Transfer-Encoding:
Content-ID:
Content-Description:
```

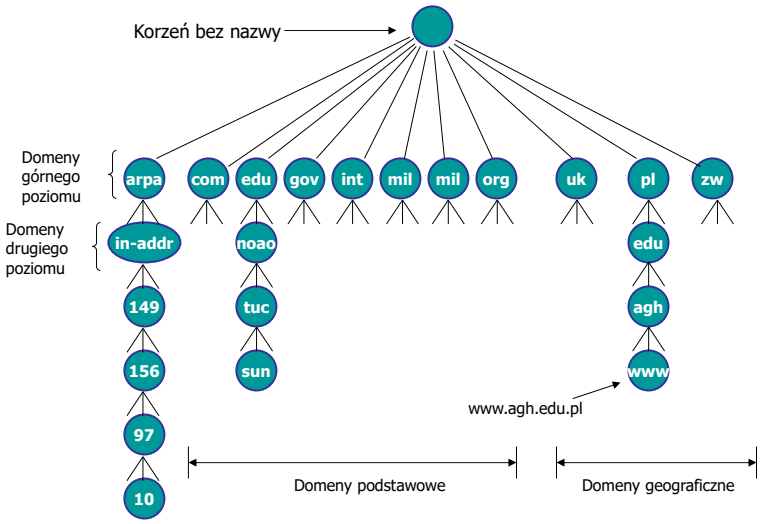
Domain Name System

System Nazw Domen

Definicja DNS

- DNS to rozproszona baza danych używana przez aplikacje TCP/IP do odwzorowywania nazw hostów na adresy IP i odwrotnie.
- Rozproszenie polega na tym, że żaden system nie posiada pełnej informacji o odwzorowaniu →
  informacja ta jest współdzielona pomiędzy niezależne serwery.

Hierarchiczna budowa nazw



Nazwy w DNS

www.cs.agh.edu.pl

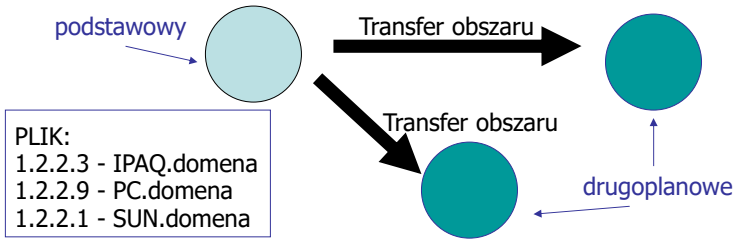
- Każdy węzeł drzewa może mieć etykietę do 63 znaków (A-Z,a-z,0-9,-). Wyjątkiem jest korzeń, który nie ma nazwy (ma nazwę pustą).
- Nazwy nie rozróżniają wielkości liter.
- Nazwa zakończona kropką jest nazywana absolutną nazwą domeny lub w pełni określoną nazwą domeny (ang. FQDN).
- Zakłada się, że nazwa bez końcowej kropki musi być uzupełniona:
  - jeśli składa się z dwóch lub więcej członów może być traktowana jako w pełni określona,
  - może ona też być uzupełniona przez dodanie nazwy zależnej od lokalizacji węzła.
- Można używać znaków narodowych.

Obszary i ich obsługa

- Obszar jest częścią drzewa DNS, która jest oddzielnie administrowana.
  - może być podzielony na mniejsze obszary — następuje wtedy delegacja odpowiedzialności.
- Delegowanie odpowiedzialności za zarządzanie etykietami sprawia, że rozwiązanie staje się skalowalne:
  - Nigdy pojedyncza jednostka nie zarządza wszystkimi etykietami w drzewie,
  - Odpowiedzialność jest delegowana w dół.

