

STATE AND SESSION MANAGEMENT

2010

Οικονομικό Πανεπιστήμιο Αθηνών

HTTP is Stateless

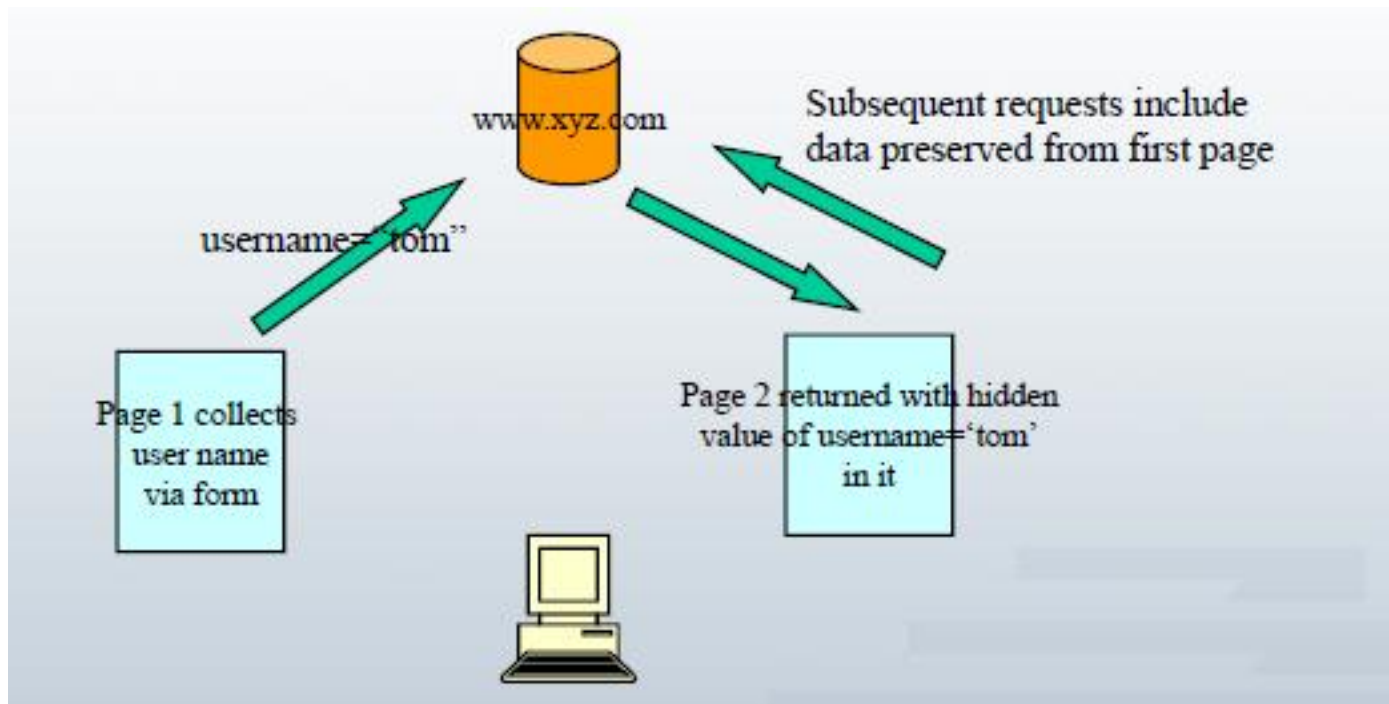
2

- Το HTTP είναι **stateless**, δηλαδή δεν υπάρχει μνήμη καθώς ο χρήστης επισκέπτεται διάφορες σελίδες. Υπάρχουν τεχνικές για την αντιμετώπιση αυτού του προβλήματος όπως:
 - ▣ Κρυφά πεδία φορμών
 - ▣ Dirty URLs
 - ▣ Cookies
 - Memory ή session cookies
 - Persistent cookies
- Ευτυχώς, τα περισσότερα προγραμματιστικά περιβάλλοντα, όπως PHP, ColdFusion, ASP.NET, JSP, κτλ παρέχουν την ιδέα της συνόδου (session) η οποία αποκρύπτει αρκετές λεπτομέρειες της διατήρησης καταστάσεων (state preservation)
 - ▣ Οι σύνοδοι υλοποιούνται συνήθως σαν cookies που αποθηκεύουν ένα μοναδικό αναγνωριστικό που συνδέεται με δεδομένα που είναι αποθηκευμένα στον εξυπηρετητή

Κρυφά πεδία φορμών

2

- Η φόρμα περιέχει `<input type="hidden" name="cartid" value="78Ccad786">` και αυτά τα δεδομένα στέλνονται κατά την υποβολή της
- Οι μεταγενέστερες σελίδες που φορτώνονται περιέχουν τα δεδομένα του χρήστη σε κρυφά πεδία



