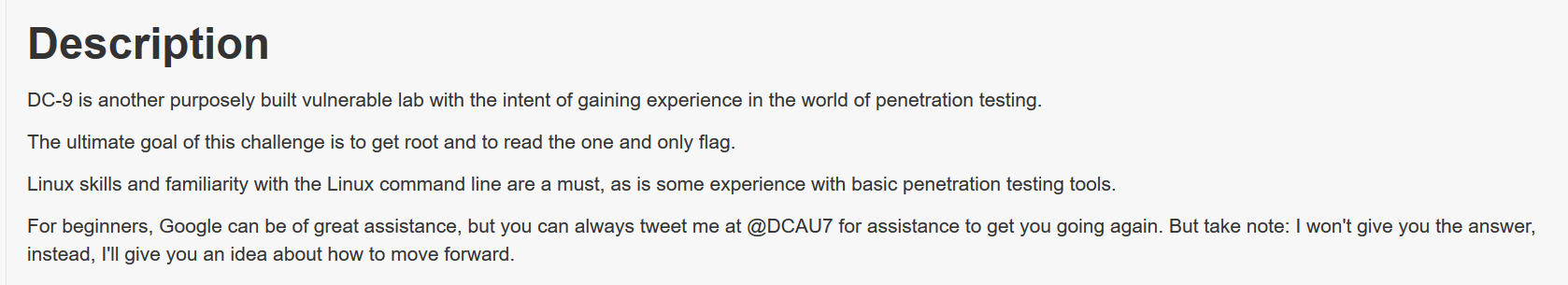
# 0x00:靶机下载

下载链接: <https://www.vulnhub.com/entry/dc-9,412/>

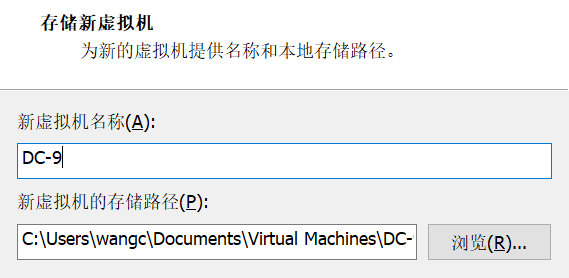
靶机的目标是获得root权限



下载完以后解压是个.ova文件



点击打开选择一个合适的地方安装



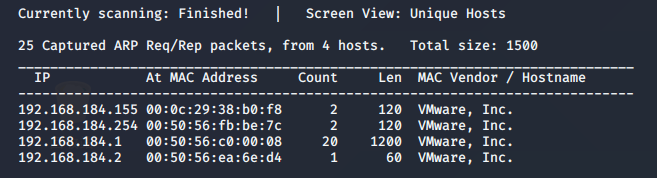
