

中南大学

网络安全线上实验报告 2

题 目 包嗅探和欺骗

学生姓名 林丹丹

指导教师 王伟平

学 院 信息科学与工程学院

专业班级 信息安全 1401 班

二〇一六 年 十二 月

目录

一. 概述.....	1
二. 实验任务.....	1
1.任务 1: 写一个包嗅探程序.....	1
问题.....	3
2. 任务 2: 欺骗.....	9
问题.....	12
三. 实验结果.....	12
四. 实验心得.....	12

包嗅探和欺骗

一. 概述

包嗅探和欺骗是网络安全中两个重要概念，是网络通信中两大主要威胁。能够理解这两种威胁对于我们理解网络安全措施至关重要。有很多的包嗅探和欺骗工具，比如 Wireshark、Tcpdump、Netwox 等。某些工具被安全专家们以及攻击者广泛使用。对学生而言，能够使用这些工具很是重要，但是对网络安全课程上的学生而言，更重要的是了解这些工具的工作原理，即包嗅探和欺骗是如何在软件上实现的。

本次实验的目的是为了让学生掌握大多数嗅探和欺骗工具的低层技术。学生应操作一些简单的嗅探和欺骗程序，阅读它们的源代码，最终能够深入了解这些程序的技术知识。本次实验结束后，学生应能够编写自己的嗅探和欺骗程序。

二. 实验任务

1.任务 1：写一个包嗅探程序

用 pcap 库可以很容易地编写出嗅探器程序。在 pcap 库下，嗅探器的任务变成在 pcap 库中调用简单序列的过程。序列的末端，数据包一旦被捕获，就会被放入缓冲中以待进一步处理。数据包捕获的所有细节都由 pcap 库处理。Tim Carstens 已经写好了一个关于如何使用 pcap 库编写嗅探器程序的教程。教程可在 <http://www.tcpdump.org/pcap.htm> 中下载。

任务 1.a：理解嗅探器。请从上面内容提到的地方下载 sniffex.c 程序，编译并运行它。提供截图以证明程序运行成功且产生了预期的结果。

实验过程：

通过调用命令 `gcc -o sniffex sniffex.c -l pcap`，编译 sniffex.c 程序，编译成功后，调用命令 `sudo ./sniffex` 运行这一嗅探器。

实验截图如下所示：

刷新浏览器，嗅探器将捕捉到的 10 个 ip 包

```
Capture complete.
[11/16/2016 04:48] seed@ubuntu:~/Documents/test$ sudo ./sniffex
sniffex - Sniffer example using libpcap
Copyright (c) 2005 The Tcpdump Group
THERE IS ABSOLUTELY NO WARRANTY FOR THIS PROGRAM.
```

```
Device: eth0
Number of packets: 10
Filter expression: ip
```

```
Packet number 1:
  From: 192.168.181.130
  To: 220.181.76.66
  Protocol: TCP
  Src port: 52916
  Dst port: 80
```

```
Packet number 2:
  From: 192.168.181.130
  To: 220.181.76.66
  Protocol: TCP
  Src port: 52917
  Dst port: 80
```

```
Packet number 3:
  From: 220.181.76.66
  To: 192.168.181.130
  Protocol: TCP
  Src port: 80
  Dst port: 52916
```

```
Packet number 4:
  From: 192.168.181.130
  To: 220.181.76.66
  Protocol: TCP
  Src port: 52916
  Dst port: 80
```

```
Packet number 5:
  From: 192.168.181.130
```

```
      From: 192.168.181.130
      To: 14.215.177.38
      Protocol: TCP
      Src port: 39249
      Dst port: 443
      Payload (378 bytes):
00000 16 03 01 01 75 01 00 01 71 03 01 58 2d 10 e6 2d ....u...q..X...-
00016 fb e0 19 03 43 80 ab 1c 17 05 58 b0 ea 21 40 92 ...C....X...!@.
00032 79 a7 3c 13 b8 17 86 4e 70 19 51 20 92 7c ac cc y.<...Np.Q..|..
00048 7f a8 96 cf b5 3e a3 b6 ee 36 82 31 e2 06 65 40 .....>..6.1..e@
00064 77 8e 28 70 d9 1e 73 9f 3e 89 61 69 00 48 00 ff w.(p..s.>.al.H..
00080 c0 0a c0 14 00 88 00 87 00 39 00 38 c0 0f c0 05 .....9.8....
00096 00 84 00 35 c0 07 c0 09 c0 11 c0 13 00 45 00 44 ...5.....E.D
00112 00 33 00 32 c0 0c c0 0e c0 02 c0 04 00 96 00 41 .3.2.....A
00128 00 05 00 04 00 2f c0 08 c0 12 00 16 00 13 c0 0d ...../.....
00144 c0 03 fe ff 00 0a 01 00 00 e0 00 00 00 12 00 10 .....
00160 00 00 0d 77 77 77 2e 62 61 69 64 75 2e 63 6f 6d ...www.baidu.com
00176 00 0a 00 08 00 06 00 17 00 18 00 19 00 0b 00 02 .....
00192 01 00 00 23 00 b0 d7 b2 41 aa c2 e9 b1 6e a2 f0 ...#....A....n..
00208 85 8b 9b dc da b5 c3 15 24 d9 dc f9 f8 47 a1 47 .....$....G.G
00224 7d 08 56 0d 05 45 5e cd 5e 13 06 0a 06 1d 02 08 }.VneE^..fj..b.
00240 51 21 36 22 8b d3 e7 22 9b 5f e2 17 d9 36 4b 7b Q!6".....6K{
00256 f8 db 10 c7 26 16 0b c3 94 93 42 33 92 2b 62 56 ...&.k...B3.+bv
00272 24 c9 ed c5 65 38 9a 90 05 9d ee 97 45 a5 ec c3 $.e8.....E...
00288 29 2a 00 83 f8 d5 24 72 dd 08 f5 f0 63 8f 00 44 )*....$r....C..D
00304 3d 6d e3 3f e5 f9 89 43 91 10 21 42 9d 4e d2 bd -n.2...C...lB.N..
00320 19 43 c2 12 fe 35 5c e6 6d 04 f8 68 57 31 81 a8 .C...5\m...hW1..
00336 e1 82 55 b9 43 91 9d dc bd 0c e7 a0 06 7f 08 a5 ..U.C.....
00352 a0 d7 52 b9 c5 c7 d5 a7 ab bb 60 8e 83 99 48 89 ..R.....'...H.
00368 ef 9e 0f 3c 16 7c 33 74 00 00 ...<.|3t..
```

```
Packet number 7:
  From: 14.215.177.38
  To: 192.168.181.130
  Protocol: TCP
  Src port: 443
  Dst port: 39249
```

```
Packet number 8:
  From: 14.215.177.38
  To: 192.168.181.130
```

```
Packet number 8:
  From: 14.215.177.38
  To: 192.168.181.130
  Protocol: TCP
  Src port: 443
  Dst port: 39249
  Payload (146 bytes):
00000 16 03 01 00 5e 02 00 00 5a 03 01 24 31 8d 8a 9c .....^...Z..$.1...
00016 5a 32 41 7b 62 30 7b 4a 3b 8e a3 b4 16 e4 17 92 ZZA{b0{J;.....
00032 df 0a 8a 3d 36 cf f4 10 94 2b a5 20 92 7c ac cc ...=6....+.|.
00048 7f a8 96 cf b5 3e a3 b6 ee 36 82 31 e2 06 65 40 ....>...6.1..e@
00064 77 8e 28 70 d9 1e 73 9f 3e 89 61 69 c0 11 00 00 w.(p..s.>.a1....
00080 12 ff 01 00 01 00 33 74 00 09 08 68 74 74 70 2f .....3t...http/
00096 31 2e 31 14 03 01 00 01 01 16 03 01 00 24 a1 e0 1.1.....S...
00112 e0 71 06 e8 00 12 c8 2b 20 d5 d1 c9 7b 38 92 c5 .q.....+ ...{8..
00128 92 a4 1d 49 00 77 96 4d b4 89 07 31 7b 90 a8 59 ...I.w.M...1{..Y
00144 b3 b0 ..

Packet number 9:
  From: 192.168.181.130
  To: 14.215.177.38
  Protocol: TCP
  Src port: 39249
  Dst port: 443

Packet number 10:
  From: 192.168.181.130
  To: 14.215.177.38
  Protocol: TCP
  Src port: 39249
  Dst port: 443
  Payload (83 bytes):
00000 14 03 01 00 01 01 16 03 01 00 48 11 ce 6d 2b 17 .....H..m+.
00016 67 5d 28 8f e6 02 dd e9 2a 73 9b 7d cc 33 7c dc gJ{.....*s.}.3|.
00032 d8 d2 29 4c dd b9 cb 28 9e af a2 1e 73 71 7c fc ..)L...(...sq|.
00048 3d bb be 8f 3a 2f da 7e 78 fc 32 bf 4d e6 4c 82 =...:/..~X.2.M.L.
00064 c2 cc 1d 41 0f c3 5f 14 f3 c7 fe 72 09 cf f3 26 ...A.....r...&
00080 f7 57 99 .W.

Capture complete.
[11/16/2016 18:07] seed@ubuntu:~/Documents/test$
```