Κρυφά πεδία φορμών

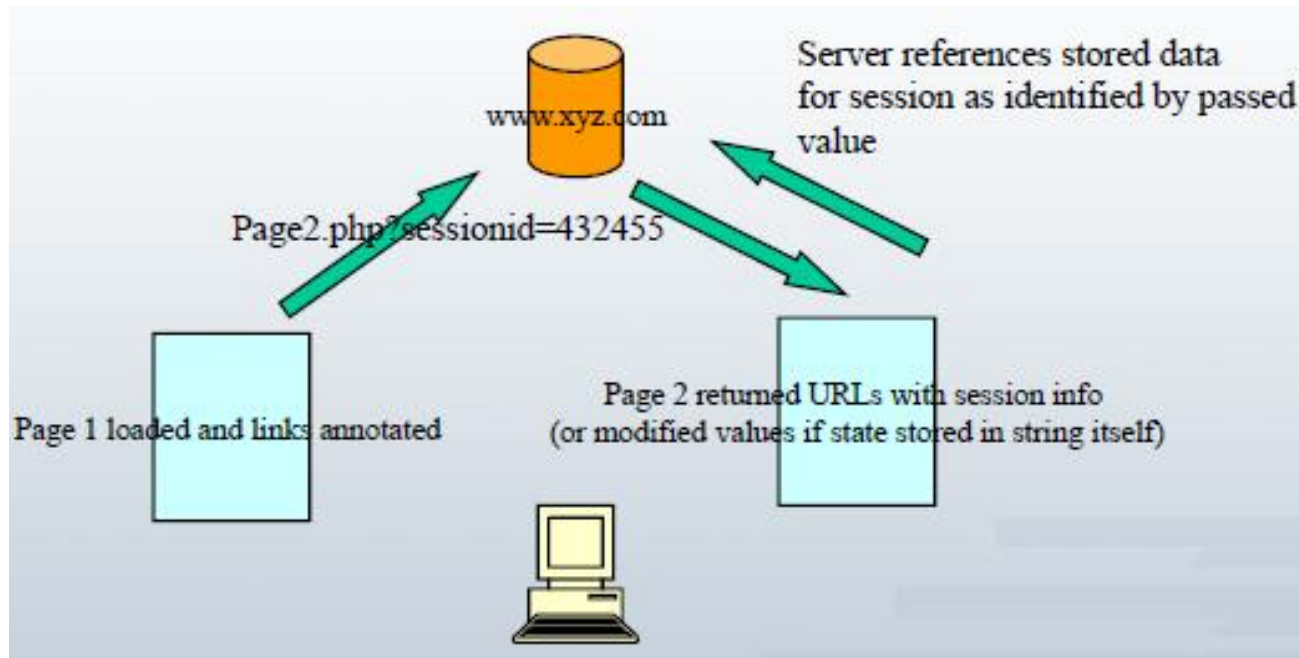
4

- Μπορούν να τροποποιηθούν από τους χρήστες
 - ▣ Λύση: Έλεγχοι ακεραιότητας στην εφαρμογή
- Δεν διατηρούνται πληροφορίες κατάστασης όλης της επίσκεψης του χρήστη στον ιστόχωρο
- Απαιτεί την ύπαρξη μιας φόρμας για την φύλαξη των κρυφών δεδομένων
 - ▣ Στη .NET χρησιμοποιούνται κρυφά πεδία φορμών και ολόκληρη η σελίδα ενσωματώνεται σε μια φόρμα

Διατήρηση Κατάστασης με "Dirty URLs"

5

- Όλοι οι σύνδεσμοι ξαναγράφονται περιέχοντας δεδομένα κατάστασης με τη μορφή ενός query string
` Page 1`
` Page 2`
- Όλες οι φόρμες περνούν τις τιμές χρησιμοποιώντας query strings (είτε άμεσα αλλάζοντας το action ή με τη χρήση κρυφών πεδίων)



Διατήρηση Κατάστασης με "Dirty URLs"

