PHP static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your PHP code

be used to end lines

Code Smell

More than one property should not be declared per statement

Code Smell

The "var" keyword should not be used

Code Smell

"<?php" and "<?=" tags should be used

Code Smell

File names should comply with a naming convention

Code Smell

Comments should not be located at the end of lines of code

Code Smell

Local variable and function parameter names should comply with a naming convention

Code Smell

Field names should comply with a naming convention

Code Smell

Lines should not end with trailing whitespaces

Code Smell

Files should contain an empty newline at the end

Code Smell

Modifiers should be declared in the correct order

Code Smell

An open curly brace should be located at the beginning of a line

Code Smell

Static members should be referenced with "static::"

Analyze your code

Bug Minor pitfall

References in a class to static class members (fields or methods) can be made using either `self::$var` or `static::$var` (introduced in 5.3). The difference between the two is one of scope. Confusingly, in subclasses, the use of `self::` references the original definition of the member, i.e. the superclass version, rather than any override at the subclass level. `static::`, on the other hand, references the class that was called at runtime.

Noncompliant Code Example

```
<?php

class Toy {

    public static function status() {
        self::getStatus();  // Noncompliant; will always pri
    }

    protected static function getStatus() {
        echo "Sticks are fun!";
    }
}

class Ball extends Toy {

    protected static function getStatus() {  // Doesn't actu
        echo "Balls are fun!";
    }
}

$myBall = new Ball();
$myBall::status();  // Prints "Sticks are fun!"
```

Compliant Solution

```
<?php

class Toy {

    public static function status() {
        static::getStatus();  // Compliant
    }

    protected static function getStatus() {
        echo "Sticks are fun!";
    }
}

class Ball extends Toy {

    protected static function getStatus() {
        echo "Balls are fun!";
    }
}

$myBall = new Ball();
$myBall::status();  // Prints "Balls are fun!"
```

Exceptions

No issue is raised when `self` is used on a constant field, a private field or a private method.

```
class A
{
    private static $somevar = "hello";
    const CONSTANT = 42;

    private static function foo()
    {
        $var = self::$somevar . self::CONSTANT;  // Should b
        self::foo();                               // Should
    }
}
```

Available In:

sonarlint sonarcloud sonarqube

An open curly brace should be located at the end of a line

Code Smell

Tabulation characters should not be used

Code Smell

Method and function names should comply with a naming convention

Code Smell

Creating cookies with broadly defined "domain" flags is security-sensitive

Security Hotspot