GBK (100)

Description

From wikipedia:

> Rock-paper-scissors is a hand game usually played between two people, in which each player simultaneously forms one of three shapes with an outstretched hand. These shapes are "rock", "paper", and "scissors". A zero-sum game, it has only two possible outcomes other than a tie: one player wins, and the other player loses.

http://tenjin.compfest.web.id:10137/

access.php

```
<?php
  $secret_key = '_________________';
  $access = false;
  if (isset($_GET['key'])) {
    $key = $_GET['key'];
    if (strcmp($key, $secret_key) == 0) {
      $access = true;
    }
  }

?>
<link rel="stylesheet" href="style.css">

<br><br><br>

<div class="container">

  <div class="info">
    <p>Hello, Developer!</p>
    <p>Please enter the key to access the game source code</p>
  </div>
  <br>
  <form action="" method="GET">
    <center>
      <input type="text" name="key" placeholder="key">
      <input type="submit">
    </center>
  </form>

  <?php if($access): ?>
    <div class="info large">
      <iframe src="play.php.sourcecode.txt" frameborder="0" style="width: 100%; height: 300p
    </div>
  <?php endif; ?>

</div>
```

Write up

- From a user comment in PHP strcmp manual:

> If you rely on strcmp for safe string comparisons, both parameters must be strings, the result is otherwise extremely unpredictable. `strcmp("foo", array()) => NULL + PHP Warning`

- We can exploit this to bypass the secret key comparison as a null can be seen as a zero because a `==` is an equal comparison with type juggling
- Thus if you visit `/access.php?key[]=` it will show the source code for `index.php`:

```
<?php
  session_start();

  $time = time();
  extract($_GET);

  srand($time);
  $enemyAction = getAction(rand());

  if (!isset($_SESSION['nWin']))
    $_SESSION['nWin'] = 0;

  if (isWin($yourAction, $enemyAction)) {
     $_SESSION['nWin']++;
  } else {
     $_SESSION['nWin'] = 0;
  }

  function getAction($i) {
    if ($i % 3 == 0) return "Gunting";
    else if ($i % 3 == 1) return "Kertas";
    else return "Batu";
  }

  function isWin($action, $enemy) {
    if ($action == "Gunting") {
      return ($enemy == "Kertas");
    } else if ($action == "Kertas") {
      return ($enemy == "Batu");
    } else if ($action == "Batu") {
      return ($enemy == "Gunting");
    }
  }

?>

<link rel="stylesheet" href="style.css">

<br><br><br>

<div class="container">
  <form action="" method="GET">
    <center>
    <input type="submit" name="yourAction" value="Gunting">
    <input type="submit" name="yourAction" value="Kertas">
    <input type="submit" name="yourAction" value="Batu">
    </center>
  </form>

  <div class="info">
    <?php
      echo "Enemy Action: " . $enemyAction;
      echo "<br>";
      echo "Your Action: " . $yourAction;
      echo "<br><br>";
      echo ($win ? "You win!!" : "You lose!!");
      echo "<br>";
      echo "Number of consecutive win: " . $_SESSION['nWin'];
      echo "<br>";
      if ($_SESSION['nWin'] > 20) echo "COMPFEST9{-----[REDACTED]-----}";
```

```
      ?>
    </div>

    <center>
      <p>
        If you are a developer, <a href="access.php">click here</a>
      </p>
    </center>
  </div>
```

- We can see that it use `rand()` to generate random action, and it is seeded with `$time` variable
- However before it was seeded, an `extract($_GET)` which is evil
- We can then specify the seed, for example 0, through adding `?time=0` in the url and thus we get: `/?yourAction=Gunting&time=0` where we always win
- Refreshing the page 21 times will get us the flag: `COMPFEST9{php_is_full_of_vulnerabilities}`