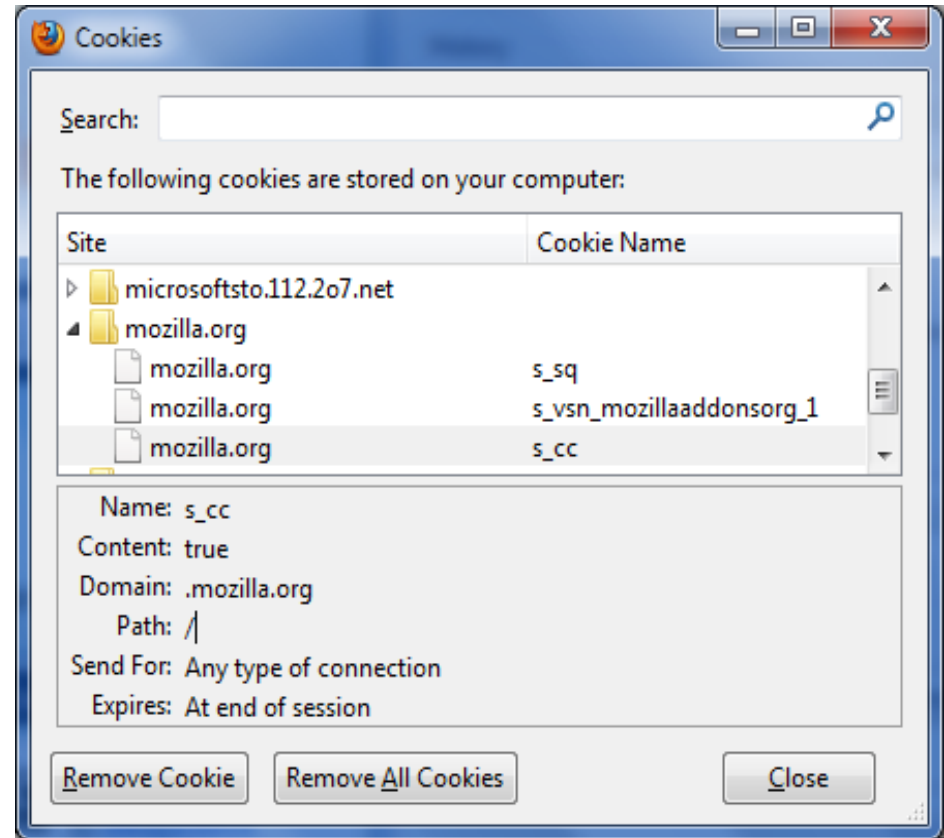
6

- Μπορούν να τροποποιηθούν από τους χρήστες
 - ▣ Λύση: Έλεγχοι ακεραιότητας στην εφαρμογή
- Δεν διατηρούνται πληροφορίες κατάστασης όλης της επίσκεψης του χρήστη στον ιστόχωρο
- Δεν συμβάλουν στην σωστή χρήση των URL, προκύπτουν προβλήματα marketing
- Περιορισμός στο μέγεθος των δεδομένων που μπορούν να αποθηκεύσουν (το όριο για ένα URL είναι 2K)

Cookies

7

- Είναι δεδομένα που αποθηκεύονται στο μηχάνημα του χρήστη
 - ▣ Τα δεδομένα ενδέχεται να αποθηκεύονται με αναφορά, δηλαδή με κάποιο δείκτη σε δεδομένα που υπάρχουν αποθηκευμένα σε ένα εξυπηρετητή



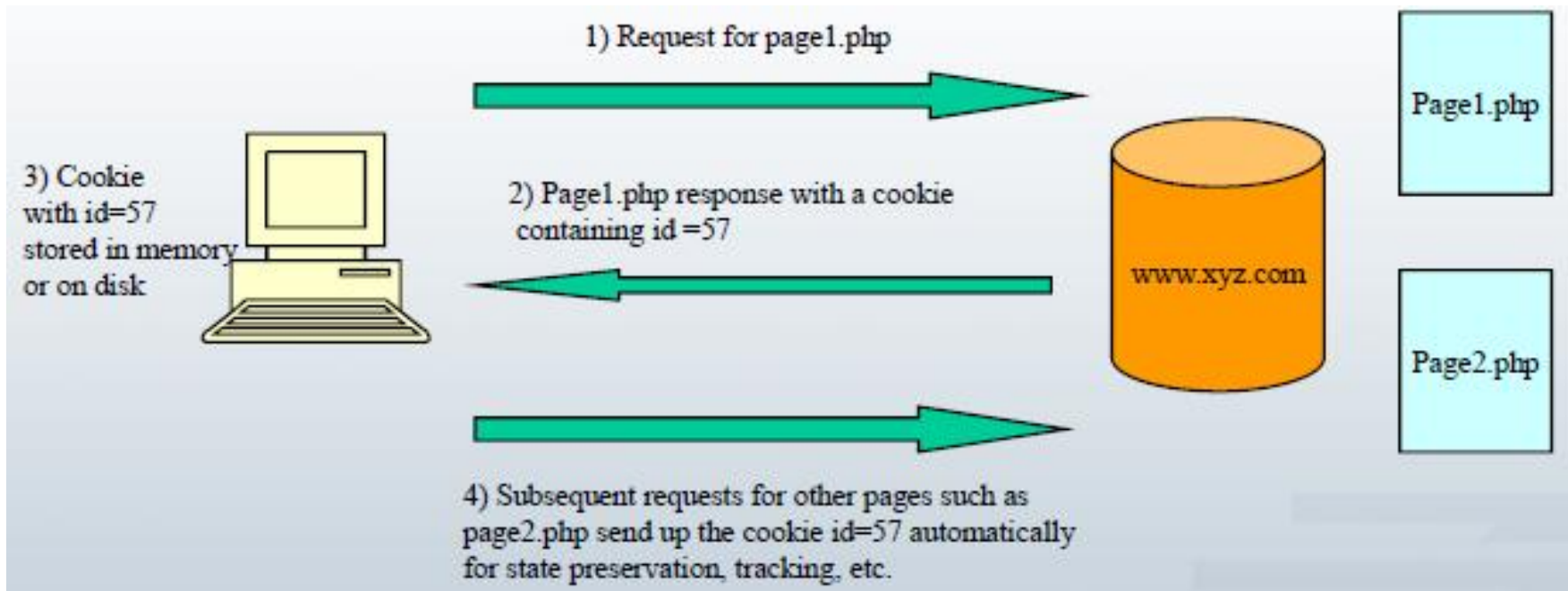
Cookies

8

- Τα cookies μεταφέρονται στο browser μέσα στο HTTP header
 - ▣ Set-Cookie: NAME=VALUE; expires=DATE; path=PATH; domain=DOMAIN_NAME; secure
- Συνηθίζεται να μην δημιουργείται manually το cookie (παρόλο που αυτό είναι δυνατό), αλλά αντίθετα το προγραμματιστικό περιβάλλον να παρέχει κάποια εντολή δημιουργίας cookie
 - ▣ PHP: `setcookie(name, value);`
 - ▣ Παράδειγμα: `setcookie('leadstooge','moe');`
- Ακόμη, είναι δυνατή η δημιουργία cookies με τη χρήση JavaScript, μέσω του αντικειμένου *document.cookie*