Podstawowy i drugoplanowy serwer nazw

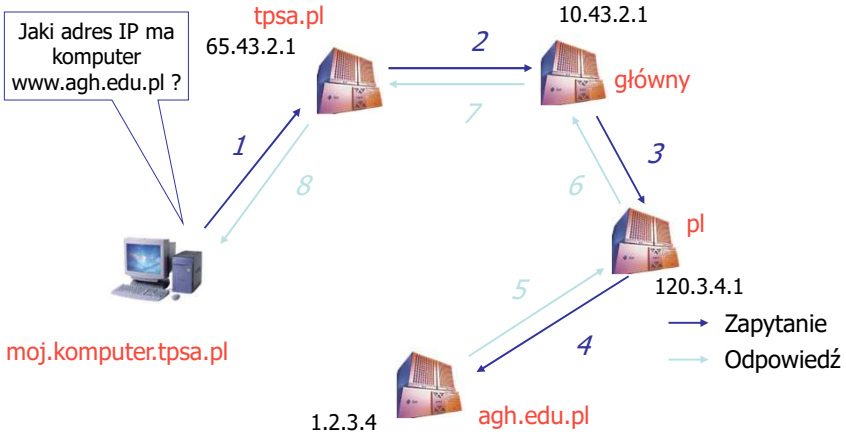
- Każdy obszar musi zawierać podstawowy serwer nazw i jeden lub więcej drugoplanowych serwerów nazw.
- Serwer drugoplanowy otrzymuje informacje poprzez transfer obszaru.



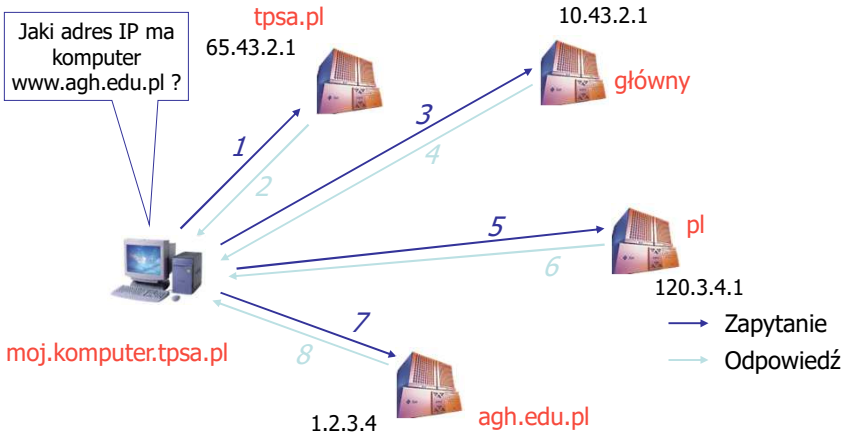
Zapytania

- Rozwiązywanie nazw polega na wysyłaniu zapytań i otrzymywaniu odpowiedzi
- Każdy serwer zna adresy hostów ze swojej domeny, adresy hostów do których delegował odpowiedzialność, adresy głównych serwerów nazw adres serwera domeny macierzystej itd... (`ftp.rs.internic.net/domain/named.root`),
- Rekurencyjne i iteracyjne

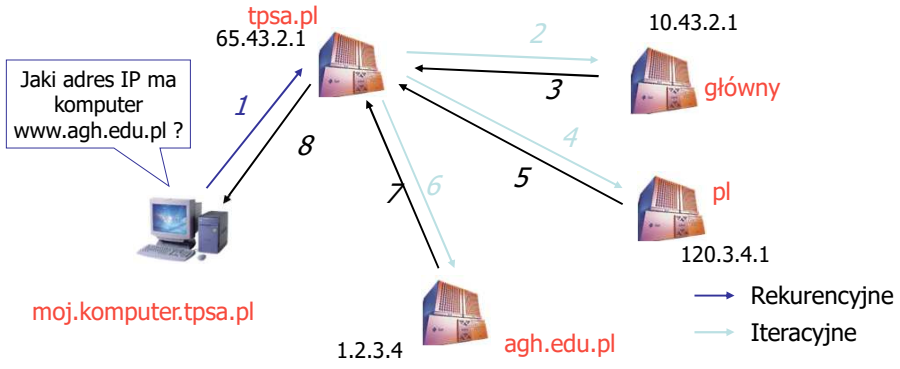
Zapytania rekurencyjne



Zapytania iteracyjne



Rzeczywistość — mieszane



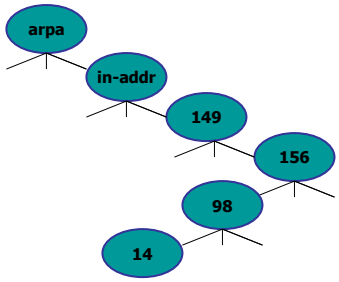
- Decyzję podejmuje oprogramowanie klienta
- Serwer jest konfigurowalny
  - Serwer główny nigdy nie jest rekurencyjny

O co pytamy?
