

9 Changelog

9.1 Version 11.3 / April 2019

Version 11.3 introduces encoding for PHP 7.3 and many new features. We were glad to receive comments and suggestions and want to thank you very much for sharing your ideas! We are looking forward to hearing about other suggestions for improving SourceGuardian. Here is a list of all the version 11.3 changes.

- Full support of PHP 7.3 encoding including all the latest language options.
- We fixed some issues with unexpected segfaults on PHP7+ and some other problems. If you experienced issues like that with your code, please re-encode with the latest version 11.3 of SourceGuardian and install the recent loader.

Note, files encoded with SourceGuardian 11.0, 10.x or older need to be re-encoded with SourceGuardian 11.3 in order to run them under PHP 7.3

- New loaders for PHP 7.3, updated loaders for older versions of PHP

We updated loaders and created new ones for PHP 7.3. Loaders for the following operating systems are available:

Windows 32-bit (VC6, VC9; VC11 PHP 5.5, 5.6; VC14 PHP 7.0, 7.1; VC15 PHP 7.2)
Windows 64-bit (VC11 PHP 5.5, 5.6; VC14 PHP 7.0, 7.1, VC15 PHP 7.2)
MacOSX (universal binaries, include i386, x86_64)
Linux (i386, x86_64)

We update the following loaders on request. Please check [our blog](#) and the [loaders page](#) for new versions.

FreeBSD (i386, x86_64)
Linux ARM (armel)
Linux ARM (armhf) Raspberry Pi (including Pi version 3) and other boards

We are happy to work with our customers if they need bespoke loaders for other custom operating system. Please contact support@sourceguardian.com if you are interested.

We do not include loaders for all supported operating systems to the GUI package. Please visit our [loaders page](#) if you can't find the loader for your OS in the [Copy Loaders](#) dialog.

New Features

- Experimental licensing for Docker. If you are installing SourceGuardian to your Docker, licensing must work correctly now letting you install and use SourceGuardian on this Docker machine. So, if you are using SourceGuardian from a Docker container during the deployment process of your files, this must work now since you installed and registered your copy of SourceGuardian as usual. Note, every separate installation of SourceGuardian still requires an additional license as one license lets you install and use SourceGuardian only on one machine.

Installing to Docker needs a special approach and mapping of `/var/run/docker.sock`. Please refer to a new section [Installing to Docker](#) in the user manual.

- We are introducing a new locking option which is available in the full version of the encoder - locking to a machine ID. Use the new loader method `sg_get_machine_id()` to obtain a machine ID on your client machine and then specify the machine ID on the Lock screen in GUI or use the new command line option `--machine-id` of the encoder or licgen tool. Encoded PHP scripts locked to a machine ID will only run on that machine (or machines if you specify multiple machine IDs). Please refer to the [Locking options](#) section for further information.

Note: as `sg_get_machine_id()` is a binary method of the loader, an appropriate loader must be installed to the client machine in order for this method to be available from your code. We recommend that you create a mini project, encode it and include the loaders for obtaining the machine ID from the client, in that case loaders will be found automatically as for any other PHP script encoded with SourceGuardian.

- We are also introducing a new locking option for CLI PHP scripts. It was always a problem to lock such the scripts as neither IP nor domain locking may be used for them. Locking to MAC address was only a solution in that case, but it's not always convenient to lock to MAC addresses. Now you may use a special verification URL to validate the CLI script and make sure it works on the same machine as your web based part of the project.

For this to work you need to do two things:

1) Create a special PHP script which is accessible via HTTP protocol, it will be used to validate the machine. The only directive you need to put into it is:

```
<?php echo sg_get_verification_id(); ?>
```

Note: if you use any frameworks, you may need to use another mechanism for sending a string to the output instead of `echo`

2) When encoding your CLI script use the new `--remote-verification-url` option to specify the full URL to your web verification script and use the same project ID as for the web part of your code and particularly the verification script.

Note 1: as the `--remote-verification-url` is mostly used only for locking of the encoded CLI scripts, consider separate encoding of the web part of your project and CLI scripts. You may create two GUI projects for that or use the command line encoder. Use the same project IDs/Keys for both!

Note 2: You may use standard locking to IP or domain for your web verification script (as usual for this to work your web server must send the information about current IP and domain to PHP via environment). Anyway, you "lock" to the domain if use the domain name in the verification URL or lock to IP if you use IP in the URL - choose the way which suits your project and the deployment process.

However, if your CLI PHP script works on its own and is not a part your your web based project, then you still may use the new [machine ID locking](#) option for it as well as good old locking to MAC addresses.

Please refer to the [Locking options](#) section for further information about using of the remote verification URL option.

- Option to add custom auto-globals. For CLI use: `--auto-global MYAUTO` If you use GUI, add this CLI option to 'additional command line options' in [Advanced options](#).

- Options to ignore IP/domain check for the scripts running with CLI PHP. It means if you have encoded the entire project with locking to a IP/domain and selected the option to ignore the IP/domain check for CLI, your protected files will require to be run under the specified IP/domain(s) for web but the same files will be run OK with CLI PHP. This simplifies running PHP CLI scripts like cron jobs but still have them encoded in the same project along with web scripts. However, you may consider the risk of using this option as it does what it does - lets run your IP/domain locked encoded files with PHP CLI bypassing the domain check. For CLI use: `--ip-ignore-cli`, `--domain-ignore-cli` or the appropriate tickboxes on the [Lock screen](#) in GUI. Regardless these are the new options, please consider locking CLI scripts to the [verification URL](#) instead.
- PHP doc comments are now removed by default. Use `--keep-doc-comments` option to keep them e.g. if the framework you use require doc comments. We added an appropriate tickbox to the [Advanced options](#) window in GUI.
- Error text for `sourceguardian.restrict_unencoded=1` updated to include the file name

Error code [08] text changed to:

PHP Fatal error: SourceGuardian Loader - This script ... or other unencoded scripts cannot run in this environment. Please contact the author of the script regarding this problem. [08]

Error code [15] text changed to:

PHP Fatal error: SourceGuardian Loader - This protected script ... can run only in conjunction with other encoded files of the same project. Please contact the author of the script regarding this problem. Error code [15]

GUI updates

- File associations work now on Windows and Mac OS. It means clicking a .sg project file in Explorer/ Finder will launch SourceGuardian and opens this project in GUI. Note, for this to work, SourceGuardian must be installed using the installer on Windows, on Mac OS, you need to launch SourceGuardian at least once for file associations to start working.
- Hex registration code is now displayed in Help/Registration information which is useful if you are using multiple installations of SourceGuardian and need to manage or reset a particular license in the online user profile, now you may easily find it there.
- You may access your SourceGuardian online user profile directly from the application, click Help/ SourceGuardian User Profile.

CLI updates

- `--days` option was replaced with `--expire 00d 00m 00h 00s`
- New command line options described above: `--ip-ignore-cli`, `--domain-ignore-cli`, `--machine-id`, `--machine-id-encrypt`, `--remote-verification-url`, `--keep-doc-comments`, `--auto-global`

Bug Fixes

- using `self::` from within nested function which in its turn is defined in a private member was not working