

LogIn.php 为登陆页面，未登陆要记录进行先登陆，每次双击打开浏览器的时候都要会分配一个 PHPSESSID，PHPSESSID 可以继承本浏览器打开的子页面，然后这个 ID 作为这个在当前浏览器下进行在这个服务器下的识别标志。如果是登陆过的用户，通过判断 PHPSESSID 可以直接进入下一页面 logged.php。

No.	Time	Source	Destination	Protocol	Length	Info
1938	285.909096835	127.0.0.1	127.0.0.1	DNS	76	Standard query response 0x6aed Refused A www.ubuntu.com
1939	285.909103918	127.0.0.1	127.0.0.1	DNS	76	Standard query response 0xfe54 Refused AAAA www.ubuntu.c...
4	0.000071908	127.0.0.1	127.0.0.1	HTTP	402	GET /logIn.php HTTP/1.1
16	0.000586392	127.0.0.1	127.0.0.1	HTTP	1472	HTTP/1.1 200 OK (text/html)
18	0.024872060	127.0.0.1	127.0.0.1	HTTP	416	GET /style.css HTTP/1.1

Transfer-Encoding: chunked\r\n
Connection: keep-alive\r\n
X-Powered-By: PHP/5.5.38\r\n
Set-Cookie: PHPSESSID=18be92294108b4b5a53085735e9cd38e; path=/\r\n
Expires: Thu, 19 Nov 1981 08:52:00 GMT\r\n

```
0000 00 00 03 04 00 06 00 00 00 00 00 00 00 00 00 .....  
0010 45 00 05 b0 3b c2 40 00 40 06 fb 83 7f 00 00 01 E...;.@.....  
0020 7f 00 00 01 00 50 c2 00 97 c1 a9 8c dc db 19 ea .....P.....  
0030 80 18 01 5e 03 a5 00 00 01 01 08 0a 00 19 92 4e .....^.....N  
0040 00 19 92 4d 48 54 54 50 2f 31 2e 31 20 32 30 30 ...MHTTP /1.1 200  
0050 20 4f 4b 0d 0a 53 65 72 76 65 72 3a 20 6e 67 69 OK..Ser ver: ngi  
0060 6e 78 2f 31 2e 35 2e 34 0d 0a 44 61 74 65 3a 20 nx/1.5.4 ..Date:  
0070 4d 6f 6e 2c 20 31 36 20 4a 61 6e 20 32 30 31 37 Mon, 16 Jan 2017  
0080 20 31 34 3a 31 33 3a 33 3a 20 47 4d 54 0d 0a 43 14:13:3 4 GMT..C  
0090 6f 6e 74 65 6e 74 2d 54 79 70 65 3a 20 74 65 78 ontent-T ype: tex  
00a0 74 2f 68 74 6d 6c 0d 0a 54 72 61 6e 73 66 65 72 t/html.. Transfer  
00b0 2d 45 6e 63 6f 64 69 6e 67 3a 20 63 68 75 6e 6b -Encodin g: chunk  
00c0 65 64 0d 0a 43 6f 6e 6e 65 63 74 69 6f 6e 3a 20 ed..Conn ection:  
00d0 6b 65 65 70 2d 61 6c 69 76 65 0d 0a 58 2d 50 6f keep-ali ve..X-Po  
00e0 7f 65 72 65 64 2d 42 79 3a 20 50 48 50 2f 35 2e wered-By : PHP/5.  
00f0 35 2e 33 38 0d 0a 53 65 74 2d 43 6f 6f 6b 69 65 5.38...Se t-Cookie  
0100 3a 20 50 48 50 53 45 53 53 49 44 3d 31 38 62 65 : PHPSES ID=18be  
0110 39 32 32 39 34 31 30 38 62 34 62 35 61 35 33 30 92294108 b4b5a530  
0120 38 35 37 33 35 65 39 63 64 33 38 65 3b 20 70 61 85735e9c d38e; pa  
0130 74 68 3d 2f 0d 0a 45 78 70 69 72 65 73 3a 20 54 th=/.Ex pires: T  
0140 68 75 2c 20 31 39 20 4e 6f 76 20 31 39 38 31 20 hu, 19 N ov 1981  
0150 30 38 3a 35 32 3a 30 30 20 47 4d 54 0d 0a 43 61 08:52:00 GMT..Ca  
0160 63 68 65 2d 43 6f 6e 74 72 6f 6c 3a 20 6e 6f 2d che-Cont rol: no-
```