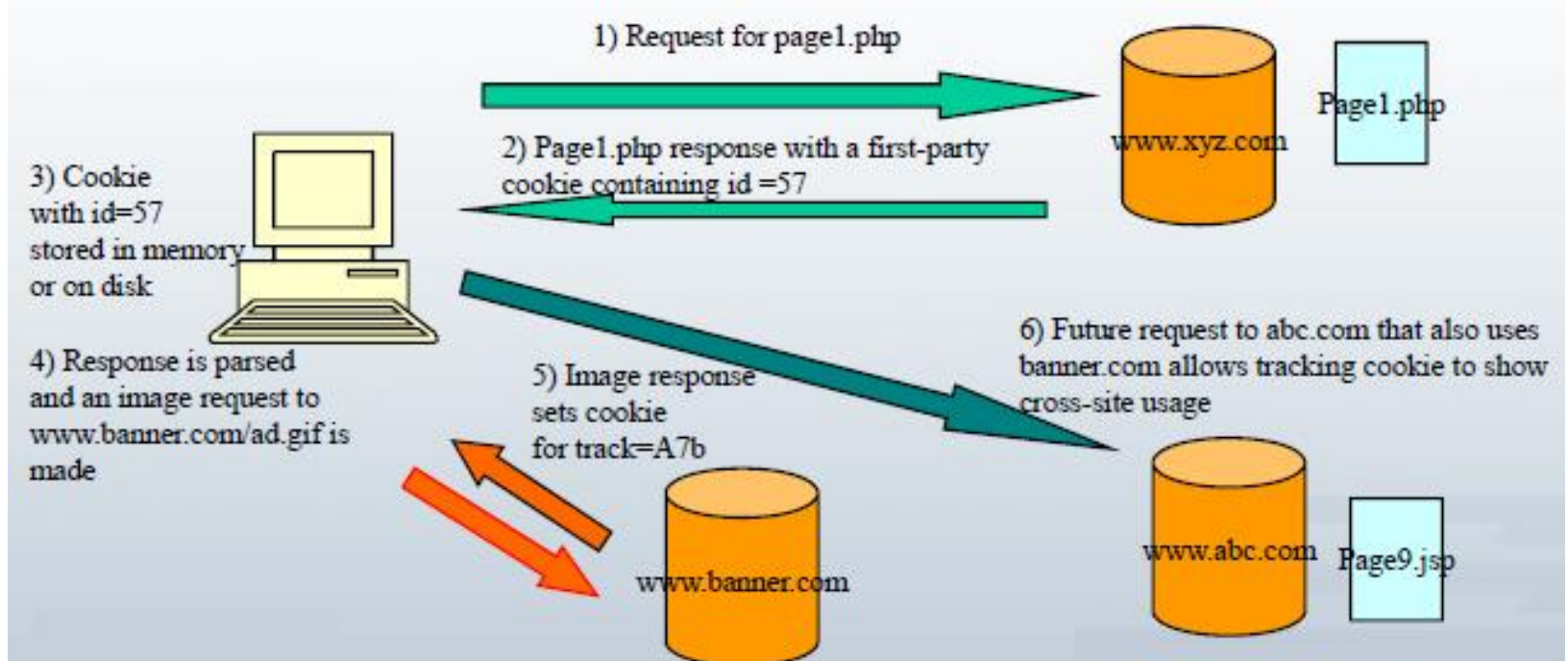
Παράδειγμα First Party Cookie

- Μόλις δημιουργηθεί το cookie μεταφέρεται στον εξυπηρετητή κάθε φορά που ο χρήστης ζητεί μια σελίδα στο καθορισμένο domain & path του Cookie.



Παράδειγμα Third Party Cookie

- Τα cookies μπορούν να δημιουργούνται από οποιοδήποτε αίτημα (request) όπως για εικόνες, CSS, scripts, κτλ
 - Αυτό συνήθως χρησιμοποιείται στην περίπτωση που δημιουργείται ένα Third party cookie για διαφημιστικούς λόγους ή για την παρακολούθηση του ιστορικού επισκέψεων του χρήστη σε ένα ιστότοπο



