PHPのセッションに関するよしなしごと

1.セッションとは?

PHPのセッションとは、いわゆる「HTTPセッション」のことであり、その実体はクッキーの一種で、目的はブラウザと鯖の間である一定期間、値を連携させるための仕掛けです。

全然分からないと思いますので、んでは実体であるところの「クッキー」から見ていきます。

2.クッキーとは?

ブラウザと鯖の間で値をやりとりするには、以前は「GET」と「POST」という二種類しかありませんでした。 どういうやり方かというと、ブラウザから一方的にGETあるいはPOSTというHTTPで値を鯖に知らせるやり方です。 んで、鯖はその値をパラメータとしてCGIを動かして、処理するわけです。

ところが、このやりかたの場合ブラウザ 鯖の一方通行だったわけです。

どーにかして、鯖 ブラウザ方向も使えるようにしたかったわけです。(HIDDENフォームを使ってCGI側で、ブラウザ側から送信されたGET/POST変数を埋め込む方法もある。けど、めんどい。)

そこで、HTTPのヘッダー部分を拡張し、「Set-Cookie」と「Cookie」という二つのフィールドを追加しました。「Set-Cookie」: 鯖 ブラウザ

「Cookie」: ブラウザ 鯖

これで、結構楽に鯖とブラウザの間で値を一定期間保持できるようになったわけです。

3.実際にクッキーのやりとりを見てみませう。

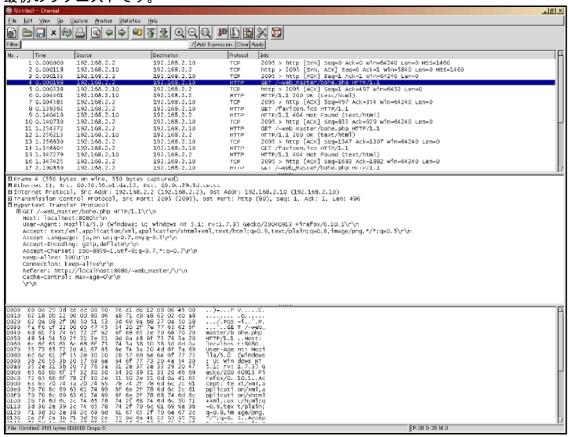
```
PHPY-スコード: bohe.php
<?php
ob_start();
if(!isset($HTTP_COOKIE_VARS["a"])) setcookie("a", 1);
else setcookie("a", ++$HTTP_COOKIE_VARS["a"]);
if(!isset($HTTP_COOKIE_VARS["b"])) setcookie("b", 1);
else setcookie("b", ++$HTTP_COOKIE_VARS["b"]);
print_r($HTTP_COOKIE_VARS);
?>
```

中身はなんて一ことはありません。クッキーにセットされてる値を1で初期化して、以降、+1してひょうじするだけです。

HTTPのやりとりにはEthereal をつかってパケットをキャプチャしてみました。フィルタリングは $tcp\ port\ 80$ で結構です。

実況中継します。

ブラウザ側 最初のリクエストです。



もうちょっと主要部分を拡大してみます。

```
4 0.000198 192.168.2.2
                                192.168.2.10
                                                                HTTP
                                                                         GET /~web_master/bohe.php HTTP/1.1
      5 0.000338
                    192.168.2.10
                                          192.168.2.2
                                                                          http > 2095 [ACK] Seq=1 Ack=497 wir
                                                                TCP
      6 0.004461
                    192.168.2.10
                                          192.168.2.2
                                                                HTTP
                                                                          HTTP/1.1 200 OK (text/html)
      7 0.004583
                    192.168.2.2
                                          192.168.2.10
                                                                          2095 > http [ACK] Seq=497 Ack=354 w
                                                                TCP.
      8 0.139392
                    192.168.2.2
                                          192.168.2.10
                                                                HTTP
                                                                          GET /favicon.ico HTTP/1.1
                                                                          HTTP/1.1 404 Not Found (text/html)
      9 0.140410
                    192.168.2.10
                                          192.168.2.2
                                                                HTTP
                                                                          2095 > http [ACK] Seq=833 Ack=929 w
     10 0.140730
                    192.168.2.2
                                          192.168.2.10
                                                                TCP
                    192.168.2.2
                                                                          GET /~web_master/bohe.php HTTP/1.1
     11 1.254372
                                          192.168.2.10
                                                                HTTP
     12 1.256213
                    192.168.2.10
                                          192.168.2.2
                                                                HTTP
                                                                          HTTP/1.1 200 OK (text/html)
                                                                          2095 > http [ACK] Seq=1347 Ack=1307
     13 1.256630
                    192.168.2.2
                                          192.168.2.10
                                                                TCP
     14 1.346604
                    192.168.2.2
                                                                         GET /favicon.ico HTTP/1.1
                                          192,168,2,10
                                                                HTTP
     15 1.347279
                    192.168.2.10
                                          192.168.2.2
                                                                HTTP
                                                                          HTTP/1.1 404 Not Found (text/html)
     16 1.347425
                                                                          2095 > http [ACK] Seq=1683 Ack=1882
                    192.168.2.2
                                          192.168.2.10
                                                                TCP
     17 2.190659
                    192.168.2.2
                                          192.168.2.10
                                                                HTTP
                                                                         GET /~web_master/bohe.php HTTP/1.1
⊞ Ethernet II, Src: 00:50:56:e1:da:12, Dst: 00:0c:29:3d:ce:cc
⊞ Internet Protocol, Src Addr: 192.168.2.2 (192.168.2.2), Dst Addr: 192.168.2.10 (192.168.2.10)
⊞ Transmission Control Protocol, Src Port: 2095 (2095), Dst Port: http (80), Seq: 1, Ack: 1, Len: 496
□ Hypertext Transfer Protocol
  Host: localhost:8080\r\n
    User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; rv:1.7.3) Gecko/20040913 Firefox/0.10.1\r\n
    Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/pnq,*/*;c
    Accept-Language: ja,en-us;q=0.7,en;q=0.3\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Charset: ISO-8859-1, utf-8; q=0.7, *; q=0.7\r\n
    Keep-Alive: 300\r\n
    Connection: keep-alive\r\n
    Referer: http://localhost:8080/~web_master/\r\n
    Cache-Control: max-age=0\r\n
    \r\n
```

普通のGETメソッドによるHTTPヘッダーです。 では、それに対する前掲PHPの応答です。

鯖応答

5 0.000338	192.168.2.10	192.168.2.2	TCP	http > 2095 [ACK] Seq=1 Ack=497				
6 0.004461	192.168.2.10	192.168.2.2	HTTP	HTTP/1.1 200 OK (text/html)				
7 0.004583	192.168.2.2	192.168.2.10	TCP	2095 > http [ACK] Seq=497 Ack=3				
8 0.139392	192.168.2.2	192.168.2.10	HTTP	GET /favicon.ico HTTP/1.1				
9 0.140410	192.168.2.10	192.168.2.2	HTTP	HTTP/1.1 404 Not Found (text/ht				
10 0.140730	192.168.2.2	192.168.2.10	TCP	2095 > http [ACK] Seq=833 Ack=9				
11 1.254372	192.168.2.2	192.168.2.10	HTTP	GET /~web_master/bohe.php HTTP/				
12 1.256213	192.168.2.10	192.168.2.2	HTTP	HTTP/1.1 200 OK (text/html)				
13 1.256630	192.168.2.2	192.168.2.10	TCP	2095 > http [ACK] Seq=1347 Ack=				
14 1.346604	192.168.2.2	192.168.2.10	HTTP	GET /favicon.ico HTTP/1.1				
15 1.347279	192.168.2.10	192.168.2.2	HTTP	HTTP/1.1 404 Not Found (text/ht				
16 1.347425	192.168.2.2	192.168.2.10	TCP	2095 > http [ACK] Seq=1683 Ack=				
17 2.190659	192.168.2.2	192.168.2.10	HTTP	GET /~web_master/bohe.php HTTP/				
<pre>⊞ Internet Protocol, Src Addr: 192.168.2.10 (192.168.2.10), Dst Addr: 192.168.2.2 (192.168.2.2) ⊞ Transmission Control Protocol, Src Port: http (80), Dst Port: 2095 (2095), Seq: 1, Ack: 497, Len: 353 □ Hypertext Transfer Protocol □ HTTP/1.1 200 OK\r\n Date: Mon, 18 Oct 2004 11:53:30 GMT\r\n Server: Apache/1.3.27 (Unix) mod_ruby/0.9.7 Ruby/1.6.4 mod_perl/1.26 DAV/1.0.3 PHP/4.2.3\r\n X=Powered=By: PHP/4.2.3\r\n Set-Cookie: a=1\r\n Set-Cookie: b=1\r\n Keep-Alive: timeout=15, max=100\r\n Connection: Keep-Alive\r\n Transfer=Encoding: chunked\r\n Content-Type: text/html; charset=euc-jp\r\n \r\n □ HTTP chunked response □ Data chunk (10 octets) □ Data chunk (last chunk) □ Line-based text data: text/html</pre>								

