Serwer WWW

Instalacja i konfiguracja serwera WWW w Linux.

Konfigurowanie wirtualnych serwerów WWW w Linux.
Blokowanie przeglądania zawartości katalogu.

Serwer WWW

- Wprowadzenie do serwera WWW
- Instalacja serwera WWW
- Konfiguracja serwera
- Moduły serwera WWW
- Konfiguracja autentykacji
- Konfiguracja PHP
- Serwery wirtualne
- Certyfikaty SSL (HTTPS)

Wprowadzenie do serwera WWW

Apache ma ponad 50 procent udziału w rynku.

Zalety:

jest stabilny

kilka największych stron takich jak amazon czy IBM go używają

całe oprogramowanie i dodatkowe elementy są na licencji open source

pracuje na wielu platformach (Unix, Linux, Windows)

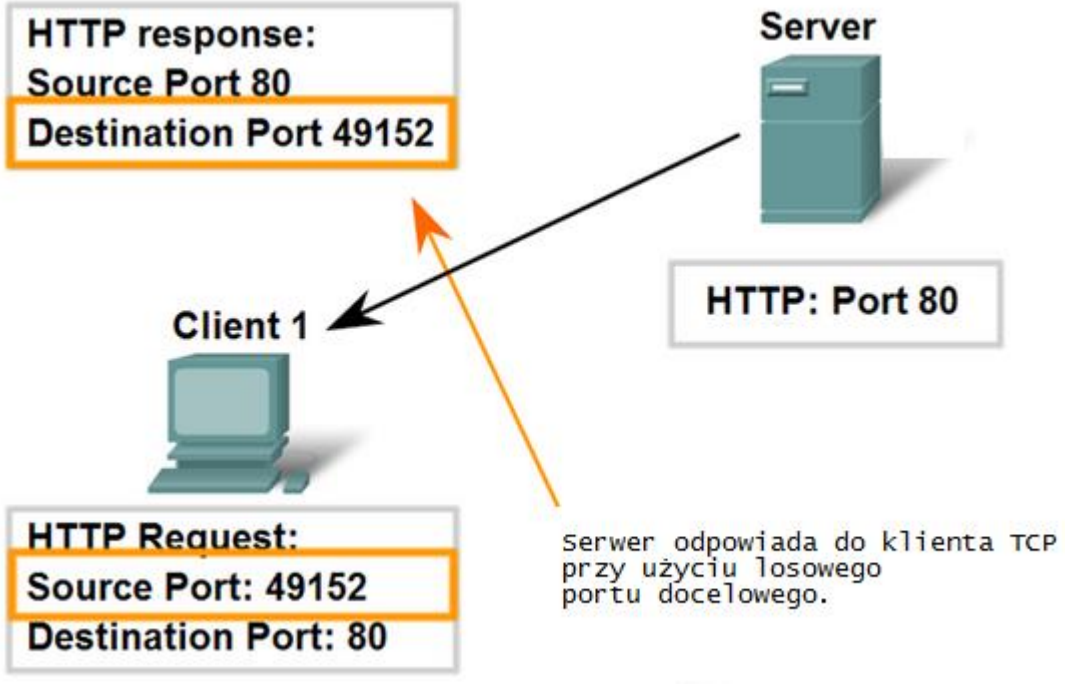
jest elastyczny

jest bezpieczny

Wprowadzenie do serwera WWW

- Apache HTTP Web Server
- Cechy serwera
  - wielowątkowość
  - skalowalność
  - bezpieczeństwo
  - kontrola dostępu/uwierzytelnianie: mod_authz_host
- Pakiety `apache2 apache2-doc apache2-utils`
- Demon /usr/sbin/apache2
- Skrypt /etc/init.d/apache2
- konfiguracja /etc/apache2/*, /var/www/*
- Porty 80 (http), 443 (https)