Zapytania proste

- Najczęściej występujące
  - Jaki jest adres IP urządzenia o nazwie www.internic.net ?

O co pytamy?
Zapytania wskazujące

- Po podaniu adresu IP dostajemy nazwę DNS: 149.156.98.14 -> x.y.z?
- Wykorzystywana jest domena *in-addr.arpa.*
  - 149.156.98.14 -> 14.98.156.149.in-addr.arpa.



Bez oddzielnej gałęzi drzewa DNS należałoby przeszukiwać wszystkie domeny.

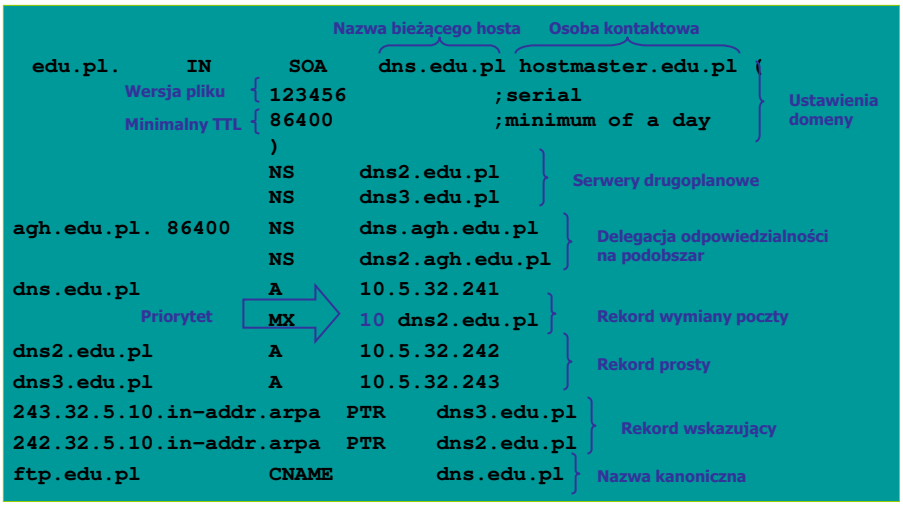
O co pytamy?
Inne typy

- MX — serwer poczty dla określonej nazwy
  - maciek@poczta.pl -> mail.poczta.pl
- NS — autorytatywny serwer DNS dla danej domeny
- CNAME — nazwa kanoniczna (alias)
- HINFO — informacje na temat komputera
- inne

Pamięć podręczna

- Wszystkie serwery DNS stosują pamięć podręczną aby zredukować wymianę komunikatów DNS i zwiększyć efektywność działania:
  - odpowiedzi autorytatywne pochodzą od serwerów z odpowiedniej domeny,
  - odpowiedzi z pamięci podręcznej nie są autorytatywne,
  - zawierają informacje na temat serwera od którego zostały uzyskane — klient ma możliwość osobistej kontroli,
  - wraz z odpowiedzią autorytatywną serwer otrzymuje TTL określający jak długo należy przechowywać daną informację w pamięci podręcznej.

Przykład konfiguracji serwera



UDP i TCP

- DNS obsługuje zarówno UDP jak i TCP:
  - Dobrze znany port UDP i TCP nr 53,
  - UDP stosowane jest najczęściej,
  - TCP wykorzystywane jest jeśli odpowiedź od serwera nazw przekracza 512 bajtów — jest to wielkość pakietu UDP jaki musi być w stanie odebrać każdy host,
  - jeśli odpowiedź UDP zawiera informację o tym, że ilość informacji została obcięta do wymaganych 512 bajtów to resolver ponawia zapytanie po TCP,
  - TCP jest używane do transmisji obszarów,
  - przy stosowaniu UDP programy muszą same obsługiwać czasy oczekiwania i retransmisje.