はい。「Set-Cookie」ヘッダーがしっかりと出現しています。対応するPHPコードがif(!isset(\$HTTP_COOKIE_VARS["a"])) setcookie("a", 1); と if(!isset(\$HTTP_COOKIE_VARS["b"])) setcookie("b", 1); です・・・たぶん。いえ、ちょっとこのスクリプト、実際のとは違うんです。このキャプチャで表示されてるスクリプト、消しちゃって・・・(汗)。んで、確かこんな感じだったなーみたいに作ったのが載せてるのです。でもでも、これ、1で初期化せずにいきなり飛んで2から始まったりするんです。・・・ま、まあ「理想的な」スクリプトだったら、このキャプチャ画面の通りになるということで。

ともかく、これでブラウザ側に「クッキー」がセットされたわけです。では、リロードさせてみます。 リロードするとき、ブラウザはどんなHTTPへッダーを鯖に送信するのか?

も一度ブラウザから・・・の送信。

し 反ファフリ	つ のた旧。			
3 0.000330	102.100.2.10	192.100.2.2	100	HCCP / 2003 [ACK] 304=1 ACK=4:
6 0.004461	192.168.2.10	192.168.2.2	HTTP	HTTP/1.1 200 OK (text/html)
7 0.004583	192.168.2.2	192.168.2.10	TCP	2095 > http [ACK] Seq=497 Ack=
8 0.139392	192.168.2.2	192.168.2.10	HTTP	GET /favicon.ico HTTP/1.1
9 0.140410	192.168.2.10	192.168.2.2	HTTP	HTTP/1.1 404 Not Found (text/h
10 0.140730	192.168.2.2	192.168.2.10	TCP	2095 > http [ACK] Seq=833 Ack=
11 1.254372	192.168.2.2	192.168.2.10	HTTP	GET /~web_master/bohe.php HTTF
12 1.256213	192.168.2.10	192.168.2.2	HTTP	HTTP/1.1 200 OK (text/html)
13 1.256630	192.168.2.2	192.168.2.10	TCP	2095 > http [ACK] Seq=1347 Ack
14 1.346604	192.168.2.2	192.168.2.10	HTTP	GET /favicon.ico HTTP/1.1
15 1.347279	192.168.2.10	192.168.2.2	HTTP	HTTP/1.1 404 Not Found (text/h
16 1.347425	192.168.2.2	192.168.2.10	TCP	2095 > http [ACK] Seq=1683 Ack
17 2.190659	192.168.2.2	192.168.2.10	HTTP	GET /~web_master/bohe.php HTTF

.⊞ Frame 11 (568 b∨	tes on wire, 568 by	tes captured)		

Host: localhost:8080\r\n

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; rv:1.7.3) Gecko/20040913 Firefox/0.10.1\r\n Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,

Accept-Language: ja,en-us;q=0.7,en;q=0.3\r\n

Accept-Encoding: gzip, deflate\r\n

Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n

Keep-Alive: 300\r\n

Connection: keep-alive\r\n

Referer: http://localhost:8080/~web_master/\r\n

Cookie: a=1; b=1\r\n Cache-Control: max-age=0\r\n

 $\r\n$

「Cookie」ヘッダーがちゃんとついてますね。 しかも、セミコロンで区切って「変数名=値」形式で送信されてます。 んではんでは、これに対する応答はどうなるのか?

[⊞] Ethernet II, Src: 00:50:56:é1:da:12, Dst: 00:0c:29:3d:ce:cc

[⊞] Internet Protocol, Src Addr: 192.168.2.2 (192.168.2.2), Dst Addr: 192.168.2.10 (192.168.2.10)

[⊞] Transmission Control Protocol, Src Port: 2095 (2095), Dst Port: http (80), Seq: 833, Ack: 929, Len:

[☐] Hypertext Transfer Protocol

鯖の応答二度目。

```
<del>.92.108.2.1</del>6
                                                                                    псср
       6 0.004461
                                                                                    HTTP/1.1 200 OK (text/html)
                      192.168.2.10
                                                192.168.2.2
                                                                         HTTP
       7 0.004583
                      192.168.2.2
                                                192.168.2.10
                                                                         TCP
                                                                                    2095 > http [ACK] Seq=497 Ack=354 win
      8 0.139392
                      192.168.2.2
                                                192.168.2.10
                                                                         HTTP
                                                                                    GET /favicon.ico HTTP/1.1
                                                                                    HTTP/1.1 404 Not Found (text/html)
      9 0.140410
                      192.168.2.10
                                                192,168,2,2
                                                                         HTTP
                                                                                    2095 > http [ACK] Seq=833 Ack=929 win
     10 0.140730
                      192.168.2.2
                                                192.168.2.10
                                                                         TCP
     11 1.254372
                      192.168.2.2
                                                192.168.2.10
                                                                         HTTP
                                                                                    GET /~web_master/bohe.php HTTP/1.1
     12 1.256213
                      192.168.2.10
                                                192.168.2.2
                                                                                    HTTP/1.1 200 OK (text/html)
                                                                         HTTP
                                                                                    2095 > http [ACK] Seq=1347 Ack=1307
     13 1.256630
                       192.168.2.2
                                                192.168.2.10
                                                                         TCP
     14 1.346604
                      192.168.2.2
                                                192.168.2.10
                                                                         HTTP
                                                                                    GET /favicon.ico HTTP/1.1
     15 1.347279
                      192.168.2.10
                                                192.168.2.2
                                                                                    HTTP/1.1 404 Not Found (text/html)
                                                                         HTTP
     16 1.347425
                                                                                    2095 > http [ACK] Seg=1683 Ack=1882 W
                      192.168.2.2
                                                192.168.2.10
                                                                         TCP
     17 2.190659
                      192,168,2,2
                                                192.168.2.10
                                                                         HTTP
                                                                                    GET /~web_master/bohe.php HTTP/1.1
■ Frame 12 (432 bytes on wire, 432 bytes captured)
⊞ Ethernet II, Src: 00:0c:29:3d:ce:cc, Dst: 00:50:56:e1:da:12
⊞ Internet Protocol, Src Addr: 192.168.2.10 (192.168.2.10), Dst Addr: 192.168.2.2 (192.168.2.2)
⊞ Transmission Control Protocol, Src Port: http (80), Dst Port: 2095 (2095), Seq: 929, Ack: 1347, Len: 378
☐ Hypertext Transfer Protocol
  ⊞ HTTP/1.1 200 OK\r\n
    Date: Mon, 18 Oct 2004 11:53:31 GMT\r\n
Server: Apache/1.3.27 (Unix) mod_ruby/0.9.7 Ruby/1.6.4 mod_perl/1.26 DAV/1.0.3 PHP/4.2.3\r\n
    X-Powered-By: PHP/4.2.3\r\n
     Set-Cookie: a=2\r\n
    Set-Cookie: b=2\r\n
    Keep-Alive: timeout=15, max=98\r\n
    Connection: Keep-Alive\r\n
    Transfer-Encoding: chunked\r\n
    Content-Type: text/html; charset=euc-jp\r\n
  ■ HTTP chunked response
□ Line-based text data: text/html
    Array
     (
          [a] => 1
         [b] => 1
      53 65 74 2d 43
0a 4b 65 65 70
65 6f 75 74 3d
0a 43 6f 6e 6e
                       6f 6f
2d 41
                                                    3d 32 0d
74 69 6d
                                      65 3a 20 62
76 65 3a 20
6d 61 78 3d
loofo
                                   69
                                                                  set-cook
                                                                            ie: b=2.
0100
                               6c
2c
74
                                   69
                                                                  .Keep-Al
                                                                            ive: tim
0110
                        31
                           35
                                   20
                                                        38 Od
                                                                  eout=15,
                                                                             max=98.
0120
                    6e 65
                           63
                                              3а
                                                    4b
                                          6e
                                                                  .Connect
                                                                            ion: Kee
0130
      70
          2d 41
                 6с
                    69
                        76
                           65
                               ٥d
                                   0a
                                       54
                                              61
                                                 6e
                                                        66 65
                                                                  p-Alive
                                                                            .Transfe
                                                                  r-Encodi ng: chun
      72
                           64
6f
                                              20
74
                                                    68
54
                                                        75 6e
79 70
0140
          2d 45
                бе
                    63 6f
                               69
                                   6e 67
                                          За
                                                 63
      6b 65 64
65 3a 20
                                       65
                                                                           tent-Typ
html; ch
                                                 2d
                0d
74
                       43
78
                               6e
2f
                                   74
lo1 50
                    0a
                                          6e
                                                                  ked..com
                           74
                                                 3b
                    65
                                   68
                                          6d
                                              60
                                                     20
                                                        63
                                                                 e: text/ html
arset=eu c-jp
0160
                                       74
                                                            68
      61
          72
             73
                 65
                    74
                        3d
                           65
                               75
                                       2d
                                              70
                                                 Οď
                                                     0a
0170
                                   63
                                          6а
                                                        0d 0a
             20
61
                0d
5d
                    0a
20
20
                        41
3d
                                          0a
20
                                                 0a
20
                                                     20
20
                           72
3e
                               72
20
29
                                       79
0a
0d
l0180
                                   61
                                                                       . Arr
0190
          5b
                                    31
01a0
```