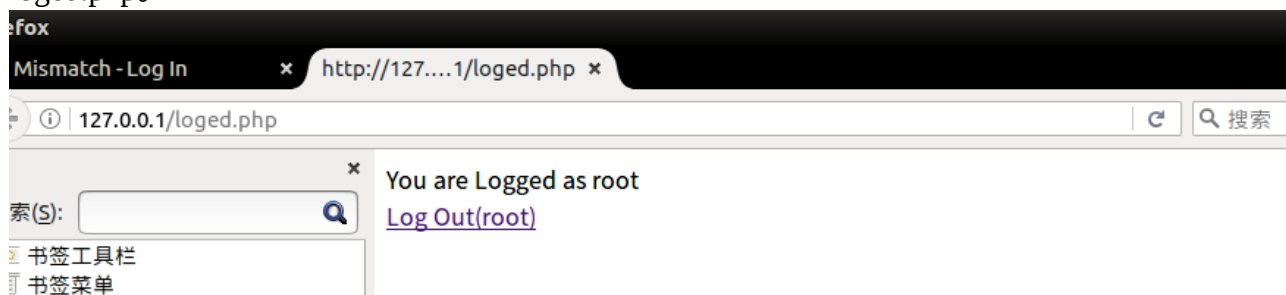
Frame (1472 bytes) De-chunked entity body (1011 bytes)

字节 246-309: Set-Cookie (http.set_cookie) 分组: 1939 · 已显示: 1939 (100.0%) 配置文件: Defau

数据库使用 mysql 数据库，存的时候对密码进行 sha 加密，conDb.php 负责数据库连接的信息

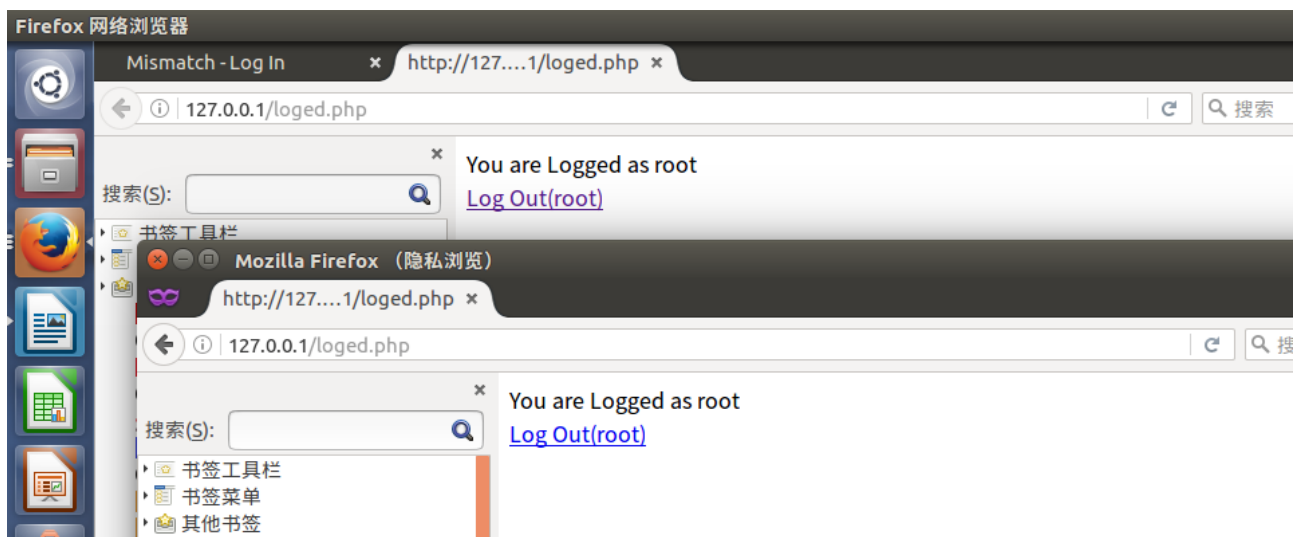
```
mysql> select * from user_info;  
+-----+-----+-----+-----+  
| username | user_id | password | name |  
+-----+-----+-----+-----+  
| liumeng | 12345678 | 7c222fb2927d828af22f592134e8932480637c0d | liumeng |  
| root | 12345678 | 7c4a8d09ca3762af61e59520943dc26494f8941b | liumeng |  
+-----+-----+-----+-----+  
2 rows in set (0.00 sec)  
  
mysql>
```

用户输入用户名和密码正确的后，设置 cookie 的内容，设置了 cookie 的 username 和 user_id。设置了 cookie 的时间。这时在旁白标签栏，输入登陆页面，直接进入跳转页面 logged.php。