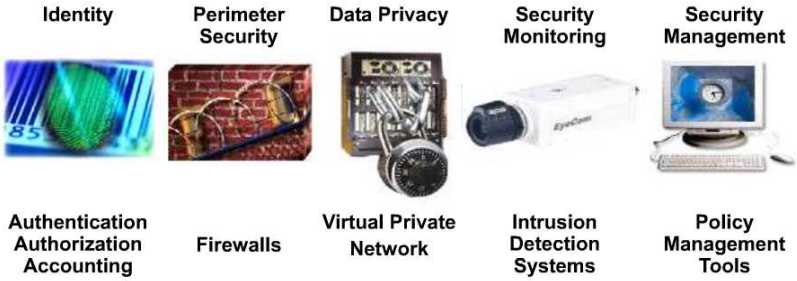
Podstawy bezpiecznych sieci komputerowych

Wprowadzenie

- Bezpieczeństwo to nie produkt
  - nie jest urządzeniem
  - nie jest oprogramowaniem
  - nie można go kupić
- Bezpieczeństwo to podejście
  - które musi nadążać za ewolucją sprzętu i oprogramowania
  - które powinno być stosowane dogłębnie
- Bezpieczeństwo to ciągła praca

Co jest przedmiotem ochrony?

- Poufność
- Integralność
- Dostępność



Rodzaje ataków

- Rodzaje ataków:
  - zależą od pomysłowości atakującego
  - zależą od warstwy, w którą wycelowany jest atak
- 4 klasy ataków
  - Rekonesans
  - Uzyskanie dostępu
  - *Denial of service* (DoS lub DDoS)
  - Robaki, wirusy, konie trojańskie

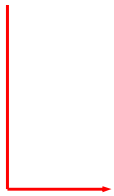
Przykłady

- przechwycenie haseł (słowniki, *brute-force*)
- *sniff*-owanie pakietów (telnet, ftp, smtp...)
- skanowanie portów, badanie poprzez *ping*
- *Man-in-the-middle*
- wirusy, robaki, konie trojańskie
- DoS or DDoS:
  - ping of death,
  - packet fragmentation
  - e-mail bomb,
  - SYN flood,
  - SMURF,
  - Stacheldraht ...

Nmap
Nessus ...

Co jest przedmiotem ochrony?

- Na uwadze trzeba mieć:
  - warstwę fizyczną
  - warstwę łącza danych
  - warstwę sieciową
  - warstwę transportową
  - warstwę aplikacji



efekt domina

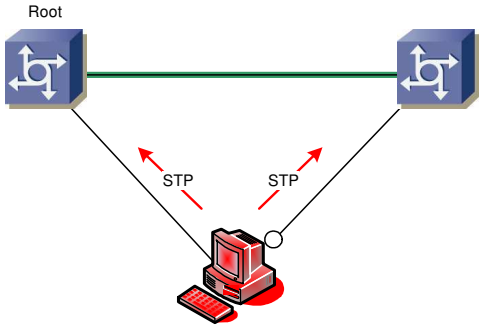
Zagrożenia fizyczne

- Sprzęt:
  - autoryzowany dostęp (konsole, porty hub-a)
    - zabezpieczenia: klucz, karta, karta/PIN, metody biometryczne
  - logowanie, kamery przemysłowe
- Środowisko:
  - temperatura, wilgotność, alarmowanie zmian
- Zasilanie:
  - UPS, zasilanie zapasowe, alarmowanie
- Utrzymanie:
  - dokumentacja, etykietowanie kabli, dostęp do konsol

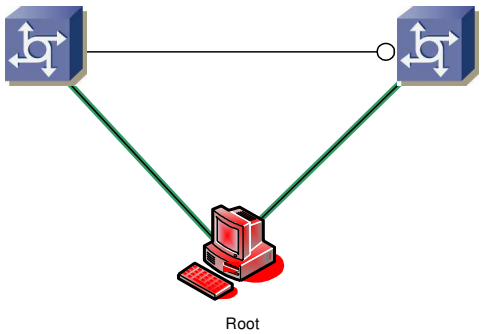
Warstwa łącza danych

- Ataki ograniczone do domeny rozgłoszeniowej
- Wycelowane w przełącznice:
  - wykorzystanie protokołu STP,
  - zalewanie tablic FDB (+ BPDU),
  - skakanie po VLAN-ach (podwójne tagowanie),
  - inne możliwości: CDP, VTP, ISL.
