实验一 应用层

09019204 曹邹颖

一．实验内容

1. 学会使用 Wireshark 抓包软件，会使用过滤器
2. 学习 Wireshark 基本操作：重点掌握捕获过滤器和显示过滤器。分析 HTTP 和 DNS 协议
3. 测试 curl 命令，访问一个 web 页面。（选做）
4. 利用 telnet 命令测试 get 命令，访问 www.baidu.com。（选做）
5. 利用 telnet 命令测试 SMTP 服务，解析其过程。（选做）
6. 测试 tracert 命令，并解析其过程。
7. 使用 nslookup 查询域名信息，简要分析。

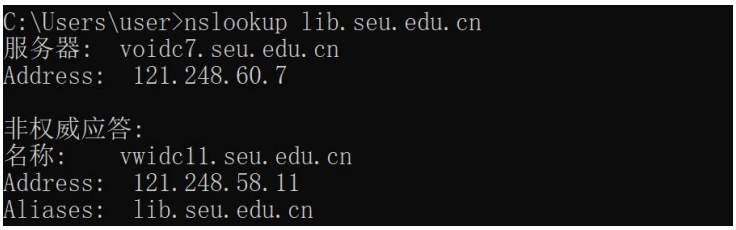
二．实验步骤与分析

1. Wireshark 实验

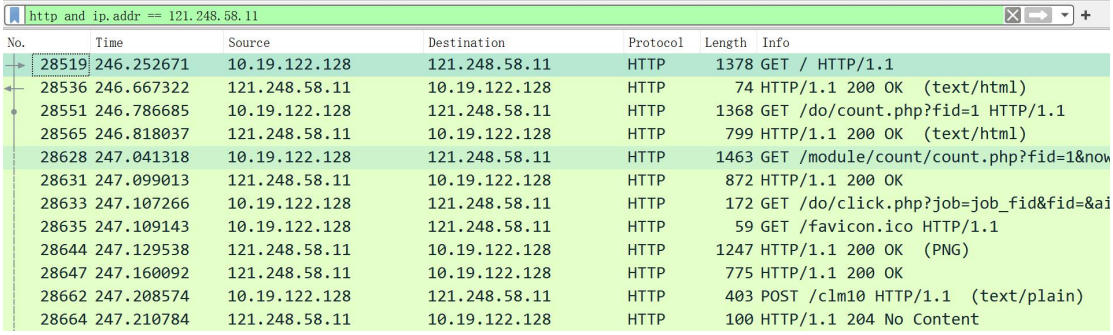
(1) HTTP 协议分析

① 打开 Wireshark 设置过滤器：http and ip.addr == 121.248.58.11

121.248.58.11 是使用 nslookup 命令查询的东南大学图书馆网站的 IP 地址。

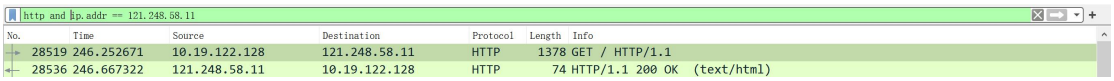


② 打开浏览器访问东南大学图书馆网站 lib.seu.edu.cn，浏览器打开网址开始抓包

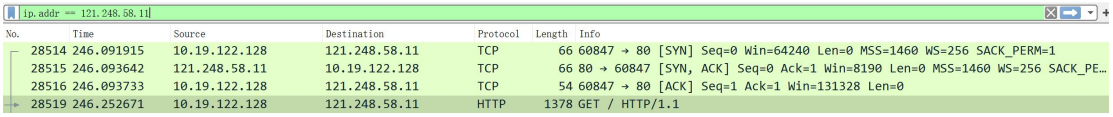


③ 这里，我以 HTTP-GET 数据包为例分析 HTTP 协议。

选取第一个客户端向服务器发送的 HTTP 请求包以及其对应的 HTTP 响应包分析。

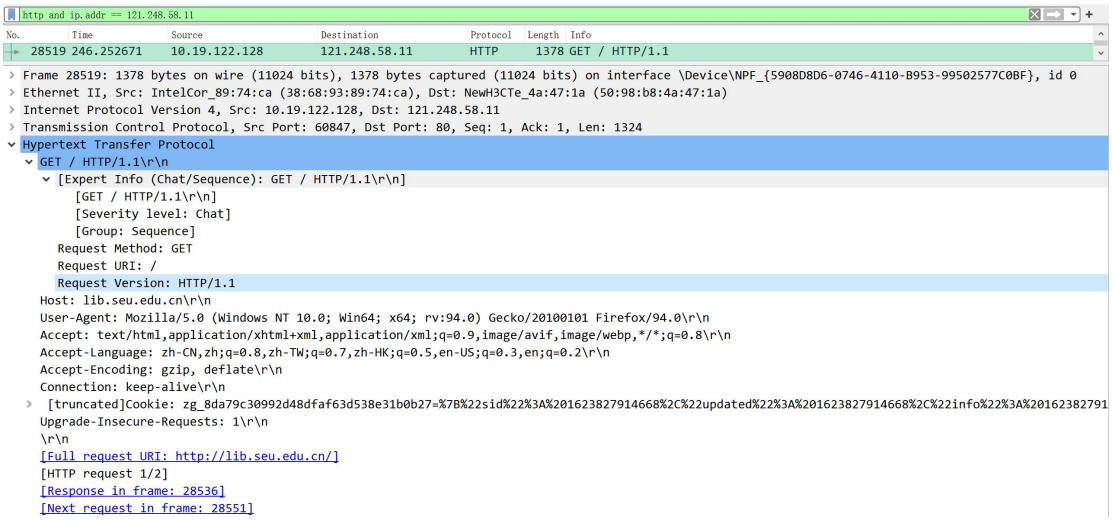


在开始分析之前，我修改显示过滤器为：ip.addr == 121.248.58.11

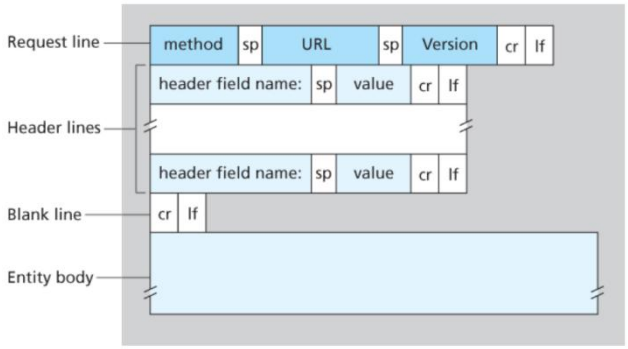


由此可见，HTTP 工作流程为客户端通过 TCP 三次握手与服务器建立连接，TCP 建立连接成功后，开始向服务器发送 HTTP 请求。

开始分析第一个 HTTP 请求包：



对照 HTTP 请求报文的格式，逐句分析：



GET / HTTP/1.1\r\n 为请求行信息
[Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n] 为专家信息
[GET / HTTP/1.1\r\n]
[Severity level: Chat]
[Group: Sequence]
Request Method: GET 表示请求方法为 GET
Request URI: / 表示请求的 URI
Request Version: HTTP/1.1 表示请求的版本为 HTTP/1.1
Host: lib.seu.edu.cn\r\n 表示请求的主机为东南大学图书馆网站
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:94.0) Gecko/20100101 Firefox/94.0\r\n 为浏览器类型
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n 表示请求的类型
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2\r\n 表示请求语言
Accept-Encoding: gzip, deflate\r\n 表示请求的编码格式
Connection: keep-alive\r\n 表示使用持久连接
\r\n 表示空行
[Full request URI: http://lib.seu.edu.cn/] 表示请求的 URI 为 http://lib.seu.edu.cn/
HTTP request 1/2
[Response in frame: 28536] 表示应答是第 28536 帧
[Next request in frame: 28551] 表示下一个请求是第 28551 帧

实验一 应用层
09019204_曹邹颖

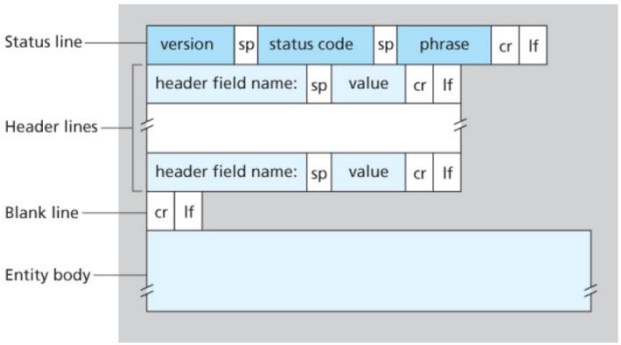
开始分析第一个 HTTP 响应包：

```