需要调整把桥接模式改为NAT模式



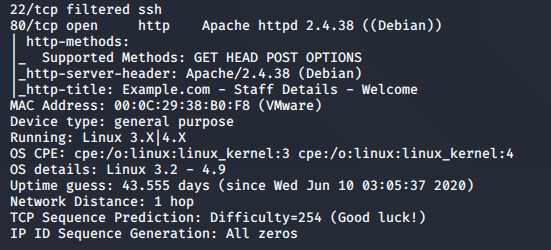
# 0x01:探测靶机

首先使用netdiscover探测靶机IP，排除.1,.2,和.254可以确认靶机IP为192.168.184.155

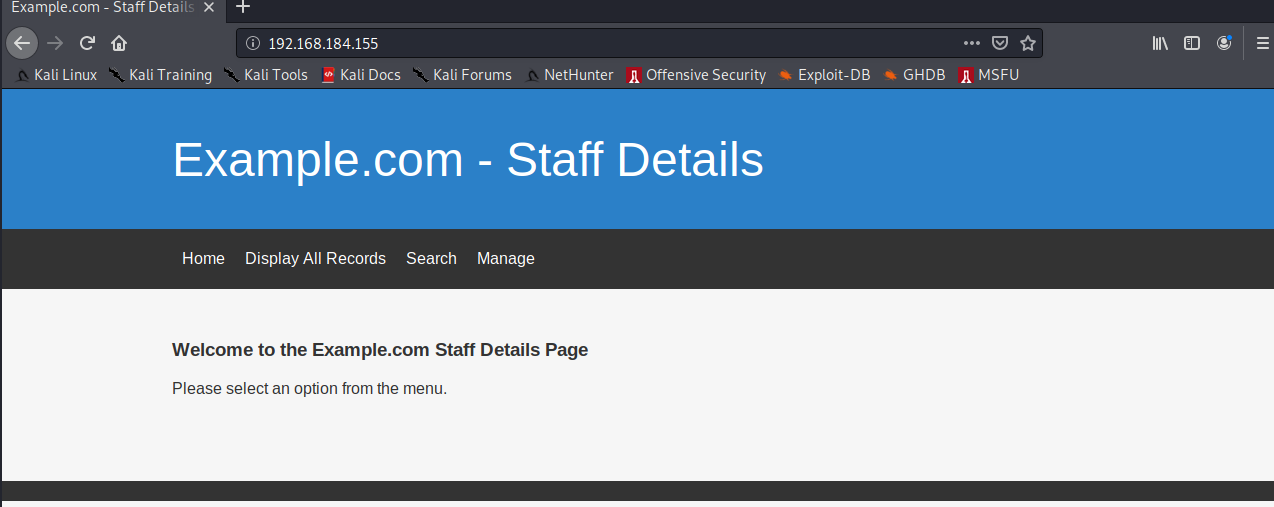
命令:netdiscover -r 192.168.184.0/16