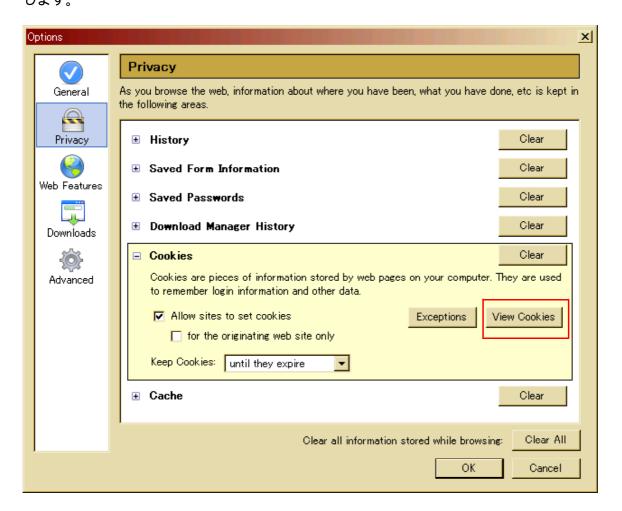
あー・・・何気に末尾の text/html が不穏な内容になってますが (だからPHPスクリプト、違うんですってばて ば。勘弁してえな)。

```
までも、一応
```

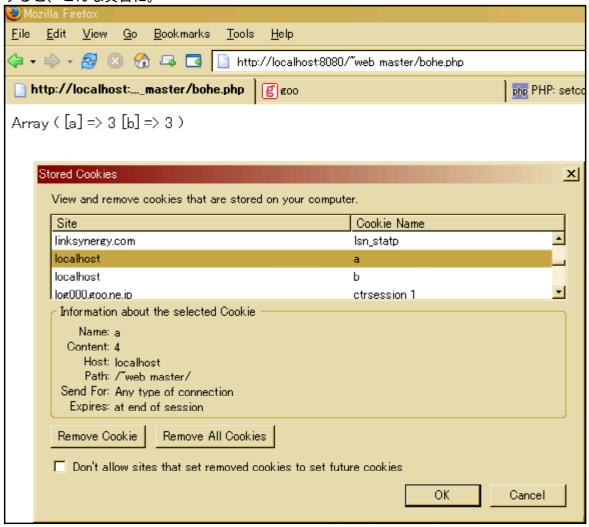
```
else setcookie("a", ++$HTTP_COOKIE_VARS["a"]);
```

else setcookie("b", ++\$HTTP_COOKIE_VARS["b"]); により、「クッキーはこれこれにしてね。」と Set-Cookie が新しい値で送信されています! で。結局、このクッキーはブラウザではどのようにして管理されているのでしょうか? 実際に実体としてどう管理されているのではなくて(ブラウザ依存だし)、ブラウザの備えるクッキー管理では どうなるのか、を見てみます。それくらいだったらどのブラウザも一緒だし。 FireFoxの場合、「Tools」 「Options」 「Privacy」の「+Cookies」を展開して「View Cookies」をクリック

します。



すると、こんな具合に。



こんな具合に、サイト毎に管理されます。ここで「Remove Cookie」とかすればブラウザ側で保持してたCookie の値が消えますので、次回リクエスト時には「Cookie」へッダーが無くなり、鯖からの「Set-Cookie」で改めてクッキーが再設定されるわけです。

いかがでしょう。

こうしてみると、要するにクッキーとはブラウザと鯖との間でHTTPへッダーを介して値をやりとりするものであり、最大の特徴は「値はブラウザ側で保持され(続け)る」点にあることが御納得頂けるかと思います。

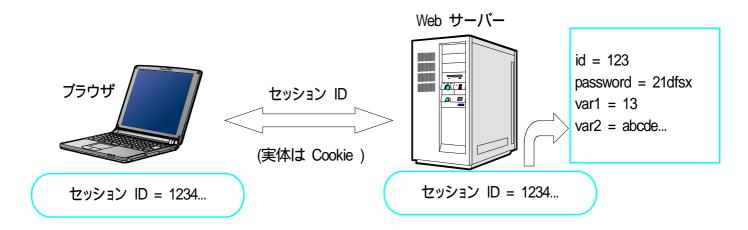
4. 再度セッションとは

んで、ですね。やがてクッキーは「ログインユーザーのユーザーIDやパスワード」を保存するのにも使われ出すわけです。これを使えば、うまい具合に一度入力されたユーザー情報を、ログイン情報としてGETやPOSTなど使わずに「恒久的に」ブラウザと鯖の間でやりとりできるわけですから。

もちろん、クッキーに制限時間をかけられることがHTTPのRFCにありますので、「タイムアウト」エラーなんかも実現できます。

ところがです。まずクッキーは基本的に「平文」です。盗聴しようと思えばいくらでもできてしまうわけです。 次に、ショッピングカートシステムなどをクッキーを使って実装しようと思うと「一変数につきーCookieへッダー」ですからCookie変数があっというまに増殖していってしまうわけです。

そこでようやく、PHPのセッション概念が登場します。これはまとめて言ってしまえば、「値はまとめて鯖側で保管。それを引き出すためのキー値をブラウザとCookieでやりとりする」という仕組みです。



まあ、通常はブラウザ側でセッションの中身を参照することは無いので。そもそもセッションの中身に基づいて HTMLをはき出すのが鯖側のCGIなりなんなりなので。こんな具合で良いわけです。

それでも「セッションID乗っ取り」はあり得るわけですが。根性出せば、の話ですが。

それはともかく、んでは

5.セッションIDの実況中継

を楽しんでみましょう。

今度のPHPはちゃんと動かした奴です(汗)。

```
<?php
session_start();

$HTTP_SESSION_VARS["hoge"]++;
printf("session_id = %s\u00e4n", session_id());
print "hoge=".\u00e4HTTP_SESSION_VARS["hoge"];
?>
```

ま、これもアクセスするたびに(リロードするたび)カウントアップしていく変数を表示するだけのサンプルです。では、行ってみましょう。キャプチャフィルタは前と同じ「tcp port 80」です。

