

实验人：谢志康
学号：22307110187
实验内容：应用层协议模拟与观察

提交内容 1：进行以上操作和获取到的应答的命令行截图，并与直接使用浏览器访问该网页的内容进行对比。

```
This message is shown once a day. To disable it please create the
/home/kurumi/.hushlogin file.
kurumi@kurumi:~$ telnet www.example.com http
Trying 2606:2800:21f:cb07:6820:80da:af6b:8b2c...
Connected to www.example.com.
Escape character is '^['.
GET / HTTP/1.1
Host: www.example.com

HTTP/1.1 200 OK
Accept-Ranges: bytes
Age: 98653
Cache-Control: max-age=604800
Content-Type: text/html; charset=UTF-8
Date: Fri, 27 Sep 2024 05:58:21 GMT
Etag: "3147526947"
Expires: Fri, 04 Oct 2024 05:58:21 GMT
Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT
Server: ECAcc (lac/55C2)
Vary: Accept-Encoding
X-Cache: HIT
Content-Length: 1256

<!doctype html>
<html>
<head>
  <title>Example Domain</title>

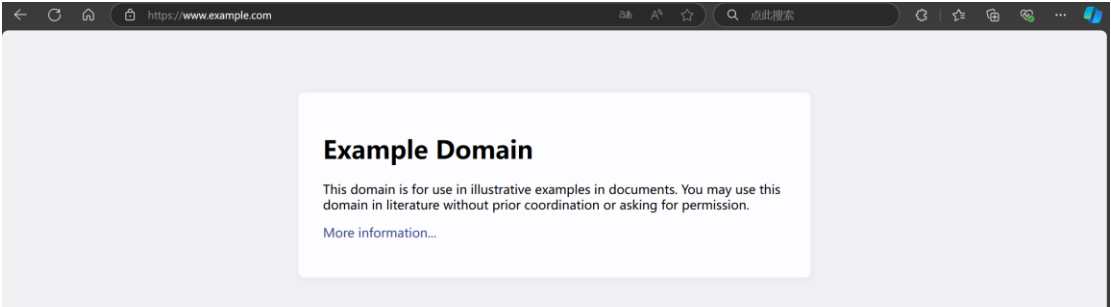
  <meta charset="utf-8" />
  <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <style type="text/css">

    body {
      background-color: #f0f0f2;
      margin: 0;
      padding: 0;
      font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;
    }

    <div>
      width: 600px;
      margin: 5em auto;
      padding: 2em;
      background-color: #fdfdff;
      border-radius: 0.5em;
      box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);
    </div>
    a:link, a:visited {
      color: #38488f;
      text-decoration: none;
    }
    @media (max-width: 700px) {
      <div>
        margin: 0 auto;
        width: auto;
      </div>
    }
  </style>
</head>

<body>
<div>
  <h1>Example Domain</h1>
  <p>This domain is for use in illustrative examples in documents. You may use this
  domain in literature without prior coordination or asking for permission.</p>
  <p><a href="https://www.iana.org/domains/example">More information...</a></p>
</div>
</body>
</html>
Connection closed by foreign host.
kurumi@kurumi:~$
```