- Zabezpieczanie przełącznic
  - Port security
  - VMPS
  - Identity Based Network Services (IBNS) – 803.1x

Atak STP



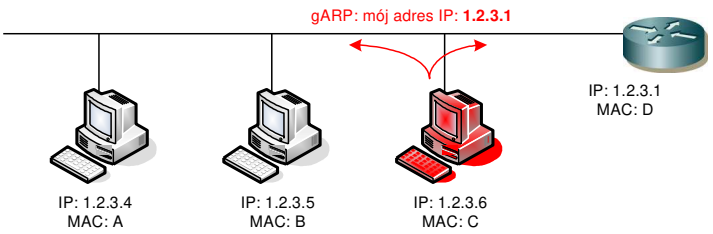
Atak STP



Warstwa sieciowa

- Atak może być prowadzony z wnętrza, jak i spoza granic sieci!
- Często opiera się na :
  - ARP Spoofing (*analogicznie - wykorzystanie protokołu DHCP*)
  - *IP Address Spoofing* - generowanie adresu źródłowego,
  - ICMP,
  - RIP, OSPF, BGP
- Może być wycelowany w:
  - host, router
  - protokół routingu

Atak przy użyciu gratuitous ARP



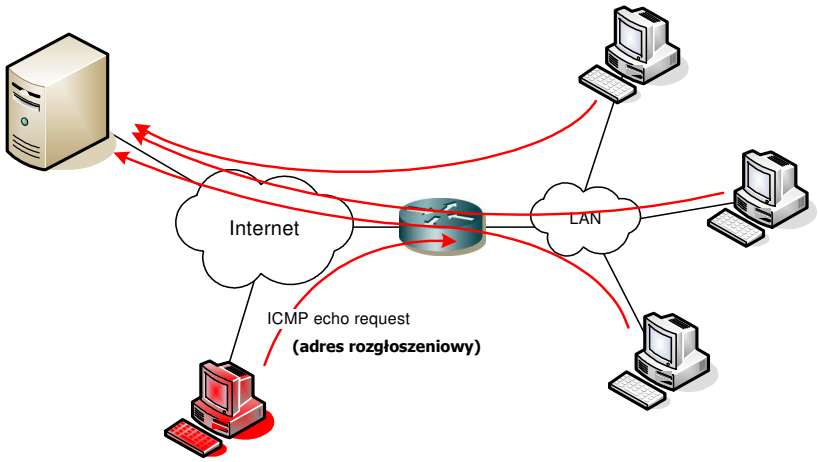
Warstwa transportowa

- Protokoły TCP i UDP
- Atak typu TCP SYN Flood
- Zapobieganie przez
  - TCP Interception
  - SYN Cookies

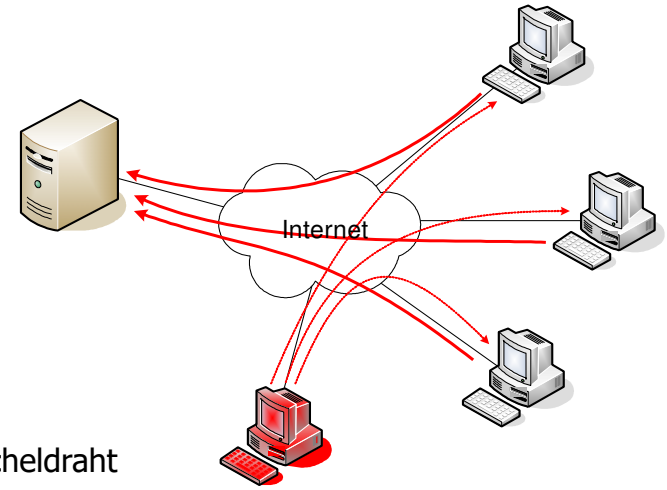
Denial-of-Service

- Powoduje zablokowanie wybranej usługi przez:
  - zużycie zasobów np.: przepustowości łącza, CPU,
  - zablokowanie przepływu informacji konfiguracyjnej np. pakietów routingu dynamicznego,
  - zablokowanie urządzenia sieciowego.