Wprowadzenie do serwera WWW



Wprowadzenie do serwera WWW

- Nagłówek klienta

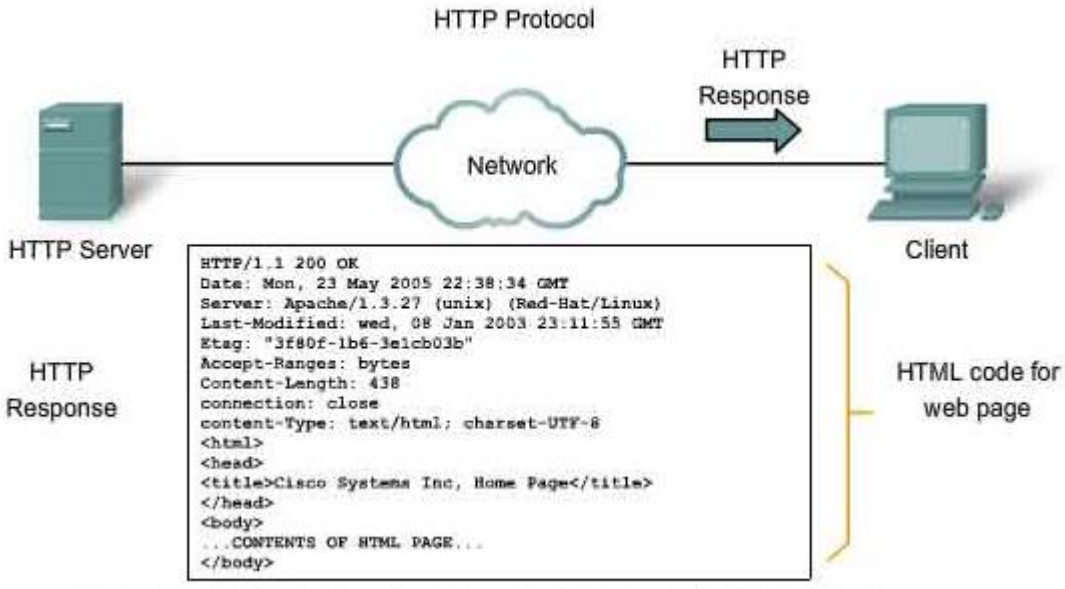
```
GET / HTTP/1.1
Connection: Keep-Alive
User-Agent: Mozilla/5.0 (X11; U; Linux i686)
Host: localhost:80
Accept: text/xml, image/gif, image/jpeg,
image/png...
Accept-Encoding: gzip,deflate
Accept-Language: en-us
Accept-Charset: iso-8859-1,*,utf-8
```

- Nagłówek serwera

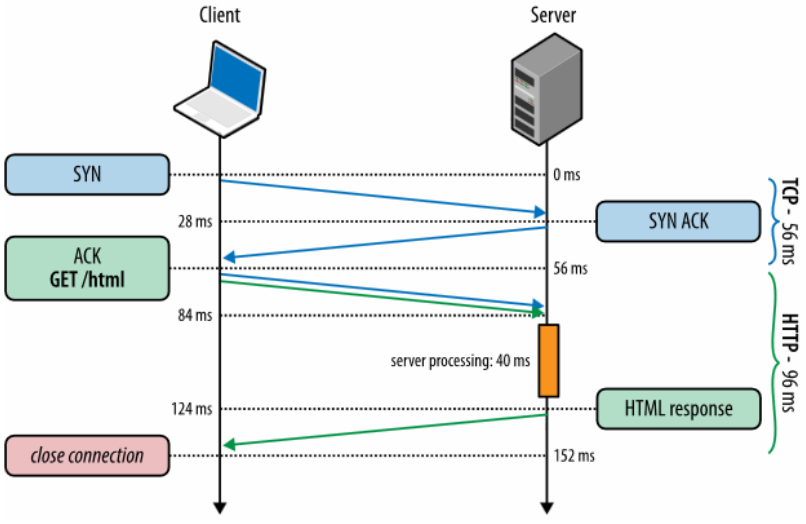
```
HTTP/1.1 200 OK
Date: Thu, 02 Jun 2009 14:03:31 GMT
Server: Apache/2.0.52 (Fedora)
Last-Modified: Thu, 02 Jun 2009 11:41:32 GMT
ETag: "3f04-1f-b80bf300"
Accept-Ranges: bytes
Content-Length: 31
Connection: close
Content-Type: text/html; charset=UTF-8
```

- Pusta linia – nagłówek kompletny
- Transmisja danych

Wprowadzenie do serwera WWW



In response to the request, the HTTP server returns code for a web page.