问题

问题 1: 请使用自己的语言描述对嗅探程序至关重要的库调用顺序。这是总结，而不是像教程中的详细解释。

问题 2: 为什么需要 root 权限运行 sniffex? 如果没有 root 权限，程序会运行到哪一步失败?

回答: 由于是最底层的系统调用，所以需要 root 权限。如果没有 root 权限，程序运行到调用 `pcap_lookupdev()` 后就会失败，显示无法找到合适的设备。

问题 3: 请在嗅探器程序中打开、关闭混杂模式。你能演示在打开或关闭这个模式时的区别吗? 请描述你是如何进行演示的。

回答: 打开混杂模式时，即使不是自身的目的地址，但只要该地址通过网卡，那么就可以捕获到他们的数据包。

`attack.py` 中调用了 `pcap_open_live()` 函数，其中参数 `promisc` 代表了是否开启混杂模式，`promisc` 为 0，处于非混杂模式；`promisc` 为 1，开启了混杂模式。`attack.py` 中，原来的 `promisc` 参数的值就为 1，即原来便开启了混杂模式。

非混杂模式下载图：

```
[11/16/2016 20:36] seed@ubuntu:~/Documents/test$ gcc -o sniffex sniffex.c -l pcap
[11/16/2016 20:37] seed@ubuntu:~/Documents/test$ su ./sniffex
Unknown id: ./sniffex
[11/16/2016 20:37] seed@ubuntu:~/Documents/test$ sudo ./sniffex
sniffex - Sniffer example using libpcap
Copyright (c) 2005 The Tcpdump Group
THERE IS ABSOLUTELY NO WARRANTY FOR THIS PROGRAM.
```

```
Device: eth0
Number of packets: 10
Filter expression: ip
```

```
Packet number 1:
  From: 220.170.182.32
  To: 192.168.181.130
  Protocol: TCP
  Src port: 443
  Dst port: 39341
  Payload (27 bytes):
00000  15 03 01 00 16 7e 00 23 38 3a a2 70 91 7a 1f 47  .....#8:.p.z.G
00016  0a 8b 5a b3 d5 b8 cb 44 31 6a 86  ..Z....D1j.
```

```
Packet number 2:
  From: 192.168.181.130
  To: 220.170.182.32
  Protocol: TCP
  Src port: 39341
  Dst port: 443
```

```
Packet number 3:
  From: 220.170.182.32
  To: 192.168.181.130
  Protocol: TCP
  Src port: 443
  Dst port: 39341
```

```
Packet number 4:
  From: 220.170.182.32
  To: 192.168.181.130
  Protocol: TCP
  Src port: 443
```

```
Packet number 4:
  From: 220.170.182.32
  To: 192.168.181.130
  Protocol: TCP
  Src port: 443
  Dst port: 39344
  Payload (27 bytes):
00000  15 03 01 00 16 b4 58 41 5f 9e 3c c3 f5 54 50 86  .....XA_<..TP.
00016  dd 19 72 97 21 29 17 30 9b fa f2  ..r.!).0...
```

```
Packet number 5:
  From: 192.168.181.130
  To: 220.170.182.32
  Protocol: TCP
  Src port: 39344
  Dst port: 443
```

```
Packet number 6:
  From: 220.170.182.32
  To: 192.168.181.130
  Protocol: TCP
  Src port: 443
  Dst port: 39344
```

```
Packet number 7:
  From: 14.215.177.38
  To: 192.168.181.130
  Protocol: TCP
  Src port: 443
  Dst port: 39714
  Payload (27 bytes):
00000  15 03 01 00 16 6f 49 9f 5e 6a 23 50 34 ca 1b a4  .....oI.^j#P4...
00016  a5 0e 70 be 11 d3 fa ed f4 81 bf  ..p.....
```

```
Packet number 8:
  From: 192.168.181.130
  To: 14.215.177.38
  Protocol: TCP
  Src port: 39714
  Dst port: 443
```

```
Packet number 8:
  From: 192.168.181.130
  To: 14.215.177.38
  Protocol: TCP
  Src port: 39714
  Dst port: 443
```

```
Packet number 9:
  From: 14.215.177.38
  To: 192.168.181.130
  Protocol: TCP
  Src port: 443
  Dst port: 39714
```

```
Packet number 10:
  From: 14.215.177.37
  To: 192.168.181.130
  Protocol: TCP
  Src port: 443
  Dst port: 44874
  Payload (27 bytes):
00000  15 03 01 00 16 f9 60 3b 68 68 25 37 a5 ea 3e cc  .....`hh#7...>.
00016  4b 05 85 cf 83 30 e8 72 80 49 f3  K....0.r.I.
```

```
Capture complete.
[11/16/2016 20:37] seed@ubuntu:~/Documents/test$
```