http and ip.addr == 121.248.58.11
No.     Time           Source          Destination     Protocol  Length  Info
  28536 246.667322     121.248.58.11   10.19.122.128   HTTP        74 HTTP/1.1 200 OK  (text/html)

Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
    [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      [HTTP/1.1 200 OK\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Response Version: HTTP/1.1
    Status Code: 200
    [Status Code Description: OK]
    Response Phrase: OK
  Date: Sun, 28 Nov 2021 07:01:09 GMT\r\n
  Server: Apache/2.4.39 (Win64) OpenSSL/1.1.1b mod_fcgid/2.3.9a\r\n
  X-Powered-By: PHP/5.6.9\r\n
  Set-Cookie: security_session_verify=d7a8555331588a1ceeff3bc7ae0f4c84; expires=Wed, 01-Dec-21 15:01:09 GMT; path=/; HttpOnly\r\n
  Set-Cookie: USR2=hbwtc0yi%090%091638082869%09http%3A%2F%2Flib.seu.edu.cn%2Findex.php; expires=Mon, 29-Nov-2021 07:01:09 GMT; Max-Age=86400; path=/\r
  Set-Cookie: security_session_verify=d7a8555331588a1ceeff3bc7ae0f4c84; expires=Wed, 01-Dec-21 15:01:09 GMT; path=/; HttpOnly\r\n
  Keep-Alive: timeout=5, max=100\r\n
  Connection: Keep-Alive\r\n
  X-Via-NSCOPI: 1.0\r\n
  Transfer-Encoding: chunked\r\n
  Content-Type: text/html; charset=utf-8\r\n
  Set-Cookie: NSC_ESNS=41e52b2b-2938-11a3-9678-00e0ed9d66bd_3534334965_3248738030_00000000005399912716; Path=/; Expires=Sun, 28-Nov-2021 07:01:24 GMT\
  Cache-Control: no-cache\r\n
  Cache-Control: private\r\n
  Content-Encoding: gzip\r\n
  Transfer-Encoding: chunked\r\n
  \r\n
  [HTTP response 1/2]

  [Time since request: 0.414651000 seconds]
  [Request in frame: 28519]
  [Next request in frame: 28551]
  [Next response in frame: 28565]
  [Request URI: http://lib.seu.edu.cn/do/count.php?fid=1]
> HTTP chunked response
  Content-encoded entity body (gzip): 16574 bytes -> 147459 bytes
  File Data: 147459 bytes
> Line-based text data: text/html (2425 lines)
```

