Elektroniczne płatności z wykorzystaniem bezpiecznego WWW

mgr inż. Krzysztof Szczypiorski

e-mail: K.Szczypiorski@tele.pw.edu.pl

Instytut Telekomunikacji Politechniki Warszawskiej

ENIGMA'98 - II Krajowa Konferencja Zastosowań Kryptografii

Tutorial III - Protokoły elektronicznych płatności

Warszawa, 26-28 maja 1998 r.

Plan prezentacji

zał. analiza środowiska WWW i istniejńcych w nim zabezpieczeń pod kńtem zastosowań do elektronicznych płatności

- Wprowadzenie
  - elektroniczny handel - dziedzina
  - czym jest WWW?
  - zwińzek z protokołami rodziny TCP/IP, URL
- Architektura WWW
  - MIME, HTTP, wymiana pomińdzy klientem a serwerem
  - architektura usługowa
- Zagrońenia
- Bezpieczeństwo
  - transportu (S-HTTP, SSL, PCT)
  - po stronie serwera (konfiguracja, umiejscowienie serwera)
  - po stronie klienta (Java, JavaScript, ActiveX)

Elektroniczny handel z wykorzystaniem WWW - dziedzina cz.1/2

Klient

Usługodawca

Bank klienta

Bank usługodawcy

Sień międzybankowa

Elektroniczny handel z wykorzystaniem WWW - dziedzina cz.2/2

Klient

Bank klienta

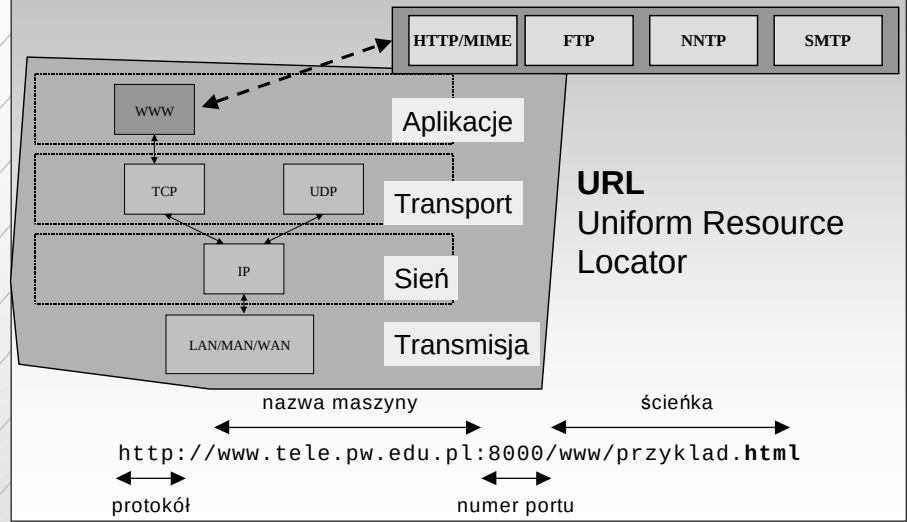
Home-banking, office-banking, direct-banking

Czym jest WWW?

- **WWW, W3 - World Wide Web - World Wide Wait**
- reakcja na frustracje zwińzane:
  - z ograniczeniami Internetu
  - i z jego niekontrolowanym rozwojem (liczba dokumentów i ich rozmieszczenie)
- w przeszłości kańdy protokół (FTP, Gopher, NNTP, WAIS, Telnet, SMTP) wymagał:
  - innego oprogramowania do obsługi (klienta)
  - rónił siń reprezentacjń przechowywanych plików
  - wymagał zaangańowania ludzi
- pomysł - unifikacja poprzez WWW (nowy protokół HTTP)
- najwańniejsze fakty z 9. letniej historii

1989 Tim Bernes-Lee z grupń fizyków z European Laboratory for Particle Physics (CERN) proponuje stworzenie nowego systemu
1990 powstaje nazwa WWW dla pierwszej implementacji na maszynach NeXT; HTTP i HTML
1990 grudzień - pierwsze wersjń oprogramowania dostńpne poza CERN
1992 styczeń - pierwsza tekstowa przeglńdarka z CERN rozumiejńca HTML 2.0