ブラウザからの最初のリクエスト

4 0.002884	192.168.2.2	192.168.2.10	HTTP	GET /~web_master/hoge.php HTTP/1.1			
5 0.003063	192.168.2.10	192.168.2.2	TCP	http > 2097 [ACK] Seq=1 Ack=452 win=			
6 0.012097	192.168.2.10	192.168.2.2	HTTP	HTTP/1.1 200 OK (text/html)			
7 0.012203	192.168.2.2	192.168.2.10	TCP	2097 > http [ACK] Seq=452 Ack=563 Wi			
8 0.200216	192.168.2.2	192.168.2.10	HTTP	GET /favicon.ico HTTP/1.1			
9 0.201329	192.168.2.10	192.168.2.2	HTTP	HTTP/1.1 404 Not Found (text/html)			
10 0.201433	192.168.2.2	192.168.2.10	TCP	2097 > http [ACK] Seq=840 Ack=1138 w			
11 2.942847	192.168.2.2	192.168.2.10	HTTP	GET /~web_master/hoge.php HTTP/1.1			
12 2.944913	192.168.2.10	192.168.2.2	HTTP	HTTP/1.1 200 OK (text/html)			
13 2.945065	192.168.2.2	192.168.2.10	TCP	2097 > http [ACK] Seq=1343 Ack=1635			
14 3.038214	192.168.2.2	192.168.2.10	HTTP	GET /favicon.ico HTTP/1.1			
15 3.039016	192.168.2.10	192.168.2.2	HTTP	HTTP/1.1 404 Not Found (text/html)			
16 3.039181	192.168.2.2	192.168.2.10	TCP	2097 > http [ACK] Seq=1731 Ack=2210			
17 3.753213	192.168.2.2	192.168.2.10	HTTP	GET /~web_master/hoge.php HTTP/1.1			
<pre>Brame 4 (505 bytes on wire, 505 bytes captured) ⊞ Ethernet II, Src: 00:50:56:e1:da:12, Dst: 00:0c:29:3d:ce:cc ⊞ Internet Protocol, Src Addr: 192.168.2.2 (192.168.2.2), Dst Addr: 192.168.2.10 (192.168.2.10) ⊞ Transmission Control Protocol, Src Port: 2097 (2097), Dst Port: http (80), Seq: 1, Ack: 1, Len: 451 □ Hypertext Transfer Protocol ⊞ GET /~web_master/hoge.php HTTP/1.1\r\n Host: localhost:8080\r\n User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; rv:1.7.3) Gecko/20040913 Firefox/0.10.1\r\n Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q= Accept-Language: ja,en-us;q=0.7,en;q=0.3\r\n Accept-Encoding: gzip,deflate\r\n Accept-Encoding: gzip,deflate\r\n Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n Keep-Alive: 300\r\n Connection: keep-alive\r\n Cache-Control: max-age=0\r\n</pre>							

なんてこたない、普通のリクエストです。 んでは。

鯖応答一回目

E A AA2A62							
5 0.003063	192.168.2.10	192.168.2.2	TCP	http > 2097 [ACK] Seq=1 Ack=45			
6 0.012097	192.168.2.10	192.168.2.2	HTTP	HTTP/1.1 200 OK (text/html)			
7 0.012203	192.168.2.2	192.168.2.10	TCP	2097 > http [ACK] Seq=452 Ack=			
8 0.200216	192.168.2.2	192.168.2.10	HTTP	GET /favicon.ico HTTP/1.1			
9 0.201329	192.168.2.10	192.168.2.2	HTTP	HTTP/1.1 404 Not Found (text/h			
10 0.201433	192.168.2.2	192.168.2.10	TCP	2097 > http [ACK] Seq=840 Ack=			
11 2.942847	192.168.2.2	192.168.2.10	HTTP	GET /~web_master/hoge.php HTTP			
12 2.944913	192.168.2.10	192.168.2.2	HTTP	HTTP/1.1 200 OK (text/html)			
13 2.945065	192.168.2.2	192.168.2.10	TCP	2097 > http [ACK] Seq=1343 Ack			
14 3.038214	192.168.2.2	192.168.2.10	HTTP	GET /favicon.ico HTTP/1.1			
15 3.039016	192.168.2.10	192.168.2.2	HTTP	HTTP/1.1 404 Not Found (text/h			
16 3.039181	192.168.2.2	192.168.2.10	TCP	2097 > http [ACK] Seq=1731 Ack			
17 3.753213	192.168.2.2	192.168.2.10	HTTP	GET /~web_master/hoge.php HTTP			
<pre> ⊞ Internet Protocol, Src Addr: 192.168.2.10 (192.168.2.10), Dst Addr: 192.168.2.2 (192.168.2.2) ⊞ Transmission Control Protocol, Src Port: http (80), Dst Port: 2097 (2097), Seq: 1, Ack: 452, Len: 562 □ Hypertext Transfer Protocol □ HTTP/1.1 200 OK\r\n Date: Mon, 18 Oct 2004 12:05:02 GMT\r\n Server: Apache/1.3.27 (Unix) mod_ruby/0.9.7 Ruby/1.6.4 mod_perl/1.26 DAV/1.0.3 PHP/4.2.3\r\n Set-Cookie: PHPSESSID=65674e910caa58ef7a38fa8661bdb832; path=/\r\n Expires: Thu, 19 Nov 1981 08:52:00 GMT\r\n Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0\r\n Pragma: no-cache\r\n Keep-Alive: timeout=15, max=100\r\n Connection: Keep-Alive\r\n Transfer-Encoding: chunked\r\n Content-Type: text/html; charset=euc-jp\r\n \r\n #HTTP chunked response □ Line-based text data: text/html session_id = 65674e910caa58ef7a38fa8661bdb832 </pre>							

「Set-Cookie」に注目してください。普通の「Set-Cookie」ヘッダーとなんら代わりありません。ただ、変数名 が「PHPSESSID」という変数になってます。(続けてpathがあります。これも実はセッションと関連しますが・

で、実際のHTMLボディを表すデータ部を見てみましょう。これは、PHPスクリプトの

printf("session_id = %s\forall n", session_id());

print "hoge=".\$HTTP SESSION VARS["hoge"];

が出力している部分です。

session_id()でセッションIDを取得しているわけです。で、セッションIDすなわちクッキーの値、というわけで す。

二回目のブラウザリクエスト

6 0.012097	192.168.2.10	192.168.2.2	HTTP	HTTP/1.1 200 OK (text/html)
7 0.012203	192.168.2.2	192.168.2.10	TCP	2097 > http [ACK] Seq=452 Ack
8 0.200216	192.168.2.2	192.168.2.10	HTTP	GET /favicon.ico HTTP/1.1
9 0.201329	192.168.2.10	192.168.2.2	HTTP	HTTP/1.1 404 Not Found (text/
10 0.201433	192.168.2.2	192.168.2.10	TCP	2097 > http [ACK] Seq=840 Ack
11 2.942847	192.168.2.2	192.168.2.10	HTTP	GET /~web_master/hoge.php HTT
12 2.944913	192.168.2.10	192.168.2.2	HTTP	HTTP/1.1 200 OK (text/html)
13 2.945065	192.168.2.2	192.168.2.10	TCP	2097 > http [ACK] Seq=1343 Ac
14 3.038214	192.168.2.2	192.168.2.10	HTTP	GET /favicon.ico HTTP/1.1
15 3.039016	192.168.2.10	192.168.2.2	HTTP	HTTP/1.1 404 Not Found (text/
16 3.039181	192.168.2.2	192.168.2.10	TCP	2097 > http [ACK] Seq=1731 Ac
17 3.753213	192.168.2.2	192.168.2.10	HTTP	GET /~web_master/hoge.php HTT

⊞ Frame 11 (557 bytes on wire, 557 bytes captured)

- ⊞ Ethernet II, Src: 00:50:56:e1:da:12, Dst: 00:0c:29:3d:ce:cc
- ⊞ Internet Protocol, Src Addr: 192.168.2.2 (192.168.2.2), Dst Addr: 192.168.2.10 (192.168.2.10)
- ⊞ Transmission Control Protocol, Src Port: 2097 (2097), Dst Port: http (80), Seq: 840, Ack: 1138, Len:
- ☐ Hypertext Transfer Protocol
 - ⊞ GET /~web_master/hoge.php HTTP/1.1\r\n

Host: localhost:8080\r\n

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; rv:1.7.3) Gecko/20040913 Firefox/0.10.1\r\n Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png

 $\label{eq:Accept-Language: ja,en-us; q=0.7,en; q=0.3\\ $$ Accept-Encoding: gzip, deflate\\ $$ n$$

Accept-Charset: ISO-8859-1, utf-8; q=0.7, *; q=0.7\r\n

Keep-Alive: $300\r\n$

Connection: keep-alive\r\n

Cookie: PHPSESSID=65674e910caa58ef7a38fa8661bdb832\r\n

Cache-Control: max-age=0\r\n

\r\n

はい。今までのクッキーと同様に、先ほどのSet-Cookieで送られてきたクッキー値を使ってCookieを送信してま す。(path変数は空なので、送信しなかったようです) 中身の「hoge」という値は、ブラウザは関知してないようです。