对照 HTTP 响应报文的格式，逐句分析：

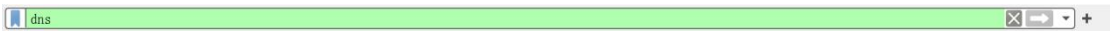


HTTP/1.1 200 OK\r\n 为响应行信息
    [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n] 为专家信息
        [HTTP/1.1 200 OK\r\n] 为 HTTP 响应信息，响应码为 200
        [Severity level: Chat]
        [Group: Sequence]
    Request Version: HTTP/1.1 为请求版本
    Status Code: 200 为状态码
    Response Phrase: OK 为响应短语
Date: Sun, 28 Nov 2021 07:01:09 GMT\r\n 表示消息发送的时间
Server: Apache/2.4.39 (Win64) OpenSSL/1.1.1b mod_fcgid/2.3.9a\r\n 表示服务器类型
X-Powered-By: PHP/5.6.9\r\n 表示网站是用 PHP 开发的
Set-Cookie: security_session_verify=d7a8555331588a1ceeff3bc7ae0f4c84; expires=
        Wed, 01-Dec-21 15:01:09 GMT; path=/; HttpOnly\r\n 设置 HTTP Cookie
Set-Cookie: USR2=hbwtc0yi%090%091638082869%09http%3A%2F%2Flib.seu.edu.
        cn%2Findex.php; expires=Mon, 29-Nov-2021 07:01:09 GMT; Max-Age=
        86400; path=/\r\n
Set-Cookie: security_session_verify=d7a8555331588a1ceeff3bc7ae0f4c84; expires=
        Wed, 01-Dec-21 15:01:09 GMT; path=/; HttpOnly\r\n

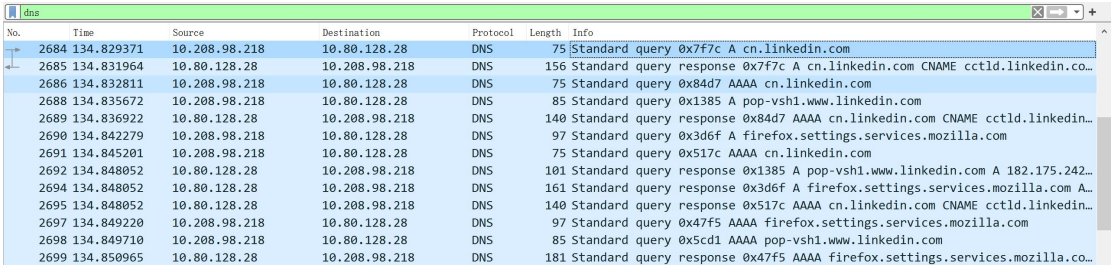
Keep-Alive: timeout=5, max=100\r\n 过期时间 5 秒，最多一百次请求强制断掉连接
Connection: Keep-Alive\r\n 表示使用持久连接
X-Via-NSCOPI: 1.0\r\n
Transfer-Encoding: chunked\r\n 使用分块编码的编码传输
Content-Type: text/html; charset=utf-8\r\n 表示响应的内容类型
Set-Cookie: NSC_ESNS=41e52b2b-2938-11a3-9678-00e0ed9d66bd_3534334965_
3248738030_00000000005399912716; Path=/; Expires=Sun,
28-Nov-2021 07:01:24 GMT\r\n
Cache-Control: no-cache\r\n 缓存控制
Cache-Control: private\r\n
Content-Encoding: gzip\r\n 表示实体数据的压缩格式
Transfer-Encoding: chunked\r\n
\r\n 表示空行
[HTTP response 1/2] 为 HTTP 响应
[Time since request: 0.414651000 seconds] 为响应使用的时间
[Request in frame: 28519] 表示请求的帧号为 28519
[Next request in frame: 28551] 表示下一个请求的帧号 28551
[Next response in frame: 28565] 表示下一个响应的帧号是 28565
[Request URI: http://lib.seu.edu.cn/do/count.php?fid=1] 表示请求的 URI
HTTP chunked response
Data chunk (6357 octets)
Data chunk (7709 octets)
Data chunk (2508 octets)
End of chunked encoding
\r\n
Content-encoded entity body (gzip): 16574 bytes -> 147459 bytes 内容编码（gzip）
File Data: 147459 bytes
Line-based text data: text/html (2425 lines) 表示基于行的文本数据

(2) DNS 协议分析

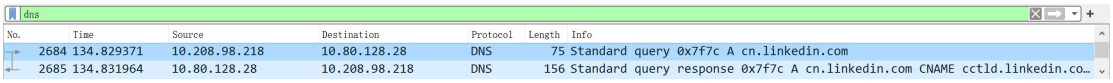
① 打开 Wireshark 设置过滤器：dns



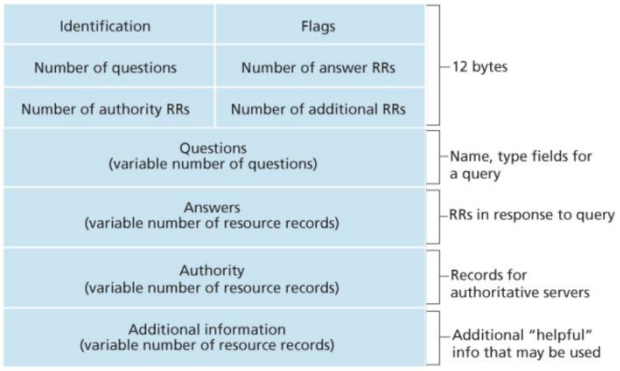
② 打开浏览器访问领英 www.linkedin.com，浏览器打开网址开始抓包