开始用nmap探测端口，开了22和80，先排除22，有过滤。

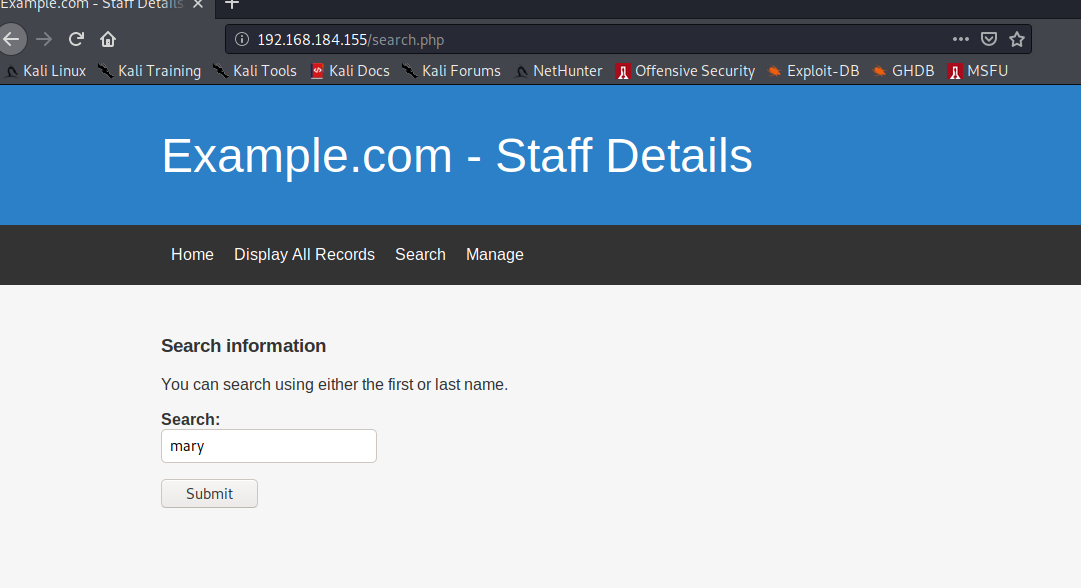


来到80端口以后可以正常访问

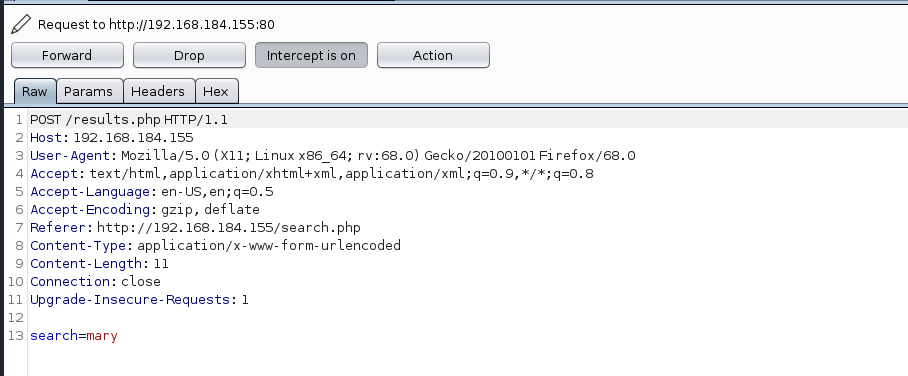


# 0x02:挖掘漏洞

来到搜索页面，正常的输入就是输入一个人名，然后判断是否存在有的话就回显，没有的话就显示空，同时网页URL可能会变成?search=xxxx这样子，然而没有，尝试用Burp抓包