WWW a protokoły TCP/IP; URL



MIME

Multipurpose Internet Mail Extensions

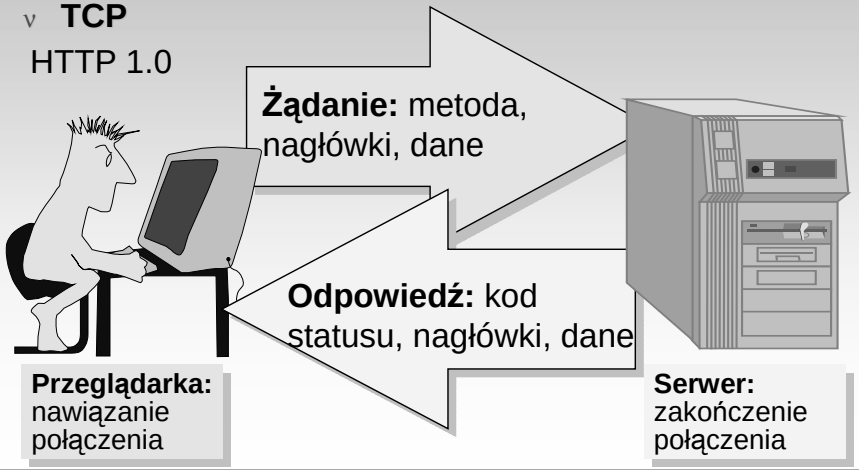
- typ pliku rozpoznawany po końcówce
- x dla eksperymentalnych typów MIME

```
application/msword            message/news
application/news-message-id   message/rfc822
application/postscript
application/x-tex             multipart/alternative
application/zip               multipart/mixed

audio/basic                   text/html
audio/x-wav                   text/plain

image/gif                     video/mpeg
image/jpeg                    video/x-msvideo
```


Idea działania HTTP

- **HyperText Transfer Protocol** - wersja 0.9, 1.0, 1.1
- **TCP**

HTTP 1.0



HTTP - żądania klienta

Żądanie: metoda

GET	pobierz zawartość danego dok.
HEAD	pobierz info. o nagłówku danego dok.
POST	potraktuj dokument tak jak skrypt - wyślij do niego dane
PUT	*zamień zawartość danego dokumentu*
DELETE	*skasuj dany dokument*

Żądanie: nagłówki

From	adres e-mail użytkownika
User-Agent	info. o przeglądarce
Accept	wspierane typy MIME
Accept-Encoding	wspierany typ kompresji
Referer	ostatnio wyświetlany URL
Authorization	kontrola dostępu
Charge-To	opłaty
If-Modified-Since	„tylko jeśli modyfikacja po”
Pragma	instrukcje wewn. serwera
Content-Length	długość danych w bajtach

HTTP - odpowiedź serwera cz.1/2

Odpowiedź: kod statusu

2xx	**sukces**
200	OK
202	Accepted
204	No response
3xx	**przekierowanie**
4xx	**błąd po stronie klienta**
400	Bad Request
403	Forbidden
404	Not Found
5xx	**błąd po stronie serwera**
500	Internal Error
501	Not Implemented

Przykłady

HTTP - odpowiedź serwera cz.2/2

Odpowiedź: nagłówki

Server	info. o serwerze
Date	aktualna data (czas GMT)
Last-Modified	ostatnia modyfikacja dokumentu
Expires	data ważności
URI	adres przekierowania
MIME-Version	wersja MIME
Content-Length	długość danych
Content-Type	typ danych
Content-Encoding	metoda kompresji
Content-Language	język dokumentu
Content-Transfer-Encoding	metoda kodowania (np. 7-bitowe, binarne)
WWW-Authenticate	kontrola dostępu
Message-Id	identyfikator wiadomości (tylko News)
Cost	koszt
Link	URL „ojca” dokumentu
Title	tytuł
Allowed	żadania żądającego użytk. mogć zostać wykonane
Public	żądania dowolnego użytk. mogć zostać wykonane