③ 这里，我选取客户端向本地 DNS 服务器发送的第一个查询报文以及其对应的回答报文分析。



首先，DNS 只有两种报文：查询报文、回答报文，两者有着相同格式，如下：



开始分析 DNS 查询报文：

```
Domain Name System (query)
    Transaction ID: 0x7f7c
    Flags: 0x0100 Standard query
        0... .... .... .... = Response: Message is a query
        .000 0... .... .... = Opcode: Standard query (0)
        .... ..0. .... .... = Truncated: Message is not truncated
        .... ...1 .... .... = Recursion desired: Do query recursively
        .... .... .0.. .... = Z: reserved (0)
        .... .... ...0 .... = Non-authenticated data: Unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
    Queries
        cn.linkedin.com: type A, class IN
            Name: cn.linkedin.com
            [Name Length: 15]
            [Label Count: 3]
            Type: A (Host Address) (1)
            Class: IN (0x0001)
    [Response In: 2685]
```

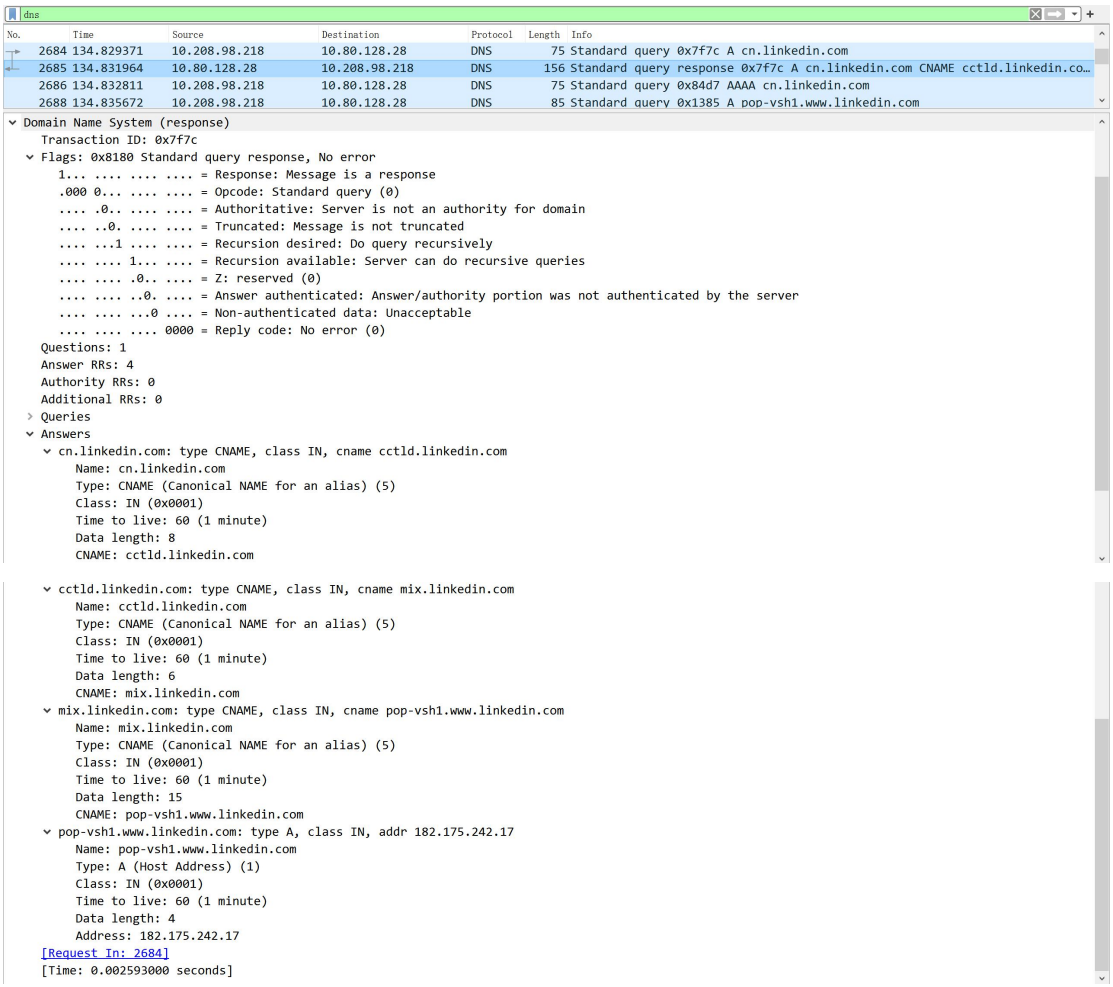
Transaction ID: 0x7f7c 标识字段，用于辨别 DNS 回答报文是哪个查询报文的响应
Flags: 0x0100 Standard query 递归请求
0... .... = Response: Message is a query 0 表示查询
.000 0... .... = Opcode: Standard query (0) 表示查询或响应类型，0 表示标准
.... ..0. .... = Truncated: Message is not truncated 截断，0 表示没有发生截断
....1 .... = Recursion desired: Do query recursively 是否希望得到递归回答
.... .0.. .... = Z: reserved (0) 保留字段
....0 .... = Non-authenticated data: Unacceptable 保留字段
Questions: 1 问题数
Answer RRs: 0 资源记录数
Authority RRs: 0 授权资源记录数
Additional RRs: 0 额外资源记录数
Queries 查询或者响应的正文部分
cn.linkedin.com: type A, class IN
Name: cn.linkedin.com 查询名称
[Name Length: 15]
[Label Count: 3]
Type: A (Host Address) (1) 查询类型，这里是主机 A 记录
Class: IN (0x0001) 类，IN 表示 Internet 数据,通常为 1
[Response In: 2685]

