## Web入口 192.168.5.11

1. web1.py ，升级olimometer或者停用olimometer插件

监控Olimometer 2.54插件状态，如果插件删除，加分但百分比少，如果升级则加分较多

“olimometer\_id”参数使用SQL方法，我们能够从example.com实例网站获得数据库名称。如果有这样的漏洞，使用不同的参数运行SQLmap，我们可以得到完整的数据库，包括用户名和密码。

https://wordpress.org/plugins/olimometer/installation/下载2.57版本

1. web2.py，监控test下的文件，如果存在就测试文件是否可以上传，如果可以上传就代表漏洞未修复，如果文件删除了也代表修复，加分，但第一种修复方案加分较多
2. Web3py 检测192.168.5.11的22端口弱口令，只判断口令是否修改。

## FTP 192.168.5.13

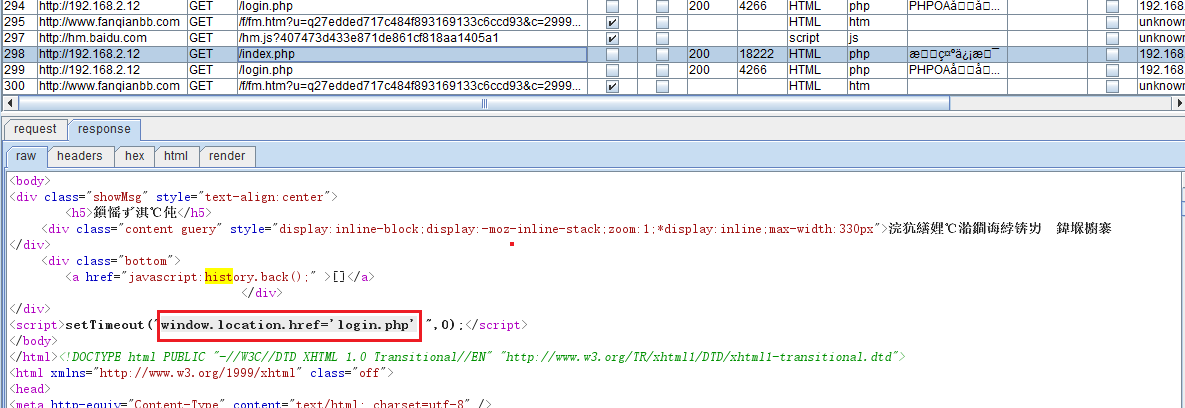
4.ftp.py监控ftp的弱口令是否修改，如果修改就代表修复

## Oa 192.168.2.12

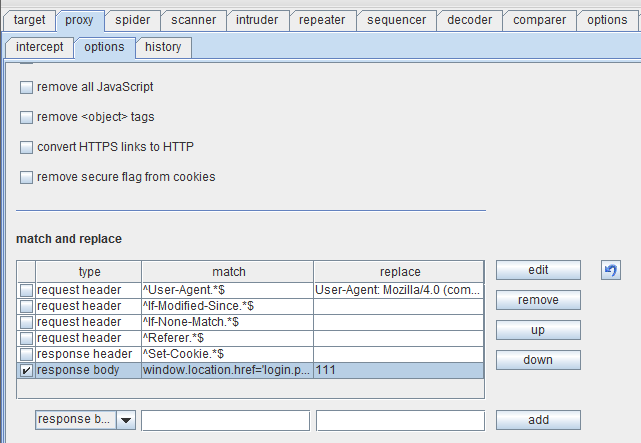
1. oa1.py，监控oa权限绕过漏洞是否修复，开启检查权限

function\_global.php

之前是//exit()权限可绕过的，绕过需用burp抓包



将框内复制到options的替换处

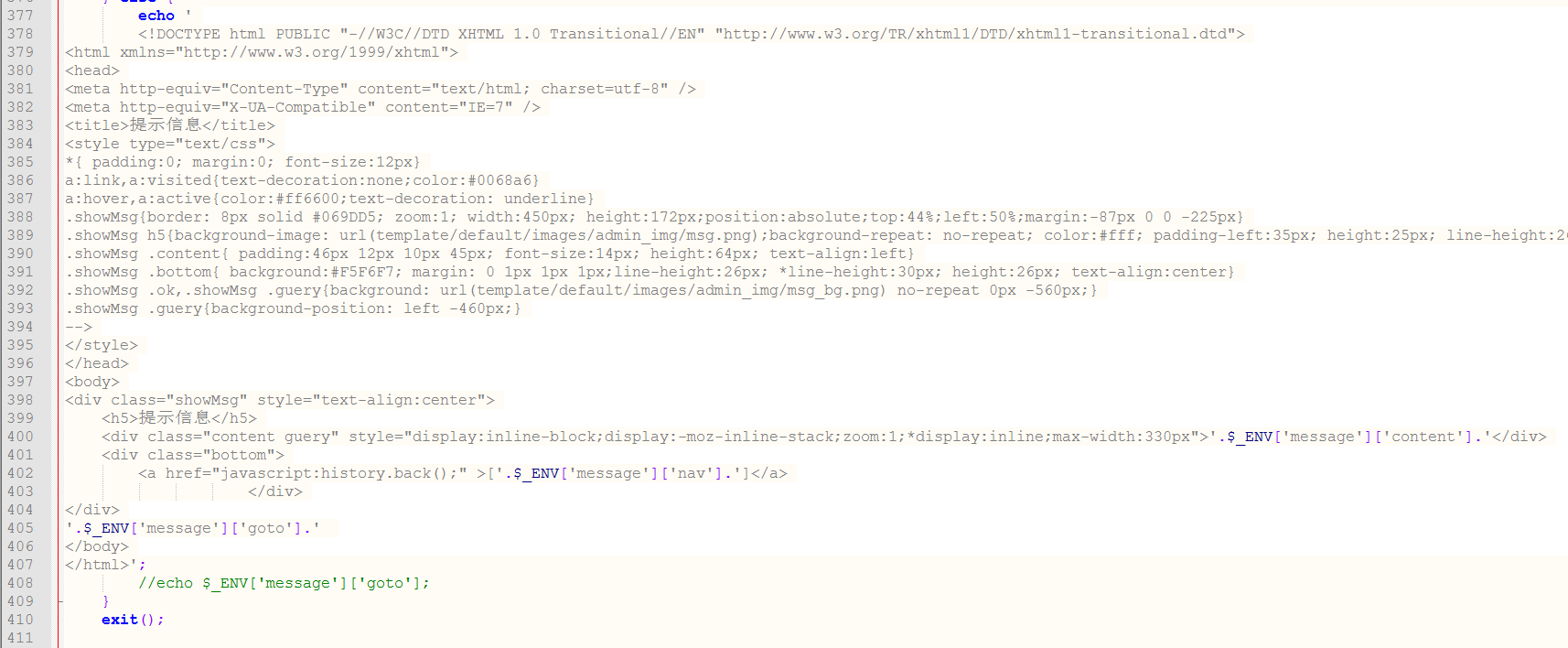


修改前：



修改方法：去掉function\_global.php里exit();前的注释，使无法绕过，则加分

修改后：



1. oa2.py

监控upload\_file.php文件上传是否修复，因为是正常功能文件，所以不能删除这个文件，如果删除就扣分

源文件的验证为

if( ( $uploaded\_type == "image/jpeg" || $uploaded\_type == "image/png" ) &&

( $uploaded\_size < 100000 ) &&

getimagesize( $uploaded\_tmp ) ) {

所以只需要加上后缀的验证即可例如下边的代码或者不验证type类型也是可以的，也就是黄色的部分可以没有。

if(( $uploaded\_type == "image/jpeg" || $uploaded\_type == "image/png" ) && ( strtolower( $uploaded\_ext ) == "jpg"|| strtolower( $uploaded\_ext ) == "jpeg" || strtolower( $uploaded\_ext ) == "png" ) &&

( $uploaded\_size < 100000 ) &&

getimagesize( $uploaded\_tmp ) ) {

1. oa3.py，监控test.php文件是否删除，或者修改密码，或者清空，不管如何操作，修改比删除加分多
2. 3389.py，修改pc密码 192.168.3.11 PC0