Przykład wymiany HTTP

```
>telnet www.tele.pw.edu.pl http
Trying 148.81.65.117
Connected to bach.tele.pw.edu.pl.
Escape character is '^]'.
GET /www/example.txt HTTP/1.0
From: lamer@any.site.pl
Accept: text/plain
Accept: text/html

HTTP/1.0 200 OK
Date: Monday, 28-May-98 12:34:56 GMT
Server: WebServer/1.0
MIME-version: 1.0
Last-modified: Sunday, 24-May-98 11:11:11 GM
Content-length: 12

Udalo sie!

Connection closed by foreign host.
>
```

Przeglądarka:
1. nawiązanie połączenia

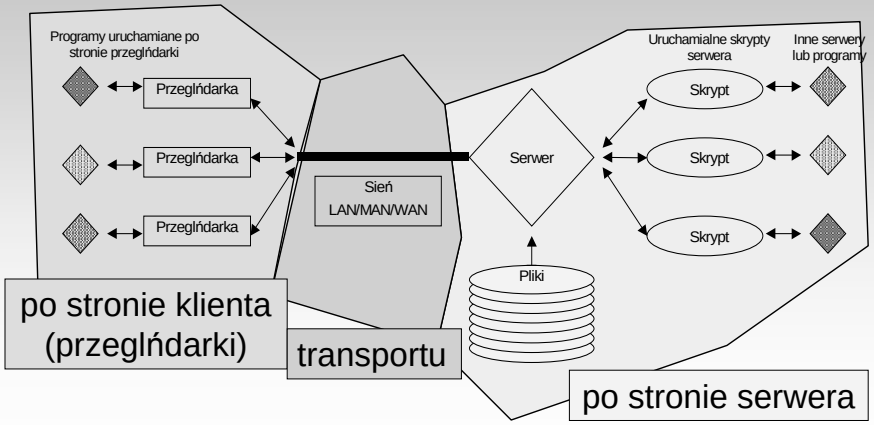
2. żądanie

Serwer:
3. odpowiedź

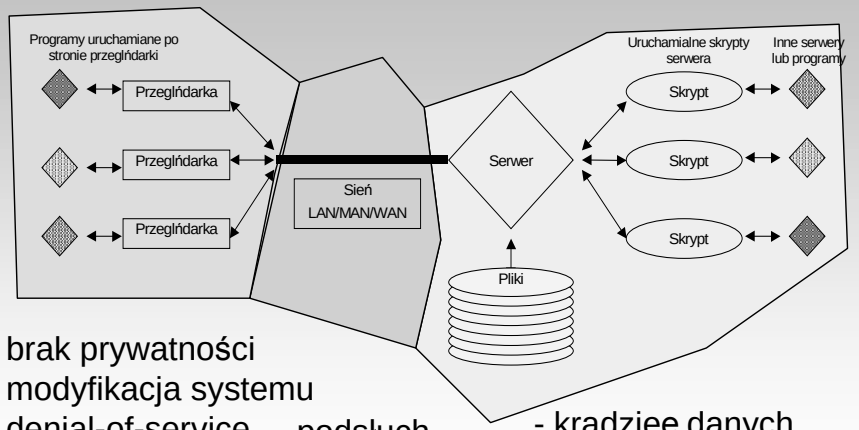
4. zakończenie połączenia

WWW - architektura usługowa

Obszary zabezpieczeń



Zagrożenia



- brak prywatności
- modyfikacja systemu
- denial-of-service

- podsłuch
- modyfikacja
- podszycie się

- kradzieę danych
- modyfikacja systemu
- denial-of-service

Kryptografia - pojęcia znane

Tutorial dla początkujących

- usługi ochrony informacji: kontrola dostępu, poufnoą , integralnoą , uwierzytelnienie, niezaprzeczalnoą
- mechanizmy ochrony informacji: szyfrowanie (symetryczne i asymetryczne), funkcja skrótu, MAC, podpis cyfrowy
- inne: certyfikat, uzgodnienie klucza, dystrybucja klucza
- algorytmy: DES, IDEA, RSA, DH, SHA, MDx