混杂模式下的截图：


```
Capture complete.
[11/16/2016 20:30] seed@ubuntu:~/Documents/test$ gcc -o sniffex sniffex.c -l pcap
[11/16/2016 20:34] seed@ubuntu:~/Documents/test$ sudo sniffex
sudo: sniffex: command not found
[11/16/2016 20:34] seed@ubuntu:~/Documents/test$ sudo ./sniffex
sniffex - Sniffer example using libpcap
Copyright (c) 2005 The Tcpdump Group
THERE IS ABSOLUTELY NO WARRANTY FOR THIS PROGRAM.
```

```
Device: eth0
Number of packets: 10
Filter expression: ip
```

```
Packet number 1:
  From: 192.168.181.130
  To: 14.215.177.38
  Protocol: TCP
  Src port: 39703
  Dst port: 443
```

```
Packet number 2:
  From: 14.215.177.38
  To: 192.168.181.130
  Protocol: TCP
  Src port: 443
  Dst port: 39703
```

```
Packet number 3:
  From: 192.168.181.130
  To: 14.215.177.38
  Protocol: TCP
  Src port: 39703
  Dst port: 443
```

```
Packet number 4:
  From: 192.168.181.130
  To: 14.215.177.38
  Protocol: TCP
  Src port: 39703
  Dst port: 443
  Payload (378 bytes):
```

```
Protocol: TCP
Src port: 39703
Dst port: 443
```

```
Packet number 4:
  From: 192.168.181.130
  To: 14.215.177.38
  Protocol: TCP
  Src port: 39703
  Dst port: 443
  Payload (378 bytes):
```

```
00000 16 03 01 01 75 01 00 01 71 03 01 58 2d 33 73 dc
00016 76 14 d9 16 96 ad 22 60 7a 6d 6e 1b 99 12 c0 49
00032 29 19 94 34 ee bf d0 41 27 38 00 20 92 7c ac cc
00048 7f a8 96 cf b5 3e a3 b6 ee 36 82 31 e2 06 65 40
00064 77 8e 28 70 d9 1e 73 9f 3e 89 61 69 00 48 00 ff
00080 c0 0a c0 14 00 88 00 87 00 39 00 38 c0 0f c0 05
00096 00 84 00 35 c0 07 c0 09 c0 11 c0 13 00 45 00 44
00112 00 33 00 32 c0 0c c0 0e c0 02 c0 04 00 96 00 41
00128 00 05 00 04 00 2f c0 08 c0 12 00 16 00 13 c0 0d
00144 c0 03 fe ff 00 0a 01 00 00 e0 00 00 00 12 00 10
00160 00 00 0d 77 77 77 2e 62 61 69 64 75 2e 63 6f 6d
00176 00 0a 00 08 00 06 00 17 00 18 00 19 00 0b 00 02
00192 01 00 00 23 00 b0 d7 b2 41 aa c2 e9 b1 6e a2 f0
00208 85 8b 9b dc da b5 c3 15 24 d9 dc f9 f8 47 a1 47
00224 7d 08 56 6d 65 45 5e cd 5e 13 66 6a 06 1d 62 08
00240 51 21 36 22 8b d3 e7 22 9b 5f e2 17 d9 36 4b 7b
00256 f8 db 10 c7 26 16 6b c3 94 93 42 33 92 2b 62 56
00272 24 c9 ed c5 65 38 9a 90 05 9d ee 97 45 a5 ec c3
00288 29 2a 00 83 f8 d5 24 72 dd 08 f5 f0 63 8f 00 44
00304 3d 6d e3 3f e5 f9 89 43 91 10 21 42 9d 4e d2 bd
00320 19 43 c2 12 fe 35 5c e6 6d 04 f8 68 57 31 81 a8
00336 e1 82 55 b9 43 91 9d dc bd 0c e7 a0 06 7f 08 a5
00352 a0 d7 52 b9 c5 c7 d5 a7 ab bb 60 8e 83 99 48 89
00368 ef 9e 0f 3c 16 7c 33 74 00 00
```

```
Packet number 5:
  From: 14.215.177.38
  To: 192.168.181.130
  Protocol: TCP
  Src port: 443
```

任务 1.b: 编写过滤器。根据以下捕获包的要求，为你自己的嗅探器程序编写过滤表达式。实验报告中，你需要提供截图以显示应用这些过滤器所获得的结果。

- ①在两个特定的主机间捕获 ICMP 数据包。
- ②捕获 TCP 数据包，目的端口范围为 10~100。

实验截图：

①通过修改 sniffex.c 中的过滤表达式，实现在两个特定的主机间捕获 ICMP 数据包。开启另一 linux 终端，调用 ping 命令，便于 sniffex.c 运行时，可以捕获