Instalacja serwera WWW

- Instalacja Apache 2

```
apt-get install apache2 apache2-doc apache2-utils
```

- Instalacja ze źródeł

wget http://www.apache.org/dist/httpd/httpd-2.4.23.tar.gz

gunzip –c httpd-2.4.23.tar.gz |tar xvzf –

./configure --prefix=/usr/local/httpd

make

make install

Konfiguracja serwera

Podstawowa konfiguracja w pliku
/etc/apache2/

```
#       |-- apache2.conf  - podstawowa konfiguracja
#       |       `--  ports.conf – konfiguracja portów 80(http) 443(https)
#       |-- mods-enabled – konfiguracja modułów wywoływanych przez
symlinki
#       |       |-- *.load
#       |       `-- *.conf
#       |-- conf-enabled
#       |       `-- *.conf
#       `-- sites-enabled
#               `-- *.conf
```

Konfiguracja serwera

- W dystrybucji Debian GNU / Linux zmieniono nazwę oprogramowania z httpd na apache2.

- W systemie Debian znajdziesz plik konfiguracyjny o nazwie apache2.conf w katalogu o nazwie /etc/apache2

- W systemie Ubuntu/Debian Apache przetwarza wszystkie pliki w /etc/apache2/sites-enabled/ (które powinny być dowiązaniami symbolicznymi do plików w katalogach sites-available, zarządzanych przez programy a2ensite i a2dissite)

Konfiguracja serwera

Uruchomienie, zatrzymanie, status, przeładowanie serwera

- polecenie init.d, w terminalu, do zarządzania serwerem Apache:

    /etc/init.d/apache2 (start/stop/status/reload)

- polecenie apache2ctl, w terminalu, do zarządzania serwerem Apache:

    apache2ctl (start/stop/status/reload)

Konfiguracja serwera

Polecenia systemctl (start/stop/status/reload), w terminalu, do zarządzania serwerem Apache:

- uruchom serwer Apache: systemctl start apache2
- zatrzymaj serwer Apache: systemctl stop apache2
- zatrzymaj, a następnie uruchom serwer Apache: systemctl reload apache2
- ponownie załaduj serwer Apache w celu zaktualizowania nowych konfiguracji: systemctl reload apache2
- uruchom serwer Apache podczas rozruchu: systemctl enable apache2
- wyłącz serwer Apache podczas uruchamiania: systemctl disable apache2

Konfiguracja serwera

- Podstawowa konfiguracja w pliku
  - /etc/apache2/apache2.conf
    - parametry web serwera
    - virtual hosts
    - definicje dostępu
    - mime-types
- Konfiguracja modułów

/etc/apache2/mods-enabled

/etc/apache2/sites-available

- włączenie strony a2ensite
- wyłączenie strony a2dissite

Konfiguracja serwera

conf-available to katalog zawierający dodatkowe lokalne pliki konfiguracyjne oraz pliki innych aplikacji, które nie są jeszcze powiązane z żadnymi modułami. Konfiguracje w tym katalogu nie są aktywne, chyba że je włączysz. Polecenie a2enconf może włączyć plik konfiguracyjny.

conf-enabled to katalog zawierający dodatkowe konfiguracje lokalne i pliki konfiguracyjne innych aplikacji powiązanych z odpowiadającymi im modułami. Pliki konfiguracyjne w tym katalogu są aktywne i wpływają na zachowanie Apache. Możesz wyłączyć plik konfiguracyjny za pomocą polecenia a2disconf.

mods-available to katalog zawierający pliki konfiguracyjne do ładowania modułów i ich konfigurowania. Jednak nie są one jeszcze włączone. Możesz je włączyć za pomocą polecenia a2enmod, np.:
a2enmod rewrite

mods-enabled to katalog zawierający pliki konfiguracyjne do ładowania modułów i ich konfigurowania. Konfiguracje w tym katalogu są już włączone i prowadzą do odpowiednich modułów. Możesz wyłączyć moduły za pomocą polecenia a2dismod, np.:

a2dismod rewrite

Testowanie nowej konfiguracji Apache apache2ctl configtest.

Konfiguracja serwera

Strona domowa

- /var/www/

Domyślna konfiguracja Apache2 odwołuje się do położenia (w przypadku Ubuntu)

- /var/www/html

Konfiguracja serwera

- Opcje konfiguracyjne
  - Server Root *directory-path* – katalog konfiguracyjny /etc/apache2
  - Listen [*IP-address*:] *portnumber* – porty
  - ServerName *fully-qualified-domain-name*[: *port*] – identyfikacja samego siebie
  - ServerAdmin *e-mail_address* – mail
  - MaxClient - maksymalna ilość jednoczesnych połączeń
  - LoadModule - używane do dodawania i korzystania z dodatkowych modułów
  - User - określa ID użytkownika z którego uprawnieniami będzie pracował web serwer
  - Group - określa grupę dla procesu Apache
  - Alias - dyrektywa pozwalająca korzystać z zawartości dokumentów trzymanych w innych lokalizacjach niż DocumentRoot
  - ScripAlias - określa docelowy katalog skryptów CGI