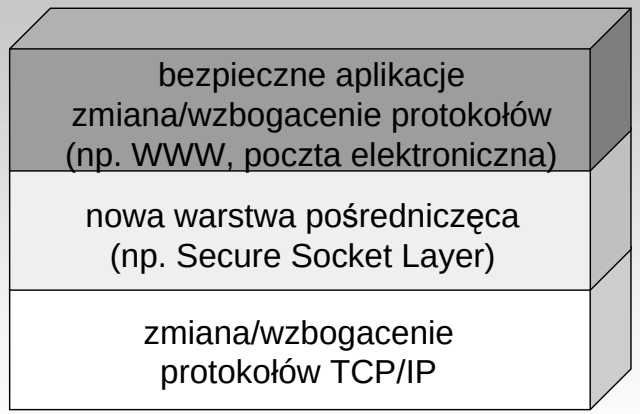
Cechy sieci Internet a bezpieczeństwo

Przypomnienie Enigma'97

- podstawowe protokoły rodziny TCP/IP nie posiadają ńadnych kryptograficznych zabezpieczeń, nie są realizowane usługi:
  - poufnońci (podsłuch wiadomońci)
  - integralnońci i uwierzytelnienia (modyfikacja danych, podszywanie sińi)
- wińkszoń aplikacji było/jest projektowanych (i pisanych) niebezpiecznie
- architektura sieci Internet wymusza:
  - zastosowanie kontroli dostńpu na poziomie warstwy sieciowej, a takńe na poziomie warstwy aplikacji (separacja sieci) - firewalling
  - zorganizowanie zaawansowanego audytu - logowanie
- konieczne jest takńe indywidualne zabezpieczenie podstawowych usług internetowych (e-mail, news, http) na poziomie uńytkownika

Kierunki zabezpieczeń w Internecie

Przypomnienie Enigma'97



+ ograniczenie penetracji sieci poprzez firewalling

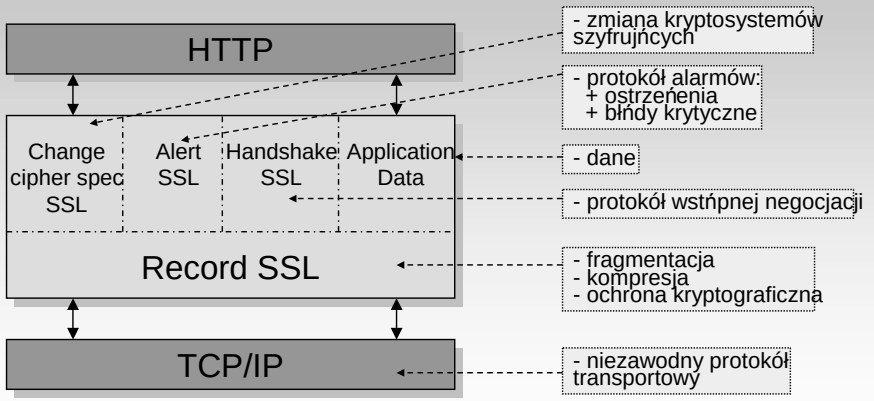
S-HTTP/1.4 (Secure HTTP)

- lipiec 1995 - Enterprise Integration Technologies Corp. - E. Rescorola i A.Schiffman
- IETF Working Group - web transaction security - draft z 11.97 (wańny do 5.98) - http://www.ietf.org/internet-drafts/draft-ietf-wts-shttp-05.txt
- RFC 2048 - Considerations for Web Transaction Security
- rozszerzenie HTTP
- shttp:// - ten sam port TCP co http://
- szyfrowanie, integralnośń (MAC), podpis cyfrowy
- ń danie: `Secure * Secure-HTTP/1.4`
- jedyna linia statusu: `Secure-HTTP/1.4 200 OK.`
- dwa typy nagłówków:
  - nagłówki ogólne - definiujń zastosowane mech. ochrony informacji - nie chronione
  - nagłówki HTTP - chronione poprzez enkapsulacjń

