Security – PHP Protection

Security · https://www.exeoutput.com/help/php-protection

During compilation, ExeOutput for PHP **compresses and encodes source files** into the final executable file. The resulting EXE file cannot be unpacked with an archiver like WinRAR or 7-Zip. When you run the compiled application, files such as PHP pages, HTML, images, and JavaScript are never unpacked to the hard disk. Therefore, it is not possible for an average user to copy them.

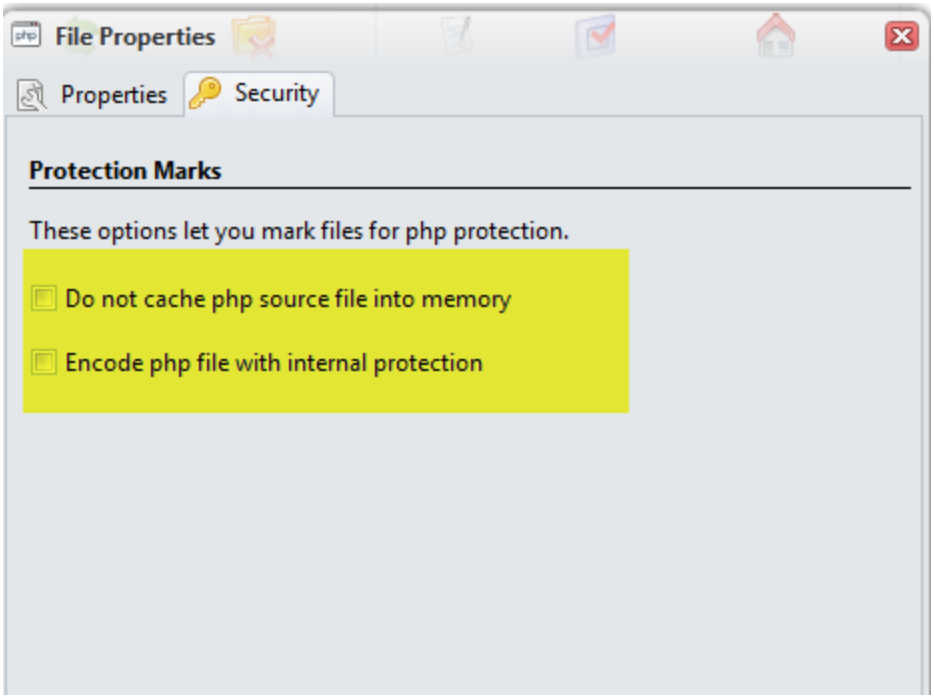
Since PHP scripts must be unpacked into memory to be interpreted by the PHP runtime, it may be possible for a skilled hacker to extract portions of the compiled PHP files. To make this task more complicated and time-consuming, ExeOutput for PHP includes several security measures.

Warning

We strongly recommend that you do not include private passwords, database login info, or other sensitive information in applications released to the public. Instead, use encryption, server authentication, HEScript calls, or at a minimum, the string protection feature.

Moreover, ExeOutput for PHP provides **additional security options for sensitive PHP scripts**.

These **options are global**. Since they should not be applied to all PHP scripts, **you must mark the specific scripts** that require protection. This is done using "Protection Marks," available in the **Security** tab of the File Properties window in the File Manager:



Encode Marked PHP Files with the Internal Protection System

ExeOutput for PHP encrypts the PHP source file so that it does not appear in clear text in memory, although the script remains functional.

Encoding is performed while ExeOutput for PHP compresses the files. The original files are not replaced; instead, they are encoded into memory and then compressed into the final EXE.

Note

The internal protection system may not be compatible with all PHP files. In such cases, ExeOutput for PHP may fail to perform the conversion correctly. If an error occurs, it is logged in the compilation log, and ExeOutput for PHP will compile the original, unencoded PHP source file instead.

Do Not Cache Marked PHP Files in Memory

PHP scripts are unpacked into memory to be interpreted by the PHP runtime. Since some PHP scripts (such as includes) may be required multiple times, the runtime module caches them in memory for better performance. This means they remain in memory until the cache is full or the application is closed. This makes the application more responsive, as the decompression step is skipped on subsequent requests.

This option lets you **specify which PHP scripts should not be kept in memory after execution**. Note that in this case, the application will have to decompress these non-cached scripts every time they are requested by the PHP runtime.