接着，分析 DNS 回答报文：

可以看到和查询报文相比，回答报文多出了一个 Answers 字段，同时 Flags 字段每一位都有定义。可见，Flags 中 Answer RRs 为 4 说明对应的 Answers 字段中将会出现 4 项解析结果。

实验一 应用层

09019204_曹邹颖



```
dns
No.     Time           Source          Destination     Protocol  Length Info
   2684 134.829371     10.208.98.218   10.80.128.28    DNS         75 Standard query 0x7f7c A cn.linkedin.com
   2685 134.831964     10.80.128.28    10.208.98.218   DNS        156 Standard query response 0x7f7c A cn.linkedin.com CNAME cctld.linkedin.co…
   2686 134.832811     10.208.98.218   10.80.128.28    DNS         75 Standard query 0x84d7 AAAA cn.linkedin.com
   2688 134.835672     10.208.98.218   10.80.128.28    DNS         85 Standard query 0x1385 A pop-vsh1.www.linkedin.com

Domain Name System (response)
    Transaction ID: 0x7f7c
    Flags: 0x8180 Standard query response, No error
        1... .... .... .... = Response: Message is a response
        .000 0... .... .... = Opcode: Standard query (0)
        .... .0.. .... .... = Authoritative: Server is not an authority for domain
        .... ..0. .... .... = Truncated: Message is not truncated
        .... ...1 .... .... = Recursion desired: Do query recursively
        .... .... 1... .... = Recursion available: Server can do recursive queries
        .... .... .0.. .... = Z: reserved (0)
        .... .... ..0. .... = Answer authenticated: Answer/authority portion was not authenticated by the server
        .... .... ...0 .... = Non-authenticated data: Unacceptable
        .... .... .... 0000 = Reply code: No error (0)
    Questions: 1
    Answer RRs: 4
    Authority RRs: 0
    Additional RRs: 0
    Queries
    Answers
        cn.linkedin.com: type CNAME, class IN, cname cctld.linkedin.com
            Name: cn.linkedin.com
            Type: CNAME (Canonical NAME for an alias) (5)
            Class: IN (0x0001)
            Time to live: 60 (1 minute)
            Data length: 8
            CNAME: cctld.linkedin.com

        cctld.linkedin.com: type CNAME, class IN, cname mix.linkedin.com
            Name: cctld.linkedin.com
            Type: CNAME (Canonical NAME for an alias) (5)
            Class: IN (0x0001)
            Time to live: 60 (1 minute)
            Data length: 6
            CNAME: mix.linkedin.com
        mix.linkedin.com: type CNAME, class IN, cname pop-vsh1.www.linkedin.com
            Name: mix.linkedin.com
            Type: CNAME (Canonical NAME for an alias) (5)
            Class: IN (0x0001)
            Time to live: 60 (1 minute)
            Data length: 15
            CNAME: pop-vsh1.www.linkedin.com
        pop-vsh1.www.linkedin.com: type A, class IN, addr 182.175.242.17
            Name: pop-vsh1.www.linkedin.com
            Type: A (Host Address) (1)
            Class: IN (0x0001)
            Time to live: 60 (1 minute)
            Data length: 4
            Address: 182.175.242.17
    [Request In: 2684]
    [Time: 0.002593000 seconds]
```

Transaction ID: 0x7f7c 标识字段，可见该回答报文是上面查询报文的响应

Flags: 0x8180 Standard query response, No error

1... .... = Response: Message is a response 1 表示回答

.000 0... .... = Opcode: Standard query (0) 表示查询或响应类型，0 表示标准

.... .0.. .... = Authoritative: Server is not an authority for domain 表示服务器不是所请求名字的权威 DNS 服务器

.... ..0. .... = Truncated: Message is not truncated 截断，0 表示没有发生截断

....1 .... = Recursion desired: Do query recursively 表示对应的查询报文是递归请求

.... 1... .... = Recursion available: Server can do recursive queries 表示递归可用

.... .0.. .... = Z: reserved (0) 保留字段

....0. .... = Answer authenticated: Answer/authority portion was not authenticated by the server 保留字段