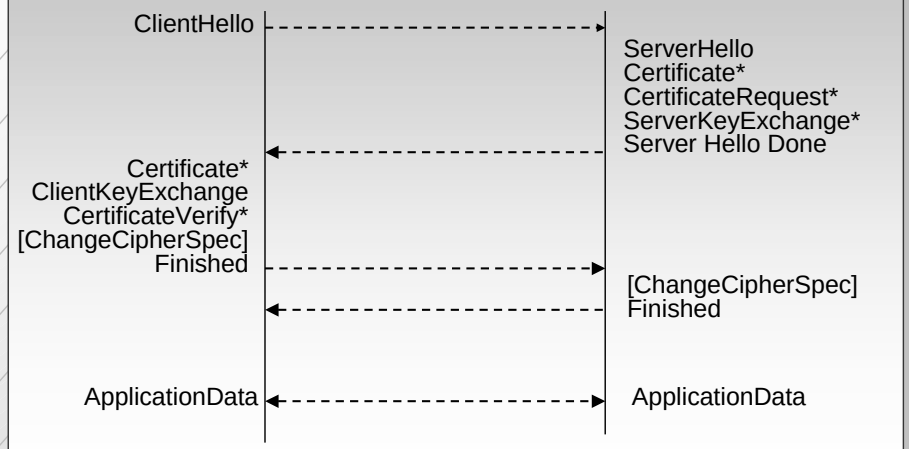
SSL

- Secure Sockets Layer - wersja 3.0
- Netscape Communications
- na popularnośń składa siń:
  - proste tworzenie aplikacji, łatwośń w uńyciu
  - bezpieczeństwo (nie chroni jednak przed analizń ruchu)
  - biblioteki (public domain SSL - SSLeay, SSLava - Java)
- https:// - port TCP 443
- poufnośń, integralnośń, uwierzytelnienie
- ulepszenie wersji 2.0
- przyszłośń: Transport Layer Security 1.0 draft IETF z 12.11.97 (wańny do 12.05.98) - http://www.ietf.org/internet-drafts/draft-ietf-tls-protocol-05.txt

SSL 3.0 - architektura



SSL 3.0 - Handshake SSL



SSL 3.0 - idea pracy cz.1/3

- Uńytkownik dokonuje zakupów w internetowym sklepie (czyli na serwerze) sprzedawcy
- Faza 1 - Handshake - ustalenie wspólnego klucza K
  1. Uńytkownik ń da, pobiera i weryfikuje certyfikat serwera.
  2. Uńytkownik tworzy losowo 160-bitowń wartośń K.
  3. Uńytkownik szyfruje K kluczem publicznym serwera.
  4. Uńytkownik wysyła szyfrogram (3) do serwera.
  5. Serwer deszyfruje szyfrogram swoim kluczem prywatnym - odzyskuje K.
  6. Serwer dokonuje skrótu K.
  7. Serwer wysyła skrót (6) do uńytkownika.
  8. Uńytkownik dokonuje skrótu K i porównuje z wartościń (7) otrzymanń.

SSL 3.0 - idea pracy cz.2/3

- Po zakończeniu fazy 1 - serwer jest uwierzytelniony przed uńytkownikiem, gdyń:
  - zna jego uwierzytelniony poprzez certyfikat klucz publiczny
  - serwer wykazał siń posiadaniem klucza prywatnego (zdolnośń do odszyfrowania K)
- K jest wspólny kluczem (SSL/TLS MasterSecret)
- Faza 2 - Bezpieczna wymiana danych przy pomocy wspólnego klucza K
  - dane transmitowane w postaci pakietów zaszyfrowanych K (poufnośń) i chronionych MAC (integralnośń)