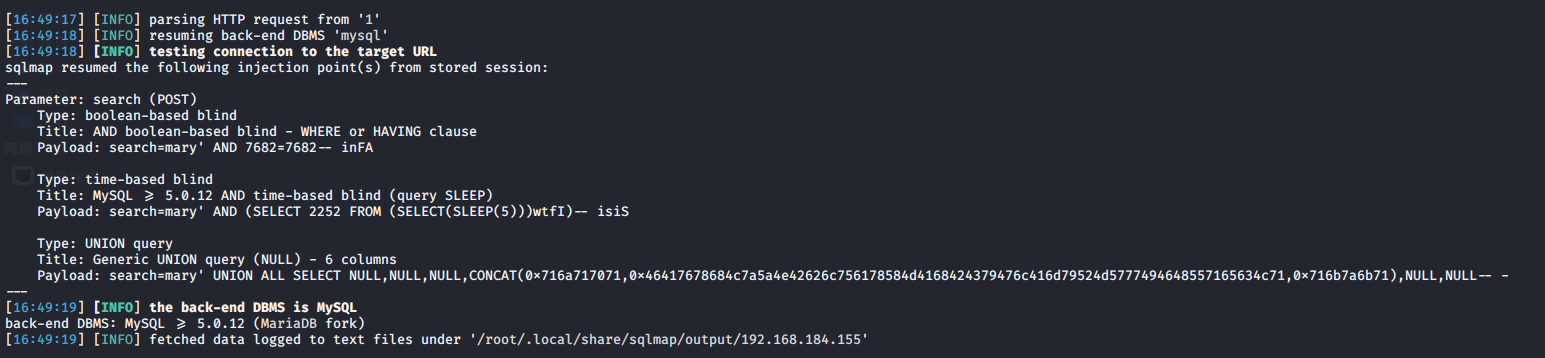


尝试用Burp可以抓到search=mary参数出来，复制数据放到txt文本里

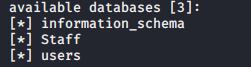


然后用kali的SQLMAP可以跑出来

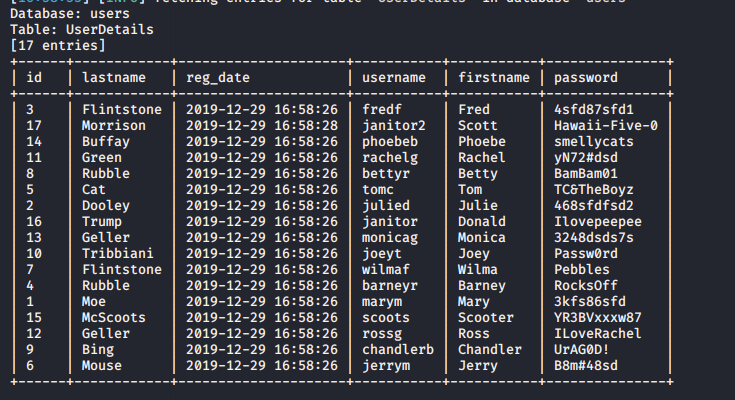
命令:sqlmap -r 1 –-batch –-level 3