....0 .... = Non-authenticated data: Unacceptable 保留字段

.... 0000 = Reply code: No error (0) 返回码表示响应的差错状态

Questions: 1 问题数

Answer RRs: 4 回答数

Authority RRs: 0 授权资源记录数

Additional RRs: 0 额外资源记录数

Queries 同查询报文处，故折叠

Answers 回答问题区域字段
cn.linkedin.com: type CNAME, class IN, cname cctld.linkedin.com 资源记录，
类型 CNAME 表示能够向请求主机提供一个主机名对应的规范主机名
Name: cn.linkedin.com 主机别名
Type: CNAME (Canonical NAME for an alias) (5)
Class: IN (0x0001)
Time to live: 60 (1 minute) 表示该资源记录的生命周期
CNAME: cctld.linkedin.com
cctld.linkedin.com: type CNAME, class IN, cname mix.linkedin.com
Name: cctld.linkedin.com
Type: CNAME (Canonical NAME for an alias) (5)
Class: IN (0x0001)
Time to live: 60 (1 minute)
Data length: 6
CNAME: mix.linkedin.com
mix.linkedin.com: type CNAME, class IN, cname pop-vsh1.www.linkedin.com
Name: mix.linkedin.com
Type: CNAME (Canonical NAME for an alias) (5)
Class: IN (0x0001)
Time to live: 60 (1 minute)
Data length: 15
CNAME: pop-vsh1.www.linkedin.com
pop-vsh1.www.linkedin.com: type A, class IN, addr 182.175.242.17 资源记录，
类型 A 表示提供了标准的主机名到 IP 地址的映射
Addr(资源数据)：返回的 IP 地址
Name: pop-vsh1.www.linkedin.com
Type: A (Host Address) (1)
Class: IN (0x0001)
Time to live: 60 (1 minute)
Data length: 4
Address: 182.175.242.17
[Request In: 2684]
[Time: 0.002593000 seconds]

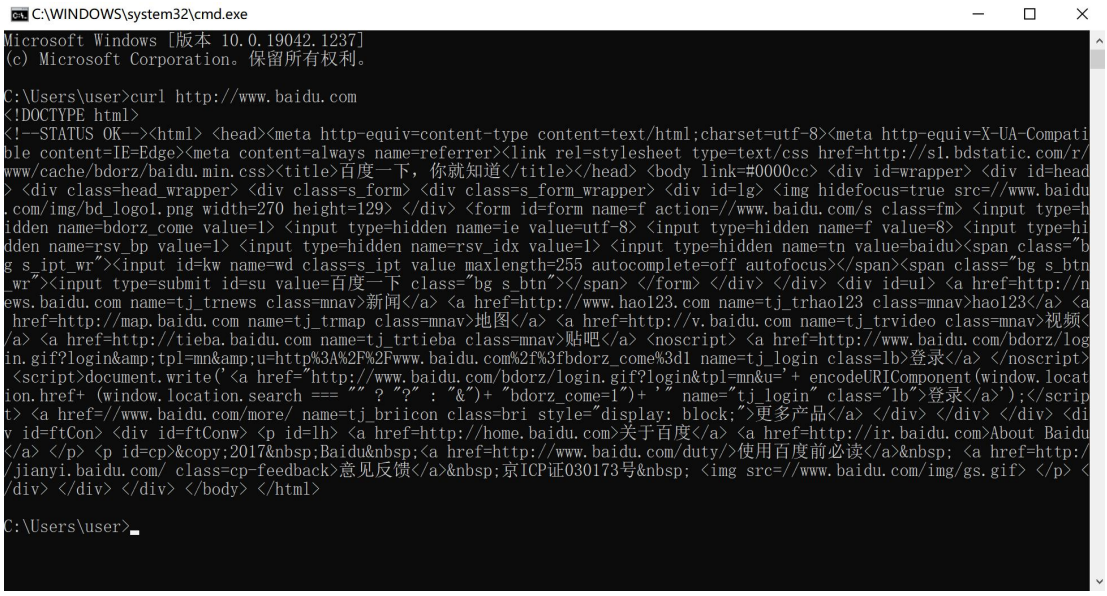
2. curl 命令测试

curl 全写是：CommandLine Uniform Resource Locator， 命令行统一资源定位器，是使用命令行访问网页 URL 的工具。

测试：

cmd 中输入：curl www.baidu.com

稍好片刻，终端会返回 Web 网站的响应源代码，效果如下:



3. telnet 命令测试 get 命令

(1) 用 win+r 打开 cmd

(2) 在 cmd 中执行 telnet www.baidu.com 80， 然后可以看到一个黑色的框框

(3) 按 ctrl +] 退出，结果为：

欢迎使用 Microsoft Telnet Client

Escape 字符是 'CTRL+]'