鯖応答二度目

- MR / U - I - I - I				
6 0.012097	192.168.2.10	192.168.2.2	HTTP	HTTP/1.1 200 OK (text/html)
7 0.012203	192.168.2.2	192.168.2.10	TCP	2097 > http [ACK] Seq=452 Ac
8 0.200216	192.168.2.2	192.168.2.10	HTTP	GET /favicon.ico HTTP/1.1
9 0.201329	192.168.2.10	192.168.2.2	HTTP	HTTP/1.1 404 Not Found (text
10 0.201433	192.168.2.2	192.168.2.10	TCP	2097 > http [ACK] Seq=840 Ac
11 2.942847	192.168.2.2	192.168.2.10	HTTP	GET /~web_master/hoge.php HT
12 2.944913	192.168.2.10	192.168.2.2	HTTP	HTTP/1.1 200 OK (text/html)
13 2.945065	192.168.2.2	192.168.2.10	TCP	2097 > http [ACK] Seq=1343 A
14 3.038214	192.168.2.2	192.168.2.10	HTTP	GET /favicon.ico HTTP/1.1
15 3.039016	192.168.2.10	192.168.2.2	HTTP	HTTP/1.1 404 Not Found (text
16 3.039181	192.168.2.2	192.168.2.10	TCP	2097 > http [ACK] Seq=1731 A
17 3.753213	192.168.2.2	192.168.2.10	HTTP	GET /~web_master/hoge.php HT

⊞ Frame 12 (551 bytes on wire, 551 bytes captured)

- ⊞ Ethernet II, Src: 00:0c:29:3d:ce:cc, Dst: 00:50:56:e1:da:12
- ⊞ Internet Protocol, Src Addr: 192.168.2.10 (192.168.2.10), Dst Addr: 192.168.2.2 (192.168.2.2)
- ⊞ Transmission Control Protocol, Src Port: http (80), Dst Port: 2097 (2097), Seq: 1138, Ack: 1343, Le
- ☐ Hypertext Transfer Protocol

⊞ HTTP/1.1 200 OK\r\n

Date: Mon, 18 Oct 2004 12:05:05 GMT\r\n

Server: Apache/1.3.27 (Unix) mod_ruby/0.9.7 Ruby/1.6.4 mod_per1/1.26 DAV/1.0.3 PHP/4.2.3\r\n

X-Powered-By: PHP/4.2.3\r\n

Expires: Thu, 19 Nov 1981 08:52:00 GMT\r\n

Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0\r\n

Pragma: no-cache\r\n

Keep-Alive: timeout=15, max=98\r\n

Connection: Keep-Alive\r\n Transfer-Encoding: chunked\r\n

Content-Type: text/html; charset=euc-jp\r\n

\r\n

⊞ HTTP chunked response

□ Line-based text data: text/html

session_id = 65674e910caa58ef7a38fa8661bdb832

hoge=2

・・・どうでしょうか。「Set-Cookie」ヘッダーが消えてます。

これは単純で、逆に何で前のCookieの実験で毎度Set-Cookieがあったのかを考えればわかりやすいと思います。

何度も言いますが、クッキーの値はブラウザ側で保持します。

Set-Cookieというのは、ブラウザ側に「新しいクッキーの値」を報知するための鯖用(たぶん)のヘッダーです。んで、先ほどの例ではクッキーの値を毎度+1してました。だから、毎度新しい値をブラウザに知らせるため、Set-Cookieを送信していたわけです。

ところがセッションIDというのは、ある一定の期間(session)同じキー値を「ブラウザ側が鯖に報知」することにより、鯖側でブラウザの身元証明代わりに使います。ですから、セッションIDは一度「Set-Cookie」すれば良いわけです。

ブラウザが「Cookie」でセッションIDを送らなくなったとき。それは、ブラウザ側で(セッションID用の)クッキーの保持期限が過ぎたことを意味しますので、「タイムアウトエラー」みたく考えて(多くのアプリでは)ログインフォームの再表示などを行うわけです。

んでは、この段階でブラウザ側ではどんなふうにセッションIDが「クッキーとして」管理されているのでしょうか?



普通のクッキーと変わりません。

こんな具合で、PHPのセッションの実体はクッキーであることが納得頂けたかと・・・って、まだもう一つ重要なの忘れてました。

これ、肝心の「hoge」という変数の値はいったいどこに保存されてるんでしょうか? ブラウザ側ではなく、鯖側にあるのは分かるのですが・・・

その秘密は、いつもの <?php phpinfo(); ?>を動かせば分かります。

6. セッションの実体(ファイルバージョン)

session

Session Support enabled

Directive	Local Value	Master Value
session.auto_start	Off	Off
session.cache_expire	180	180
session.cache_limiter	nocache	nocache
session.cookie_domain	no value	no value
session.cookie_lifetime	0	0
session.cookie_path	1	1
session.cookie_secure	Off	Off
session.entropy_file	no value	no value
session.entropy_length	0	0
session.gc_maxlifetime	1440	1440
session.gc_probability	1	1
session.name	PHPSESSID	PHPSESSID
session.referer_check	no value	no value
session.save_handler	files	files
session.save_path	/tmp	/tmp
session.serialize_handler	php	php
session.use_cookies	On	On
session.use_trans_sid	0	0

「session.save_path」という、いかにもそれっぽいのがPHP変数としてリストアップされています。では、早速見てみましょう。

ありました。一目でわかります。FireFoxのクッキー管理画面やsession_id()の出力で得られた「セッションID」と同じ値で構成された名前です。

Г	OLAX	z root	root	4036	JUI	10	20:47	orbit-root/
	-rw	1 nobody	nobody	9	Oct.	18	21:12	sess_2c4a102c680bede7847a153ff93315dd
ı	-rw	1 nobody	nobody	9	Oct.	18	21:08	sess_65674e910caa58ef7a38fa8661bdb832
ı	-rw	1 nobody	nobody	9	Oct.	18	20:33	sess_a17e6e187ce3043ad13c17c547d02dcc
	-rw	1 nobody	nobody	9	Oct.	18	21:10	sess_ada53343e1ff36a980056f239710188c
ı	-rw	1 nobody	nobody	52	Oct.	8	00:42	sess_bcebbcaa0f40131c3c1550a5c5133c0e
ı	-rw	1 nobody	nobody	44	Oct.	16	09:45	sess_cc3c729b02b8474d2371458af3985843
L	-rw	1 nobody	nobody	9	0ct	18	21:01	sess_eb2a3a6eeaa9b5180aa5fbc012e50d0f

ユーザーもnobody。間違いありません。中身を見てみましょう。

[root@hogasumi /tmp]# more sess_65674e910caa58ef7a38fa8661bdb832 hoge|i:4; [root@hogasumi /tmp]# more sess_2c4a102c680bede7847a153ff93315dd hoge|i:4;

ようやく見えてきました。

変数名|変数の型:値;

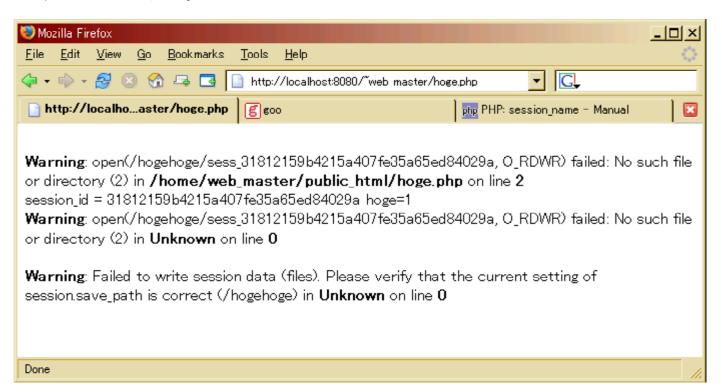
で一つの変数を表しているようです。

ではここでちょっと遊んでみます。save_pathを変更して存在しないディレクトリにし、セッションファイルが 作成できないようにしてみましょう。PHPスクリプトと同じディレクトリに .htaccess をおき、中身を以下のよ うにしてみます。

もちろん、/hogehoge は存在しないディレクトリです。 これでphpinfoの出力は・・・

_		
session.save_path	/hogehoge	/tmp

大丈夫そうです。左がローカル設定、右が全体の基本設定です。 では、アクセスしてみます。



見事!予想通り、いかにも「セッションファイルが作れないからセッションが取り出せないよう」って具合のワーニングメッセージです。

実は、Windows用のPHPパッケージのデフォルトのphp.iniでは、このsession.save_pathが正しく設定されていません。(まあ、Windowsではテンポラリフォルダの位置がUnixほど定型的ではないので・・・とか。)したがって、Windows上でPHPのセッション機構を利用す

るときは、まずはphp.iniのsession.save_pathでフォルダを設定する必要があることをよ く注意していください。

7.セッションIDのクッキー変数名を変更する

さて。セッションIDが保存されるクッキー変数名、PHPSESSIDっていかにも味気ないです。 同じ鯖上でことなるシステムが使うことを想定すると、PHPSESSIDという変数名、いかにも変更できるっぽいで す。

