

# ZenCrypt PHP Encoder

## Users Manual

ZenCrypt is a simple yet efficient PHP encoder that encodes your PHP source files into unreadable form.

ZenCrypt makes it difficult for non technical people to disassemble and modify your source code. It does not require any special tools or libraries to be installed on a server making it very portable way to encode and run encoded files to any hosting account.

ZenCrypt is written on PHP and for it to run you need to have PHP installed on your computer or server.

Advantage of that is that ZenCrypt is supported by any platform and any operating system in existence (where PHP command line tools are supported and installed).

ZenCrypt PHP Encoder is a command line tool allowing you to build your projects and product in a batch mode.

### Usage:

To run ZenCrypt issue this command from shell/command prompt:

```
php -f zencrypt.php infile=input_filename.php outfile=output_filename.php  
[encycles=number_of_encoding_cycles] [stripcomments=yes] [replace:with=AAA:BBB]  
[replace:with=CCC:DDD] ...
```

Source file that you want to be encoded needs to include special tokens that would mark areas of file that you want to be processed. This makes it convenient for you to choose whether you want to encode everything inside your source code or only parts of the code.

ZenCrypt will encode only chunks of code inside your PHP source file that are included in between properly formatted **PHP\_ENCODE** tokens. Example:

```
<?php  
// test.php  
// -----  
// To encode this file run:  
// php -f zencrypt.php infile=test.php outfile=test2.php  
  
// This line will not be encoded  
$a=1;  
  
//{{{PHP_ENCODE}}}  
// This line will be encoded  
$b=2;  
//{{{/PHP_ENCODE}}}  
  
// This line will not be encoded  
echo $a, $b;
```

?>

Running ZenCrypt to encode this file will produce the following result:

```
<?php
// test.php
// -----
// To encode this file run:
// php -f zencrypt.php infile=test.php outfile=test2.php

// This line will not be encoded
$a=1;

$yQJTYOqQZfiuOL=='AAGIbNymkUlHeTJ9kzLlMSVpESRlMzPjSVLxscyIbFLInMCXd90';$kcUVrbzltwSG=';))

// This line will not be encoded
echo $a, $b;
```

?>

By default ZenCrypt runs through 1 cycle of encoding. To make more secure encryption and make your code more challenging to disassemble you may choose to increase the number of encoding cycles.

To choose 5 cycles to encode your source will require this command:

```
php -f zencrypt.php infile=test.php outfile=test2.php encycles=5
```

...and will produce the following result:

```
<?php
// test.php
// -----
// To encode this file run:
// php -f zencrypt.php infile=test.php outfile=test2.php

// This line will not be encoded
$a=1;

$wTgNEODmvmRmC_lkco='H8vO6PF/6g65OV38P7fHBNStTA+vkfIgcDb8oERg+EF/
r3+d4zX6CNCD/
ftrAhzh3h3oG2r6j2NGXTvCrF24XTzQsVN6PkGrg28J6fYYqPgs+qJUoE391th3SD/vSF9zp/
kxW4F2fe98P/
53V3DEjek3mbwXBhyf8wfMTlaf19qC7X5eJmtdaeiq0Kh+55n3xVMVtYgOnkrxwqpF8E8ZmdXvGyNrWjxmsFN0dxPb
Ifm5L23064u731lac+vJT1fYF16k9GtAVXhg4EF6H0vth01ZvuGL6RUT6dGy/4TO/
oQmImJ5eagdK+puykKCKnJXyAh8N+Wzw1VpPNNmJqAg4Ir1QArlRQ5YuOlpgSWRFqXxYs67mmXW+9dbxyde0zohVyU
uNTPN4bm2UmweXBFP51rSHZiM8iLv1CALr525SJckqslXY4MoMxbAT1x6vfgOND2XynqOOUspsBgp200o1pgNDJC66
rgb0R65OBCE1JHRkGWAnYzUY2s0aQKJ6Kuf1A0m1P/0px0wFZQXhrqTC/hYUcl1f+zrTJGO/
MBITOWtYuBkKsXjoOlq3na1b79ftu51NOMYIj4GDcwzZwDzwxCrMSAMAcscIlJ/REAAnqbj/
Yb';$fBPqvdNldoUHFmuw=';))bpxy_PzEzizQBRatGj$(ireegf(rqbprq_46rfno(rgnysavmt(ynir';$DbWB
```

```
// This line will not be encoded
echo $a, $b;
```

```
?>
```

ZenCrypt currently supports following tokens:

**PHP\_ENCODE** - as we saw above

**PHP\_ENCODE\_xyz** - variation of above token allowing you to relocate encoded chunks of code into another area of your source file.