可以看到 HTTP/1.1 200 OK 表示服务器正常地应答了请求。浏览器打开 example.com



可以看到内容是一致的，我们请求获取到的信息和浏览器实际打开这个页面相同，获取到的

是一个 html 文件：

<title>Example Domain</title> 定义这个页面在浏览器中显示的 title



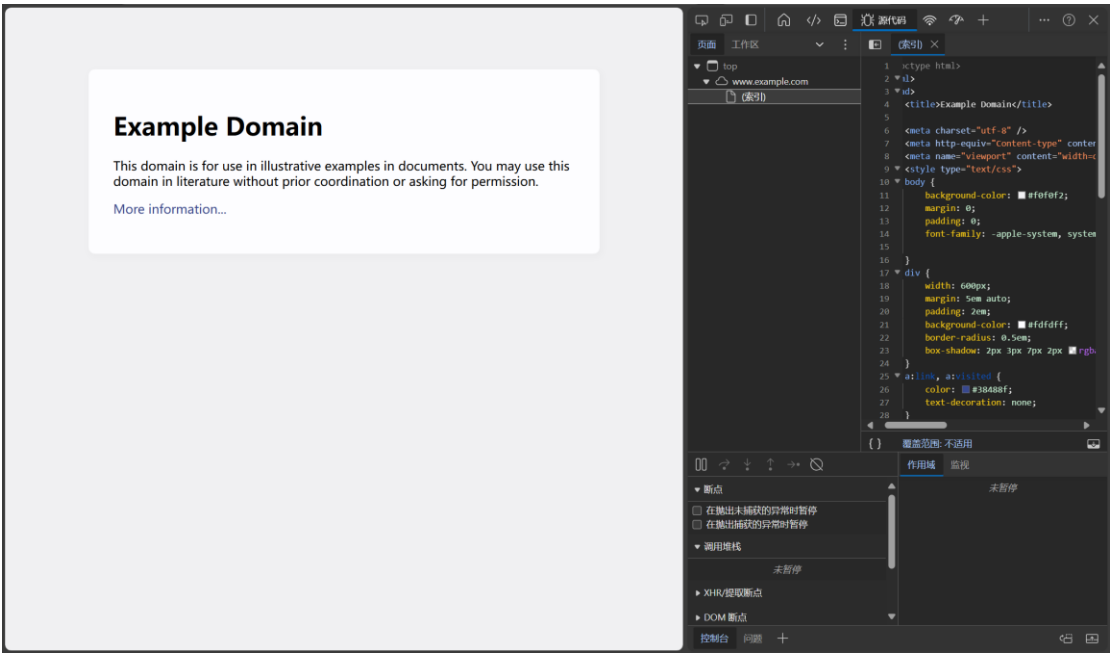
接下来一段只是定义了一下 css 样式

之后 body 部分就是内容，与浏览器上显示的一样

```
<body>
<div>
  <h1>Example Domain</h1>
  <p>This domain is for use in illustrative examples in documents. You may use this
  domain in literature without prior coordination or asking for permission.</p>
  <p><a href="https://www.iana.org/domains/example">More information...</a></p>
</div>
</body>
```

H1 加粗的 Example Domain 字样，底下是<p>小字，href 插入超链接，www.iana.org..... 这个网站即是链接的目的网址，点进去发现确实是这个网站，More information 是显示的字样。

进一步检查，f12 打开浏览器的 console，可以看到这个界面的 html 和我们 telnet 获取到的一模一样：



提交内容 2: 使用 telnet 发送邮件的过程截图，以及网页客户端中对方收到的邮件的截图。

P3/SMTP/IMAP

开启服务：

IMAP/SMTP服务

已开启 | 关闭

POP3/SMTP服务

已关闭 | 开启

POP3/SMTP/IMAP服务能让你在本地客户端上收发邮件，[了解更多 >](#)

