Hosting WWW
Bezpieczeństwo hostingu WWW

Dr Michał Tanaś (http://mtanas.web.amu.edu.pl/)

HTTPS

- Szyfrowana wersja protokołu HTTP
- Kiedyś używany do specjalnych zastosowań (np. banki internetowe), obecnie stosowany wszędzie
- Działa na zasadzie enkapsulacji protokołu HTTP w protokole TLS
- … tzn.
  - Najpierw uzgadniane jest połączenie TLS
  - … potem w jego ramach wysyłane są żądania/odpowiedzi HTTP

HTTPS

- TLS (Transport Layer Security) – biblioteka umożliwiająca tworzenie szyfrowanych tuneli pomiędzy dwoma hostami
- … dawniej zwana SSL (Secure Socket Layer)
- TLS nie należy mylić z
  - VPN (Virtual Private Network)
  - IPSec (Internet Protocol Security)
- Obecnie używana jest wersja 1.3 (z 2018 r.)

HTTPS

- Procedura nawiązywania połączenia TLS:
  1. Wymiana informacji o wersji i opcjach TLS obsługiwanych przez klienta i przez serwer
  2. Sprawdzenie certyfikatu serwera przez klienta (opcjonalne, praktycznie zawsze używane)
  3. Sprawdzenie certyfikatu klienta przez serwer (opcjonalne, rzadko używane)
  4. Uzgodnienie wspólnego szyfru
  5. … i można wymieniać dane

HTTPS

- Dlaczego stosuje się szyfrowanie?
  - Zabezpieczenie przed podsłuchem na poziomie warstwy fizycznej (np. przy połączeniach radiowych)
  - Zabezpieczenie przed podsłuchem pakietów IP przez któreś z urządzeń pośredniczących (routerów) – tzw. IP sniffing
  - Zabezpieczeniem przed atakiem na cache ARP switcha (tzw. ARP cache poisoning)
    - MAC man in the middle
    - MAC flooding

HTTPS

- Dlaczego stosuje się szyfrowanie?
  - Zabezpieczeniem przed włączeniem się w ciąg komunikacji HTTP (tzw. transparent proxy)
  - Możliwość weryfikacji tożsamości hostów (tzw. certyfikaty)
  - Możliwość weryfikacji prawdziwości danych (tzw. podpisy cyfrowe)

HTTPS

- Rodzaje szyfrów
  - Blokowe – działają na blokach danych o stałej długości (najczęściej 128b, 196b lub 256b)
  - Strumieniowe – działają na pojedynczych bajtach
  - Istnieją metody zamiany szyfru blokowego w strumieniowy
  - … najczęściej stosowany jest tzw. tryb licznika (ang. Counter Mode - CM)

HTTPS

- Rodzaje szyfrów
  - Symetryczne
    - Odszyfrowywanie odbywa się przy pomocy tego samego klucza co szyfrowanie
    - Szybkie
    - Bezpieczne dla stosunkowo małej długości klucza (256b)
    - Stosowane do zabezpieczenia danych wymienianych w już nawiązanym połączeniu

HTTPS

- Rodzaje szyfrów
  - Symetryczne – najważniejsze szyfry
    - DES (Data Encryption Standard) – 1975
      - nieoficjalna nazwa: „Lucifer”
      - pierwszy szyfr powszechnie używany w Internecie
      - obecnie przestarzały i nie używany
      - długość klucza 56b

HTTPS

- Rodzaje szyfrów
  - Symetryczne – najważniejsze szyfry
    - 3DES (triple DES)
      - potrójne szyfrowanie szyfrem DES z trzema różnymi kluczami
      - co teoretycznie dawało szyfr o długości klucza 168b
      - … ale …
      - … nikomu nie udało się udowodnić że 3DES jest istotnie silniejszy od DES!
      - Obecnie przestarzały i nie używany

HTTPS

- Rodzaje szyfrów
  - Symetryczne – najważniejsze szyfry
    - RC4 (Rivest Cipher 4) - 1987
      - Inne nazwy: ARC4, ARCFOUR
      - Długość klucza do 256b
      - Początkowo uważany za następcę DES
      - … ale szybko wykryto skuteczne metody jego łamania
      - Obecnie uznawany za przestarzały
      - Był używany w WEP i WPA (ale nie WPA2 i WPA3)

HTTPS

- Rodzaje szyfrów
  - Symetryczne – najważniejsze szyfry
    - AES (Advanced Encryption Standard) - 1997
      - Nieoficjalna nazwa: „Rijndael”
      - Długość klucza: 128b, 192b lub 256b
      - Obecnie nie jest znany żaden możliwy do praktycznego wykonania atak na sam algorytm
      - … aczkolwiek znane są ataki na niektóre implementacje sprzętowe
      - Używany: w TLS, WPA2 (w wersji CCMP - Counter Mode Cipher Block Chaining Message Authentication Code Protocol) i WPA3 (w wersji GCM – Galois Counter Mode)

HTTPS

- Rodzaje szyfrów
  - Asymetryczne (szyfry z kluczem publicznym)
    - Używają pary kluczy, przy czym
      - Dane zaszyfrowane kluczem A mogą być odszyfrowane jedynie kluczem B
      - … ale nie samym kluczem A!
      - Podobnie dane zaszyfrowane kluczem B mogą być odszyfrowane jedynie kluczem A
      - … ale nie samym kluczem B!

HTTPS

- Rodzaje szyfrów
  - Asymetryczne (szyfry z kluczem publicznym)
    - Zazwyczaj jeden z kluczy udostępnia się publiczne (tzw. klucz publiczny)
    - … a drugi trzyma w sekrecie (tzw. klucz prywatny).
    - Dane zaszyfrowane kluczem publicznym może odczytać jedynie posiadacz klucza prywatnego
    - … czyli mamy gwarancję że dane odczyta tylko właściwy odbiorca
    - … i nikt inny, nawet nadawca danych!

