Wireless Ad-hoc Network Exp1 report
刘昱辰 0840042
Wiresharck Packet Sniff

Q1. Can you get any detail information from received packets for HTTP, FTP, Telnet and SSH via Open Access AP!?

(1) HTTP:

```
141 1.600817443   192.168.0.100        192.168.0.1          HTTP      605 GET /IFLWCCMAWSAPJVNC/userRpm/StatusRpm.htm HTTP/1.1
```

```
1··User- Agent: M
ozilla/5 .0 (X11;
 Ubuntu;  Linux x
86_64; r v:40.0)
Gecko/20 100101 F
irefox/4 0.0··Acc
```

接收端使用火狐浏览器，在 ubuntu64 位操作系统下向网址请求 html 页面。

```
Accept-Language: en-US,en;q=0.5\r\n
Accept-Encoding: gzip, deflate\r\n
```

同样也可以看到接受的语言和编码方式。

```
[Full request URI: http://192.168.0.1/IFLWCCMAWSAPJVNC/userRpm/StatusRpm.htm]
```

以及完整的 URL 请求，192.168.0.1 是路由器保留地址。

```
 Source Port: 33767
 Destination Port: 80
```

端口是 http 常用端口 80。

```
[Stream index: 0]
[TCP Segment Len: 479]
Sequence number: 1    (relative sequence number)
[Next sequence number: 480    (relative sequence number)]
Acknowledgment number: 1    (relative ack number)
```

因为 http 是基于 tcp 的，会有滑动窗口的大小，封包序列号等信息。

(2) FTP

```
2298 12.927394544  140.113.9.151    192.168.0.100    FTP    210 Response: 220 ProFTPD 1.3.5a Server (ProFTPD Default Installation) [::ffff:140.113.9.151]
2913 17.345983188  192.168.0.100    140.113.9.151    FTP    138 Request: USER 12345
2920 17.354442359  140.113.9.151    192.168.0.100    FTP    163 Response: 331 \351\234\200\350\246\201\347\202\272 12345 \346\217\220\344\276\233\345\257\206\347\242\274
3203 19.819920092  192.168.0.100    140.113.9.151    FTP    138 Request: PASS 12345
3212 19.850133270  140.113.9.151    192.168.0.100    FTP    150 Response: 530 \347\231\273\345\205\245\344\270\215\346\255\243\347\242\272
3215 19.853839197  192.168.0.100    140.113.9.151    FTP    132 Request: SYST
3219 19.858045531  140.113.9.151    192.168.0.100    FTP    148 Response: 215 UNIX Type: L8
```

用户名：12345
密码：12345
通过三次交互建立连接，基于 TCP

```
 ProFTPD  1.3.5a
Server ( ProFTPD
Default  Installa
tion) [: :ffff:14
0.113.9. 151]··8·
```

(3) Telnet

```
3352 31.617470045  140.112.172.4    192.168.0.100    TELNET    130 Telnet Data ...
3378 31.915616159  192.168.0.100    140.112.172.4    TELNET    127 Telnet Data ...
3385 31.924366313  140.112.172.4    192.168.0.100    TELNET    130 Telnet Data ...
3400 32.037360959  192.168.0.100    140.112.172.4    TELNET    127 Telnet Data ...
3408 32.045000439  140.112.172.4    192.168.0.100    TELNET    130 Telnet Data ...
```

连续的几个 telnet 封包，封包未加密，可以看到其中的 data。
在 telnet 中每次敲击一下键盘，就会产生一个 telnet 封包，因此我们可以看到这几个封

包的内容均为一个字母。

```
Telnet
    Data: a
Telnet
    Data: s
Telnet
    Data: d
```

(4) SSH

```
SSH Protocol
    Protocol: SSH-2.0-OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.3
```

可以从封包内容中看到 ssh 所用的协议版本号为 OpenSSH_6.6.1，在 ubuntu 系统上运行。

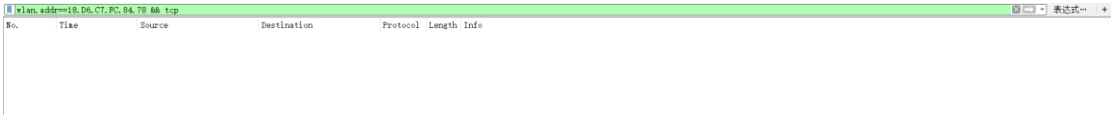
SSH 交换的过程中还会有登陆密钥交换的过程。

```
4384 42.076372794  192.168.0.100      140.113.13.191     SSHv2     646 Client: Key Exchange Init

4399 42.087646609  140.113.13.191     192.168.0.100      SSHv2     409 Server: Diffie-Hellman Key Exchange Reply, New Keys
4403 42.092551238  192.168.0.100      140.113.13.191     SSHv2     142 Client: New Keys
4417 42.138760916  192.168.0.100      140.113.13.191     SSHv2     182 Client: Encrypted packet (len=56)
```

之后的封包会被加密。

Q2. Can you get any detail information from received packets for HTTP, FTP, Telnet and SSH via Encrypted AP!?



在加密的传输中进行封包嗅探，可以接收到封包却无法进行解析，使用过滤器搜索 tcp 未显示任何结果说明了这点（HTTP, FTP, Telnet, SSH 都是基于 TCP 的）。

搜索 AP 的 MAC 地址，显示出的协议类型仅为 802.11，无法显示更多信息。

```
21 0.155086394   Tp-LinkT_fc:84:78   Broadcast           802.11    299 Beacon frame, SN=2708, FN=0, Flags=........C, BI=100, SSID=bunlab
42 0.637287152   Tp-LinkT_fc:84:78   IntelCor_7e:8a:d3   802.11    408 Probe Response, SN=2713, FN=0, Flags=........C, BI=100, SSID=bunlab
43 0.638906493                       Tp-LinkT fc:84:78 (… 802.11     50 Acknowledgement, Flags=........C
```

能解析出来的封包只有以下几种不进行加密的类型。

```
Beacon frame, SN=2708, FN=0, Flags=........C, BI=100, SSID=bunlab
```

广播的 Beacon 封包。

```
Request-to-send, Flags=........C
```

RTS 封包。

```
Acknowledgement, Flags=........C
```

ACK 封包。

以下的这个 QoS Data 表示的是收到的加密封包，因为上一题的四种封包均是有服务质量保障的。但是因为进行了加密，无法对具体内容进行解析，显示如下。

```
QoS Data, SN=775, FN=0, Flags=.p....F.C
```

```
Data (1516 bytes)
    Data: 8a42b2fa559a5b8da7330ac37d2ffb2031743fbffc9e173b…
    [Length: 1516]
```