到 ICMP 数据包。

```
Src port: 44335
Dst port: 80

Capture complete.
[11/16/2016 22:52] seed@ubuntu:~/Documents/test$ gcc -o sniffex sniffex.c -l pcap
[11/16/2016 22:55] seed@ubuntu:~/Documents/test$ sudo ./sniffex
sniffex - Sniffer example using libpcap
Copyright (c) 2005 The Tcpdump Group
THERE IS ABSOLUTELY NO WARRANTY FOR THIS PROGRAM.

Device: eth0
Number of packets: 10
Filter expression: icmp and src 192.168.181.130 and dst 192.168.1.1

Packet number 1:
  From: 192.168.181.130
  To: 192.168.1.1
  Protocol: ICMP
Packet number 2:
  From: 192.168.181.130
  To: 192.168.1.1
  Protocol: ICMP
Packet number 3:
  From: 192.168.181.130
  To: 192.168.1.1
  Protocol: ICMP
Packet number 4:
  From: 192.168.181.130
  To: 192.168.1.1
  Protocol: ICMP
Packet number 5:
  From: 192.168.181.130
  To: 192.168.1.1
  Protocol: ICMP
Packet number 6:
  From: 192.168.181.130
  To: 192.168.1.1
  Protocol: ICMP

  From: 192.168.181.130
  To: 192.168.1.1
  Protocol: ICMP
Packet number 4:
  From: 192.168.181.130
  To: 192.168.1.1
  Protocol: ICMP
Packet number 5:
  From: 192.168.181.130
  To: 192.168.1.1
  Protocol: ICMP
Packet number 6:
  From: 192.168.181.130
  To: 192.168.1.1
  Protocol: ICMP
Packet number 7:
  From: 192.168.181.130
  To: 192.168.1.1
  Protocol: ICMP
Packet number 8:
  From: 192.168.181.130
  To: 192.168.1.1
  Protocol: ICMP
Packet number 9:
  From: 192.168.181.130
  To: 192.168.1.1
  Protocol: ICMP
Packet number 10:
  From: 192.168.181.130
  To: 192.168.1.1
  Protocol: ICMP
Capture complete.
[11/16/2016 22:56] seed@ubuntu:~/Documents/test$
```