SSL 3.0 - idea pracy cz.3/3

- Handshake - CipherSuite - parametry:
  - algorytmy symetryczne: ńaden, RC4 (40-bitowe i 128-bitowe), RC2 (40-bitowe), IDEA, DES (40-bitowy), DES, TripleDES
  - algorytmy asymetryczne: RSA, Fortezza i Diffie-Hellman (z certyfikatami opartymi o RSA, DSS albo bez nich)
  - funkcje skrótu: MD5, SHA
- Odnowienie sesji, sensowne gdyń:
  - operacje wykorzystujńce klucz prywatny sń kosztowne
  - zmniejszenie ruchu w sieci
  - łatwa multipleksacja równoległych połńczeń

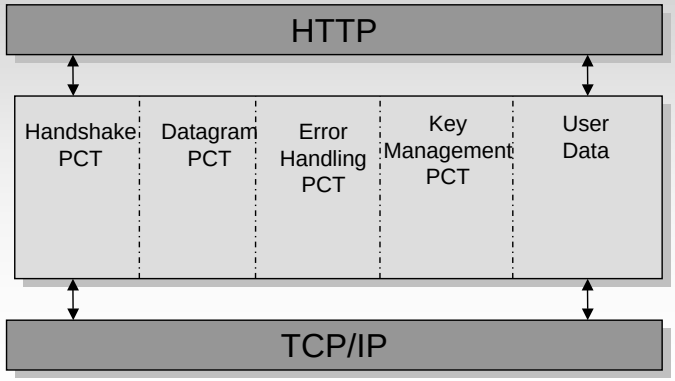
Najważniejsze problemy związane z SSL

wg. Paula Kochera - współtwórcy SSL

1. Brak wsparcia dla systemów pośredniczęcych (proxy).
2. Obcię enie obliczeniowe klienta i serwera.
3. Dodatkowy ruch w sieci (handshake).
4. Trudna migracja w stronę systemu symetrycznego.
5. Nie współpracuję dobrze z istniejęcymi tokenami kryptograficznymi (Kerberos).
6. Zarzędzanie kluczami kosztowne (często wymaga hardware'u).
7. Wymaga urzędu ds. certyfikacji z określonę politykę.
8. Zaszyfrowane informacje nie daję się kompresowaę (modemy).
9. Międzynarodowe restrykcje na algorytmy kryptograficzne.

PCT v2.0

- Private Communication Technology
- 1995 - Microsoft - wersja 1.0
- mutacja SSL 2.0
- nie zyskał popularności (wspierane tylko przez Microsoft)

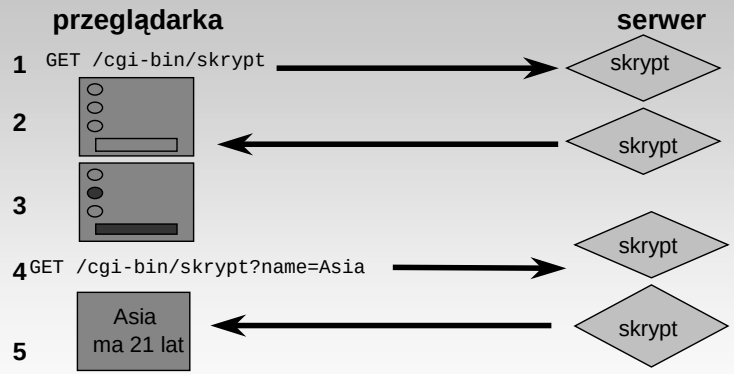


Bezpieczeństwo po stronie serwera

- „bezpieczna maszyna"
  - usunięcie zbędnych usług i uaytkowników,
  - rozszerzone logowanie, aktualne łaty
- bezpieczny serwer HTTP
  - wybór bezpiecznego serwera (producent, produkt, wersja)
  - konfiguracja:
    - wyłączenie automatycznego listowania katalogów
    - wyłączenie Symbolic Link Following
    - katalogi uaytkowników - szczególne restrykcje na symlinki i skrypty
    - skrypty: serwer uruchomiony z prawami „nobody"
    - anonimowe FTP bez moąliwoąci zapisu
    - ograniczenie praw na plikach konfiguracyjnych
    - uruchomienie w ąrodowisku chroot
    - zablokowanie odczytu plików access-log i error-log
    - kontrola dostępu na poziomie serwera
- bezpieczne skrypty
- anonimowoą (robots.txt)