**PHP\_INSERT\_xyz** - use this token to mark the spot inside of your source file where encoded data will be moved to (relocated) from it's original location.

**PHP\_MELT** - allows you to eliminate marked pieces of source code completely.

Example:

```
<?php

//{{{{PHP_MELT}}}
// Secret notes to myself:
// - Fix bugs
// - Do not give plain source code to the client before payment in full is
made!
//{{{{/PHP_MELT}}}

$a = 1234;
$b = $a * 2;

//{{{{PHP_ENCODE}}}
// This is secret operation #1
$c = ($a + $b) / 4;
//{{{{/PHP_ENCODE}}}

//{{{{PHP_ENCODE_a1}}}
// This is secret operation #2
$d = ($c * 8) + $a * $b;
$e = my_func ($d);
echo $e;
//{{{{/PHP_ENCODE_a1}}}

function my_func ($x)
{
    return $x * $x;
}

// Do not worry about this piece.
//{{{{PHP_INSERT_a1}}}
// "a1" encoded chunk will be inserted here
//{{{{/PHP_INSERT_a1}}}
```

?>

Let use this command to encode this file:

```
php -f zencrypt.php infile=test.php outfile=test2.php enccycles=3
```

Encoding above file with 3 cycles of encoding will produce the following result:

```
<?php
```

```
$a = 1234;  
$b = $a * 2;
```

```
$qRPaSVTuentEUTWSEj='=Ewfgf7Y5rpkBxnFA/nMZ/bbG//V/nTAlDEmOQxnq+45xOzPFY5/  
MDCuH76pFHTmfZaS8kkn8sbvkU0Gx7NFz0tP3nWBwVGsXaxOfb0JSzYGqI6iOa+M6Cw1FsBFwrvTx73cOJXK4u4YWu/  
DpUn/abMCx9EysGGER+YbEfGtZDJcBWCDKV8GdwinOVcpv0WEc/  
RUSTOY2v3gYGFFdUZl4B2zh+LpXbgwW9zhYWFc+3K8n+Ev6WDF0QIjWwb8rsQLKcOei64qtocPsg7pdhMcappN/  
m2LLrRGLmaYoBqaahMYChKMDYRJuafE+DcDofdjegLSZanlXmgpjKVu/  
l1qmWcmwIsDNCUD40rjQOE65eqt244yeKK6amJ3cevv7ue9++9XWQRBYB12cNspDjyyC1CzXAB0VV7CMSkx/  
VEAApqjJ/  
Yb';$Sak_SBskcZISiKBZ=';))wRFJGHrgarhGIFnCEd$(ireegf(rqbprq_46rfno(rgnysavmt(ynir';$BYkb
```

```
function my_func ($x)  
{  
    return $x * $x;  
}
```

```
// Do not worry about this piece.  
$QoTwFeGCRB='==wP280r8TsApZSyR+VmjrjSNp/  
n2PdloQGo1gUdpkhsrNW+GKpQTtVa7lon3NempSZpDJKfQespQNVQXztMDVSxkYhF6T6XKQqMWmEJQGjHy1giA8hxx/  
gXkffR37XRQ0yE85OcmkIePD970/  
SHaeNo80GuQfaE74dUcjdfeFcmq3Kuf16qaB9ha390w2FMUUrqXWNmUoIfN49eqcFE+D2wQW0QZR1Xckax1uLetr7j/  
VEAApqjJDZb';$cRdXlxlMaPjR=';))OEPTrSjGbd$(ireegf(rqbprq_46rfno(rgnysavmt(ynir';$vwlV_c
```

?>

This should help you to get up and running.  
More details will be added soon!

## Contact:

For more information please contact us at: [info@zencrypt.com](mailto:info@zencrypt.com)