HTTPS

- Rodzaje szyfrów
  - Asymetryczne (szyfry z kluczem publicznym)
    - Można je wykorzystać do potwierdzenia autentyczności danych (tzw. podpisy cyfrowe)
      1) Oblicza się jakąś funkcję skrótu danych (np. MD5 czy SHA-1)
      2) … a następnie szyfruje się ją kluczem prywatnym
      3) … dzięki czemu każdy może sprawdzić kluczem publicznym czy nadawca zna właściwy klucz prywatny
      4) … i czy dane nie zostały zmienione

HTTPS

- Rodzaje szyfrów
  - Asymetryczne (szyfry z kluczem publicznym)
    - Można je wykorzystać do potwierdzenia autentyczności hostów (tzw. certyfikaty)

1) Klient wysyła przypadkowe dane (tzw. „wyzwanie”, ang. challenge)
2) Serwer szyfruje wyzwanie swoim kluczem prywatnym
3) … a klient odszyfrowuje je znanym kluczem publicznym
4) … dzięki czemu wie że serwer zna właściwy klucz prywatny

HTTPS

- Rodzaje szyfrów
  - Asymetryczne (szyfry z kluczem publicznym)
    - Można je wykorzystać do potwierdzenia autentyczności hostów (tzw. certyfikaty)
      - Autentyczność klucza publicznego potwierdza w analogiczny sposób organizacja zwana CA (Certificate Authority)

HTTPS

- Rodzaje szyfrów
  - Asymetryczne (szyfry z kluczem publicznym)
    - Są powolne
    - Wymagają stosunkowo dużych długości klucza (minimum 2Kb)
    - Dlatego są używane do potwierdzania tożsamości i do nawiązania połączenia (tzn. uzgodnienia klucza szyfru symetrycznego)
    - … ale nie do samej wymiany danych

HTTPS

- Rodzaje szyfrów
  - Asymetryczne (szyfry z kluczem publicznym)
    - Najważniejszym szyfrem asymetrycznym jest RSA (Rivest Shamir Alderman) – 1977
      - … natomiast istnieje algorytm Shora
      - … który bardzo szybko rozkłada liczby na czynniki pierwsze
      - … ale wymaga on tzw. komputera kwantowego
      - … co jest jednym z powodów intensywnych prac nad komputerami kwantowymi

HTTPS w Apache

1) Włączyć moduł „SSL”
2) Stworzyć nowy certyfikat SSL
3) … i samemu go podpisać (tzw. self-signed certificate)

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048
    -keyout mysitename.key -out mysitename.crt
```

HTTPS w Apache

1) Stworzyć VirtualHost działający na porcie 443
2) … i w jego konfiguracji
   a) Włączyć SSL/TLS: „SSLEngine On”
   b) Podać ścieżkę do certyfikatu: „SSLCertificateFile”
   c) Podać ścieżkę do pliku z kluczem prywatnym: „SSLCertificateKeyFile”
3) Zrestartować Apache
4) … i gotowe.

HTTPS w Apache

Żeby uzyskać publiczny certyfikat serwera należy wygenerować tzw. CSR – Certificate Signing Request

```
openssl req -new -newkey rsa:4096 -nodes -keyout mysite.key -out mysite.csr
```

HTTPS w Apache

W czasie generowania CSR należy podać następujące informacje (openssl zada pytania na terminalu):

- Kod kraju (PL dla Polski)
- Województwo (Province)
- Miasto (Locality)
- Organizacja/firma będąca właścicielem strony (Organization name). Uwaga, chodzi o właściciela strony a nie serwera!
- Działu, wydziału, BU, itp. (Organization Unit Name)

HTTPS w Apache

W czasie generowania CSR należy podać następujące informacje (openssl zada pytania na terminalu):

- Adres DNS'owy strony, tzw. FQDN (common name). Uwaga! przeglądarka go sprawdza!
- Adres mailowy webmastera (Email address)
- Challenge password: **zostawić pusty (wcisnąć tylko enter),** inaczej będzie trzeba ręcznie podać hasło przy każdorazowym starcie Apache!

HTTPS w Apache

Warto sprawdzić czy CSR jest poprawny

```
openssl req -text -in mysite.csr -noout -verify
```

Następnie wysyła się plik CSR do wybranego CA i otrzymuje się plik z certyfikatem na określony czasu (usługa płatna).

HTTPS w Apache

Jak wygenerować CSR dla strony mającej więcej niż jedną nazwę (aliasy DNS) lub więcej niż 1 adres IP ?

Tworzymy plik tekstowy np. mysite.cfg z następującą zawartością:

```
[req]
distinguished_name = req_distinguished_name
req_extension = v3_req
```

HTTPS w Apache

[req_distinguished_name]

C = PL

ST = Wielkopolska

L = Poznań

O = UAM

OU = Wydział Fizyki

CN = www.amu.edu.pl

HTTPS w Apache

```
[req_v3]
subjectAltName = @alt_names
[alt_names]
DNS.1 = fizyka.amu.edu.pl
DNS.2 = www.fizyka.amu.edu.pl
```

HTTPS w Apache

Używamu pliku mysite.cfg do wygenerowania CSR:

openssl req -new -newkey rsa:4096 -nodes -keyout mysite.key **-out** mysite.csr **-config** mysite.cfg

Który wysyłamy do wybranego CA otrzymując gotowy certyfikat (usługa płatna).