Microsoft Telnet>

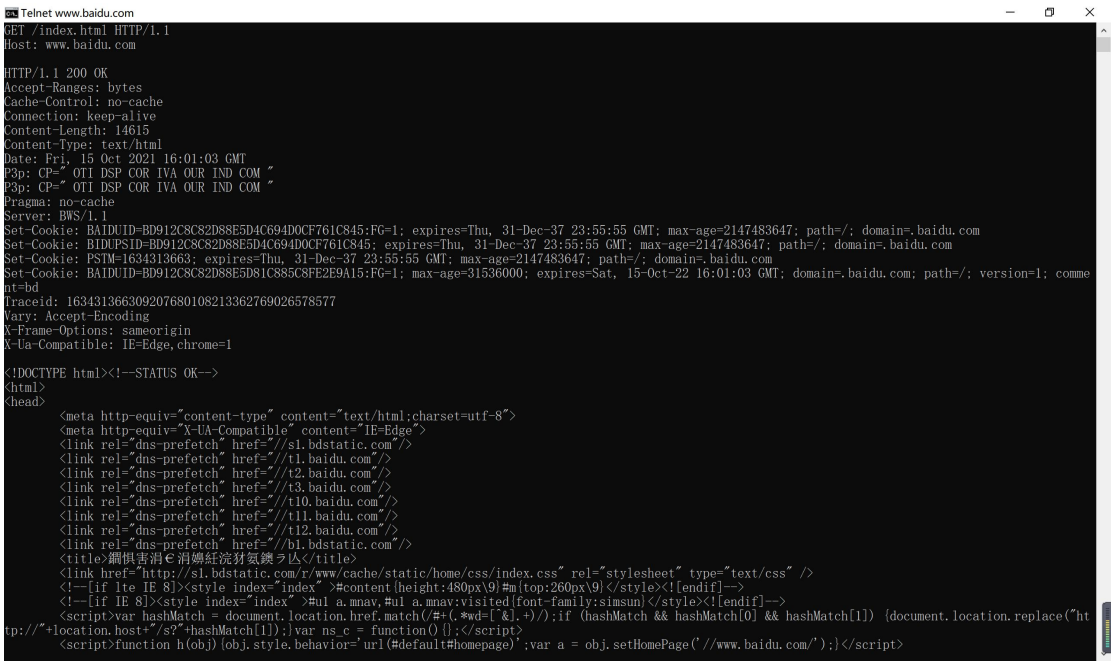
(4) 按 enter, 进入到输入框

(5) 输入如下内容（有时间限制，因此先写好然后整体拷贝进去）

GET /index.html HTTP/1.1

Host: www.baidu.com

(6) 连续按两下 enter 键盘， 得到结果如下：



4. telnet 命令测试 SMTP 服务

(1) cmd 中输入：telnet smtp.qq.com 25
客户端 TCP 连接邮件服务器 25 端口，如下图：

```
C:\Users\user>telnet smtp.qq.com 25
```

(2) 三次握手以后，连接建立成功，SMTP 服务器（邮件服务器 S）发送服务就绪信息，这里 220 代表服务就绪，后接服务器的主机名；

```
C:\WINDOWS\system32\cmd.exe
220 newxmesmtplogicsvrsza23.qq.com XMail Esmtp QQ Mail Server.
```

(3) 客户端通过 helo 命令向服务器表明身份，交代自己认证 SMTP 服务器的域名，这里采用我自己的 QQ 邮箱；

```
C:\WINDOWS\system32\cmd.exe
220 newxmesmtplogicsvrsza23.qq.com XMail Esmtp QQ Mail Server.
helo qq.com
```

(4) 如果身份有效，则服务器进入等待认证状态，下面三行是 QQ 邮箱发送的内容；

```
C:\WINDOWS\system32\cmd.exe
220 newxmesmtplogicsvrsza23.qq.com XMail Esmtp QQ Mail Server.
helo qq.com
250-newxmesmtplogicsvrsza23.qq.com-10.57.80.90-19858592
250-SIZE 73400320
250 OK
```

(5) 客户端发送 auth login，向服务器请求认证；

```
C:\WINDOWS\system32\cmd.exe
220 newxmesmtplogicsvrsza23.qq.com XMail Esmtp QQ Mail Server.
helo qq.com
250-newxmesmtplogicsvrsza23.qq.com-10.57.80.90-19858592
250-SIZE 73400320
250 OK
auth login
```

(6) 如果认证请求合理，服务器将进入等待用户输入状态，这里 334 表示等待客户端输入，VXNlcm5hbWU6 表示等待输入用户名；

```
C:\WINDOWS\system32\cmd.exe
220 newxmesmtplogicsvrsza23.qq.com XMail Esmtp QQ Mail Server.
helo qq.com
250-newxmesmtplogicsvrsza23.qq.com-10.57.80.90-19858592
250-SIZE 73400320
250 OK
auth login
334 VXNlcm5hbWU6
```

(7) 客户端向服务器发送 Base64 编码后的 QQ 邮箱用户名(caozouying@qq.com)；

```
C:\WINDOWS\system32\cmd.exe
220 newxmesmtplogicsvrsza23.qq.com XMail Esmtp QQ Mail Server.
helo qq.com
250-newxmesmtplogicsvrsza23.qq.com-10.57.80.90-19858592
250-SIZE 73400320
250 OK
auth login
334 VXNlcm5hbWU6
MTc5OTc3MTY0NUBxcS5jb20=
```

(8) 服务器再次进入等待用户输入状态,这里 334 表示等待客户端输入，UGFzc3dvcmQ6 表示等待输入密码；

```
C:\WINDOWS\system32\cmd.exe
220 newxmesmtplogicsvrsza23.qq.com XMail Esmtp QQ Mail Server.
helo qq.com
250-newxmesmtplogicsvrsza23.qq.com-10.57.80.90-19858592
250-SIZE 73400320
250 OK
auth login
334 VXNlcm5hbWU6
MTc5OTc3MTY0NUBxcS5jb20=
334 UGFzc3dvcmQ6
```

(9) 客户端向服务器发送 Base64 编码后的密码(开启 IMAP/SMPT 时授权码的 base64 编码)；

```
C:\WINDOWS\system32\cmd.exe
220 newxmesmtplogicsvrsza23.qq.com XMail Esmtp QQ Mail Server.
helo qq.com
250-newxmesmtplogicsvrsza23.qq.com-10.57.80.90-19858592
250-SIZE 73400320
250 OK
auth login
334 VXNlcm5hbWU6
MTc5OTc3MTY0NUBxcS5jb20=
334 UGFzc3dvcmQ6
cWJ0dGRieHFsZmx6ZGRpYQ==
```

