Error messages leading to potential data exfiltration

Low **romm** published **GHSA-5pgm-3j3g-2rc7** on Jul 10

Package

php **cuyz/valinor** (Composer)

Affected versions	Patched versions
<0.12.0	0.12.0

Description

```php
<?php

namespace My\App;

use CuyZ\Valinor\Mapper\MappingError;
use CuyZ\Valinor\Mapper\Tree\Node;
use CuyZ\Valinor\Mapper\Tree\NodeTraverser;
use CuyZ\Valinor\MapperBuilder;

require_once __DIR__ . '/Valinor/vendor/autoload.php';

final class Money
{
    private function __construct(public readonly string $amount)
    {
    }

    public static function fromString(string $money): self
    {
        if (1 !== \preg_match('/^\d+ [A-Z]{3}$/', $money)) {
            throw new \InvalidArgumentException(\sprintf('Given "%s" is not a recognized monetar
        }

        return new self($money);
    }
}

class Foo
```

```
{
    public function __construct(
        private readonly Money $a,
        private readonly Money $b,
        private readonly Money $c,
    ) {}
}

$mapper = (new MapperBuilder())
    ->registerConstructor([Money::class, 'fromString'])
    ->mapper();

try {
    var_dump($mapper->map(Foo::class, [
        'a' => 'HAHA',
        'b' => '100 EUR',
        'c' => 'USD 100'
    ]));
} catch (MappingError $e) {
    $messages = (new NodeTraverser(function (Node $node) {
        foreach ($node->messages() as $message) {
            var_dump([
                '$message',
                $message->path(),
                $message->body()
            ]);
        }
        return '';
    }))->traverse($e->node());

    iterator_to_array($messages);
}
```

Now, this is quite innocent: it produces following output:

```
❯ php value-object-conversion.php
array(3) {
  [0]=>
  string(8) "$message"
  [1]=>
  string(1) "a"
  [2]=>
  string(48) "Given "HAHA" is not a recognized monetary amount"
}
array(3) {
  [0]=>
  string(8) "$message"
  [1]=>
  string(1) "c"
  [2]=>
  string(51) "Given "USD 100" is not a recognized monetary amount"
}
```

The problem is that nowhere I told valinor that it could use `Throwable#getMessage()`.

This is a problem with cases where you get:

- an SQL exception showing an SQL snippet
- a DB connection exception showing DB ip address/username/password
- a timeout detail / out of memory detail (exploring DDoS possibilities)

This allows for potential data exfiltration, DDoS, enumeration attacks, etc.

Severity

Low

CVE ID

CVE-2022-31140

Weaknesses

No CWEs