Bezpieczne CGI

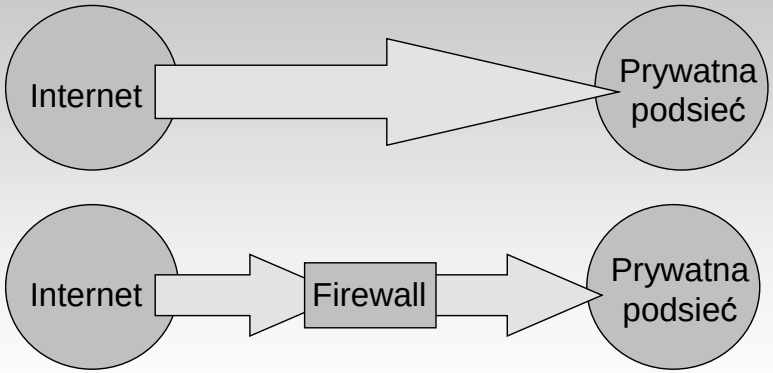
Common Gateway Interface



- wybór języka (Perl vs. C/C++)
- czego naleay unikaą pisząc programy?
- jak weryfikowaą „ąciągnięte" skrypty?

Firewalle - kontrola dostępu, audyt

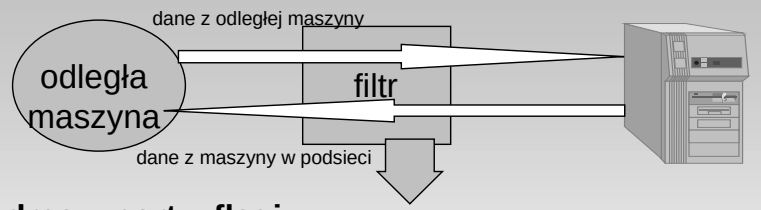
Przypomnienie Enigma'97



- separacja sieci
- pierwsza generacja produktów do zabezpieczeń

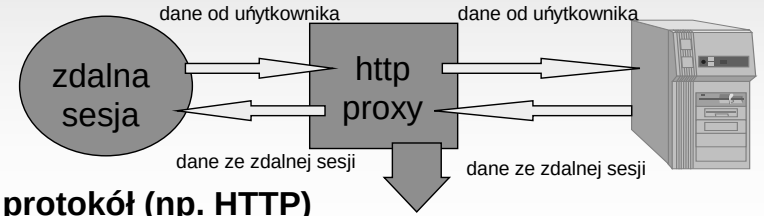
Filtr, proxy

Przypomnienie Enigma'97



adresy, porty, flagi

Logowanie połńczeń

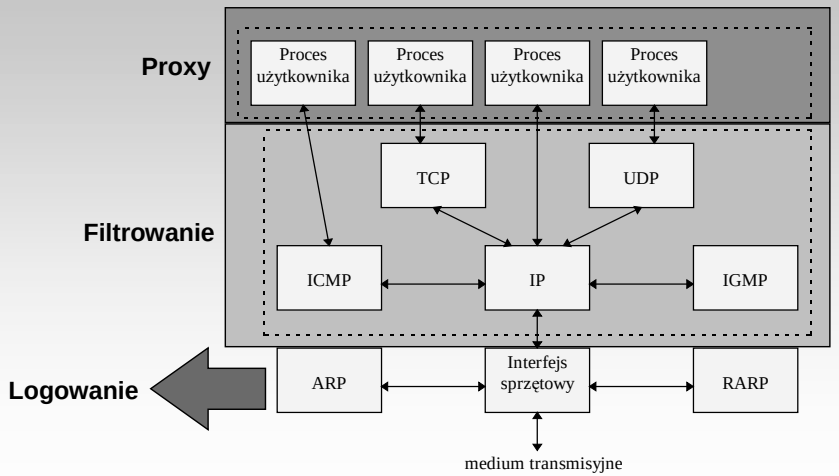


protokół (np. HTTP)