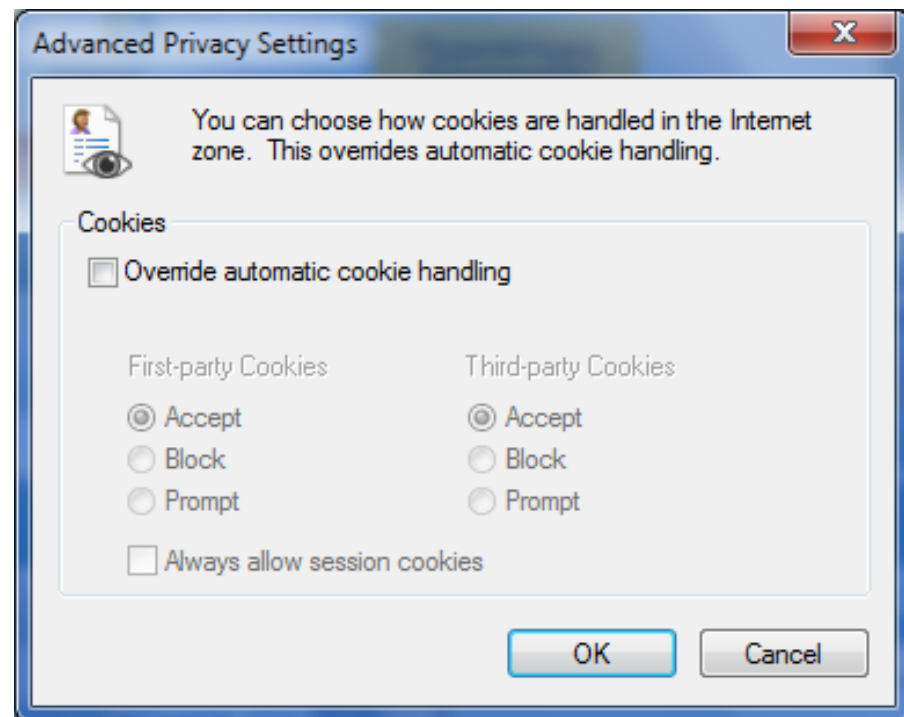
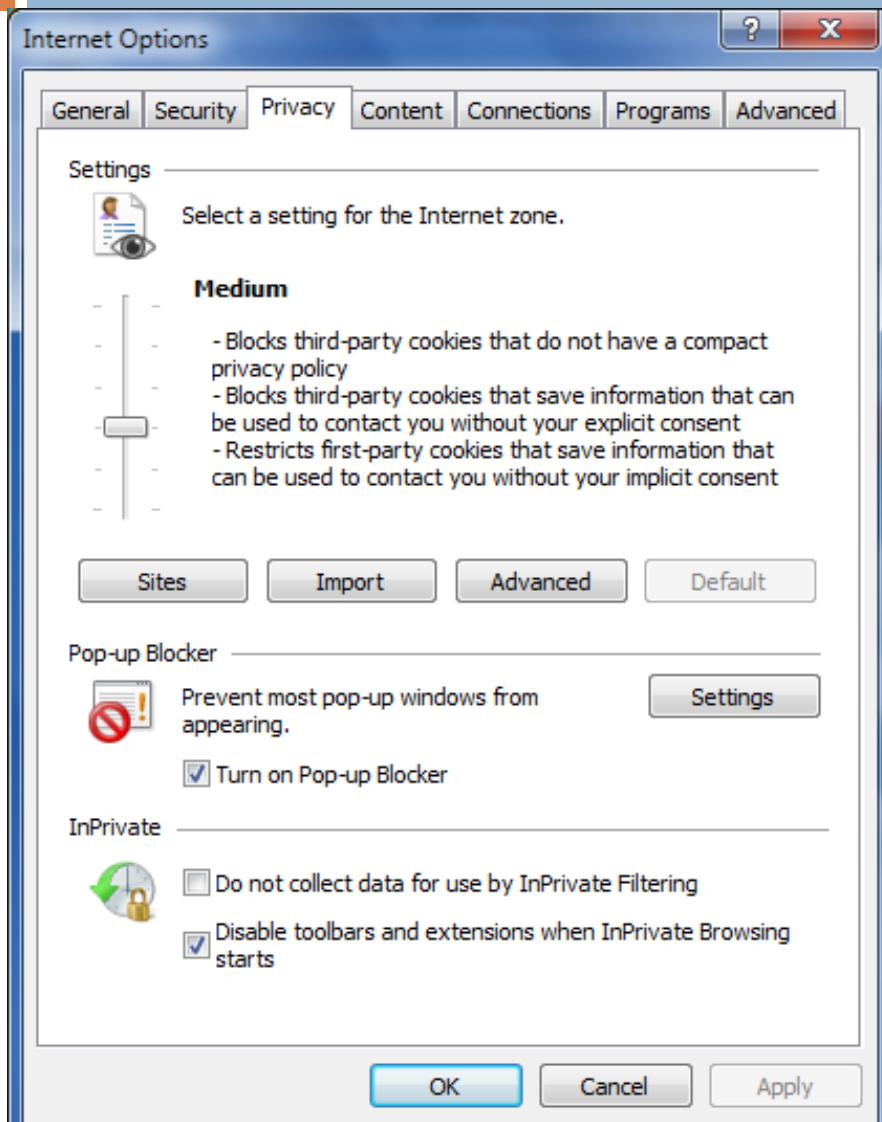
Ζητήματα Ιδιωτικότητας που σχετίζονται με τη χρήση των Cookies

11

- Τα Third party cookies δίνουν την δυνατότητα παρακολούθησης της δραστηριότητας του χρήστη σε διάφορους ιστότοπους
- Εδώ τα cookies συνήθως δημιουργούνται με την χρήση εικόνων GIF μεγέθους ενός pixel, ώστε να διαφεύγουν της προσοχής ενός χρήστη που θέλει να τις εμποδίσει
 - Συνήθως αναφέρονται ως "Web bugs" ή "bugs"
 - Συνήθως αυτές οι πρακτικές χρησιμοποιούνται από υπηρεσίες ανάλυσης του Web (Web analytics), όπως η Hitbox της WebSideStory
- Τα ζητήματα ιδιωτικότητας της χρήσης των cookies πηγάζουν από το γεγονός ότι ένας χρήστης ενδέχεται να εγγραφεί σε έναν ιστόχωρο, όπως προηγουμένως στον xyz.com και να συσχετίσει προσωπικά δεδομένα με ένα cookie-id. Αν τα προσωπικά δεδομένα γίνουν διαθέσιμα σε άλλους ιστότοπους είναι δυνατόν να δημιουργηθεί ένα προφίλ των συνηθειών του χρήστη στο Web
 - Έτσι πολλοί χρήστες θέλουν να εμποδίζουν την δημιουργία Third party cookies, όμως το πραγματικό πρόβλημα αφορά την παροχή προσωπικών δεδομένων από τους χρήστες χωρίς να είναι γνωστό τι συμβαίνει σε αυτές
 - Οι browsers, εργαλεία και τεχνολογία σαν την P3P, μπορούν να βοηθήσουν στον περιορισμό τους, παρόλο που το πρόβλημα είναι κοινωνικό και όχι τεχνολογικό

