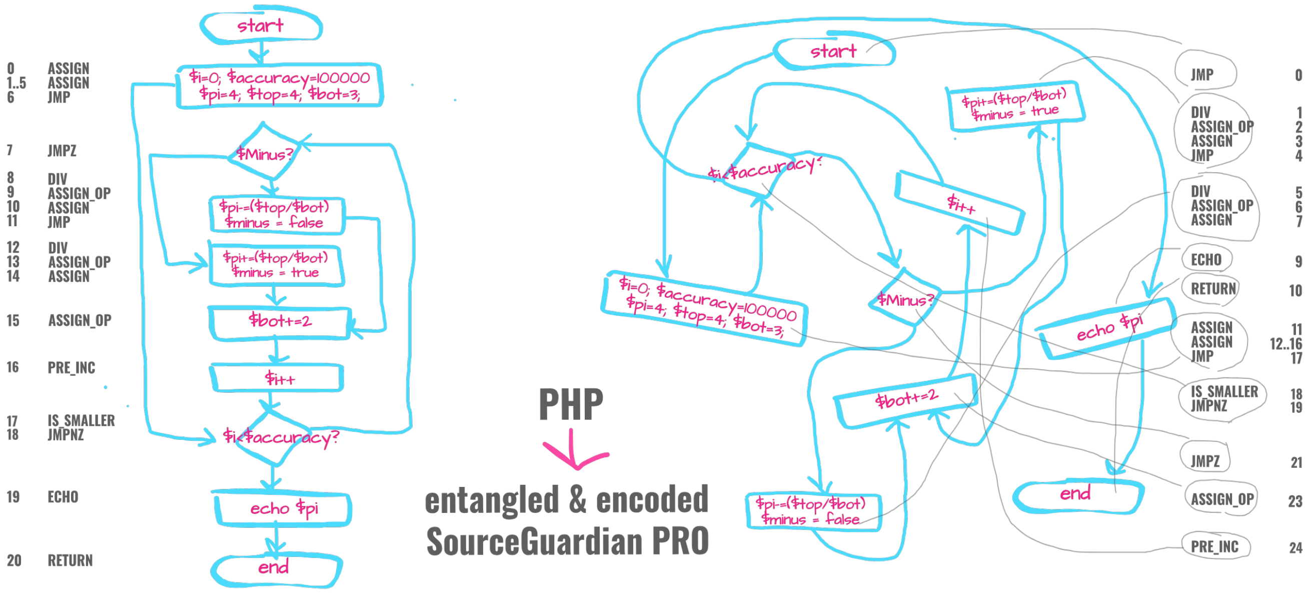
SourceGuardian Pro

Innovative code protection with bytecode entangling

SourceGuardian Pro innovative code protection uses bespoke bytecode entangling algorithms. This is an addition to what standard version of SourceGuardian provides in terms of code protection. SourceGuardian Pro performs a deep analysis of the compiled PHP bytecode which is followed by splitting bytecode into logical blocks, randomly mixing them carefully keeping all the links between the blocks, encrypting, and producing an encoded PHP file. Protected code is running as the original, but internally, it becomes dramatically hard to understand or decode. Bytecode entangling makes it very hard to follow the bytecode with a debugger or a custom modified PHP interpreter. This creates an additional serious barrier for a potential code attacker.



Dynamic encoder license support for CI/CD

SourceGuardian Pro supports a new dynamic license, which makes it possible to integrate the encoder and tools into dynamic CI/CD environments. This requires subscribing to SourceGuardian dynamic license. Once the license is enabled in the user profile, specify new options in the encoder or tools command line to use it.

Thank you for choosing SourceGuardian Pro

We appreciate your comments as well as any questions and suggestions. This is a new version of SourceGuardian and it uses innovative ideas for code protection. We spent many months working on it and many weeks testing and debugging it with various versions of PHP and different target platforms. New algorithms for deep bytecode analysis, code entangling, and randomization are quite difficult. We believe they are very well debugged and tested, but if you experience any issues, inconsistencies, or see a difference in how the encoded code runs in comparison to the unencoded source, please do not hesitate to contact us:

support@sourceguardian.com

As always, in case of problems, we appreciate you sending a sample PHP code, making it possible for us to reproduce the issue and debug and fix it quickly.