んで、実際に変更できます。session_name()で設定できます。

```
<?php
session_name("SampleSession");
session_start();
$HTTP_SESSION_VARS["hoge"]++;
printf("session_id = %s\forall n", session_id());
print "hoge=".$HTTP_SESSION_VARS["hoge"];
```

鯖の初回 Set-Cookie

?>

4 0.000280	192.168.2.2	192.168.2.10	HTTP	GET /~web_master/hoge2.php H
5 0.000386	192.168.2.10	192.168.2.2	TCP	http > 2102 [ACK] Seq=1 Ack=
6 0.004643	192.168.2.10	192.168.2.2	HTTP	HTTP/1.1 200 OK (text/html)
7 0.004781	192.168.2.2	192.168.2.10	TCP	2102 > http [ACK] Seq=453 Ad
8 0.146568	192.168.2.2	192.168.2.10	HTTP	GET /favicon.ico HTTP/1.1
9 0.147667	192.168.2.10	192.168.2.2	HTTP	HTTP/1.1 404 Not Found (text
10 0.148047	192.168.2.2	192.168.2.10	TCP	2102 > http [ACK] Seq=845 Ad
11 0.613531	192.168.2.2	192.168.2.10	HTTP	GET /~web_master/hoge2.php H
12 0.615547	192.168.2.10	192.168.2.2	HTTP	HTTP/1.1 200 OK (text/html)
13 0.616054	192.168.2.2	192.168.2.10	TCP	2102 > http [ACK] Seq=1353 A
14 0.706746	192.168.2.2	192.168.2.10	HTTP	GET /favicon.ico HTTP/1.1
15 0.707429	192.168.2.10	192.168.2.2	HTTP	HTTP/1.1 404 Not Found (text
16 0.707728	192.168.2.2	192.168.2.10	TCP	2102 > http [ACK] Seq=1745 A
17 1.085360	192.168.2.2	192.168.2.10	HTTP	GET /~web_master/hoge2.php H

```
⊞ Frame 6 (620 bytes on wire, 620 bytes captured)
```

- Ethernet II, Src: 00:0c:29:3d:ce:cc, Dst: 00:50:56:e1:da:12
- ⊞ Internet Protocol, Src Addr: 192.168.2.10 (192.168.2.10), Dst Addr: 192.168.2.2 (192.168.2.2)
- ⊞ Transmission Control Protocol, Src Port: http (80), Dst Port: 2102 (2102), Seq: 1, Ack: 453, Len:
- ☐ Hypertext Transfer Protocol

Date: Mon, 18 Oct 2004 12:12:25 GMT\r\n

Server: Apache/1.3.27 (Unix) mod_ruby/0.9.7 Ruby/1.6.4 mod_perl/1.26 DAV/1.0.3 PHP/4.2.3\r\n

× Powered By: PHP/4.2.3\r\n

Set-Cookie: SampleSession=2c4a102c680bede7847a153ff93315dd; path=/\r\n

Expires: Thu, 19 Nov 1981 08:52:00 GMT\r\n Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0\r\n

Pragma: no-cache\r\n

Keep-Alive: timeout=15, max=100\r\n

Connection: Keep-Alive\r\n Transfer-Encoding: chunked\r\n

Content-Type: text/html; charset=euc-jp\r\n

\r\n

⊞ HTTP chunked response

□ Line-based text data: text/html

session_id = 2c4a102c680bede7847a153ff93315dd

hoge=1

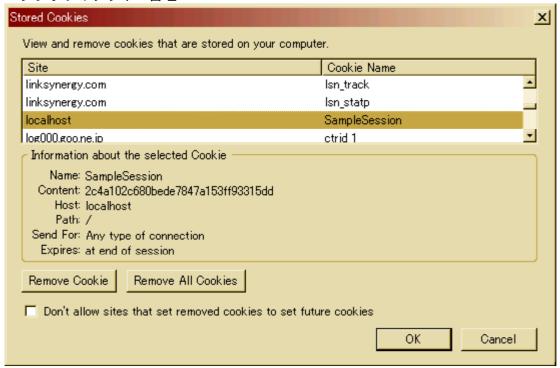
見事にセッションクッキーの変数名が変更されています。

ブラウザのリクエスト(Set-Cookie後)

4 0.00028		192.168.2.10	HTTP	GET /~web_master/hoge2.php HTTP/1.1					
5 0.00038		192.168.2.2	TCP	http > 2102 [ACK] Seq=1 Ack=453 win=6					
6 0.00464		192.168.2.2	HTTP	HTTP/1.1 200 OK (text/html)					
7 0.00478		192.168.2.10	TCP	2102 > http [ACK] Seq=453 Ack=567 Wir					
8 0.14656		192.168.2.10	HTTP	GET /favicon.ico HTTP/1.1					
9 0.14766		192.168.2.2	HTTP	HTTP/1.1 404 Not Found (text/html)					
10 0.14804		192.168.2.10	TCP	2102 > http [ACK] Seq=845 Ack=1142 Wi					
11 0.61353		192.168.2.10	HTTP	GET /~web_master/hoge2.php HTTP/1.1					
12 0.61554		192.168.2.2	HTTP	HTTP/1.1 200 OK (text/html)					
13 0.61605		192.168.2.10	TCP	2102 > http [ACK] Seq=1353 Ack=1639 w					
14 0.70674		192.168.2.10	HTTP	GET /favicon.ico HTTP/1.1					
15 0.70742		192.168.2.2	HTTP	HTTP/1.1 404 Not Found (text/html)					
16 0.70772		192.168.2.10	TCP	2102 > http [ACK] Seq=1745 Ack=2214 v					
17 1.08536	0 192.168.2.2	192.168.2.10	HTTP	GET /~web_master/hoge2.php HTTP/1.1					
	bytes on wire, 562 by								
		?, Dst: 00:0c:29:3d:ce:		2 4 6 0 2 4 0 64 0 2 4 0 0 2 4 0 0					
				2.168.2.10 (192.168.2.10)					
		Purt: 2102 (2102), DSt	Pont: necp	(80), Seq: 845, Ack: 1142, Len: 508					
☐ Hypertext Trai	nsier Protocol Master/hoge2.php HTTP/	7 7\p\p							
	host:8080\r\n	1.17/1/11							
		. II. Windows NT 5 1. m	1 7 3) coc	ko/20040913 Firefox/0.10.1\r\n					
	uage: ja,en-us;q=0.7,			q=0.9,text/plain;q=0.8,image/png,*/*;q=0					
	Accept-Encoding: gzip, deflate\r\n								
	Accept-Charset: ISO-8859-1, utf-8; q=0.7, *; q=0.7\r\n								
	Keep-Alive: 300\r\n								
	Connection: keep-alive\r\n Cookie: SampleSession=2c4a102c680bede7847a153ff93315dd\r\n								
	nl: max-age=0\r\n	opede,04/91131133311da	(1 (1)						
\r\n	max=age=u v vi								
VVI									

ちゃんとセッションクッキー変数名が取れているようです。

ブラウザのクッキー管理



大丈夫そうです。

8.HTML出力バッファリング(output_buffering, ob_start())

さて、PHPでは header関数や setcookie, session_start などでHTTPへッダーの要素を出力できます。 できるのですが・・・もしも。もしも、途中でエラーなどが起こって、HTTPへッダーを出力する「前に」無関係なエラー文字列や、HTMLコンテンツが出力されたらどうなるのでしょうか?