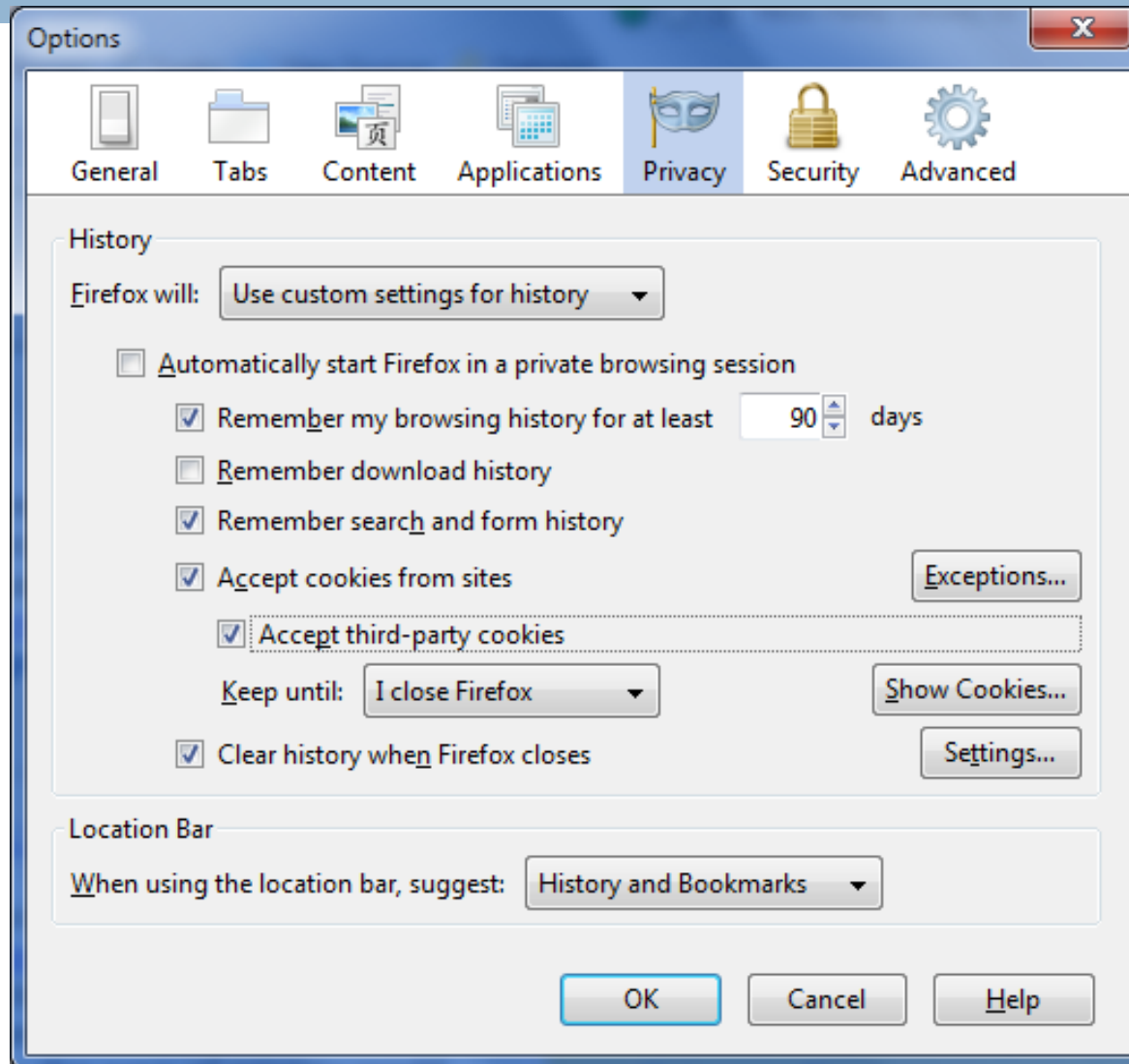
Σχετικές Ρυθμίσεις στον ΙΕ

12



Σχετικές Ρυθμίσεις στον Firefox

12



Δημιουργία Cookie με PHP (1/2)

14

```
<?php
setcookie('name','Moe');
setcookie('numstooges','3');
?>
<html>
<head><title>Simple Cookie</title></head>
<body>
  <h1>I just "baked" a cookie</h1>
  <a href="cookie2.php">See what's in the cookie</a>
  <form>
    <input type="button" value="JS Cookie Check"
      onclick="alert(document.cookie);">
  </form>
</body>
</html>
```

Δημιουργία Cookie με PHP (2/2)

15

```
<html>
<head><title>Show Cookie</title></head>
<body>
<?php
    print ("There are " . $_COOKIE['numstooges'] . " stooges and the main
    one is named " . $_COOKIE['name']);
?>
</body>
</html>
```

- Σε αυτό το παράδειγμα δημιουργήθηκε ένα session cookie (αλλιώς memory cookie)
 - Θα διαγραφεί μόλις τερματιστεί ο browser
 - Δεν είναι κοινό μεταξύ των browsers, αλλά είναι κοινό μεταξύ διαφορετικών παραθύρων του browser
 - Αυτά τα cookies δεν προσβάλλουν ιδιαίτερα την ιδιωτικότητα των χρηστών και είναι απαραίτητα για την δημιουργία των σύγχρονων ιστότοπων (
 - παρόλα αυτά κάποιοι χρήστες εξακολουθούν να απαγορεύουν την δημιουργία τους)



