《软件安全》实验报告

姓名：梁婧涵 学号：2112155

**实验名称：**

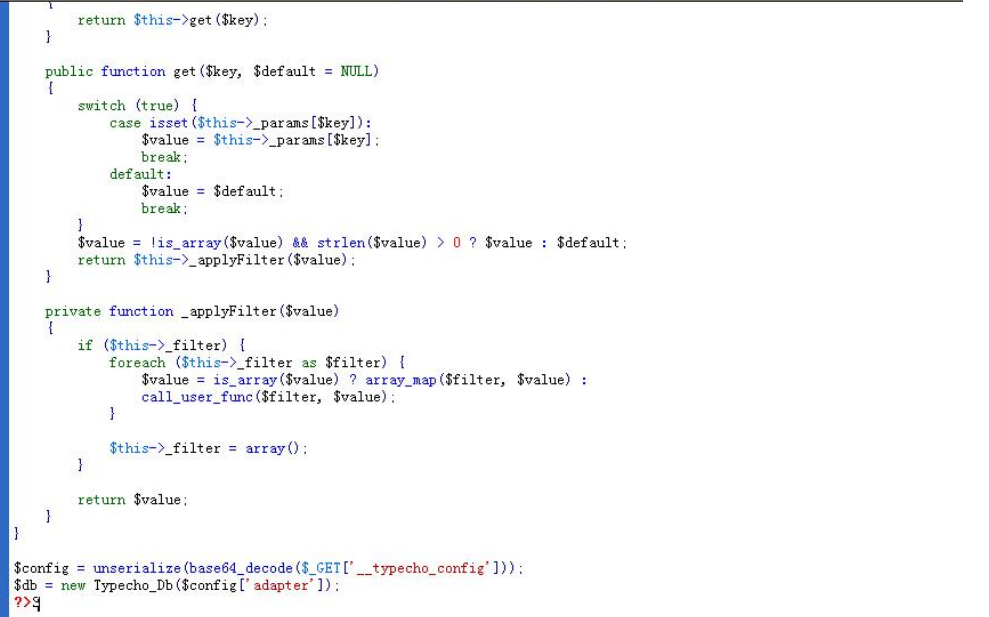
PHP反序列化漏洞

**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：**

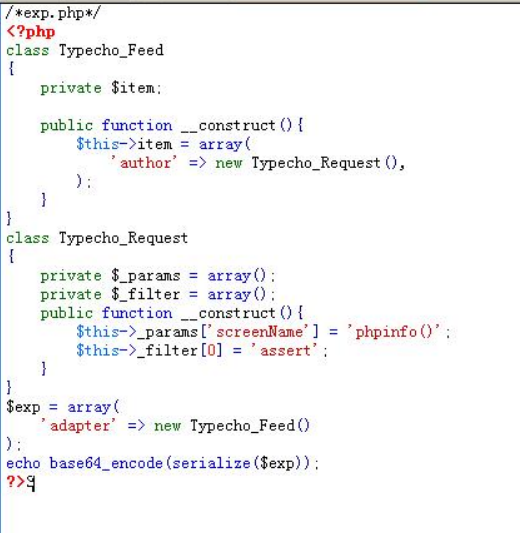
typecho.php代码复现



可以看到，此代码的unserialize的参数是用GET从用户处获得的，那么就满足了反序列化漏洞的基本条件。

程序实例化了类 Typecho\_Db，类的参数是通过反序列化得到的$config。在类 Typecho\_Db 的构造函数中，进行了字符串拼接的操作，而在 PHP 魔术方法中，如果一个类 被当做字符串处理，那么类中的\_\_toString()方法将会被调用。全局搜索，发现类 Typecho\_Feed 中存在\_\_toString()方法。 在类 Typecho\_Feed 的\_\_toString()方法中，会访问类中私有变量$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

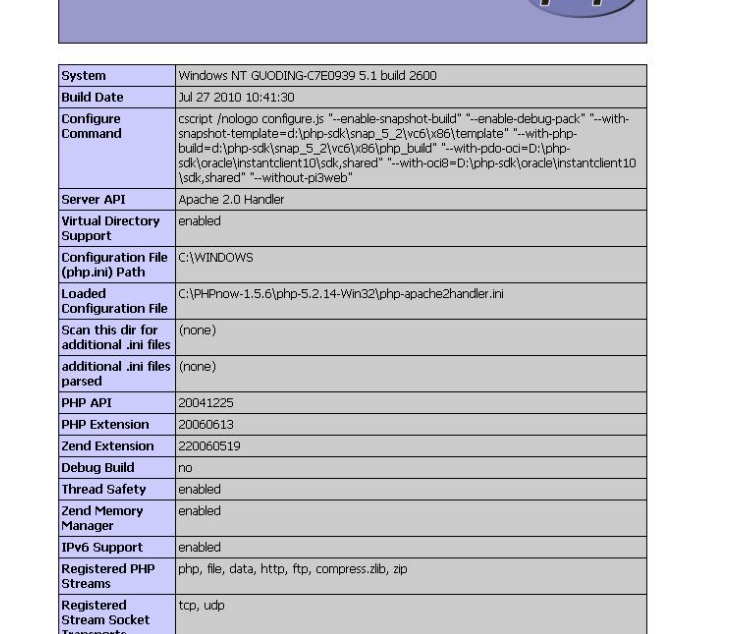
Typecho\_Request 类中，正好定义了\_\_get()方法。 类 Typecho\_Request 中的\_\_get()方法会返回 get()，get()中调用了\_applyFilter()方法，而在 \_applyFilter()中，使用了 PHP 的 call\_user\_function()函数，其第一个参数是被调用的函数，第 二个参数是被调用的函数的参数，在这里$filter，$value 都是我们可以控制的，因此可以用 来执行任意系统命令。至此，一条完整的利用链构造成功。



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();是我们要执行的命令，同理的，想执行其他的命令只需要更换此部分即可

get请求的方式传递给typecho.php,成功执行phpinfo



'fopen(\'newfile.txt\', \'w\');'，依然先访问exp.php得到内容，再用get传给typecho.php，运行后再htdocs文件夹下发现

**心得体会：**

通过本次实验，初步理解了PHP反序列化漏洞的原理，并且初步学习了如何对其进行利用。在未来的编程中也会进一步注意防范此类漏洞