(10) 服务器返回 235 Authentication successful，表示认证成功；

```
C:\WINDOWS\system32\cmd.exe
220 newxmesmtplogicsvrsza23.qq.com XMail Esmtp QQ Mail Server.
helo qq.com
250-newxmesmtplogicsvrsza23.qq.com-10.57.80.90-19858592
250-SIZE 73400320
250 OK
auth login
334 VXNlcm5hbWU6
MTc5OTc3MTY0NUBxcS5jb20=
334 UGFzc3dvcmQ6
cWJ0dGRieHFsZmx6ZGRpYQ==
235 Authentication successful
```

(11) 客户端通过 mail from 命令告诉服务器邮件来自何方；

(12) 服务端返回 250 OK 表示成功；

(13) 客户端通过 rcpt to 命令告诉服务器邮件去往何地；

(14) 服务端返回 250 OK 表示成功；

(15) 客户端通过 data 命令告诉服务器自己准备发送邮件正文；

(16) 服务器返回 354 表示准备接受邮件并提醒客户端开始发送邮件并以“.”结束；

```
mail from:<zouyingcao@qq.com>
250 OK.
rcpt to:<1048037069@qq.com>
250 OK
data
354 End data with <CR><LF>.<CR><LF>.
```

(17) 客户端发送邮件正文；客户端发送完正文以后，紧接着发送结束符“.”；

```
Subject:this is my computer network experiment
from:<zouyingcao@qq.com>
to:<1048037069@qq.com>

oh my dear friend~
such a cute girl~
.
```

(18) 如果合理，服务端返回“250 OK:queued as.\r\n”表示发送成功；

(19) 客户端通过 quit 命令表示邮件发送结束，客户端请求断开连接；

(20) 服务器返回“211 Bye.”表示断开申请被采纳并主动断开连接，邮件发送过程结束。

```
250 OK: queued as.
quit
221 Bye.
```

查看邮件服务结果：

---------- Original ----------

From: zouyingcao <zouyingcao@qq.com>
Date: Sat,Oct 16,2021 11:21 AM
To: 1048037069 <1048037069@qq.com>
Subject: Fw: this is my computer network experiment

oh my dear friend~
such a cute girl~
.

5. tracert 命令测试

① **cmd 中输入：**tracert www.baidu.com

表示追踪当前 IP 到 www.baidu.com 域名指向的 IP 地址所经过的路由地址列表。

```
C:\Users\user>tracert www.baidu.com

通过最多 30 个跃点跟踪
到 www.a.shifen.com [112.80.248.76] 的路由:

  1    16 ms     3 ms     *      10.208.64.1
  2     3 ms     2 ms     5 ms   10.80.128.141
  3     8 ms     4 ms     6 ms   10.80.128.149
  4     1 ms     1 ms     1 ms   10.80.3.10
  5     3 ms     3 ms     2 ms   153.3.60.1
  6     4 ms     7 ms     7 ms   221.6.2.173
  7     3 ms     3 ms     3 ms   112.86.192.134
  8     3 ms     3 ms     3 ms   182.61.216.0
  9     *        *        *      请求超时。
 10     3 ms     2 ms     2 ms   112.80.248.76

跟踪完成。
```

首先，会自动将 www.baidu.com 域名找到其对应的 ip 地址——112.80.248.76，并提示到达目的地址的路由跃点估算——最多 30 个；

接下来，每一行为所经过的一个路由地址，包括序号、3 次实验的往返时延、路由 IP；其中出现序号后面是*号，且有请求超时的提示，可能原因是路由跃点禁 PING 或者路由跃点不对 TTL 超时做响应处理，直接丢弃；

最后提示追踪完成表示命令执行完毕。

② **解析原理过程：**

tracert 命令用 IP 生存时间 (TTL) 字段和 ICMP 错误消息来确定从一个主机到网络上其他主机的路由。

首先，tracert 发出一个 TTL=1 的数据包到目的地，当路径上的第一个路由器收到这个数据包时，它将其 TTL 减 1。此时，TTL 变为 0，所以该路由器会将此数据包丢掉，并送回一个「ICMP time exceeded」消息（包括发 IP 包的源地址，IP 包的所有内容及路由器的 IP 地址），tracert 便知道这个路由器为追踪路径上的第一个路由，接着 tracert 再发出一个 TTL=2 的数据包，发现第 2 个路由器.....

就这样，tracert 每次将发出的数据包 TTL 加 1 来发现路径中下一个路由器，这个重复的动作一直持续到某个数据包抵达目的地。当数据包到达目的地后，该主机则不会返回「ICMP time exceeded」消息，一旦到达目的地，由于 tracert 通过 UDP 数据包向不常见端口(30000 以上)发送数据包，因此会收到「ICMP port unreachable」消息，故可判断到达目的地。

6. nslookup 查询

Nslookup 全称为 name server lookup(域名查询)，

测试：

① 直接查询：cmd 中输入：nslookup www.baidu.com

② 查询其它记录：

nslookup -qt = type domain [dns-server]

选择 type：MX→邮件服务器记录

cmd 中输入：nslookup -qt=mx www.baidu.com

三．实验体会

在 Wireshark 实验中，选择好过滤器，不然会有很多妨碍观察 HTTP/DNS 协议的数据包，增加无谓的寻找时间。分析 HTTP 协议时，还可以将访问的 IP 地址作为过滤条件。在使用 Wireshark 抓取 HTTP 和 DNS 数据包时，对 HTTP 与 DNS 报文内容有了更全面的学习与了解。

同时，学习了多种在命令行中经常使用的命令，例：curl 与服务器之间传输数据，telnet 测试 get 命令以及远程登录 SMTP 服务器发邮件，tracert 追踪到达某个网站的路由信息，nslookup 从域名中解析 IP 地址等。