- Zwykle przez ręczny wpis adresu źródłowego
- Warianty:
  - atak typu *smurf*
  - atak rozproszony tzw. *DDoS*

Atak typu *smurf*



Distributed DoS



np.: Stacheldraht

Koń trojański

- Oprogramowanie „kuszące”, by je uruchomić
  - np. prosta gra komputerowa
- Zawiera złośliwy kod
- Działa poza świadomością użytkownika
  - Może niszczyć dane
  - Może tworzyć luki w bezpieczeństwie tzw. backdoor
- Wirusy (kod dołączony), robaki (samoklonujące)...

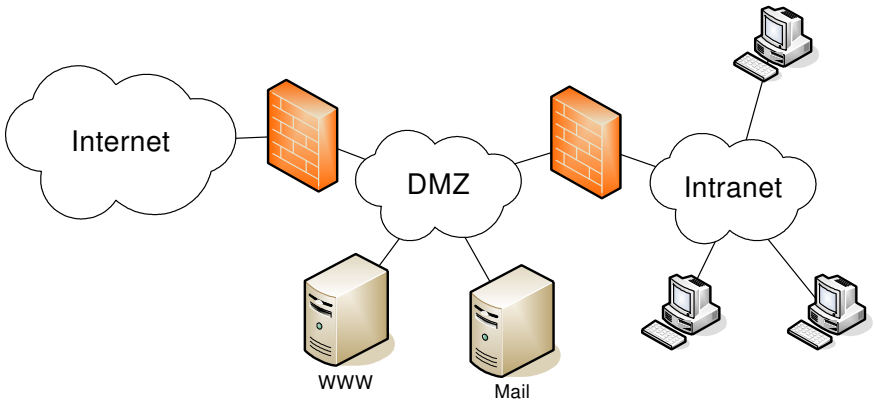
Metody zabezpieczeń

- szyfrowanie (klucz symetryczny/asymetryczny),
- SSH, HTTPS
- kontrola integralności (*MD5 hash*), certyfikaty,
- autoryzacja pakietów protokółów rutingu
- AAA – Autentykacja, Autoryzacja, Audyt
  - RADIUS, TACACS, Kerberos
- ACL – listy kontroli dostępu
- IDS – systemy wykrywania ataków - *sygnatury*
- TCP interception – kontrola połączeń TCP

Podział sieci

- Wprowadzenie podsieci
  - możliwość umieszczenia firewalla pomiędzy podsieciami
- Sieci tego samego typu należy budować na tego samego typu switch'ach
  - sieci zaufane łączymy na „zaufanym" switchu,
  - sieci strefy DMZ łączymy na switchu DMZ,
  - sieci niezaufane na „niezaufanym" switchu.
- Wydzielenie specjalnej podsieci do zarządzania i administracji

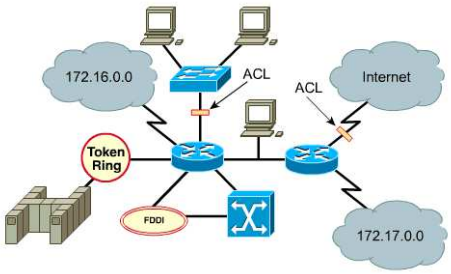
Strefy w sieci



Access Control Lists

listy kontroli dostępu

Listy kontroli dostępu (ACL)



- w celu uniemożliwienia niektórych rodzajów transmisji (przepuszczanie/blokowanie)
- zestaw reguł
- definiowany dla KAŻDEGO rutowanego protokołu
- do/z określonego portu rutera

ACL - zastosowania

- filtr bezpieczeństwa
  - zapewnia bezpieczeństwo sieci, do/z której kontroluje ruch
  - ochrona przed niepowołanym dostępem
- filtr ruchu
  - zapobiega niepotrzebnemu ruchowi w łączach o ograniczonej przepustowości (zmniejszenie obciążenia)
- identyfikacja pakietów
  - narzędzie wyboru pakietów dla innych mechanizmów ruterów Cisco (dialer list, route maps itp)

ACL - zastosowania

- przykłady:
  - stworzenie ‘strefy zdemilitaryzowanej’
  - udostępnienie tylko usługi HTTP na zewnątrz danej sieci
  - zablokowanie możliwości konfiguracji routera poprzez telnet
  - „droga jednokierunkowa”
  - ...