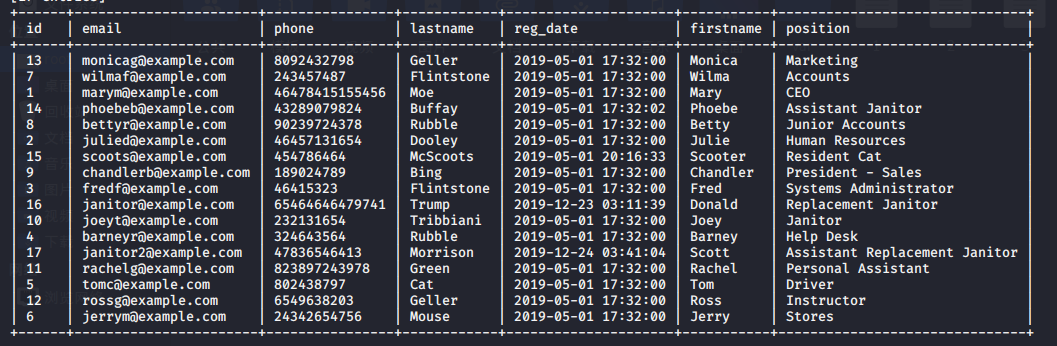
输入—dbs可以跑出数据库名出来

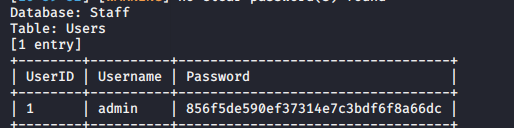


在user数据库里面只有一个UserDetails表里面有大量账号密码

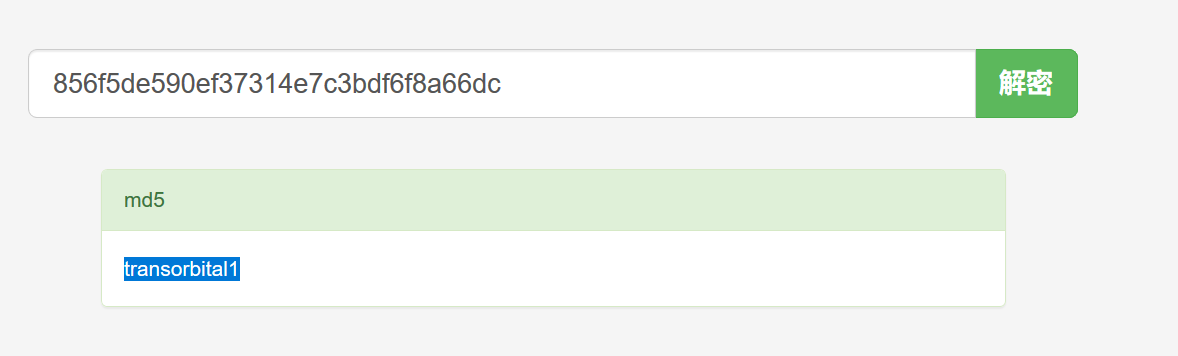


而在Staff数据库里面，有两个表，同时在Users表里面包含admin关键词