ちょっとやってみましょう。

まずは、正常系です。

<?php
header("Custom-Header: Sample");
print "body contents";
?>

これにアクセスしたときの、鯖のレスポンスを下に示します。

		–							
14 2.973823	192.168.2.10	192.168.2.2	HTTP	HTTP/1.1 200 OK (text/html)					
15 2.973930	192.168.2.2	192.168.2.10	TCP	2126 > http [ACK] Seq=1160 Ack=149					
16 11.083273	192.168.2.2	192.168.2.10	HTTP	GET /favicon.ico HTTP/1.1					
7									
□ Enamp 14 (200 by # s	s on using 200 by	tos santumed)	******						
⊞ Frame 14 (398 byte			2						
		, Dst: 00:50:56:e1:da:1		03 160 3 3 (103 160 3 3)					
				192.168.2.2 (192.168.2.2)					
		Port: nttp (80), DSt Po	rt: 2126 (2	2126), seq: 1152, Ack: 1160, Len: 34					
⊟ Hypertext Transfer									
⊞ HTTP/1.1 200 OK									
	ct 2004 14:24:00 G								
		.ruby/0.9.7 Ruby/1.6.4 r	mod_perl/1.:	26 DAV/1.0.3 PHP/4.2.3\r\n					
X Powered By: P									
Custom-Header: :									
Keep-Alive: time	eout=15, max=9 8\r\	,n							
Connection: Kee	p-Alive\r\n								
Transfer-Encodi	ng: chunked\r\n								
Content-Type: to	ext/html; charset=	euc-jp\r\n							
\r\n									
#HTTP chunked response									
□ Line-based text da	ta: text/html								
body contents	,								

Webブラウザ出力を下に示します。

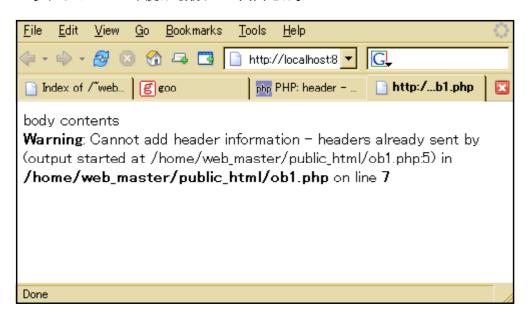


まずはスクリプト通り、「Custom-Header: Sample」というHTTPヘッダーが出力され、次に通常のHTMLボディコンテンツに相当する"body contents"文字列が出力され、ブラウザ画面にも表示されています。

では、順番を入れ替えてみます。

```
<?php
print "body contents";
header("Custom-Header: Sample");
?>
```

こうすると・・・今度は最初にWeb画面をば。



はい。PHP側で、「ヲイヲイ、HTMLコンテンツを送信した後にHTTPヘッダー送信されても困るんだよ、チミ。」ってな具合の警告が表示されちゃいます。

実際のHTTPパケットはどうなったのでしょうか?

35 24.354920 192.168.2.10 192.168.2.2 192.168.2.10 TCP 2126 > http [ACK] Seq=3689 / 37 24.466325 192.168.2.2 192.168.2.10 HTTP GET /favicon.ico HTTP/1.1 38 24.467052 192.168.2.10 192.168.2.2 HTTP HTTP/1.1 404 Not Found (text 39 24.467350 192.168.2.2 192.168.2.10 TCP 2126 > http [ACK] Seq=4025 / 40 41.203381 192.168.2.10 192.168.2.2 TCP http > 2126 > http [ACK] Seq=4025 / 40 41.203381 192.168.2.10 192.168.2.2 TCP http > 2126 FIN. ACK] Seq=6					 	
37 24.466325 192.168.2.2 192.168.2.10 HTTP GET /favicon.ico HTTP/1.1 38 24.467052 192.168.2.10 192.168.2.2 HTTP HTTP/1.1 404 Not Found (text 39 24.467350 192.168.2.2 192.168.2.10 TCP 2126 > http [ACK] seq=4025 / 40 41.203381 192.168.2.10 192.168.2.2 TCP http > 2126 FIN. ACK] seq=6	35 24.354920	192.168.2.10	192.168.2.2	HTTP		
37 24.466325 192.168.2.2 192.168.2.10 HTTP GET /favicon.ico HTTP/1.1 38 24.467052 192.168.2.10 192.168.2.2 HTTP HTTP/1.1 404 Not Found (text 39 24.467350 192.168.2.2 192.168.2.10 TCP 2126 > http [ACK] seq=4025 / 40 41.203381 192.168.2.10 192.168.2.2 TCP http > 2126 FIN. ACK] seq=6	36 24.355026	192.168.2.2	192.168.2.10	TCP	2126 > http [ACK] Seq=3689 A	
39 24.467350 192.168.2.2 192.168.2.10 TCP 2126 > http [ACK] seq=4025 A 40 41.203381 192.168.2.10 192.168.2.2 TCP http > 2126 [FIN. ACK] seq=6025 A 40 41.203381 192.168.2.10 192.168.2.2 TCP http > 2126 [FIN. ACK] seq=603	37 24.466325	192.168.2.2	192.168.2.10	HTTP		
39 24.467350 192.168.2.2 192.168.2.10 TCP 2126 > http [ACK] seq=4025 A 40 41.203381 192.168.2.10 192.168.2.2 TCP http > 2126 [FIN. ACK] seq=6025 A 40 41.203381 192.168.2.10 192.168.2.2 TCP http > 2126 [FIN. ACK] seq=603	38 24.467052	192.168.2.10	192.168.2.2	HTTP	HTTP/1.1 404 Not Found (text	
### 40 41.203381 192.168.2.10 192.168.2.2 TCP http > 2126 \[FIN. ACK] seq=6 \] ###################################	39 24.467350	192.168.2.2	192.168.2.10	TCP		
<pre>⊞ Ethernet II, Src: 00:0c:29:3d:ce:cc, Dst: 00:50:56:e1:da:12 ⊞ Internet Protocol, Src Addr: 192.168.2.10 (192.168.2.10), Dst Addr: 192.168.2.2 (192.168.2.2) ⊞ Transmission Control Protocol, Src Port: http (80), Dst Port: 2126 (2126), Seq: 4946, Ack: 3689, L □ Hypertext Transfer Protocol</pre>	40 41.203381	192.168.2.10	192.168.2.2	TCP	http > 2126 FIN. ACK1 Sea=6	
<pre>⊞ Ethernet II, Src: 00:0c:29:3d:ce:cc, Dst: 00:50:56:e1:da:12 ⊞ Internet Protocol, Src Addr: 192.168.2.10 (192.168.2.10), Dst Addr: 192.168.2.2 (192.168.2.2) ⊞ Transmission Control Protocol, Src Port: http (80), Dst Port: 2126 (2126), Seq: 4946, Ack: 3689, L □ Hypertext Transfer Protocol</pre>	ব					
<pre>⊞ Ethernet II, Src: 00:0c:29:3d:ce:cc, Dst: 00:50:56:e1:da:12 ⊞ Internet Protocol, Src Addr: 192.168.2.10 (192.168.2.10), Dst Addr: 192.168.2.2 (192.168.2.2) ⊞ Transmission Control Protocol, Src Port: http (80), Dst Port: 2126 (2126), Seq: 4946, Ack: 3689, L □ Hypertext Transfer Protocol</pre>	N					
<pre>⊞ Internet Protocol, Src Addr: 192.168.2.10 (192.168.2.10), Dst Addr: 192.168.2.2 (192.168.2.2) ⊞ Transmission Control Protocol, Src Port: http (80), Dst Port: 2126 (2126), Seq: 4946, Ack: 3689, Louis Entrology</pre>	⊞Frame 35 (584 bytes on wire, 584 bytes captured)					
<pre>⊞ Internet Protocol, Src Addr: 192.168.2.10 (192.168.2.10), Dst Addr: 192.168.2.2 (192.168.2.2) ⊞ Transmission Control Protocol, Src Port: http (80), Dst Port: 2126 (2126), Seq: 4946, Ack: 3689, Louis Entrology</pre>	⊞ Ethernet II. Src: 00:0c:29:3d:ce:cc. Dst: 00:50:56:e1:da:12					
<pre>⊞ Transmission Control Protocol, Src Port: http (80), Dst Port: 2126 (2126), Seq: 4946, Ack: 3689, Lot □ Hypertext Transfer Protocol</pre>						
□ Hypertext Transfer Protocol						
⊞HTTP/1.1 200 OK\r\n Date: Mon, 18 Oct 2004 14:24:22 GMT\r\n Server: Apache/1.3.27 (Unix) mod_ruby/0.9.7 Ruby/1.6.4 mod_perl/1.26 DAV/1.0.3 PHP/4.2.3\r\n X-Powered-By: PHP/4.2.3\r\n Keep-Alive: timeout=15, max=91\r\n Connection: Keep-Alive\r\n Transfer-Encoding: chunked\r\n Content-Type: text/html; charset=euc-jp\r\n \r\n ⊞HTTP chunked response □Line-based text data: text/html body contents body contents HTTP Data Data Data Data Data Data						
Date: Mon, 18 Oct 2004 14:24:22 GMT\r\n Server: Apache/1.3.27 (Unix) mod_ruby/0.9.7 Ruby/1.6.4 mod_perl/1.26 DAV/1.0.3 PHP/4.2.3\r\n X-Powered-By: PHP/4.2.3\r\n Keep-Alive: timeout=15, max=91\r\n Connection: Keep-Alive\r\n Transfer-Encoding: chunked\r\n Content-Type: text/html; charset=euc-jp\r\n \r\n B HTTP chunked response B Line-based text data: text/html body contents br />						
Server: Apache/1.3.27 (Unix) mod_ruby/0.9.7 Ruby/1.6.4 mod_perl/1.26 DAV/1.0.3 PHP/4.2.3\r\n X-Powered-By: PHP/4.2.3\r\n Keep-Alive: timeout=15, max=91\r\n Connection: Keep-Alive\r\n Transfer-Encoding: chunked\r\n Content-Type: text/html; charset=euc-jp\r\n \r\n B HTTP chunked response B Line-based text data: text/html body contents br						
X-Powered-By: PHP/4.2.3\r\n Keep-Alive: timeout=15, max=91\r\n Connection: Keep-Alive\r\n Transfer-Encoding: chunked\r\n Content-Type: text/html; charset=euc-jp\r\n \r\n B HTTP chunked response B Line-based text data: text/html body contents br						
Keep-Alive: timeout=15, max=91\r\n Connection: Keep-Alive\r\n Transfer-Encoding: chunked\r\n Content-Type: text/html; charset=euc-jp\r\n \r\n ⊞ HTTP chunked response ⊟ Line-based text data: text/html body contents 						
Connection: Keep-Alive\r\n Transfer-Encoding: chunked\r\n Content-Type: text/html; charset=euc-jp\r\n \r\n ⊞ HTTP chunked response ⊟ Line-based text data: text/html body contents 						
Transfer-Encoding: chunked\r\n Content-Type: text/html; charset=euc-jp\r\n \r\n ⊞ HTTP chunked response ⊟ Line-based text data: text/html body contents 						
Content-Type: text/html; charset=euc-jp\r\n \r\n						
\r\n	Transfer-Encoding: chunked\r\n					
⊞ HTTP chunked response ⊟ Line-based text data: text/html body contents 	Content-Type: text/html; charset=euc-jp\r\n					
□ Line-based text data: text/html body contents 	\r\n					
□ Line-based text data: text/html body contents 	■ HTTP chunked response					
body contents 						
Tooman ning 7 by. Cannot add neader information - neaders an eady sent by (output stafted at 700m			ler information - beader	es already s	ent by Coutnut started at /bor	
	<	. Cammoc add mead	iei iiiioiimacioni – neauei	s an eauy s	ent by Courbor Started at Anom	