Περιορισμοί των Cookies

16

- Υπάρχουν περιορισμοί του browser ή του server όσον αφορά τα cookies
 - ▣ Συνήθως ένας browser
 - Μπορεί να αποθηκεύσει ένα περιορισμένο αριθμό cookies (~300)
 - Όταν αυτό το όριο ξεπεραστεί το νέο cookie αντικαθιστά το λιγότερο χρησιμοποιούμενο cookie
 - Το μέγεθος του cookie δεν μπορεί να είναι μεγαλύτερο από 4K
 - Μεγαλύτερα σε μέγεθος cookie "κόβονται" ώστε να μην ξεπερνούν το μέγιστο επιτρεπόμενο μέγεθος
 - Είναι σύνηθες ότι ένας browser δεν θα αποθηκεύσει ή/και ένας server δεν θα δεχθεί περισσότερα από 20 cookies για ένα συνδυασμό host-domain
 - Παρόλα αυτά τα web.xyz.com και store.xyz.com θεωρούνται διαφορετικά και μπορούν να έχουν 20 cookies το καθένα
 - Στην πράξη αυτοί οι περιορισμοί μπορεί να μην τηρούνται καθώς εξαρτώνται από τον τρόπο υλοποίησής τους
 - Συνήθως κατά τον προγραμματισμό εφαρμογών στον ιστό αποθηκεύουμε λίγα cookies για κάθε domain και μόνο ένα sessionid ή κάποια άλλη τιμή σχετική με τα αποθηκευμένα δεδομένα στον server

Δημιουργία Persistent Cookies

17

- Δημιουργούνται με τον προσδιορισμό της ημερομηνίας λήξης του cookie
 - ▣ PHP: `setcookie('name','value',expiration,'path','domain',secure);`
 - ▣ Η λήξη υπολογίζεται με τον προσδιορισμό των δευτερολέπτων από την τρέχουσα ώρα
 - ▣ Παραδείγματα

```
setcookie('color','green',time()+3600);  
// expires in 1 hour  
setcookie('userid','admin',time()+3600, '/admin');  
// limit to admin directory path  
setcookie('userid','admin',time()+3600,'/admin',  
'', 1);  
// use security in transmit
```

Δημιουργία Persistent Cookie

18


```
<?php
    setcookie('favoritestoooge', 'Moe', time()+3600);
?>


<html>
<head><title>Simple
    Cookie</title></head>
<body>
<h1>I just "baked" a
    persistent cookie so
    check your disk</h1>
</body>
</html>
```



persistent.php

Cookie Information - http://localhost/test_lab/state%

 Collapse All

 Expand All

http://localhost/test_lab/state%20management/persistent.php

 1 cookie

NAME	favoritestoooge
VALUE	Moe
HOST	localhost
PATH	/test_lab/state%20management/
SECURE	No
EXPIRES	Sun, 11 Apr 2010 10:20:09 GMT

 [Edit Cookie](#)

 [Delete Cookie](#)

Διαγραφή των Cookies

10

- Τα **cookies συνόδου** διαγράφονται αφού **τερματίσει ο browser**, ενώ τα **persistent cookies** διαγράφονται όταν επέλθει η **λήξη** τους
 - ▣ Τα persistent cookies διαγράφονται πρόωρα αλλάζοντας την ημερομηνία λήξης τους ή αλλάζοντας την σε κενό ή καλύτερα κάνοντας και τα δυο
 - `setcookie('favoritestoooge','');`
 - `setcookie('favoritestoooge','',time()-3600);`
- Σε μια εφαρμογή στο ιστό πρέπει να ενθαρρύνεται, για λόγους ασφάλειας, ο χρήστης να βγαίνει από αυτή (logout) αντί να πηγαίνει σε άλλη σελίδα/εφαρμογή
 - ▣ Για παράδειγμα σκεφτείτε τι θα συνέβαινε σε ένα σύστημα ελεύθερης πρόσβασης αν έμεναν εκτεθειμένα cookies από τραπεζικές συναλλαγές

Cookies & Security

20

- Όπως είπαμε τα cookies αν δεν διαγραφούν από την εφαρμογή τότε μπορούν να χρησιμοποιηθούν από ένα κακόβουλο χρήστη που θα χρησιμοποιήσει το ίδιο υπολογιστικό σύστημα για να νικήσει τους μηχανισμούς αυθεντικοποίησης
 - ▣ Session hijacking
- Επιπρόσθετα το γεγονός ότι τα cookies συμπεριλαμβάνονται σε κάθε header ενός HTTP request μπορεί να χρησιμοποιηθεί από έναν επιτιθέμενο για να εισάγει εντολές σε μια εφαρμογή στον ιστό εκ μέρους και χωρίς την έγκριση του χρήστη
 - ▣ Cross Site Request Forgery Attack
 - περισσότερα στην αντίστοιχη διάλεξη!

Σύνοδοι(Sessions)

21

- Παρόλο που όλη η διαχείριση ενός ιστόχωρου μπορεί να πραγματοποιηθεί με τη χρήση cookies (ή ακόμα με κρυφά πεδία ή dirty URLs) συνίσταται η χρήση συνόδων
- Ένα σωστά υλοποιημένο σύστημα διαχείρισης συνόδων αποκρύπτει τον τρόπο που τα δεδομένα διατηρούνται καθώς ο χρήστης πλοηγείται από σελίδα σε σελίδα (συνήθως με χρήση cookie) και αποθηκεύει τα δεδομένα του χρήστη στον εξυπηρετητή συνδέοντας τα με ένα SESSIONID που αποθηκεύεται στο cookie
 - Προφανώς είναι δυνατή η υλοποίηση αυτού του συστήματος με την χρήση μιας βάσης δεδομένων, όπου αποθηκεύονται ή τροποποιούνται τα δεδομένα κατάστασης του χρήστη, και αποθήκευση ενός κλειδιού στο cookie
- Ένα σημαντικό πλεονέκτημα της χρήσης συνόδων είναι ότι κάνει δυσκολότερη την επίθεση **session hijacking** δεδομένου ότι δεν αποθηκεύονται σε cookie τα δεδομένα κατάστασης του χρήστη
 - Υποθέτοντας ότι ένας επιτιθέμενος δεν μπορεί να μαντέψει τα session ids εύκολα

