

PHP Memory Leaks, How to Find and Fix Them

[Matthew Thompson](#) on May 29, 2020



[Engineering Php](#)

Memory leaks can happen in any language, including [PHP](#). These memory leaks may happen in small increments that take time to accumulate, or in larger jumps that manifest quickly. Either way, if your app has a memory leak, sooner or later it will cause problems. The source of and solution to PHP memory leaks aren't always obvious, so you may need to try a few strategies before you eliminate the problem.

PHP Memory Usage: How Memory Leaks Happen

What is a memory leak in PHP?

A memory leak in PHP is a condition that causes sections of code to continue using memory even though that memory is no longer needed. There are several ways for memory leaks to occur. Variables that never go out of scope, cyclical references, extensions in C that ``malloc`` instead of ``emalloc`` and for whatever reason don't ``free``, to name a few. There are surprising and quite subtle ways of using and holding on to memory in PHP. If enough memory is leaking, your application will eventually run into memory limits imposed by PHP settings or by the OS itself, and crash.

Does PHP have garbage collection?

Yes, PHP does have garbage collection that should help prevent memory leaks. However, several factors can prevent the garbage collector from fulfilling its task. For example, if an object's refcount increases and never decreases, then the object is still technically in use and is not garbage. The garbage collector, therefore, cannot identify it as such and will not free up the object.

Garbage collection is active in PHP by default, but it can be disabled through settings in `php.ini`. If garbage collection is disabled, you'll quickly accumulate unused memory that's never freed. Disabling garbage collection can improve performance for scripts that are short-lived and completely exit (and therefore free all used memory), but for longer scripts and daemons you'll probably want the garbage collector enabled.

How to Find PHP Memory Leaks

Identifying the source of your PHP memory leak is the first step to finding a solution. You can't fix a problem until you understand its underlying cause.

Option One: Log Your Scripts

If you're running multiple scripts, then you need to determine which one is causing the leak. Use `auto_append_file` and `memory_get*` to generate a log of your scripts. Then, review the scripts to see which ones use the most memory.

Option Two: Discover Peak Usage

Use `memory_get_peak_usage` to find out [how much memory](#) has been allocated to your script. If it looks abnormally high, then you might first look at your PHP script for code that may be unintentionally loading or iterating over more data than anticipated. If so, break down the loading or processing of the data into manageable chunks instead of all-at-once.

Option Three: Use a PHP Memory Profiler

Use the [php-memprof extension](#) to learn how much memory is still in use after a script runs.

Monitoring Memory Usage with a PHP Memory Profiler

Monitoring memory usage with a PHP memory profiler can make it much easier for you to spot problems within your scripts. Several PHP profilers include features that will detect memory leaks.

What is memory profiling?

Memory profiling scans your PHP scripts to see precisely how each function uses memory. The level of depth that you get will depend on the PHP memory profiler that you choose. Some, for example, will show you how much memory your functions use and continue use while running a script. Others will point you directly to problematic functions that cause PHP memory leaks and other performance issues.

Finding the right memory profiler is an essential part of debugging PHP scripts with memory leaks.

Xhprof PHP Memory Profiler

[Xhprof](#) has a simple user interface that will help you discover PHP memory leaks. It can also identify the performance issues that make PHP memory leaks happen.

Xdebug PHP Profiler

[Xdebug](#) is a standard PHP profiler that you can use to discover a variety of performance issues in your scripts. The lightweight profiler doesn't use much memory, so you can run it alongside your PHP scripts for real-time performance debugging.

PHP-memprof

[PHP-memprof](#) is a stand-alone PHP memory profiler that can tell you exactly how much memory each of your functions uses. It can even trace an allocated byte back to a function.

Dig in!

It's inevitable that at some point you'll have to track down a memory issue with your PHP app. Unfortunately, memory leaks are the type of bug that can be notoriously difficult to hunt and fix. With the options covered here, we hope you have a good starting point to uncover and resolve any potential memory problems you encounter with your PHP application!

Most popular

- [New Relic vs. Scout: Which Is The Right APM For You?](#)
- [Monitor a Laravel app with Scout](#)
- [Monitor a Django app with Scout](#)
- [Finding slow ActiveRecord queries with Scout](#)
- [Five tools for profiling Rails apps](#)

Subscribe for the latest APM news and tech tips.

<input type="text" value="First Name"/>	<input type="text" value="Email Address"/>
<input type="button" value="Join"/>	