②通过修改 sniffex.c 中的过滤表达式，实现在端口范围为 10~100 内捕获 TCP 数据包。刷新浏览器，便于运行 sniffex.c 时，可以捕获 TCP 数据包。


```
[11/16/2016 22:52] seed@ubuntu:~$ cd ./Documents/test
[11/16/2016 22:52] seed@ubuntu:~/Documents/test$ gcc -o sniffex sniffex.c -l pcap
[11/16/2016 22:52] seed@ubuntu:~/Documents/test$ sudo ./sniffex
[sudo] password for seed:
sniffex - Sniffer example using libpcap
Copyright (c) 2005 The Tcpdump Group
THERE IS ABSOLUTELY NO WARRANTY FOR THIS PROGRAM.
```

```
Device: eth0
Number of packets: 10
Filter expression: tcp and dst portrange 10-100
```

```
Packet number 1:
  From: 192.168.181.130
  To: 128.230.208.76
  Protocol: TCP
  Src port: 44332
  Dst port: 80
```

```
Packet number 2:
  From: 192.168.181.130
  To: 128.230.208.76
  Protocol: TCP
  Src port: 44333
  Dst port: 80
```

```
Packet number 3:
  From: 192.168.181.130
  To: 128.230.208.76
  Protocol: TCP
  Src port: 44332
  Dst port: 80
```

```
Packet number 4:
  From: 192.168.181.130
  To: 128.230.208.76
  Protocol: TCP
  Src port: 44332
  Dst port: 80
  Payload (432 bytes):
```

```
Protocol: TCP
Src port: 44332
Dst port: 80
```

```
Packet number 4:
  From: 192.168.181.130
  To: 128.230.208.76
  Protocol: TCP
  Src port: 44332
  Dst port: 80
  Payload (432 bytes):
```

```
00000 47 45 54 20 2f 7e 77 65 64 75 2f 73 65 65 64 2f
00016 69 6e 64 65 78 2e 68 74 6d 6c 20 48 54 54 50 2f
00032 31 2e 31 0d 0a 48 6f 73 74 3a 20 77 77 77 2e 63
00048 69 73 2e 73 79 72 2e 65 64 75 0d 0a 55 73 65 72
00064 2d 41 67 65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f
00080 35 2e 30 20 28 58 31 31 3b 20 55 62 75 6e 74 75
00096 3b 20 4c 69 6e 75 78 20 69 36 38 36 3b 20 72 76
00112 3a 32 33 2e 30 29 20 47 65 63 6b 6f 2f 32 30 31
00128 30 30 31 30 31 20 46 69 72 65 66 6f 78 2f 32 33
00144 2e 30 0d 0a 41 63 63 65 70 74 3a 20 74 65 78 74
00160 2f 68 74 6d 6c 2c 61 70 70 6c 69 63 61 74 69 6f
00176 6e 2f 78 68 74 6d 6c 2b 78 6d 6c 2c 61 70 70 6c
00192 69 63 61 74 69 6f 6e 2f 78 6d 6c 3b 71 3d 30 2e
00208 39 2c 2a 2f 2a 3b 71 3d 30 2e 38 0d 0a 41 63 63
00224 65 70 74 2d 4c 61 6e 67 75 61 67 65 3a 20 65 6e
00240 2d 55 53 2c 65 6e 3b 71 3d 30 2e 35 0d 0a 41 63
00256 63 65 70 74 2d 45 6e 63 6f 64 69 6e 67 3a 20 6f
00272 7a 69 70 2c 20 64 65 66 6c 61 74 65 6d 0a 43 6f
00288 6e 6e 65 63 74 69 6f 6e 3a 20 6b 65 65 70 2d 61
00304 6c 69 76 65 0d 0a 49 66 2d 4d 6f 64 69 66 69 65
00320 64 2d 53 69 6e 63 65 3a 20 46 72 69 2c 20 31 30
00336 20 4a 75 6e 20 32 30 31 36 20 31 37 3a 34 39 3a
00352 34 37 20 47 4d 54 0d 0a 49 66 2d 4e 6f 6e 65 2d
00368 4d 61 74 63 68 3a 20 22 35 35 38 35 65 63 2d 31
00384 30 37 64 2d 35 33 34 66 30 32 61 31 65 61 34 63
00400 30 22 0d 0a 43 61 63 68 65 2d 43 6f 6e 74 72 6f
00416 6c 3a 20 6d 61 78 2d 61 67 65 3d 30 6d 0a 6d 0a
```

```
Packet number 5:
  From: 192.168.181.130
```

```
GET /~wedu/seed/
index.html HTTP/
1.1..Host: www.c
is.syr.edu..User
-Agent: Mozilla/
5.0 (X11; Ubuntu
; Linux i686; rv
:23.0) Gecko/201
00101 Firefox/23
..Accept: text
/html,application
n/xhtml+xml,appl
ication/xml;q=0.
9,*/*;q=0.8..Acc
ept-Language: en
-US,en;q=0.5..Ac
cept-Encoding: g
zip, deflate..Co
nnection: keep-a
live..If-Modifie
d-Since: Fri, 10
Jun 2016 17:49:
47 GMT..If-None-
Match: "5585ec-1
07d-534f02a1ea4c
0"..Cache-Contro
l: max-age=0....
```

```

Src port: 44332
Dst port: 80

Packet number 7:
  From: 192.168.181.130
  To: 128.230.208.76
  Protocol: TCP
  Src port: 44333
  Dst port: 80
  Payload (445 bytes):
00000 47 45 54 20 2f 7e 77 65 64 75 2f 73 65 65 64 2f GET /-wedu/seed/
00016 73 74 79 6c 65 5f 68 6f 6d 65 2e 63 73 73 20 48 style_home.css H
00032 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 77 TTP/1.1..Host: w
00048 77 77 2e 63 69 73 2e 73 79 72 2e 65 64 75 0d 0a ww.cis.syr.edu..
00064 55 73 65 72 2d 41 67 65 6e 74 3a 20 4d 6f 7a 69 User-Agent: Mozi
00080 6c 6c 61 2f 35 2e 30 20 28 58 31 31 3b 20 55 62 lla/5.0 (X11; Ub
00096 75 6e 74 75 3b 20 4c 69 6e 75 78 20 69 36 38 36 untu; Linux i686
00112 3b 20 72 76 3a 32 33 2e 30 29 20 47 65 63 6b 6f ; rv:23.0) Gecko
00128 2f 32 30 31 30 30 31 30 31 20 46 69 72 65 66 6f /20100101 Firefo
00144 78 2f 32 33 2e 30 0d 0a 41 63 63 65 70 74 3a 20 x/23.0..Accept:
00160 74 65 78 74 2f 63 73 73 2c 2a 2f 2a 3b 71 3d 30 text/css,*/*;q=0
00176 2e 31 0d 0a 41 63 63 65 70 74 2d 4c 61 6e 67 75 .1..Accept-Langu
00192 61 67 65 3a 20 65 6e 2d 55 53 2c 65 6e 3b 71 3d age: en-US,en;q=
00208 30 2e 35 0d 0a 41 63 63 65 70 74 2d 45 6e 63 6f 0.5..Accept-Enco
00224 64 69 6e 67 3a 20 67 7a 69 70 2c 20 64 65 66 6c ding: gzip, defl
00240 61 74 65 0d 0a 52 65 66 65 72 65 72 3a 20 68 74 ate..Referer: ht
00256 74 70 3a 2f 2f 77 77 77 2e 63 69 73 2e 73 79 72 tp://www.cis.syr
00272 2e 65 64 75 2f 7e 77 65 64 75 2f 73 65 65 64 2f .edu/-wedu/seed/
00288 69 6e 64 65 78 2e 68 74 6d 6c 0d 0a 43 6f 6e 6e Index.html..Conn
00304 65 63 74 69 6f 6e 3a 20 6b 65 65 70 2d 61 6c 69 ection: keep-all
00320 76 65 0d 0a 49 66 2d 4d 6f 64 69 66 69 65 64 2d ve..If-Modified-
00336 53 69 6e 63 65 3a 20 54 68 75 2c 20 31 32 20 4d Since: Thu, 12 M
00352 61 79 20 32 30 31 36 20 31 33 3a 33 36 3a 30 33 ay 2016 13:36:03
00368 20 7d 4d 54 0d 0a 49 66 2d 4e 6f 6e 65 2d 4d 61 GMT..If-None-Ma
00384 74 63 68 3a 20 22 38 34 30 30 38 2d 32 34 61 64 tch: "84008-24ad
00400 2d 35 33 32 61 35 33 64 34 63 66 36 63 30 22 0d -532a53d4cf6c0".
00416 0a 43 61 63 68 65 2d 43 6f 6e 74 72 6f 6c 3a 20 .Cache-Control:
00432 6d 61 78 2d 61 67 65 3d 30 0d 0a 0d 0a max-age=0....

Packet number 8:
  From: 192.168.181.130

