Variable rewrite with boolean and GMP Type Confusion

To build exploit for GMP Type Confusion bug we need to find in code:
1) Class that implements Serializable interface
2) Code line executed from __destruct method to rewrite object property
3) Find an object to rewrite props

Symfony package “symfony/dependency-injection” was taken for analysis.
$ cat composer.json
```
{
   "require": {
      "psr/container":"1.0.0",
      "symfony/dependency-injection": "3.4.47",
      "symfony/routing": "3.4.47"
   }
}
```
$ composer install

Package “symfony/dependency-injection” has small number of __destruct methods. And it has no code line to write property field into another property field, reachable from __destruct.
But package has line:
$this->removedBindingIds[(int) $bindingId] = true;
in removeBindings method.
In PHP, boolean variable is represented in memory as 0 or 1 integer. It is enough to rewrite handle of GMP object with value 0x1. In the finish of GMP deserialization we rewrite Composer object props, because it has handle = 0x1. Exploit build process is very similar with exploitation discussed in previous advisory.

File symfony/dependency-injection/Loader/Configurator/ServiceConfigurator.php

```
58      public function __destruct()
59      {
60          parent::__destruct();
61
62          $this->container->removeBindings($this->id);
63
64          if (!$this->definition instanceof ChildDefinition) {
65              $this->container->setDefinition($this->id, $this->definition->setInstan
66          } else {
67              $this->container->setDefinition($this->id, $this->definition);
68          }
69      }
```

File symfony/dependency-injection/ContainerBuilder.php

```
1533      public function removeBindings($id)
1534      {
1535          if ($this->hasDefinition($id)) {
1536              foreach ($this->getDefinition($id)->getBindings() as $key => $binding) {
1537                  list(, $bindingId) = $binding->getValues();
1538                  $this->removedBindingIds[(int) $bindingId] = true;
1539              }
1540          }
1541      }
```

File symfony/routing/Route.php

```
 86      public function unserialize($serialized)
 87      {
 88          $data = unserialize($serialized);
 89          $this->path = $data['path'];
 90          $this->host = $data['host'];
 91          $this->defaults = $data['defaults'];
 92          $this->requirements = $data['requirements'];
 93          $this->options = $data['options'];
 94          $this->schemes = $data['schemes'];
 95          $this->methods = $data['methods'];
 96
 97          if (isset($data['condition'])) {
 98              $this->condition = $data['condition'];
 99          }
100          if (isset($data['compiled'])) {
101              $this->compiled = $data['compiled'];
102          }
103      }
```

Generated exploits can be found here.