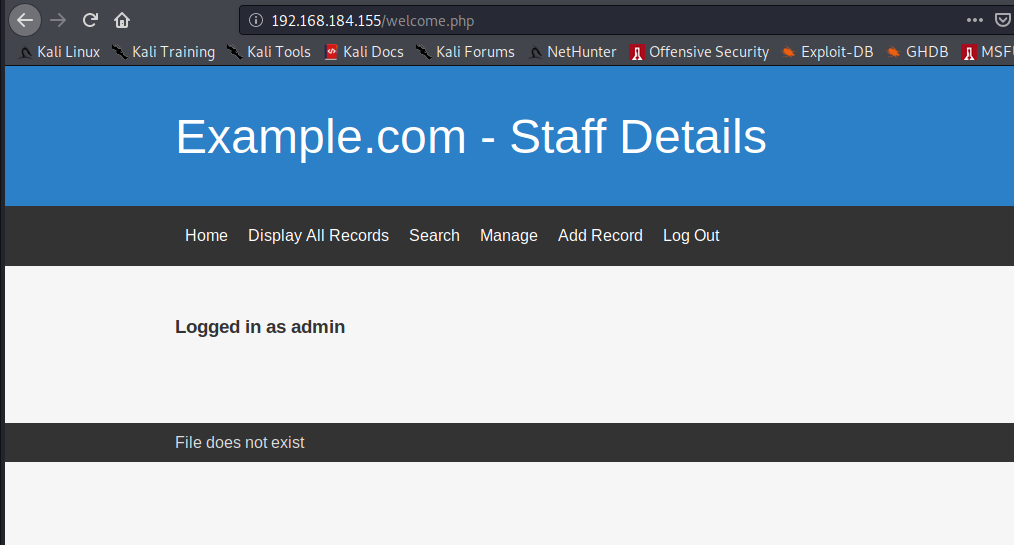




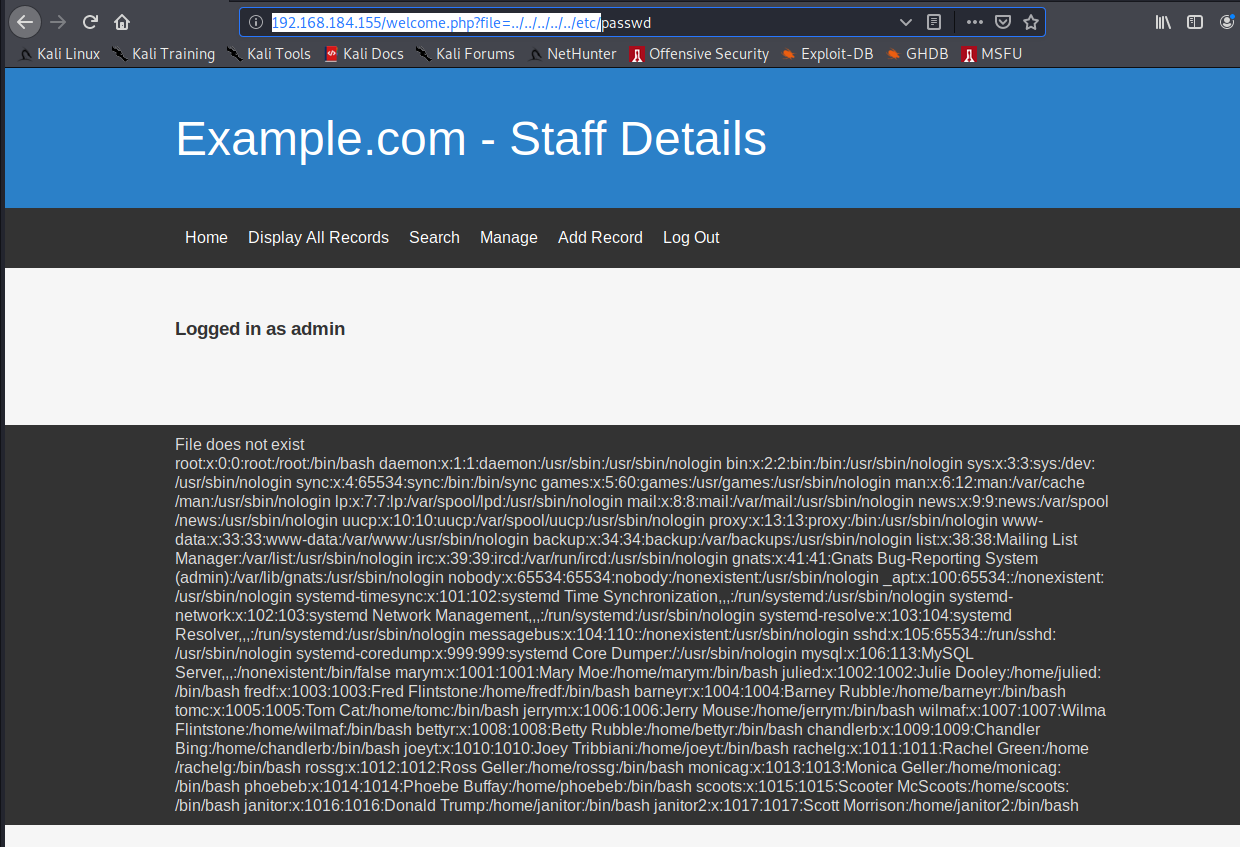
通过在线MD5解密得到密码



成功登录后台

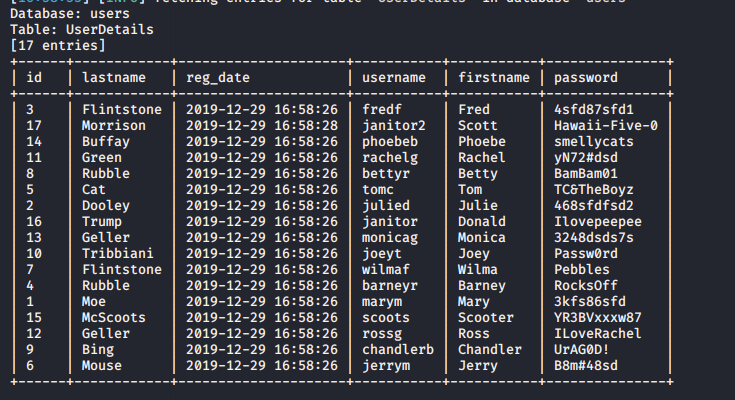


登录后台以后有发现有个提示File does not exist这里猜测莫非是本地文件包含漏洞尝试使用/etc/passwd 和控制../的数量，最后爆出很多账号来