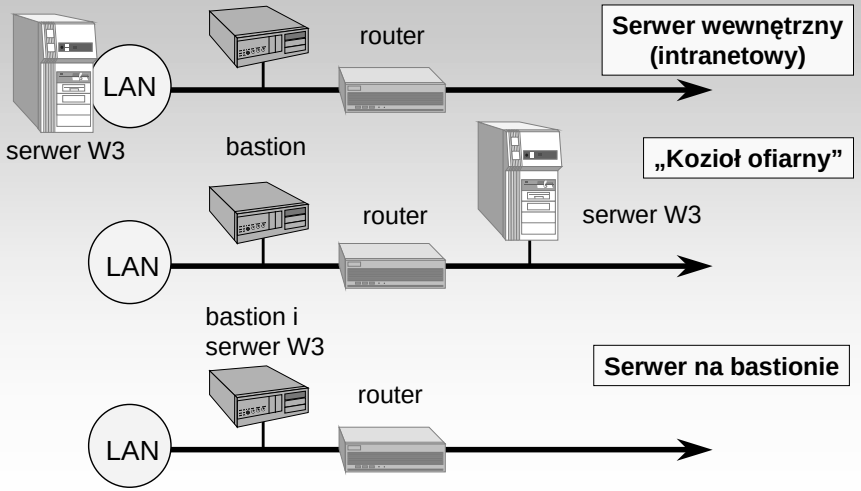
Logowanie połńczeń

Firewalling a model sieci Internet

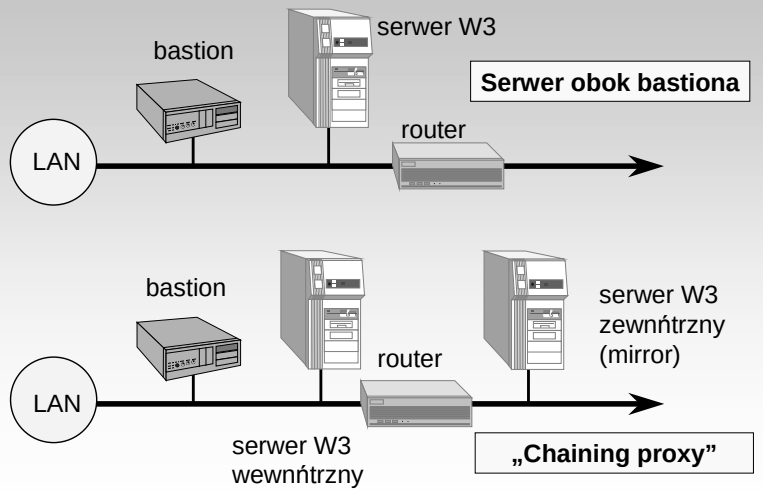
Przypomnienie Enigma'97



Serwer WWW a firewalle cz.1/2



Serwer WWW a firewalle cz.2/2



Bezpieczeństwo po stronie przeglądarki

- prywatność (m.in. ą ądaą klienta, haseł, posiadanych zasobów)
- problemy z:
  - Java (Sun)
  - JavaScript (Netscape)
  - ActiveX (Microsoft)
- inne błądy w implementacjach przeglądarek:
  - Microsoft Internet Explorer
  - Netscape Navigator

Podsumowanie

- bezpieczne WWW - nie jest objńte ńadnym standardem internetowym (RFC)
- architektura nie była tworzona pod kńtem elektronicznego handlu - ochrona styku klient-serwer to odpowiedń na typowe zagrońenia wystńpujńce w sieciach TCP/IP
- problem bezpieczeństwa serwera sprzedawcy, prawdziwości certyfikatu
- bezpieczne WWW - jako platforma - wsparcie dla innych protokołów
- co dalej? SET i inne protokoły

KONIEC

Czy mają Państwo pytania?