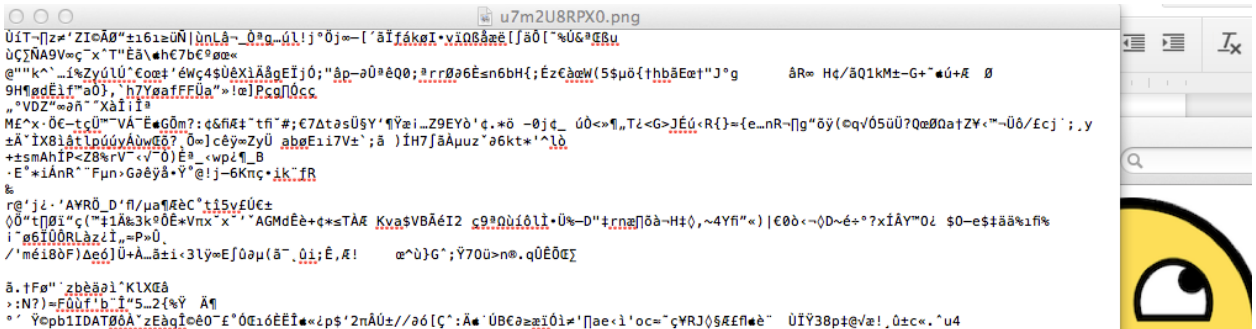
Capture The Flag - Comp116 2013
Team 11

On October 24, Team 11 began it's attempt at Capture The Flag, created by Ming Chow and played by all students in Comp116: Computer Security. The purpose of the game was to give students a better understanding of how security vulnerabilities can be exploited to find information or alter a web application, particularly after the time spent learning how to use security tools.

We started with a few simple rules: No hacking into the scoreboard, no DDOSing the server, and no trying to cheat in other ways. We were looking for keys, hidden inside of a web application, in the format key{thisisthekeyinformationhere}.

Flag 1:
We wouldn't have been able to find Flag 1 without the hint, which indicated that it could be hidden inside an image. We pulled down the weird smiling face off the site, a png file, and decided to look at it's source code (by opening the png in a text editor. We did a search for "key," and lo and behold, we found it:



Key one, obtained. Moving on...

Flag 2:
This one was a little more tricky, and required playing around with the message board a little bit. We started to play with it and realized that it worked almost like a blog posting site; you could put parameters into the URL to dictate which post you wanted to see. And you could set those parameters equal to either a number or, as we found, a SQL injection designed to force it to get set to true, which spit out the key.
We used the following SQL injection:

http://67.23.79.113/index.php?id=33%20OR%201=1;

And out came our key.

Flag 3:
This one required some searching. After hitting the server with some pings, we figured out that the server was an Apache web server. We then figured that the parameters were being eval()ed, as any php that was put in the parameter was evaluated, and thus ran.
We started to use the glob() and var_dump() functions together to sift through the filesystem, which eventually led us to the root user information: (root/Wh@t3verWh@t3ver!). Using that

information, we devised a SQL Query to dump the entire board to the screen. When we did this dump, the key presented itself. Below is the URL we used.:
http://67.23.79.113/index.php?id=$result%20=%20mysqli_query($con=mysqli_connect(%22localhost%22,%22root%22,%22Wh@t3ver!Wh@t3ver!%22,%22board%22),%22SELECT%20*%20FROM%20users%22);while($row=mysqli_fetch_array($result)){var_dump($row);}
After searching for "key" in all the dumped data, we found it.

Flag 4:
This one took a couple parts.
We went to the admin page, and tried to use a SQL injection to break in to the site. Doing the SQL injection on the username appeared to work, however the submit took us to a custom 404 error page. We looked at the source code, and saw an interesting note waiting for us in the comments:

<-- The plot thickens… -->

So we cleared our history and cache and tried again. This time, we realized that the logon was setting a session ID for our computer, which meant that cookies were being transfered. We decided to switch over to firefox and use TamperData to see what cookies were being transmitted and received. We eventually found one that looked suspicious. After changing one of the parameters of that cookie from false to true, the site revealed flag number 4 to us.

Flag 5:
Using the URL parameters to pass php functions into the app so that the functions would run, we continued to use glob() and var_dump() to explore the file system. Along the way, we found a file name with a base-64 hash. We opted to go to a base64 decoder online and found the key that way.

Flag 6 and 7:
We kept moving about the file system, looking for files that might have some info. We used var_dump(glob("/home/pheller/*") to find two interesting looking files: README.txt and namegame. We used file_get_contents() to open each, and in doing so, got the keys for both Flag 6 and Flag 7.

Flag 8:
Use var_dump(glob("*")); to see that there's a php file named "logout.php". Using file_get_contents("logout.php"), we were able to run that script. After looking at the source code for the returned HTML, the final key was hiding there!

```
<?php
        session_start();
        if ($_SESSION['login'] == null)
        {
                header("Location: admin.php");
                exit;
        }
        session_destroy();
?>
<!DOCTYPE html>

<html>
<head>
<title>Login</title>
<link rel="stylesheet" type="text/css" href="yui.2.css" />
<link rel="stylesheet" type="text/css" href="global.6.css" />
</head>
<body>
<h4>You have successfully logged out of the system.</h4>
<h4><a href="admin.php">Return to login</a></h4>
<br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/>
<br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/>
<br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/>
<br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/>
<br/><br/><br/><p>key{04e3895d689bbbe88f80021e3a8428eb3b626b6f}</p>
</body>
</html>
```