温馨提示：在第三方登录网易邮箱，可能存在邮件泄露风险，甚至危害Apple或其他平台账户安全

首先网易邮箱要开启 SMTP 服务才行，不然登录报错 550 User has no permission
然后网易会给一个仅展示一次的特殊密码，将其 base64 转换后登录即可。

```

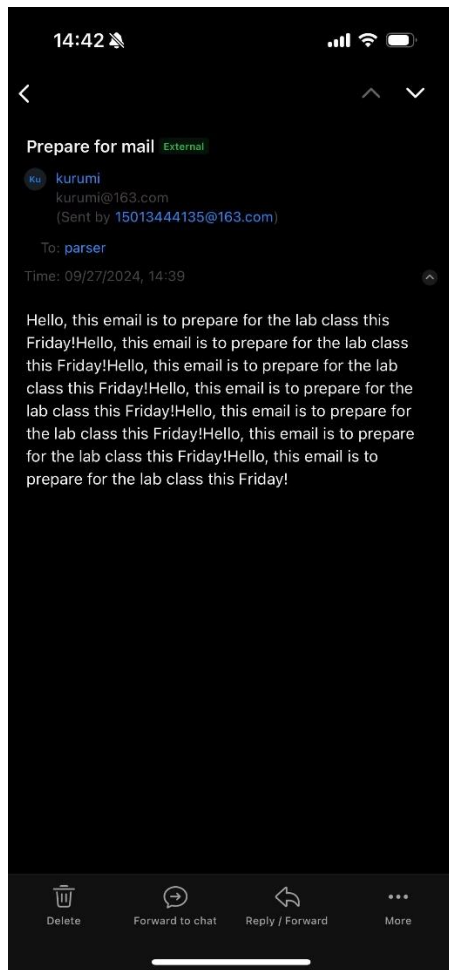
kurumi@kurumi:~$ telnet smtp.163.com 25
Trying 1.95.20.21...
Connected to smtp163.mail.ntes53.netease.com.
Escape character is '^]'.
220 163.com Anti-spam GT for Coremail System (163com[20141201])
helo kurumi
250 OK
auth login
334 dXNlcm5hbWU6
MTUwMTM0NDQxMzVAMTYzLmNvbQ==
334 UGFzc3dvcmQ6
UUbRdu5kK4te7cFZ
535 Error: authentication failed
auth login
334 dXNlcm5hbWU6
MTUwMTM0NDQxMzVAMTYzLmNvbQ==
334 UGFzc3dvcmQ6
VVViUmR1NWtLNHRlN2NGWg==
235 Authentication successful

```

```
vovvumRlNwELNHRRkZNCng==  
235 Authentication successful  
mail from:<15013444135@163.com>  
250 Mail OK  
rcpt to:<22307110187@m.fudan.edu.cn>  
250 Mail OK  
data  
354 End data with <CR><LF>.<CR><LF>  
subject: Prepare for mail  
from:kurumi@163.com  
to:parser@fudan.edu.cn  
  
Hello, this email is to prepare for the lab class this Friday!Hello, this email is to prepare for the lab class this Friday!Hello, this email is to prepare for the lab class this Friday!Hello, this email is to prepare for the lab class this Friday!Hello, this email is to prepare for the lab class this Friday!  
. . . s this Friday!  
  
250 Mail OK queued as gzga-smtp-mta-g3-3,_____wDHBnV6UvZm+mFoFg--..25574S2 1727419189  
Connection closed by foreign host.
```