ACL – różne rodzaje

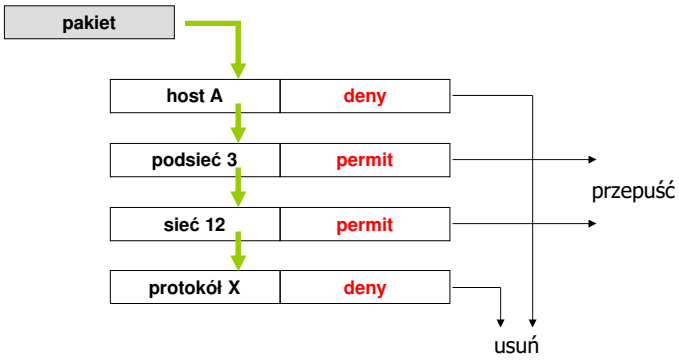
- ACL standardowy
- ACL rozszerzony
- Reflexive ACL
  - CBAC - Context-based Access Control
- Czasowe ACL
- Lock-and-Key ACL
- Turbo ACL

ACL – podstawy

- Listy kontroli dostępu składają się z sekwencji warunków (filtrów) określających sposób postępowania z danym pakietem
- Kryterium mogą być:
  - adres źródłowy/docelowy,
  - protokół warstwy wyższej,
  - numer portu,
  - ustawienie flagi...
- Akcje są dwojakiego rodzaju:
  - permit – przepuszcza pakiet,
  - deny – gubi pakiet.

ACL – działanie listy

- Pakiet ‘przechodzi’ sekwencyjnie przez listę dostępu

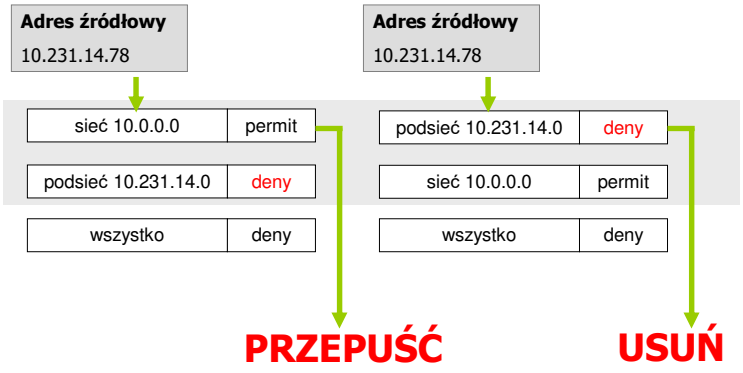


Zasady funkcjonowania ACL

- lista sprawdzana jest do momentu napotkania pierwszego warunku dotyczącego danego pakietu
- polecenie permit any na początku listy powoduje, że dalej nie jest sprawdzana
- jeśli lista zostanie sprawdzona do końca, a mimo to odpowiedni dla danego pakietu warunek nie zostanie znaleziony, pakiet jest gubiony - na końcu listy znajduje się domyślne (implicit) polecenie deny any

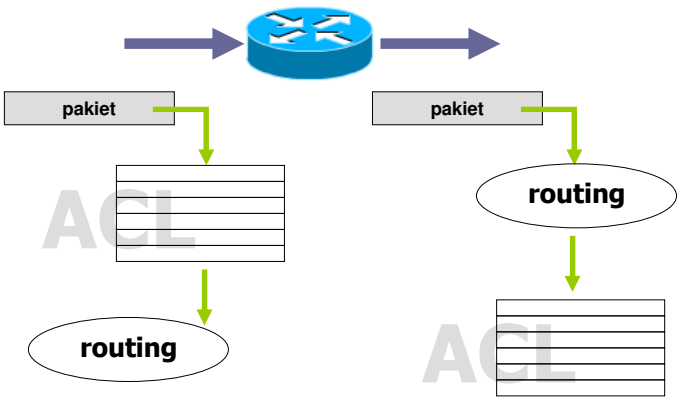
Kolejność warunków

- Ze względu na sekwencyjne przetwarzanie listy fundamentalne znaczenie ma kolejność warunków



ACL – ruch wchodzący i wychodzący

- Ruch wchodzący - **in** Ruch wychodzący - **out**



KONIEC