```

```

  To: 128.230.208.76
  Protocol: TCP
  Src port: 44332
  Dst port: 80
  Payload (464 bytes):
00000 47 45 54 20 2f 7e 77 65 64 75 2f 73 65 65 64 2f GET /-wedu/seed/
00016 69 6d 67 2f 73 65 65 64 5f 6c 6f 6f 6f 2e 70 6e ing/seed_logo.pn
00032 67 20 48 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 g HTTP/1.1..Host
00048 3a 20 77 77 77 2e 63 69 73 2e 73 79 72 2e 65 64 : www.cis.syr.ed
00064 75 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 4d u..User-Agent: M
00080 6f 7a 69 6c 6c 61 2f 35 2e 30 20 28 58 31 31 3b ozilla/5.0 (X11;
00096 20 55 62 75 6e 74 75 3b 20 4c 69 6e 75 78 20 69 Ubuntu; Linux i
00112 36 38 36 3b 20 72 76 3a 32 33 2e 30 29 20 47 65 686; rv:23.0) Ge
00128 63 6b 6f 2f 32 30 31 30 30 31 30 31 20 46 69 72 cko/20100101 Fir
00144 65 66 6f 78 2f 32 33 2e 30 0d 0a 41 63 63 65 70 efox/23.0..Accep
00160 74 3a 20 69 6d 61 67 65 2f 70 6e 67 2c 69 6d 61 t: image/png,ima
00176 67 65 2f 2a 3b 71 3d 30 2e 38 2c 2a 2f 2a 3b 71 ge/*;q=0.8,*/*;q
00192 3d 30 2e 35 0d 0a 41 63 63 65 70 74 2d 4c 61 6e =0.5..Accept-Lan
00208 67 75 61 67 65 3a 20 65 6e 2d 55 53 2c 65 6e 3b guage: en-US,en;
00224 71 3d 30 2e 35 0d 0a 41 63 63 65 70 74 2d 45 6e q=0.5..Accept-En
00240 63 6f 64 69 6e 67 3a 20 67 7a 69 70 2c 20 64 65 coding: gzip, de
00256 66 6c 61 74 65 0d 0a 52 65 66 65 72 65 72 3a 20 flate..Referer:
00272 68 74 74 70 3a 2f 2f 77 77 2e 63 69 73 2e 73 http://www.cis.s
00288 79 72 2e 65 64 75 2f 7e 77 65 64 75 2f 73 65 65 yr.edu/-wedu/see
00304 64 2f 69 6e 64 65 78 2e 68 74 6d 6c 0d 0a 43 6f d/index.html..Co
00320 6e 6e 65 63 74 69 6f 6e 3a 20 6b 65 65 70 2d 61 nnection: keep-a
00336 6c 69 76 65 0d 0a 49 66 2d 4d 6f 64 69 66 69 65 live..If-Modifile
00352 64 2d 53 69 6e 63 65 3a 20 53 61 74 2c 20 30 33 d-Since: Sat, 03
00368 20 4a 61 6e 20 32 30 31 35 20 30 34 3a 35 34 3a Jan 2015 04:54:
00384 32 38 20 47 4d 54 0d 0a 49 66 2d 4e 6f 6e 65 2d 28 GMT..If-None-
00400 4d 61 74 63 68 3a 20 22 37 33 63 32 33 30 2d 33 Match: "73c230-3
00416 37 65 30 2d 35 30 62 62 38 34 30 38 36 38 39 30 7e0-50bb84086890
00432 30 22 0d 0a 43 61 63 68 65 2d 43 6f 6e 74 72 6f 0"..Cache-Contro
00448 6c 3a 20 6d 61 78 2d 61 67 65 3d 30 0d 0a 0d 0a l: max-age=0....

Packet number 9:
  From: 192.168.181.130
  To: 128.230.208.76
  Protocol: TCP
  Src port: 44334
  Dst port: 80

```



```

00080  6f 7a 69 6c 6c 61 2f 35 2e 30 20 28 58 31 31 3b  ozilla/5.0 (X11;
00096  20 55 62 75 6e 74 75 3b 20 4c 69 6e 75 78 20 69  Ubuntu; Linux i
00112  36 38 36 3b 20 72 76 3a 32 33 2e 30 29 20 47 65  686; rv:23.0) Ge
00128  63 6b 6f 2f 32 30 31 30 30 31 30 31 20 46 69 72  cko/20100101 Fir
00144  65 66 6f 78 2f 32 33 2e 30 0d 0a 41 63 63 65 70  efox/23.0..Accep
00160  74 3a 20 69 6d 61 67 65 2f 70 6e 67 2c 69 6d 61  t: image/png,ima
00176  67 65 2f 2a 3b 71 3d 30 2e 38 2c 2a 2f 2a 3b 71  ge/*;q=0.8,*/*;q
00192  3d 30 2e 35 0d 0a 41 63 63 65 70 74 2d 4c 61 6e  =0.5..Accept-Lan
00208  67 75 61 67 65 3a 20 65 6e 2d 55 53 2c 65 6e 3b  guage: en-US,en;
00224  71 3d 30 2e 35 0d 0a 41 63 63 65 70 74 2d 45 6e  q=0.5..Accept-En
00240  63 6f 64 69 6e 67 3a 20 67 7a 69 70 2c 20 64 65  coding: gzip, de
00256  66 6c 61 74 65 0d 0a 52 65 66 65 72 65 72 3a 20  fLate..Referer:
00272  68 74 74 70 3a 2f 2f 77 77 77 2e 63 69 73 2e 73  http://www.cls.s
00288  79 72 2e 65 64 75 2f 7e 77 65 64 75 2f 73 65 65  yr.edu/~wed/see
00304  64 2f 69 6e 64 65 78 2e 68 74 6d 6c 0d 0a 43 6f  d/index.html..Co
00320  6e 6e 65 63 74 69 6f 6e 3a 20 6b 65 65 70 2d 61  nnection: keep-a
00336  6c 69 76 65 0d 0a 49 66 2d 4d 6f 64 69 66 69 65  live..If-Modifie
00352  64 2d 53 69 6e 63 65 3a 20 53 61 74 2c 20 30 33  d-Since: Sat, 03
00368  20 4a 61 6e 20 32 30 31 35 20 30 34 3a 35 34 3a  Jan 2015 04:54:
00384  32 38 20 47 4d 54 0d 0a 49 66 2d 4e 6f 6e 65 2d  28 GMT..If-None-
00400  4d 61 74 63 68 3a 20 22 37 33 63 32 33 30 2d 33  Match: "73c230-3
00416  37 65 30 2d 35 30 62 62 38 34 30 38 36 38 39 30  7e0-50bb84086890
00432  30 22 0d 0a 43 61 63 68 65 2d 43 6f 6e 74 72 6f  0"..Cache-Contro
00448  6c 3a 20 6d 61 78 2d 61 67 65 3d 30 0d 0a 0d 0a  l: max-age=0....

Packet number 9:
  From: 192.168.181.130
  To: 128.230.208.76
  Protocol: TCP
  Src port: 44334
  Dst port: 80

Packet number 10:
  From: 192.168.181.130
  To: 128.230.208.76
  Protocol: TCP
  Src port: 44335
  Dst port: 80

Capture complete.
[11/16/2016 22:52] seed@ubuntu:~/Documents/test$

```

2.任务 2：欺骗

当一个正常的用户发送数据包时，操作系统通常不会允许用户在协议报头上设置所有字段（例如，TCP、UDP 和 IP 报头）。操作系统将设置大部分的字段，同时只允许用户设置小部分字段，比如目的 IP 地址、目的端口号等。然而，如果用户拥有 root 权限，他们就可以在数据包头设置任意字段。这就是包欺骗，可以通过原始套接字实现。