Notes for using a dynamic license and its support for CI/CD

In order to use this option, please subscribe to SourceGuardian dynamic license.

Once the license is enabled in the user profile, you may specify new options in the encoder or tools command line to use it. A dynamic license is automatically acquired when SourceGuardian Pro encoder, a license generator, or a code information tool is run. The license is automatically renewed as necessary while your SourceGuardian Pro dynamic license subscription remains active. The internet connection is required on the machine that runs the encoder or tools with a dynamic license. For performance reasons, once acquired, the license file remains valid for 1 day while the environment is unchanged.

```
>./sourceguardian --license-dynamic --license-username user@domain.com --license-password random
```

Notes for using bytecode entangling

Configuring entangling

Bytecode entangling is enabled by default with all the options. It is configurable. Use the Pro --entangle X,Y option to change it and adapt to your code. The default setting provides maximum security by splitting bytecode into the smallest possible blocks before randomizing them.

```
>./sourceguardian --entangle X{,Y} . . .
```

X factor is a bit flag. Normally you will use X=7 for maximum security and PHP7+ compatibility.

Bit 0 - enables bytecode blocks randomization.
Bit 1 - enables JMP randomization. Using this option requires even more integration of SourceGuardian loader with the PHP engine running on the target machine. PHP on Linux may use PHP compiled to run in a so-called hybrid mode. Please find below further information about JMP randomization, encoding, and running scripts if they may potentially be run in such environment.
Bit 2 - enables bytecode entangling for PHP7+. Entangling for PHP5 is always on. If your code need to be run both with PHP5 and PHP7+ and you are experiencing any issues with entangling for PHP7+, it may be turned off by setting this bit to 0. Note that in normal conditions, if set to 0, bytecode will not be entangled for PHP7+ which probably is not what you want from SourceGuardian Pro.

Y factor is an optional bytecode entangling granularity value. Normally, you will use a default (unspecified) or Y=4 or Y=8 as an empirical choice. Keeping it a power of 2 is not required, though.

Random mixing of many small code blocks makes potential decoding and debugging very hard, if ever possible, but a trade-off for this may be code performance. If performance degradation is noticed, the Y factor may be increased. Note, you need to specify a valid X (normally X=7) and you cannot specify only Y.

Please note, that every PHP function or method is a bytecode sequence that is entangled and encoded separately. If your code mainly consists of short functions or methods, keeping the default or using a small Y is probably the best choice. If you are protecting a code with relatively large functions, increasing Y may improve performance. Using a larger Y factor still gives additional protection to your code because try/catch/finally blocks randomization. Choosing a relatively large Y turns off entangling, ending up with a single bytecode block that equals the original compiled one.

Y factor defines a minimum number of PHP opcodes in a block. The encoder will try not to generate blocks smaller than Y number of bytecodes. A lower value up to default zero provides more security but an empirical value of Y=8 may be an optimal choice in many cases.

If your code is experiencing performance issues and if this is crucial, like if your code is doing heavy math calculations, you may need to experiment and choose an optimal Y factor for your code. It's also possible to encode files with different Y factors by running the encoder multiple times for every set of files with different options.

Bytecode entangling code compatibility notes

JMP randomization in PHP7+ (on by default) adds yet another barrier for a potential code attacker, but handling this option by SourceGuardian loader requires additional and special bytecode processing. The performance of the encoded PHP files may be affected if this option is used. This is expected.

Running on Linux may need a new setting in php.ini, please read below. Running on Windows and MacOS platforms is not affected by the PHP hybrid engine, but SourceGuardian loader will still use special bytecode processing for running JMP randomized bytecode.

–

Due to the complexity of the randomized bytecode exception handling in PHP8+, only the last exception information is available if a new **un**handled exception is raised by the exception handler processing the current exception. This must cause no problems for most PHP code and save additional bytecodes. This reproduces the default behaviour of exception handling in PHP7.