Δημιουργία Συνόδων στην PHP

22

- Η συνάρτηση **session_start()** χρησιμοποιείται στην PHP για την δημιουργία ενός cookie με όνομα PHPSESSID και κάποια τιμή
- Η συνάρτηση είναι πολύ πιθανό να χρησιμοποιείται σε κάθε σελίδα της εφαρμογής
 - ▣ Εναλλακτικά μπορεί να ρυθμιστεί στο php.ini το session.auto_start() για την αποφυγή της κλήσης της συνάρτησης session_start() συνεχώς
- Είναι δυνατός ο προσδιορισμός διαφορετικού ονόματος για το cookie αντί για PHPSESSID
 - ▣ Παράδειγμα:
session_name('MYID');
session_start();
- Για την δημιουργία μιας μεταβλητής συνόδου χρησιμοποιείται η εντολή `$_SESSION['var'] = 'value';`
 - ▣ Παράδειγμα: `$_SESSION['stooge'] = 'moe';`

Παράδειγμα Συνόδων στην PHP(1/2)

22

```
<?php
    session_start();
    $_SESSION[ 'fav_stooge' ]='Moe';
    $_SESSION[ 'num_stooges' ]=3;
?>
<html
<head>
<title>Session Fun</title>
</head>
<body>
    <h1>Just set some session values</h1>
    <a href="session2.php">Next page</a>
    <!-- eventually read this later on session3.php -->
</body>
</html>
```


Παράδειγμα Συνόδων στην PHP(2/2)

2.4

```
<html
<head>
<title>Session Fun</title>
</head>
<body>
    <h1>Reading session variables</h1>
<?php
    session_start();
    print "There are ".$_SESSION['num_stooges'] ." and
    my favorite is ". $_SESSION['fav_stooge'];
?>
</body>
</html>
```

- Τι πρόβλημα έχει το απόσπασμα κώδικα αυτής της διαφάνειας, όσον αφορά το session management;

- Hint: Θυμηθείτε πως λειτουργεί το πρωτόκολλο HTTP

-  session1.php, session2.php

Διαγραφή Πληροφοριών Συνόδων στην PHP

25

- Για την διαγραφή μιας μεταβλητής συνόδου στην PHP χρησιμοποιούμε την εντολή `unset($_SESSION['thevar'])`, με τον ίδιο τρόπο που θα διαγράφαμε μια οποιαδήποτε μεταβλητή στην PHP
- Για την διαγραφή όλων των μεταβλητών της συνόδου καταστρέφουμε όλη τη δομή δεδομένων που τις φιλοξενεί
 - `$_SESSION = array();`
- Για την διαγραφή όλων των δεδομένων της συνόδου από τον εξυπηρετητή χρησιμοποιούμε την συνάρτηση `session_destroy();`
- Προφανώς για να διαγραφεί μια σύνοδος με τις παραπάνω επιλογές πρέπει πρώτα να δημιουργηθεί μια με την χρήση της `session_start()`
- Όπως αναφέραμε και στα cookies οι σύνοδοι πρέπει να καταστρέφονται καθώς ο χρήστης φεύγει από την εφαρμογή μας (logout)

Λεπτομέρειες Συνόδων στην PHP

26

- Εξ' ορισμού οι σύνοδοι στην PHP υλοποιούνται ως memory cookies που λήγουν με τον τερματισμό του browser
- Παρόλα αυτά είναι δυνατή η χρήση της παρακάτω συνάρτησης για τον ορισμό των τιμών που βρίσκονται σε ένα cookie
 - ▣ `session_set_cookie_params(expiration, 'path', 'domain', secure)`
 - ▣ Παράδειγμα:
`session_name('mysid');`
`session_set_cookie_params(3600, '/whoismgr', 'www.xyz.com');`
`session_start();`
 - ▣ Σημειώστε ότι σε αυτή τη περίπτωση δεν ορίζεται η τρέχουσα ώρα, αλλά προσδιορίζεται το χρονικό διάστημα που είναι έγκυρο το cookie

Λεπτομέρειες Συνόδων

27

- Session Hijack
 - ▣ Επιλογή IDs που δεν είναι προβλέψιμα
- Απενεργοποιημένα Cookies
 - ▣ Εξ' ορισμού χρήση Dirty URL, όπου το SESSIONID μπαίνει δυναμικά σε κάθε σύνδεσμο σαν query string
 - ▣ Προσοχή: Δεν είναι αυτοματοποιημένη διαδικασία και υποθέτει ότι όλες οι σελίδες περνούν από ένα script σε αντίθεση με τα cookies
- Sessions σε Server Farm
 - ▣ Πώς μοιράζονται τα δεδομένα στους διάφορους servers;
 - Χρήση ενός sticky server αντί να διαμοιράζονται τα δεδομένα
 - Διαμοιρασμός τους με χρήση:
 - ▣ Βάσης δεδομένων
 - ▣ Αποθήκευση τους σε ένα load balancer