原始套接字为程序员提供了绝对的控制包建设的可能，允许程序员构建任意数据包，包括设置头字段和有效载荷。使用原始套接字很简单，它包括 4 个步骤：(1)创建一个原始套接字；(2)设置套接字选项；(3)构建数据包；(4)通过原始套接字发送数据包。

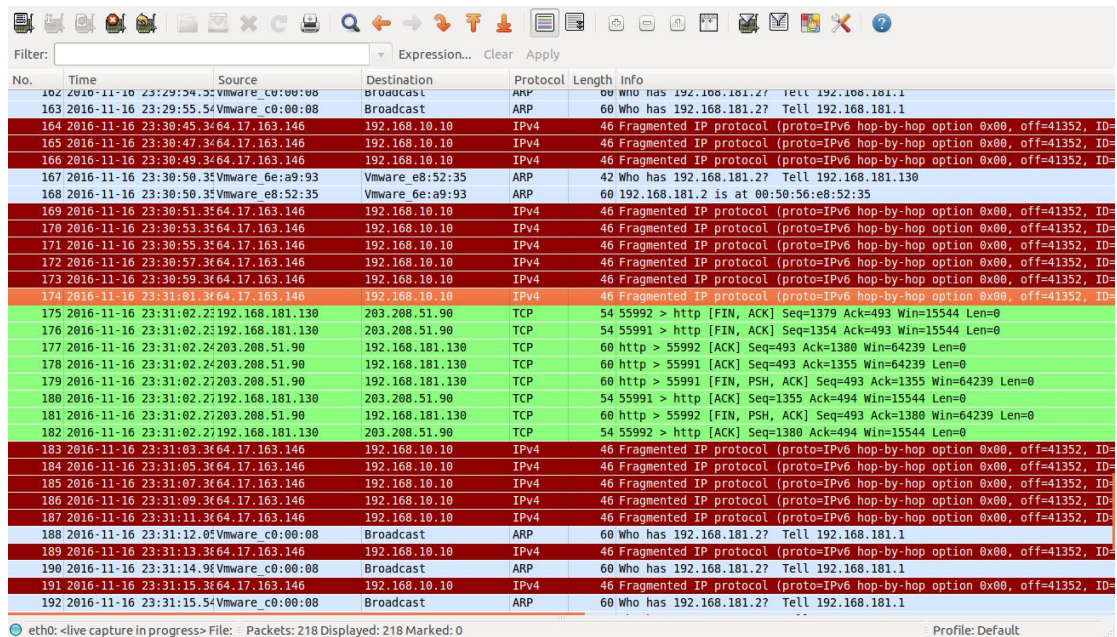
任务 2.a：编写 spoofing 程序。编写自己的包程序，或下载这样一个程序。请提供截图（比如，Wireshark 数据包跟踪）证明你的程序已成功地发送欺骗 IP 数据包。

截图如下：

此截图为运行下载的欺骗程序 rawudp.c，通过 21 号端口向目的 IP 地址为：192.168.10.10 发送的 20 个数据包


```
[11/16/2016 23:30] seed@ubuntu:~/Documents/test$ sudo ./rawudp 192.168.10.10 21
socket() - Using SOCK_RAW socket and UDP protocol is OK.
setsockopt() is OK.
Trying...
Using raw socket and UDP protocol
Using Source IP: 192.168.10.10 port: 21, Target IP: 203.106.93.91 port: 8080.
Count #1 - sendto() is OK.
Count #2 - sendto() is OK.
Count #3 - sendto() is OK.
Count #4 - sendto() is OK.
Count #5 - sendto() is OK.
Count #6 - sendto() is OK.
Count #7 - sendto() is OK.
Count #8 - sendto() is OK.
Count #9 - sendto() is OK.
Count #10 - sendto() is OK.
Count #11 - sendto() is OK.
Count #12 - sendto() is OK.
Count #13 - sendto() is OK.
Count #14 - sendto() is OK.
Count #15 - sendto() is OK.
Count #16 - sendto() is OK.
Count #17 - sendto() is OK.
Count #18 - sendto() is OK.
Count #19 - sendto() is OK.
Count #20 - sendto() is OK.
[11/16/2016 23:31] seed@ubuntu:~/Documents/test$
```