* You may notice a difference in backtrace output in cases of multiple exceptions because of this.

Note that the change affects sending exception information only for unhandled exceptions. We are sure, your code normally handles exceptions. Please contact us in case this difference in protected PHP code execution is critical for your code. Let us know the details, and we are happy to think about it and discuss it with you.

As a result:

* Linked to the availability of only the last exception information, it's not recommended that your code throws a potentially unhandled exception from any class destructor code. We are sure, you keep destructor code safe and handle exceptions within the destructor, which is fine.

* Result of throwing an unhandled exception from a destructor may be unexpected.

* Exceptions thrown by a destructor may be called in a different order than in the source.

Running on Linux with PHP hybrid bytecode engine

Running PHP files encoded by SourceGuardian Pro with JMP randomization option (default on) on Linux with PHP hybrid engine requires an additional setting in php.ini:

```
sourceguardian.enable_vm_hybrid=1
```

If not enabled and PHP uses a hybrid engine and the encoded code requires it, SourceGuardian loader will let you know by displaying an error message:

```
SourceGuardian Loader - add 'sourceguardian.enable_vm_hybrid=1' to php.ini which is required for running this protected
script with hybrid vm PHP on this platform.
```

We do not enable this setting by default because, when enabled, the loader initializes additional handlers with PHP, which may slow down the execution of unencoded files. If you mix encoded and unencoded files in your project and they run in such an environment, and if your code experiences performance issues that are critical for your application, you may re-encode the project with X=5 (bit 1 set to 0) in --entangle instead of the default X=7. In this case, you do not need to set the above option to php.ini.

The sample code in the picture on the first page

Here are the details of the sample code in the picture on the first page if you are interested. This is quite simple, and it uses a simplified bytecode for the diagram. Yet, we think it may still be useful for demo purposes and a better understanding of what SourceGuardian Pro does for entangling your PHP code.

source	unencoded			entangled		
	line	#	bytecode	line	#	bytecode
`3  $pi = 4; $top = 4; $bot = 3;`	3	0	ASSIGN	9	1	DIV
`$minus = true;`		1	ASSIGN		2	ASSIGN_OP
		2	ASSIGN	10	3	ASSIGN
		3	ASSIGN	8	4	JMP
`4  $accuracy = 100000;`	4	4	ASSIGN			
`6  for($i = 0; $i < $accuracy; $i++) {`	6	5	ASSIGN	12	5	DIV
		6	JMP		6	ASSIGN_OP
				13	7	ASSIGN
`8    if ($minus) {`	8	7	JMPZ			
`9      $pi -= ($top/$bot);`	9	8	DIV	17	9	ECHO
		9	ASSIGN_OP	18	10	RETURN
`10      $minus = false;`	10	10	ASSIGN			
	8	11	JMP			
				3	11	ASSIGN
`} else {`					12	ASSIGN
					13	ASSIGN
`12      $pi += ($top/$bot);`	12	12	DIV		14	ASSIGN
		13	ASSIGN_OP	4	15	ASSIGN
`13      $minus = true;`	13	14	ASSIGN	6	16	ASSIGN
`}`					17	JMP
`15   $bot += 2;`	15	15	ASSIGN_OP			
`}`	6	16	PRE_INC		18	IS_SMALLER
					19	JMPNZ
		17	IS_SMALLER			
		18	JMPNZ	8	21	JMPZ
`17 echo $pi;`	17	19	ECHO	15	23	ASSIGN_OP
	18	20	RETURN	6	24	PRE_INC

Please note that this is still a simplified code. Line numbers are kept only for demo purposes. SourceGuardian removes references to source code line numbers from the bytecode. Bytecode entangling is an innovative encoding process that is still followed by standard SourceGuardian bytecode encoding, encryption, and compression. Code blocks in the entangled bytecode are mixed randomly. This means every time you encode, your code will be encoded in a new way. The sample above is just one iteration.

Best Regards,
SourceGuardian Team