

Track your visitors, using PHP

This is an article on *Track your visitors, using PHP* in *PHP*.

There are many different traffic analysis tools, ranging from simple counters to complete traffic analyzers. Although there are some free ones, most of them come with a price tag. Why not do it yourself? With PHP, you can easily create a log file within minutes. In this article I will show you how!

Getting the information

The most important part is getting the information from your visitor. Thankfully, this is extremely easy to do in PHP (or any other scripting language for that matter). PHP has a special global variable called `$_SERVER` which contains several environment variables, including information about your visitor. To get all the information you want, simply use the following code:

PHP Code:

```
// Getting the information
$ipaddress = $_SERVER['REMOTE_ADDR'];
$page = "http://{$_SERVER['HTTP_HOST']}{$_SERVER['PHP_SELF']}";
$page .= iif(!empty($_SERVER['QUERY_STRING']), "?{$_SERVER['QUERY_STRING']}",
  "");
$referrer = $_SERVER['HTTP_REFERER'];
$datetime = mktime();
$useragent = $_SERVER['HTTP_USER_AGENT'];
$remotehost = @getHostByAddr($ipaddress);
```

As you can see the majority of information comes from the `$_SERVER` variable. The `mktime()` and `getHostByAddr()` functions are used to get additional information about the visitor.

Note: I used a function in the above example called `iif()`. You can get this function at <http://www.phpit.net/code/iif-function>.

Logging the information

Now that you have all the information you need, it must be written to a log file so you can later look at it, and create useful graphs and charts. To do this you need a few simple PHP function, like `fopen` and `fwrite`.

The below code will first create a complete line out of all the information. Then it will open the log file in "Append" mode, and if it doesn't exist yet, create it.

If no errors have occurred, it will write the new logline to the log file, at the bottom, and finally close the log file again.

PHP Code:

```
// Create log line
$logline = $ipaddress . '|' . $referrer . '|' . $datetime . '|' . $useragent
. '|' . $remotehost . '|' . $page . "\n";

// Write to log file:
$logfile = '/some/path/to/your/logfile.txt';

// Open the log file in "Append" mode
if (!$handle = fopen($logfile, 'a+')) {
    die("Failed to open log file");
}

// Write $logline to our logfile.
if (fwrite($handle, $logline) === FALSE) {
    die("Failed to write to log file");
}

fclose($handle);
```

Now you've got a fully function logging module. To start tracking visitors on your website simply include the logging module into your pages with the include() function:

PHP Code:

```
include ('log.php');
```

Okay, now I want to view my log file

After a while you'll probably want to view your log file. You can easily do so by simply using a standard text editor (like Notepad on Windows) to open the log file, but this is far from desired, because it's in a hard-to-read format.

Let's use PHP to generate useful overviews for is. The first thing that needs to be done is get the contents from the log file in a variable, like so:

PHP Code:

```
// Open log file
$logfile = "/some/path/to/your/logfile.txt";

if (file_exists($logfile)) {

    $handle = fopen($logfile, "r");
    $log = fread($handle, filesize($logfile));
```

```

        fclose($handle);
    } else {
        die ("The log file doesn't exist!");
    }
}

```

Now that the log file is in a variable, it's best if each logline is in a separate variable. We can do this using the `explode()` function, like so:

PHP Code:

```

// Seperate each logline
$log = explode("\n", trim($log));

```

After that it may be useful to get each part of each logline in a separate variable. This can be done by looping through each logline, and using `explode` again:

PHP Code:

```

// Seperate each part in each logline
for ($i = 0; $i < count($log); $i++) {
    $log[$i] = trim($log[$i]);
    $log[$i] = explode('|', $log[$i]);
}

```

Now the complete log file has been parsed, and we're ready to start generating some interesting stuff.

The first thing that is very easy to do is getting the number of pageviews. Simply use `count()` on the `$log` array, and there you have it;

PHP Code:

```

echo count($log) . " people have visited this website.";

```

You can also generate a complete overview of your log file, using a simple `foreach` loop and tables. For example:

PHP Code:

```

// Show a table of the logfile
echo '<table>';
echo '<th>IP Address</th>';
echo '<th>Referrer</th>';
echo '<th>Date</th>';
echo '<th>Useragent</th>';
echo '<th>Remote Host</th>';

foreach ($log as $logline) {
    echo '<tr>';

    echo '<td>' . $logline['0'] . '</td>';
}

```

```
echo '<td>' . urldecode($logline['1']) . '</td>';
echo '<td>' . date('d/m/Y', $logline['2']) . '</td>';
echo '<td>' . $logline['3'] . '</td>';
echo '<td>' . $logline['4'] . '</td>';

echo '</tr>';

}

echo '</table>';
```

You can also use custom functions to filter out search engines and crawlers. Or create graphs using PHP/SWF Charts. The possibilities are endless, and you can do all kinds of things!

In Conclusion...

In this article I have shown you how to create a logging module for your own PHP website, using nothing more than PHP and its built-in functions. To view the log file you need to parse it using PHP, and then display it in whatever way you like. It is up to you to create a kick-*** traffic analyzer.

If you still prefer to use a pre-built traffic analyzer, have a look at <http://www.hotscripts.com>.

Source: <http://www.go4expert.com/articles/track-visitors-using-php-t195/>