# 0x03:爆破22端口

其中有些账号与刚才用SQLMAP抄库到的username有重合



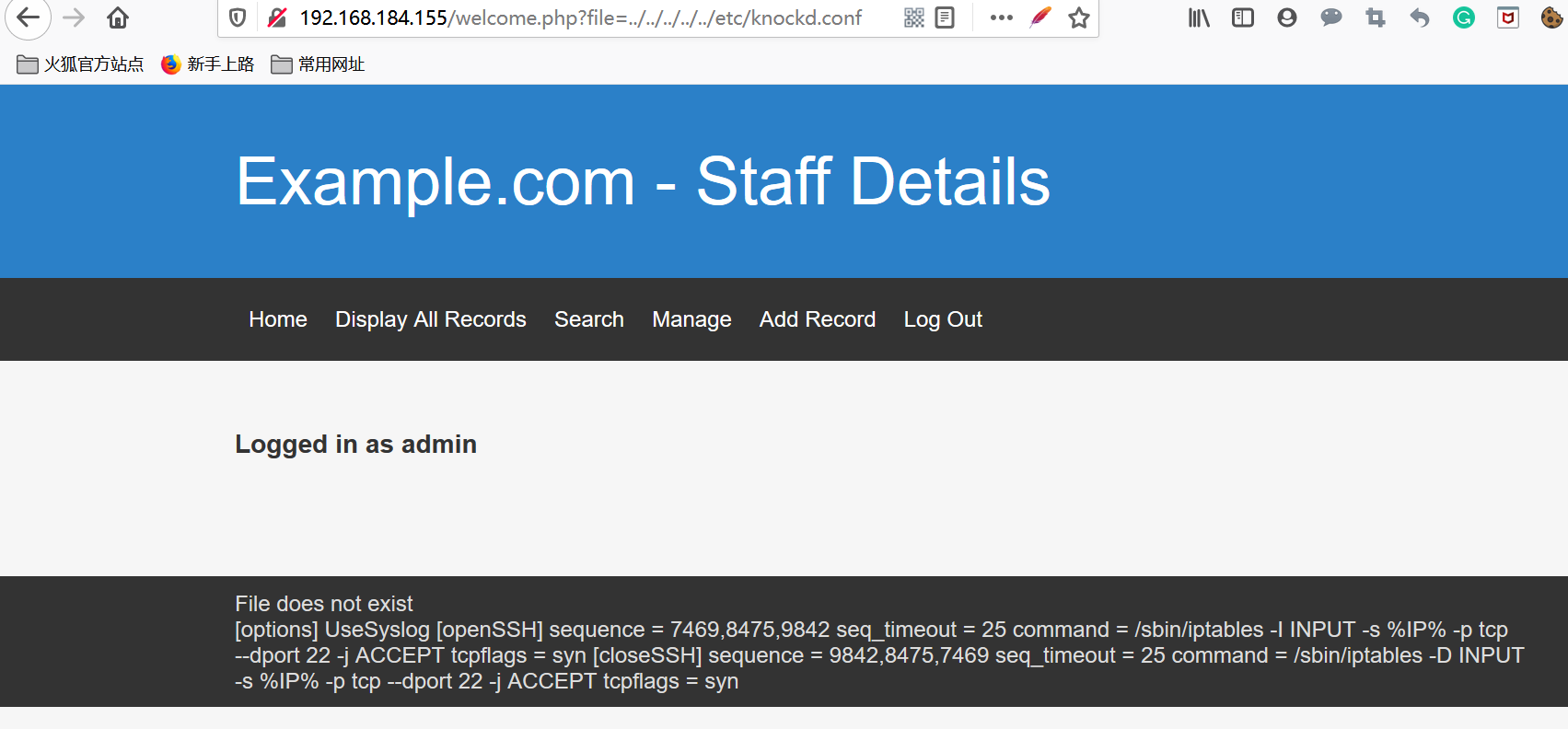
那就把username和password分成两个文件保存下来



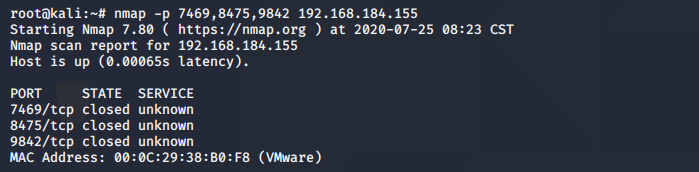
接下来的步骤应该就是爆破22端口了，然而得注意到一开始用nmap扫的时候22端口是一个处于过滤状态。并不是处于开放的状态。这里靶机可能对于22端口做了一定的防护。



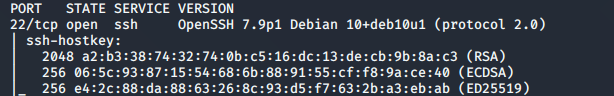
这里我查阅了一下资料<https://www.ibm.com/developerworks/cn/aix/library/au-sshlocks/index.html>，总结一下就是有种技术叫端口敲门。只有敲对了几个端口号以后22端口才能重新处于开放状态。而需要敲哪几个特定的端口是/etc/knockd.conf决定的。这里直接访问/etc/knockd.conf,可以得知端口为7469，8475和9842