这两张截图为 Wireshark 捕捉到的发送的 20 个欺骗包



No.	Time	Source	Destination	Protocol	Length	Info
162	2016-11-16 23:29:54.35	Vmware_c0:00:08	Broadcast	ARP	60	Who has 192.168.181.2? Tell 192.168.181.1
163	2016-11-16 23:29:55.54	Vmware_c0:00:08	Broadcast	ARP	60	Who has 192.168.181.2? Tell 192.168.181.1
164	2016-11-16 23:30:45.34	192.168.10.10	192.168.10.10	IPv4	46	Fragmented IP protocol (proto=IPv6 hop-by-hop option 0x00, off=41352, ID=)
165	2016-11-16 23:30:47.34	192.168.10.10	192.168.10.10	IPv4	46	Fragmented IP protocol (proto=IPv6 hop-by-hop option 0x00, off=41352, ID=)
166	2016-11-16 23:30:49.34	192.168.10.10	192.168.10.10	IPv4	46	Fragmented IP protocol (proto=IPv6 hop-by-hop option 0x00, off=41352, ID=)
167	2016-11-16 23:30:50.35	Vmware_e8:52:35	Vmware_e8:52:35	ARP	42	Who has 192.168.181.2? Tell 192.168.181.130
168	2016-11-16 23:30:50.35	Vmware_e8:52:35	Vmware_e8:52:35	ARP	60	192.168.181.2 is at 00:50:56:e8:52:35
169	2016-11-16 23:30:51.35	192.168.10.10	192.168.10.10	IPv4	46	Fragmented IP protocol (proto=IPv6 hop-by-hop option 0x00, off=41352, ID=)
170	2016-11-16 23:30:53.35	192.168.10.10	192.168.10.10	IPv4	46	Fragmented IP protocol (proto=IPv6 hop-by-hop option 0x00, off=41352, ID=)
171	2016-11-16 23:30:55.35	192.168.10.10	192.168.10.10	IPv4	46	Fragmented IP protocol (proto=IPv6 hop-by-hop option 0x00, off=41352, ID=)
172	2016-11-16 23:30:57.35	192.168.10.10	192.168.10.10	IPv4	46	Fragmented IP protocol (proto=IPv6 hop-by-hop option 0x00, off=41352, ID=)
173	2016-11-16 23:30:59.35	192.168.10.10	192.168.10.10	IPv4	46	Fragmented IP protocol (proto=IPv6 hop-by-hop option 0x00, off=41352, ID=)
174	2016-11-16 23:31:01.35	192.168.10.10	192.168.10.10	IPv4	46	Fragmented IP protocol (proto=IPv6 hop-by-hop option 0x00, off=41352, ID=)
175	2016-11-16 23:31:02.23	192.168.181.130	203.208.51.90	TCP	54	55992 > http [FIN, ACK] Seq=1379 Ack=493 Win=15544 Len=0
176	2016-11-16 23:31:02.23	192.168.181.130	203.208.51.90	TCP	54	55991 > http [FIN, ACK] Seq=1354 Ack=493 Win=15544 Len=0
177	2016-11-16 23:31:02.24	203.208.51.90	192.168.181.130	TCP	60	http > 55992 [ACK] Seq=493 Ack=1380 Win=64239 Len=0
178	2016-11-16 23:31:02.24	203.208.51.90	192.168.181.130	TCP	60	http > 55991 [ACK] Seq=493 Ack=1355 Win=64239 Len=0
179	2016-11-16 23:31:02.27	203.208.51.90	192.168.181.130	TCP	60	http > 55991 [FIN, PSH, ACK] Seq=493 Ack=1355 Win=64239 Len=0
180	2016-11-16 23:31:02.27	192.168.181.130	203.208.51.90	TCP	54	55991 > http [ACK] Seq=1355 Ack=494 Win=15544 Len=0
181	2016-11-16 23:31:02.27	203.208.51.90	192.168.181.130	TCP	60	http > 55992 [FIN, PSH, ACK] Seq=493 Ack=1380 Win=64239 Len=0
182	2016-11-16 23:31:02.27	192.168.181.130	203.208.51.90	TCP	54	55992 > http [ACK] Seq=1380 Ack=494 Win=15544 Len=0
183	2016-11-16 23:31:03.36	192.168.10.10	192.168.10.10	IPv4	46	Fragmented IP protocol (proto=IPv6 hop-by-hop option 0x00, off=41352, ID=)
184	2016-11-16 23:31:05.36	192.168.10.10	192.168.10.10	IPv4	46	Fragmented IP protocol (proto=IPv6 hop-by-hop option 0x00, off=41352, ID=)
185	2016-11-16 23:31:07.36	192.168.10.10	192.168.10.10	IPv4	46	Fragmented IP protocol (proto=IPv6 hop-by-hop option 0x00, off=41352, ID=)
186	2016-11-16 23:31:09.36	192.168.10.10	192.168.10.10	IPv4	46	Fragmented IP protocol (proto=IPv6 hop-by-hop option 0x00, off=41352, ID=)
187	2016-11-16 23:31:11.36	192.168.10.10	192.168.10.10	IPv4	46	Fragmented IP protocol (proto=IPv6 hop-by-hop option 0x00, off=41352, ID=)
188	2016-11-16 23:31:12.05	Vmware_c0:00:08	Broadcast	ARP	60	Who has 192.168.181.2? Tell 192.168.181.1
189	2016-11-16 23:31:13.36	192.168.10.10	192.168.10.10	IPv4	46	Fragmented IP protocol (proto=IPv6 hop-by-hop option 0x00, off=41352, ID=)
190	2016-11-16 23:31:14.96	Vmware_c0:00:08	Broadcast	ARP	60	Who has 192.168.181.2? Tell 192.168.181.1
191	2016-11-16 23:31:15.36	192.168.10.10	192.168.10.10	IPv4	46	Fragmented IP protocol (proto=IPv6 hop-by-hop option 0x00, off=41352, ID=)
192	2016-11-16 23:31:15.54	Vmware_c0:00:08	Broadcast	ARP	60	Who has 192.168.181.2? Tell 192.168.181.1

问题

问题 4 不管实际的数据包有多大，你都能将 IP 包的长度字段设置为任意值吗？

回答：IP 数据包的最大长度是 64k（65535），因为在 IP 包头中用 2 个字节描述报文长度，2 个字节所能表达的最大数字就是 65535。所以，我觉得，不管实际的数据包有多大，都可以将 IP 包的长度字段设置为任意值。

问题 5 使用原始套接字编程，你需要为 IP 头计算校验和吗？

回答：需要。原始套接字中包含了 IP 头和 TCP 头的的数据，而 IP/TCP 本身就是需要校验和，以此来保证在传输过程中不会出现差错，可以通过检验和判断发过来的包是否正确，对于一些简单的错误，经过校验后可以确定其出错位置并进行纠正，所以使用原始套接字编程，也需要为 IP 头计算校验和。

问题 6 为什么需要 root 权限运行使用了原始套接字的程序？如果没有使用 root 权限执行程序，它运行到哪一步就会失败？

回答：原始套接字用于低层协议的访问，所以需要使用原始套接字的程序。

三. 实验结果

实验截图在二 实验任务中。

四. 实验心得

本次实验较之“心脏滴血”攻击而言，难度提升了。实验过程不再仅仅是执行已有的代码，得到结果，还包括了阅读源码，分析源码中的相关语句，依据语句判断出相关的功能，再通过修改语句，执行实验指导书上所要求的功能。例如，通过对过滤表达式的修改，满足捕获不同包的要求。

通过本次实验，我还了解到混杂模式与非混杂模式的区别。这两种方式的区别很大。一般来说，非混杂模式的嗅探器中，主机仅嗅探那些跟它直接有关的通信，如发向它的，从它发出的，或经它路由的等都会被嗅探器捕捉。而在混杂模

式中则嗅探传输线路上的所有通信。但因本次实验，只使用了一台虚拟机，所以即使在混杂模式下也只能抓到自己的数据包。

实践的过程中，我收获了许多新知识。Seed project 不仅帮助我们锻炼自己的动手能力，而且还为我们提供了主动吸收新知识的渠道与动力。