

The Truth about Sessions

Chris Shiflett
Principal, OmniTI
chris@omniti.com

Talk Outline

- ▶ HTTP and Statelessness
- ▶ GET, POST, and Cookies
- ▶ PHP Sessions
- ▶ Debugging Sessions
- ▶ Session Security
- ▶ Questions and Answers



HTTP

Hypertext Transfer Protocol



HTTP Request

```
GET / HTTP/1.1  
Host: shiflett.org
```

```
POST /post.php HTTP/1.1  
Host: shiflett.org  
Content-Type: application/x-www-form-urlencoded  
Content-Length: 7
```

```
foo=bar
```

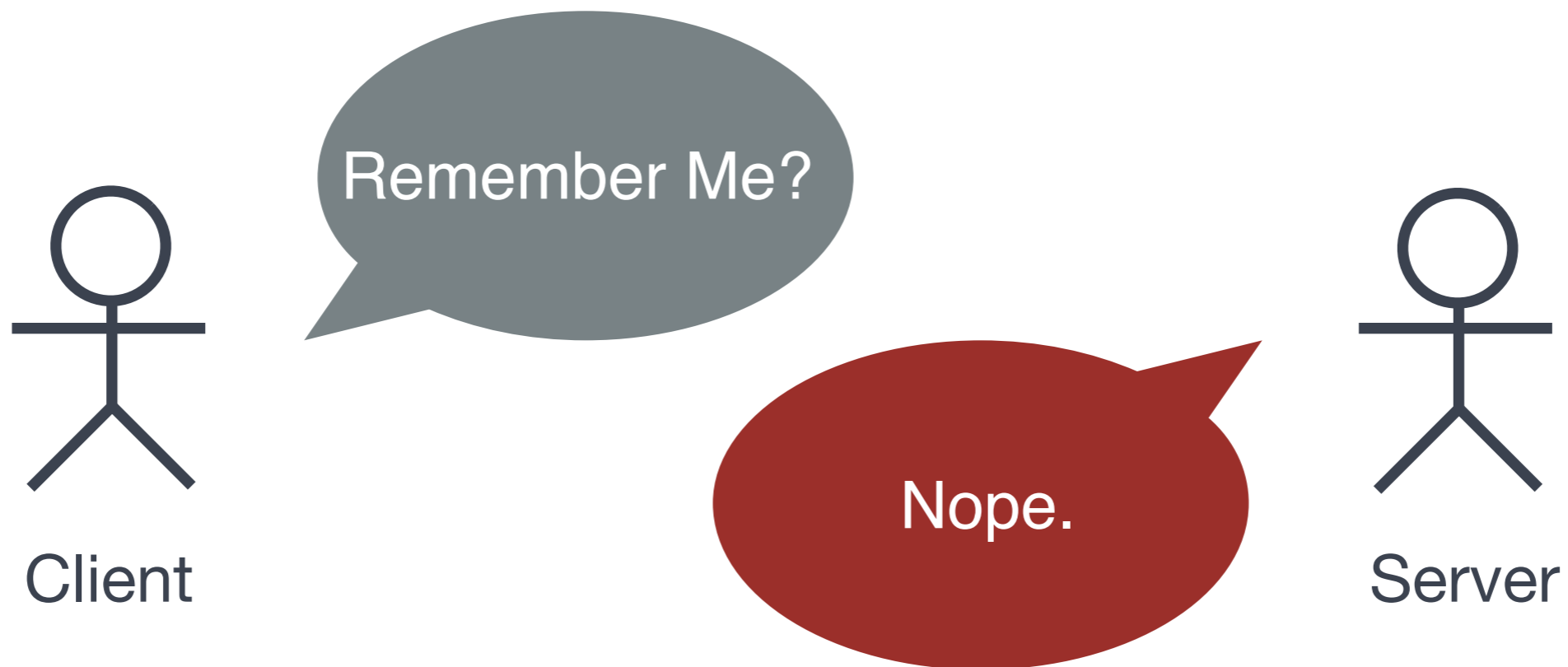
HTTP Response

```
HTTP/1.1 200 OK
Date: Thu, 15 Mar 2007 14:30:00 GMT
Server: Apache
Content-Type: text/html; charset=UTF-8
Content-Length: 35

<html>
<p>Hello, World!</p>
</html>
```

Statelessness

"HTTP is a stateless protocol."



GET Data

```
GET /next.php?id=1234 HTTP/1.1  
Host: shiflett.org
```

```
<?php  
  
$_GET['id'] = '1234';  
  
?>
```

GET Data

```
<?php  
output_add_rewrite_var('id', '1234');  
?>
```

```
<a href="next.php">Next Page</a>
```



```
<a href="next.php?id=1234">Next Page</a>
```

POST Data

```
POST /next.php HTTP/1.1
Host: shiflett.org
Content-Type: application/x-www-form-urlencoded
Content-Length: 7
```

```
id=1234
```

```
<?php
```

```
$_POST['id'] = '1234';
```

```
?>
```

Cookies

"HTTP State Management Mechanism"

- ▶ RFC 2965
- ▶ RFC 2109
- ▶ "Persistent Client State HTTP Cookies"
http://wp.netscape.com/newsref/std/cookie_spec.html

Cookies

```
bool setcookie ( string $name [, string $value [, int $expire [,  
string $path [, string $domain [, bool $secure [, bool  
$httponly]]]]]] )
```

```
<?php  
  
setcookie( 'id' , '1234' );  
  
?>
```

Cookies

```
void header ( string $string [, bool $replace [, int  
$http_response_code]] )
```

```
<?php  
header( 'Set-Cookie: id=1234' );  
?>
```

Cookies



```
session_start();
```

PHP Sessions

```
$_SESSION[ 'name' ] = 'Chris' ;
```



```
echo $_SESSION[ 'name' ] ;  
/* Chris */
```

"That was easy. Almost too easy."

Debugging PHP Sessions

```
<?php  
  
session_set_save_handler( '_open' ,  
                           '_close' ,  
                           '_read' ,  
                           '_write' ,  
                           '_destroy' ,  
                           '_clean' );  
  
?>
```

Debugging PHP Sessions

```
<?php  
  
function _open()  
{  
    echo '<p>_open()</p>';  
    /* ... */  
}  
  
?>
```

Debugging PHP Sessions

The "write" handler is not executed until after the output stream is closed.

– http://php.net/session_set_save_handler

Debugging PHP Sessions

```
<?php  
  
function _open()  
{  
    error_log("_open() \n", 3, '/tmp/debug');  
    /* ... */  
}  
  
?>
```

Debugging PHP Sessions

```
<?php  
  
include '../inc/session.inc';  
  
$_SESSION['name'] = 'Chris';  
  
?>
```

```
_open()  
_read(058c616b8ed35c16ccaf97fd1e1ba395)  
_write(058c616b8ed35c16ccaf97fd1e1ba395, name|s:5:"Chris");  
_close()
```

Debugging PHP Sessions

```
<?php  
  
header('Location: http://example.org/');  
session_write_close();  
  
?>
```

Session Security

```
http://example.org/index.php?PHPSESSID=1234
```

Session Security

```
<?php
/* $_SESSION['auth'] = FALSE; */

if (auth($_POST['username'],
        $_POST['password'])) {
    $_SESSION['auth'] = TRUE;
    session_regenerate_id(TRUE);
}

?>
```

Session Security



Thanks for Listening!

- ▶ <http://shiflett.org/>
- ▶ <http://omniti.com/>