点击 Log Out，销毁 session，然后删掉 cookie。

再次进入登陆页面，输入密码后才能进入。而同时打开了多个浏览器，会发现每打开一个浏览器会生成一个 PHPSESSION，



打开另一个浏览器后出现不一样的 sessionID

```

Accept-Encoding: gzip, deflate\r\n
Referer: http://127.0.0.1/login.php\r\n
Cookie: PHPSESSID=a4884da5b4ed2d6f740d585c69d4f942; user_id=12345678; username=root\r\n
Connection: keep-alive\r\n
3b 20 72 76 3a 35 30 2e 30 29 20 47 65 63 6b 6f ; rv:50.0) Gecko
2f 32 30 31 30 30 31 30 31 20 46 69 72 65 66 6f /2010010 1 Firefo
78 2f 35 30 2e 30 0d 0a 41 63 63 65 70 74 3a 20 x/50.0.. Accept:
74 65 78 74 2f 68 74 6d 6c 2c 61 70 70 6c 69 63 text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8.
61 74 69 6f 6e 2f 78 68 74 6d 6c 2b 78 6d 6c 2c ation/xhtml+xml,
61 70 70 6c 69 63 61 74 69 6f 6e 2f 78 6d 6c 3b applicat ion/xml;
71 3d 30 2e 39 2c 2a 2f 2a 3b 71 3d 30 2e 38 0d q=0.9,*/*;q=0.8.
0a 41 63 63 65 70 74 2d 4c 61 6e 67 75 61 67 65 .Accept- Language
3a 20 65 6e 2d 55 53 2c 65 6e 3b 71 3d 30 2e 35 : en-US, en;q=0.5
0d 0a 41 63 63 65 70 74 2d 45 6e 63 6f 64 69 6e ..Accept -Encodin
67 3a 20 67 7a 69 70 2c 20 64 65 66 6c 61 74 65 g: gzip, deflate
0d 0a 52 65 66 65 72 65 72 3a 20 68 74 74 70 3a ..Refere r: http:
2f 2f 31 32 37 2e 30 2e 30 2e 31 2f 6c 6f 67 49 //127.0. 0.1/logI
6e 2e 70 68 70 0d 0a 43 6f 6f 6b 69 65 3a 20 50 n.php..C ookie: P
48 50 53 45 53 53 49 44 3d 61 34 38 38 34 64 61 HPSESSID =a4884da
35 62 34 65 64 32 64 36 66 37 34 30 64 35 38 35 5b4ed2d6 f740d585
63 36 39 64 34 66 39 34 32 3b 20 75 73 65 72 5f c69d4f94 2; user_
69 64 3d 31 32 33 34 35 36 37 38 3b 20 75 73 65 id=12345 678; use
72 6e 61 6d 65 3d 72 6f 6f 74 0d 0a 43 6f 6e 6e rname=ro ot..Conn
65 63 74 69 6f 6e 3a 20 6b 65 65 70 2d 61 6c 69 ection: keep-ali
76 65 0d 0a 55 70 67 72 61 64 65 2d 49 6e 73 65 ve..Upgr ade-Inse
63 75 72 65 2d 52 65 71 75 65 73 74 73 3a 20 31 cure-Req uests: 1
0d 0a 0d 0a
HTTP Cookie (http.cookie), 85 字节

```

总结：

Session 是另一种记录客户状态的机制，不同的是 Cookie 保存在客户端浏览器中，而 Session 保存在服务器上。客户端浏览器访问服务器的时候，服务器把客户端信息以某种形式记录在服务器上。这就是 Session。客户端浏览器再次访问时只需要从该 Session 中查找该客户的状态就可以了。如果说 Cookie 机制是通过检查客户身上的“通行证”来确定客户身份的话，那么 Session 机制就是通过检查服务器上的“客户明细表”来确认客户身份。Session 相当于程序在服务器上建立的一份客户档案，客户来访的时候只需要查询客户档案表就可以了。

当程序需要为某个客户端的请求创建一个 session 的时候，服务器首先检查这个客户端的请求里是否已包含了一个 session 标识 - 称为 session id，如果已包含一个 session id 则说明以前已经为此客户端创建过 session，服务器就按照 session id 把这个 session 检索出来使用（如

果检索不到，可能会新建一个），如果客户端请求不包含 session id，则为此客户端创建一个 session 并且生成一个与此 session 相关联的 session id，session id 的值应该是一个既不会重复，又不容易被找到规律以伪造的字符串，这个 session id 将被在本次响应中返回给客户端保存

保存这个 session id 的方式可以采用 cookie，这样在交互过程中浏览器可以自动的按照规则把这个标识发回给服务器。一般这个 cookie 的名字都是类似于 PHPSESSID，而对 session 来说，除非程序通知服务器删除一个 session，否则服务器会一直保留，程序一般都是在用户做 log off 的时候发个指令去删除 session。然而浏览器从来不会主动在关闭之前通知服务器它将要关闭，因此服务器根本不会有机会知道浏览器已经关闭，之所以会有这种错觉，是大部分 session 机制都使用会话 cookie 来保存 session id，而关闭浏览器后这个 session id 就消失了，再次连接服务器时也就无法找到原来的 session。如果服务器设置的 cookie 被保存到硬盘上，或者使用某种手段改写浏览器发出的 HTTP 请求头，把原来的 session id 发送给服务器，则再次打开浏览器仍然能够找到原来的 session。恰恰是由于关闭浏览器不会导致 session 被删除，迫使服务器为 session 设置了一个失效时间，当距离客户端上一次使用 session 的时间超过这个失效时间时，服务器就可以认为客户端已经停止了活动，才会把 session 删除以节省存储空间。