可以看到已发送邮件确实有刚刚发的邮件,但是真实发件人还是被 detect 出来,是我本人。我另一个邮箱也确实收到了邮件。



提交内容 3: 在发送邮件的过程中，通过 Wireshark 抓包，截图并分析 SMTP 发送邮件的过程。

接下来从新发送一次，使用 wireshark 抓包：

Capturing from WLAN						
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
smtp						
No.	Time	Source	Destination	Protocol	Length	Info
326	5.125625	1.95.20.21	10.223.28.252	SMTP	119	S: 220 163.com Anti-spam GT for Coremail System (163com[20141201])
508	8.352876	10.223.28.252	1.95.20.21	SMTP	67	C: helo kurumi
544	8.404809	1.95.20.21	10.223.28.252	SMTP	62	S: 250 OK
806	15.296983	10.223.28.252	1.95.20.21	SMTP	66	C: auth login
808	15.342084	1.95.20.21	10.223.28.252	SMTP	72	S: 334 d00lcn5hblu6
1892	18.403620	10.223.28.252	1.95.20.21	SMTP	84	C: User: MTUuH7MNDQHzVAHTYzLmNvbQ==
1131	18.448594	1.95.20.21	10.223.28.252	SMTP	72	S: 334 UGfzc3dvcmQ6
1358	22.554596	10.223.28.252	1.95.20.21	SMTP	80	C: Pass: VVVuMR1NwtLNHR1N2NGhg==
1361	22.656240	1.95.20.21	10.223.28.252	SMTP	85	S: 235 Authentication successful
2207	37.892292	10.223.28.252	1.95.20.21	SMTP	87	C: mail from:<15013444135@163.com>
2209	37.940948	1.95.20.21	10.223.28.252	SMTP	67	S: 250 Mail OK
3240	55.218787	10.223.28.252	1.95.20.21	SMTP	92	C: rcpt to:<23307140055@m.fudan.edu.cn>
3249	55.264085	1.95.20.21	10.223.28.252	SMTP	67	S: 250 Mail OK
3802	62.934392	10.223.28.252	1.95.20.21	SMTP	60	C: data
3803	62.977349	1.95.20.21	10.223.28.252	SMTP	91	S: 354 End data with <CR><LF>.<CR><LF>
4726	77.716056	10.223.28.252	1.95.20.21	SMTP	81	C: DATA fragment, 27 bytes
5946	98.092134	10.223.28.252	1.95.20.21	SMTP	75	C: DATA fragment, 21 bytes
7532	124.716015	10.223.28.252	1.95.20.21	SMTP	96	C: DATA fragment, 42 bytes
8114	135.148683	10.223.28.252	1.95.20.21	SMTP	56	C: DATA fragment, 2 bytes
9870	163.215801	10.223.28.252	1.95.20.21	SMTP	88	C: DATA fragment, 34 bytes
10664	176.695127	10.223.28.252	1.95.20.21	SMTP	62	C: DATA fragment, 8 bytes
10719	178.023518	10.223.28.252	1.95.20.21	SMTP	62	C: DATA fragment, 8 bytes
10814	179.498088	10.223.28.252	1.95.20.21	SMTP	62	C: DATA fragment, 8 bytes
10974	182.607373	10.223.28.252	1.95.20.21	SMTP/L	57	subject: Prepare for mail, from: kurumi@163.com, , hello!my bro was hacked! hiahial , kurumi , kurumi , kurumi C: .
10976	182.673927	1.95.20.21	10.223.28.252	SMTP	129	S: 250 Mail OK queued as gzsmt2,s5gvCg8U0kFVvZmjyAAAQ--..2587852 1727420088
12182	201.588636	10.223.28.252	1.95.20.21	SMTP	56	C: DATA fragment, 2 bytes
12184	201.634417	1.95.20.21	10.223.28.252	SMTP	88	S: 421 closing transmission channel

和我的操作步骤一样：

最开始建立连接，我输入 helo kurumi，对面响应正常，回复 250 OK

然后我登录，可以看到 user 和 pass 都是我输入的账号密码的 base64 编码，理论上只用 base64 解码即可获得明文账号密码，但是网易邮箱还是提供了保护机制的，我的实际密码登录它会显示 no permission，开启 SMTP 服务后它会给我一个临时的密码才能 SMTP 登录，但是这个密码并不是我设置的实际密码，所以还是安全的。