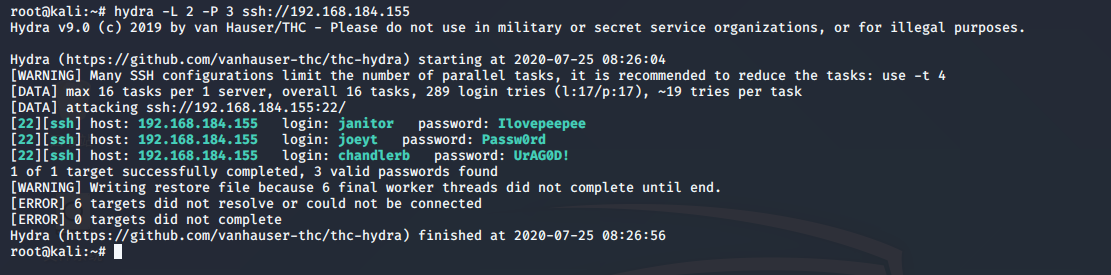
这里直接用nmap扫描这3个端口



在重新扫一遍发现22端口已经变为开放状态

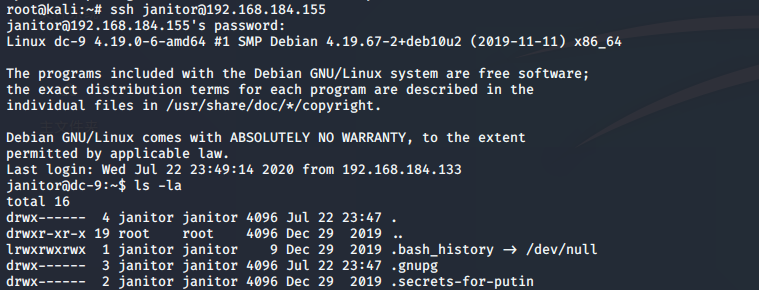


用hydra可以爆出3个用户和密码出来

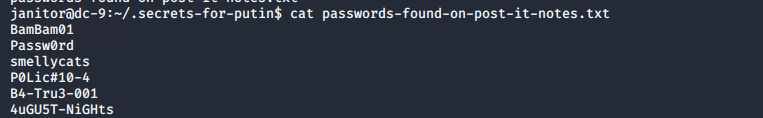


# 0x04:挖掘信息

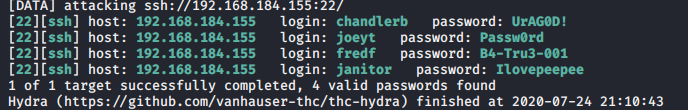
这里我用janitor账号可以成功登录然后用ls -la发现一些隐藏目录。这里跟进.secrets-for-putin



跟进了一下发现有个passwords-found-on-post-it-notes.txt文件，可能还是密码，放回密码本里

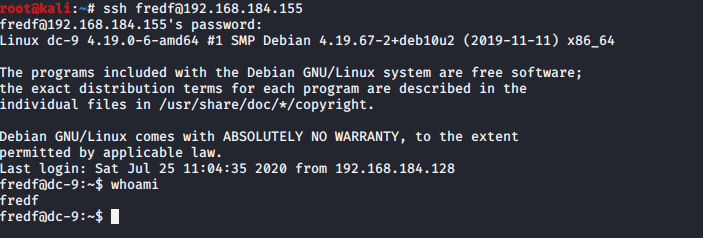


继续重新跑一次，这次跑多一个账号出来fredf

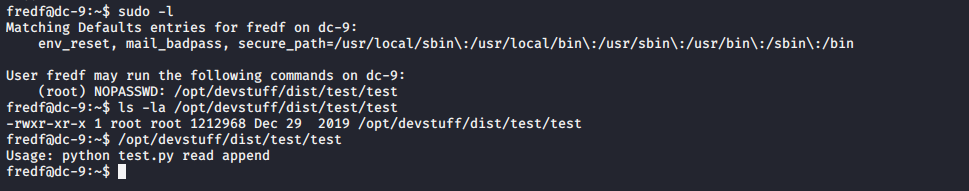


# 0x04:提权

用SSH输入账号密码后成功登录



然后用sudo -l,发现可以已root权限运行/opt/devstuff/dist/test/test 的文件，而且给了一个用法 read append。有个想法就是既然可以以root权限运行read append,理论上是可以添加一个root权限的账号来读取flag的





同时在跳回上两级目录可以发现test的脚本内容

先用kali本地的perl进行加密一波



然后随意构造一个用户名再把刚才得到的加密值作为root用户的格式放在/tmp下



开始启用脚本



输入刚才的密码成功读取flag