Moduły serwera WWW

- Większość konfiguracji plik apache2.conf
- Inne dyrektywy zawarte w modułach
- Ilość modułów zależna od wersji
- Kompilacja wszystkich modułów

  ./configure -enable-module=all
- Instalacja modułów

  apt install [module-name]

Moduły serwera WWW

mod_cgi – pozwala na wykonywanie skryptów CGI na web serwerze

mod_perl – włącza interpreter Perl w web serwerze

mod_aspdotnet – dostarcza interfejs ASP.NET dla hostów Microsoft obsługujących silnik (engine) ASP.NET

mod_autz_ldap – pozwala na autentykację do serwera Apache na podstawie LDAP

mod_ssl - pozwala na szyfrowanie za pomocą SSL oraz TSL

Konfiguracja autentykacji

- Start z uprawnieniami root
- Praca na koncie innego użytkownika
- Kontrola na poziomie plików i katalogów
- Dostęp kontrolowany opcje allow, order lub deny
  - allow,deny – pozwala na dostęp określonym klientom wszystkim innym zabrania
  - deny, allow – zabrania na dostęp określonym klientom i zezwala wszystkim innym.
- .htaccess - plik zawierający dane autentykacyjne użytkowników

Konfiguracja autentykacji

- .htaccess
  - uwzględniony gdy istnieje wpis AllowOverride w Directory:

    <Directory />
    Options None
    AllowOverride None
    </Directory >
- AllowOveride może mieć następujące opcje:
  - AuthConfig – można zmieniać opcje dotyczące autoryzacji dostępu do katalogu,
  - Limit – kontrola dostępu na podstawie IP (Deny, Allow, Order),
  - Options – możliwość używania Options,
  - All – wszystkie z powyższych,
  - None – żadne z powyższych.

Konfiguracja autentykacji

Przykład 1 dla zawartości pliku .htaccess:

AuthName "Podaj hasło"

Access allow all valid-user

Udziela dostępu wszystkim użytkownikom z dowolnego adresu IP, po poprawnej autoryzacji za pomocą nazwy użytkownika i hasła, (ustawienia ukrytego katalogu).

Przykład 2 dla zawartości pliku .htaccess:

AuthName "Strefa chroniona"

Access allow all users janek zosia

Udziela dostępu z dowolnego adresu IP użytkownikom “janek” oraz “zosia”.

Konfiguracja autentykacji

Przykład 3 dla zawartości pliku .htaccess:

AuthName "Identyfikacja"

Access allow 212.85.112.3

Udziela dostępu użytkownikowi łączącemu się z konkretnego IP (212.85.112.3), odmawiając innym tego dostępu.

Przykład 4 dla zawartości pliku .htaccess:

AuthName "X-Files"

Access allow all groups wtajemniczeni

Udziela dostępu łączącemu się z dowolnego IP i jednocześnie autoryzującego się loginem należącym do grupy “wtajemniczeni”. W tym przypadku nazwa użytkownika oraz hasło muszą być dodane do grupy „wstajemniczeni”, aby móc poprawnie przejść proces autoryzacji.

Konfiguracja autentykacji

- Przykład kontrolowania dostępu z użyciem hasła

```
# .htaccess
AuthType Basic                                  # lub Digest
AuthName "Secret"
AuthUserFile password_file
Require user dummy spamer       # lub "valid -user „
```

- Plik z hasłami password_file
  - format – użytkownik:hasło
  - hasło szyfrowane MD5 lub crypt
  - generowanie lub zmiana hasła htpasswd2

Konfiguracja PHP

- Instalacja PHP i modułu Apache PHP

apt -y install php php-cgi libapache2-mod-php php-common php-pear php-mbstring

php-cgi - zawiera interpreter CGI /usr/lib/cgi-bin/php zbudowany do użytku w Apache2 z mod_actions lub dowolnym innym httpd CGI obsługującym podobny mechanizm.

libapache2-mod-php - zawiera moduł PHP dla serwera WWW Apache 2.

php-common - zawiera wspólne narzędzia wspólne dla wszystkich spakowanych wersji PHP.

php-pear - PEAR to framework i system dystrybucji komponentów PHP wielokrotnego użytku.

php-mbstring - funkcja wyrażeń regularnych z obsługą znaków wielobajtowych.