返回登录成功后开始写邮件 from 和 to 是实际的发件人和收件人。后面的一段 data fragment 是我发件的信息，随便打开一个查看：

有这个 data 创建时间信息

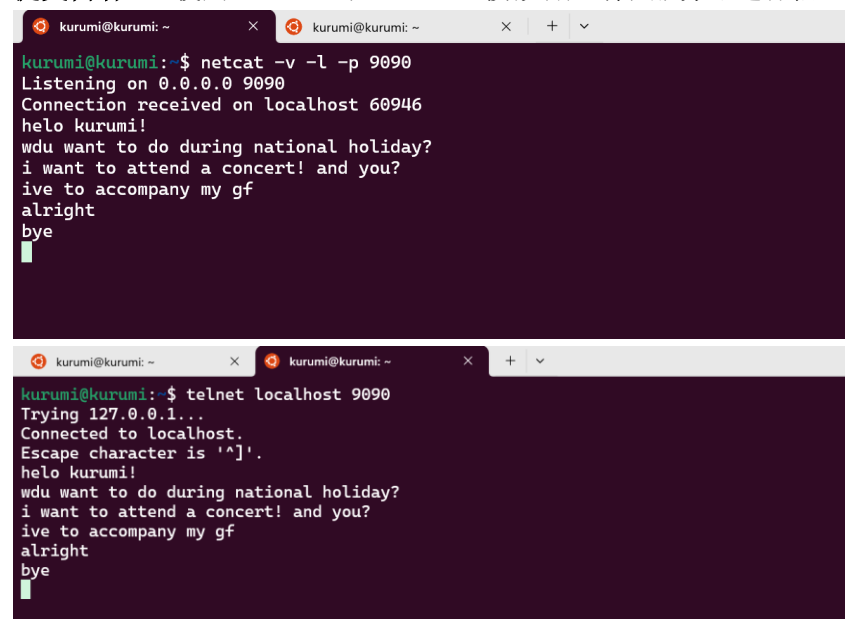
```
Section number: 1
> Interface id: 0 (\Device\NPF_{3ACA0BC3-D7EC-4E35-8D43-F1637074A865})
Encapsulation type: Ethernet (1)
Arrival Time: Sep 27, 2024 14:53:50.481542000 中国标准时间
UTC Arrival Time: Sep 27, 2024 06:53:50.481542000 UTC
Epoch Arrival Time: 1727420030.481542000
[Time shift for this packet: 0.000000000 seconds]
[Time delta from previous captured frame: 0.093331000 seconds]
[Time delta from previous displayed frame: 26.623881000 seconds]
[Time since reference or first frame: 124.716015000 seconds]
```

再看最后的 SMTP：

```
> Internet Protocol Version 4, Src: 10.223.28.252, Dst: 1.95.20.21
> Transmission Control Protocol, Src Port: 60946, Dst Port: 25, Seq: 207, Ack: 204, Len: 42
  Simple Mail Transfer Protocol
    Line-based text data (1 lines)
      to:what_can_i_say@yinzhitao.fudan.edu.cn\r\n
      [Reassembled DATA in frame: 10974]
```

可以看到这一行我写的信件的真实内容。

提交内容 4：使用 Telnet 和 Netcat 模拟客户端和服务端进行信息传输的截图。



The image shows two terminal windows side-by-side. The top window is a netcat listener on port 9090, and the bottom window is a telnet client connecting to localhost 9090. Both windows show the same conversation: 'helo kurumi!', 'wdu want to do during national holiday?', 'i want to attend a concert! and you?', 'ive to accompany my gf', 'alright', and 'bye'.

```
kurumi@kurumi: ~  
kurumi@kurumi:~$ netcat -v -l -p 9090  
Listening on 0.0.0.0 9090  
Connection received on localhost 60946  
helo kurumi!  
wdu want to do during national holiday?  
i want to attend a concert! and you?  
ive to accompany my gf  
alright  
bye
```

```
kurumi@kurumi: ~  
kurumi@kurumi:~$ telnet localhost 9090  
Trying 127.0.0.1...  
Connected to localhost.  
Escape character is '^]'.  
helo kurumi!  
wdu want to do during national holiday?  
i want to attend a concert! and you?  
ive to accompany my gf  
alright  
bye
```