・・・「Custom-Header」など影も形も無くなりました。

このように、ヘッダー出力関連(代表的な関数だと header, setcookie, session_start, session_name)のPHP 関数は、呼び出す前に通常出力が行われてはなりません。

が。

よーくこの一文を読んでみてください。「出力が」行われてはならないのです。なぜわざわざ「出力処理」(つまりエラーメッセージ出力とか print 系諸々)と書かずに、「出力」と書くのか。

これが、「出力バッファリング」という裏技(いえ、実際は裏でも何でも無いですが。まあ初心者向けの解説で

はまず出てこない・・・ん?最近の解説は質がよいから、出てくるかも?)があるわけです。

「出力バッファリング」とは、print 系やエラーメッセージ出力系の、「出力処理」によって出力された文字列をPHPの内部バッファにためておき、HTTPヘッダーが全部出力し終わってから実際にパケットとして送信してくれる便利な機能です。

mb_string系の文字コード変換機能も、この「出力バッファリング」を利用してます。

実際にどう使うのか?二種類、方法があります。

ob_start()関数を使う。

<?php

ob_start();

print "body contents";

header("Custom-Header: Sample");

?>

ob_start()が実行された時点で出力バッファリングが始まります。出力バッファをクリアしたいときは、ob_flush()を使うそうです(未確認)。というわけで、使い方によってはすんごい便利そう。

output_buffering PHP環境変数を "on" にセットする。

php.iniを書き換える手もありますが、結構影響範囲が大きいと思います。そういうときは、先ほども登場した . htaccess ファイルを用いた局所化を行います。

これで、.htaccess と同じディレクトリにあるPHPスクリプト全部で出力バッファリングが有効化されます。

このいずれかの方法で、間違いなく、Warning は消えて「Custom-Header」も復活するはずです。

9. 載せきれなかったヒントとか、おまけとか。

さて、同じセッション認証ライブラリを用いているシステムが同じドメイン内で動いていたりするとき、下手すると、「同じ鯖で同じセッションクッキー変数名を使ってしまう」みたいになりかねません。たとえば、http://www.xxxx.com/webapp1/というPHPアプリケーションと、http://www.xxx.com/webapp2/というPHPアプリケーションとではセッションクッキー変数名を分ける必要が出てくるわけです。

そのための識別要素が、「path」要素です。

・鯖レスポンスより

X-Powered-By: PHP/4.2.3\r\n Set-Cookie: PHPSESSID=65674e910caa58ef7a38fa8661bdb832; path=/\r\n Expires: Thu. 19 Nov 1981 08:52:00 GMT\r\n

ブラウザ画面より

Information about the selected Cookie

Name: SampleSession

Content: 2c4a102c680bede7847a153ff93315dd

Host: localhost
Path: /
Path: /
Send For: Any type of connection

Expires: at end of session

・http://jp2.php.net/manual/ja/function.session-set-cookie-params.phpより

session_set_cookie_params

(PHP 4, PHP 5)

session_set_cookie_params -- セッションクッキーバラメータを設定する

説明

void **session_set_cookie_params** (int lifetime [, string path [, string domain [, bool secure]]])

pathをうまく使いこなせば、セッションの識別に相当幅が出てくることでしょう。

これまで見てきたとおりセッションに関してはPHPの文法知識だけでなく、HTTP(Hyper Text Transport Protoco I)についても知っておく必要があります。PHPだけではなく、Webアプリケーション開発にはHTTPやTCP/IPの知識が欠かせません。何かトラブった時に、これらの知識やネットワーク関連のツール(特にパケットキャプチャソフト)があると非常に助かると思います。

(Webアプリ開発者にとって)ネットワーク関連は勉強しておいて損になることはありません!

最後に、クッキーやセッション関連の解説サイトを紹介しておきます。

http://www.rfs.jp/sitebuilder/perl/03/04.html

・・・Cookieの基本的な仕様が日本語で掲載されています。

http://jp2.php.net/manual/ja/ref.session.php

・・・PHPのセッション管理の概要が書かれています。まず、これを読むことをおすすめします。

http://jp2.php.net/manual/ja/ref.http.php

・・・クッキーやHTTPヘッダー関連の関数リファレンスです。

http://www.w3.org/Protocols/rfc2616/rfc2616

・・・header関数のリファレンスから飛べます。HTTP/1.1(最新版)のRFC。HTTPの全てはここに(多分)詰まってる。

10.忘れてた、も一つちょー重要なおまけ。

デフォルトの(敢えて「デフォルト」と付けるところがミソ)セッション保存形式には、実はいろいろと困る点があります。

第一に、何よりも session.save_path から容易に保存ディレクトリが取得でき、しかもしかも、平文なので容易く他のセッションデータがみれてしまうこと。

第二に、セッションの有効期限と関連して、ファイルシステムのタイムスタンプにまつわる話。というのは、ファイルシステムによっては「最終アクセス時刻」がまともに取得できないものもあり、有効期限の厳密な運用に 差し障りが出てくるからです。

どうしてもデフォルトのファイルベースでは、平文でしかも設定値から容易くファイルの保管場所が知れてしまうと言うこまった点があります。(sesison_save_path()で変更可能であるし、session.save_pathで特殊な指示方法: '5;/tmp' 等を用いることによりある程度階層化できるとはいえ。)

そこで出てくるのが、session set save handler という関数。

これ、セッション情報の書き出しに使用する関数を自前の関数に切り替えられます。

というわけで、世に使われている有名どころなCMSやらセッション管理PHPライブラリやらDBライブラリでは、みなこぞってこの関数を利用してセッション情報をDBに格納するよう調整しているわけです。

サンプルがぼろぼろ

http://jp2.php.net/manual/ja/function.session-set-save-handler.php

に掲載されていますし、有名どころのCMSやDB管理用ライブラリでは普通に使われてますので、是非一度そこらあたりをHackしてみてください。あるいは上記サンプルを参考に自前で実装してみて、いろいろ遊ぶのもおもしるいと思います。

ってなぐあいで、以上です。おつきあい頂き、誠に有り難うございました。