Konfiguracja PHP

mod_perl - pozwala na wykorzystanie interpretera Perl poprzez uruchomiony proces httpd w celu redukcji tworzenia procesów.

mod_php - przyspiesza parsowanie stron zawierających kod PHP.

mod_speling – naprawia błędy ortograficzne i literówki w URL, które mógł wprowadzić użytkownik

- Restart serwera Apache service apache2 restart lub systemctl restart apache2
- Przeładowanie nowej konfiguracji Apache2 systemctl reload apache2
- Test PHP

  vi /var/www/html/info.php

```
<?php
        phpinfo();
?>
```

  http://192.168.0.100/info.php

Serwery wirtualne

Pliki konfiguracyjne vhost znajdują się w położeniu

- /etc/apache2/sites-available/ (tu znajdują się pliki konfiguracyjne dostępnych vhostów ale jeszcze nieaktywnych)
- /etc/apache2/sites-enabled/ (tu znajdują się pliki już aktywnych vhostów)

Serwery wirtualne konfiguracja serwera

ports-conf – Jest to plik określający porty dostępne dla hostów wirtualnych i porty TCP, których nasłuchuje Apache.

sites-available – Jest to katalog zawierający dostępne pliki konfiguracyjne dla wirtualnych hostów Apache. Wirtualne hosty pozwalają Apache obsługiwać różne strony internetowe. Pliki w tym katalogu nie są jeszcze aktywne. Możesz włączyć wirtualny plik hosta za pomocą polecenia a2ensite, np.: *a2ensite example.com.conf*

sites-enabled – Jest to katalog zawierający aktywowane pliki konfiguracyjne wirtualnego hosta. Zwykle zawiera dowiązania symboliczne do plików w katalogu witryn dostępnych. Możesz wyłączyć wirtualny plik hosta za pomocą polecenia a2dissite, np.: *a2dissite example.com.conf*

Serwery wirtualne

Apache udostępnia wiele opcji wirtualnego hostingu

/etc/apache2/sites-available/przykładowy_plik.conf

Przykładowy plik wirtualnego hosta.
Zalecane jest używanie raczej IP niż nazwy (adres IP nie musi być podany)

<VirtualHost 192.168.0.1>

ServerAdmin webmaster@another-example.org

DocumentRoot /www/docs/another-example.org

ServerName www.another-example.org

ErrorLog *${APACHE_LOG_DIR}*/another-example.org.error_log
CustomLog *${APACHE_LOG_DIR}*/another-example.org.*access_log*
</VirtualHost>

Apache serwery wirtualne – przykłady

```
<VirtualHost firma.com.pl>
ServerAdmin administartork@firma.com.pl
DocumentRoot /usr/Apache2/httdocs
SerwerName firma.com.pl
(…) # tu inne parametry
ErrorLog ${APACHE_LOG_DIR}/error_log
CustomLog ${APACHE_LOG_DIR}/access_log
</VirtualHost>

<VirtualHost biuro. firma.com.pl>
ServerAdmin administartor@firma.com.pl
DocumentRoot /usr/www/biuro
SerwerName biuro.firma.com.pl
(…) # tu inne parametry
ErrorLog ${APACHE_LOG_DIR}/biuro.error_log
CustomLog ${APACHE_LOG_DIR}/biuro.access_log
</VirtualHost>
```

Serwery Wirtualne można tworzyć na podstawie adresów IP czyli do jednego urządzenia sieciowego przypisanych jest kilka adresów IP z odpowiadającym innym adresem domenowym lub poprzez identyfikowanie serwerów różnymi adresami URL związanymi z tym samym adresem IP. Rozwiązanie wykorzystujące identyfikacje serwerów poprzez nazwy domenowe jest najbardziej polecane i bazuje ono na protokole HTTP/1.1. Rozwiązanie to wykorzystuje jeden adres IP, a rozróżnianie polega na zdefiniowaniu rożnych nazw domeny np. forma.com.pl oraz biuro.firma.com.pl oczywiście nazwy te muszą być zarejestrowane w systemie DNS.

Apache serwery wirtualne – przykłady

<VirtualHost 83.210.159.2**>**
ServerAdmin administartork@firma.com.pl
DocumentRoot /usr/Apache2/httdocs
SerwerName firma.com.pl
ErrorLog *${APACHE_LOG_DIR}*/error_log
***Custom*Log** *${APACHE_LOG_DIR}*/access_log
</VirtualHost>
<VirtualHost 83.210.159.3**>**
ServerAdmin administartor@firma.com.pl
DocumentRoot /usr/www/biuro
SerwerName biuro.firma.com.pl
ErrorLog *${APACHE_LOG_DIR}/* biuro.error_log
***Custom*Log** *${APACHE_LOG_DIR}/* biuro.access_log
</VirtualHost>

Identyfikacja serwerów wirtualnych za pomocą adresów IP jest w Internecie ciągłe dość popularna mimo iż dostępny zakres adresów drastycznie zmalał.
W protokole HTTP/1.0 identyfikacja stron możliwa była jedynie w oparciu o numery IP. Dopiero protokół HTTP/1.1 wprowadził metodę umożliwiającą przesłanie przez przeglądarkę nazwy domenowej serwera. Wielu administratorów korzysta z adresu IP ze względu na to, że starsze przeglądarki nie będą udostępniały możliwości dostępu do serwera Apache

Apache serwery wirtualne – przykłady

Listen 8080
<VirtualHost 83.210.159.2:80**>**
ServerAdmin administartork@firma.com.pl
DocumentRoot /usr/Apache2/httdocs
SerwerName firma.com.pl
ErrorLog *${APACHE_LOG_DIR}/*error_log
***Custom*Log** *${APACHE_LOG_DIR}/*access_log
</VirtualHost>
<VirtualHost 83.210.159.2:8080**>**
ServerAdmin administartor@firma.com.pl
DocumentRoot /usr/www/biuro
SerwerName biuro.firma.com.pl
ErrorLog *${APACHE_LOG_DIR}/*biuro.error_log
***Custom*Log** *${APACHE_LOG_DIR}/*biuro.access_log
</VirtualHost>

Ostatnią metodą konfiguracji serwerów wirtualnych jest identyfikacja oparta na numerach portów. Zaletą tej techniki jest uruchomienie większej ilości serwerów wirtualnych z użyciem jednego adresu IP i pojedynczej nazwy domenowej.
Ze względu na to, że metoda jest oparta na metodzie opierającej się na adresach IP do obsługi nie jest potrzebny protokół HTTP/1.1.
Jeżeli będziemy chcieli obejrzeć stronę biuro.firma.pl, to w przeglądarce musimy podać adres: firma.pl:8080.
Z tego powodu ta metoda nie jest zbyt popularna wśród użytkowników Internetu.

Apache serwery wirtualne – przykłady

<VirtualHost 83.210.159.2**>**
ServerAdmin administartork@firma.com.pl
DocumentRoot /usr/Apache2/httdocs
SerwerName firma.com.pl
ServerAlias www.firma.pl
ErrorLog *${APACHE_LOG_DIR}*/error_log
***Custom*Log** *${APACHE_LOG_DIR}*/access_log
</VirtualHost>

Czasami istnieje potrzeba, by dany serwer wirtualny był widoczny pod wieloma nazwami ze względu na przyzwyczajenia użytkowników podczas wpisywania adresów w przeglądarce. Najczęstszym przypadkiem jest wpisywanie adresu bez znaków WWW. Rozwiązanie tego problemu stanowi dyrektywa ServerAlias, która umożliwia zdefiniowanie aliasów identyfikujących serwer wirtualny.

Certyfikaty SSL (HTTPS)

Secure Socket Layer (SSL) port 443

- SSL ochrona komunikacji między przeglądarką a web-serwerem
- Wymaga stworzenia i podpisania certyfikatów (CSR)
- Integralność certyfikatu jest potwierdzana przez urząd Certificate Authority (CA)
- Własny podpis będzie uznawany ale przeglądarka będzie informować o błędzie

Jak działa SSL

Certyfikaty posiada klucz publiczny i prywatny

- Klucz publiczny – szyfrowanie informacji
- Klucz prywatny – deszyfrowanie
- Klucz prywatny powinien być tajny

Rola Certificate Authority

- Ograniczona liczba instytucji poświadczająca certyfikaty
- Przeglądarki posiadają klucze publiczne
- Walidacja informacji zaszyfrowanej kluczem publicznym
- Przeglądarka deszyfruje i sprawdza czy certyfikat jest podpisany przez CA

KONIEC