提交内容 5: 观察并分析 DNS 查询过程, 判断查询采用的是递归查询还是迭代查询, 并说明理由。

```
kurumi@kurumi:~$ dig +trace www.baidu.com
; <<>> DiG 9.18.28-0ubuntu0.22.04.1-Ubuntu <<>> +trace www.baidu.com
;; global options: +cmd
.      344729 IN      NS      a.root-servers.net.
.      344729 IN      NS      b.root-servers.net.
.      344729 IN      NS      c.root-servers.net.
.      344729 IN      NS      d.root-servers.net.
.      344729 IN      NS      e.root-servers.net.
.      344729 IN      NS      f.root-servers.net.
.      344729 IN      NS      g.root-servers.net.
.      344729 IN      NS      h.root-servers.net.
.      344729 IN      NS      i.root-servers.net.
.      344729 IN      NS      j.root-servers.net.
.      344729 IN      NS      k.root-servers.net.
.      344729 IN      NS      l.root-servers.net.
.      344729 IN      NS      m.root-servers.net.
;; Received 811 bytes from 10.255.255.254#53(10.255.255.254) in 29 ms

com.   172800 IN      NS      c.gtld-servers.net.
com.   172800 IN      NS      h.gtld-servers.net.
com.   172800 IN      NS      b.gtld-servers.net.
com.   172800 IN      NS      g.gtld-servers.net.
com.   172800 IN      NS      f.gtld-servers.net.
com.   172800 IN      NS      m.gtld-servers.net.
com.   172800 IN      NS      a.gtld-servers.net.
com.   172800 IN      NS      l.gtld-servers.net.
com.   172800 IN      NS      d.gtld-servers.net.
com.   172800 IN      NS      j.gtld-servers.net.
com.   172800 IN      NS      k.gtld-servers.net.
com.   172800 IN      NS      i.gtld-servers.net.
com.   172800 IN      NS      e.gtld-servers.net.
com.   86400  IN      DS      19718 13 2 8ACBB8CD28F41250A80A491389424D341522D946B0DA8C0291F2D307 71D7805A
com.   86400  IN      RRSIG  DS 8 1 86400 20241009170000 20240926160000 20038 . Tkmg/jXUIpsnGjxBgIe4bIrxKcFkpCrel7hcNzvVla
43IcPwNo1Cv67ZL 18136G6iGtIPX8LCH/rKjijfz9IC9okjmo5CkcpmHYLUsUp9+OHFEJmRc zp2qq3LNgz6fwhatoz390sA/xIzGpUfrBhiEJ01HfqnuLLmeco7Y4tr2 xybN9mYHAF
5TrhXSmFTYtp0D/4/RrWkZeiF6GD0rK6On9P85wnRIP8I nfvu+ue1AhrTliq3BPRHPRosIFdZMAsgJto5Upks5YD/xCG+KVUDt1cw ZABZrt+SM/zD2edxnH3d5xYn90jQwzJrHgV
sg8CY2F+GdU7mQ0c3j03k 7WsnWm==
;; Received 1201 bytes from 2001:dc3::35#53(m.root-servers.net) in 29 ms

baidu.com. 172800 IN      NS      ns2.baidu.com.
baidu.com. 172800 IN      NS      ns3.baidu.com.
baidu.com. 172800 IN      NS      ns4.baidu.com.
baidu.com. 172800 IN      NS      ns1.baidu.com.
baidu.com. 172800 IN      NS      ns7.baidu.com.
CK0P0JMG874LJREF7EFN8430QVIT8BSM.com. 86400 IN NSEC3 1 1 0 - CK0Q3UDG8CEKAE7RUKPGCT1DVSSH8LL NS SOA RRSIG DNSKEY NSEC3PARAM
CK0P0JMG874LJREF7EFN8430QVIT8BSM.com. 86400 IN RRSIG NSEC3 13 2 86400 20241003002550 20240925231550 59354 com. xyjdNvYr7t+PA8uOz3Zx0T0L5jJN6s
bdas8TxWJbNhhjK0SgWwtcjymJ6 +c77A05p8MksLFdyqcp0d0ZeS0t3Hg==
HPV1V1UNKTCF9TD77I2AUR73709T975GH.com. 86400 IN NSEC3 1 1 0 - HPVVP23QU08FP9R0A0HUR5IC3PESK09J NS DS RRSIG
HPV1V1UNKTCF9TD77I2AUR73709T975GH.com. 86400 IN RRSIG NSEC3 13 2 86400 20241002004732 20240924233732 59354 com. N0nBU5qYLApgZC/2oocGBdwACu5EH
tNSZ2mYd+IKzoe9j2C4rsgFLG7b sP+kyaf9nTJilbESsXJh0dUXrCSMqA==
;; Received 657 bytes from 2001:503:d2d::30#53(k.gtld-servers.net) in 329 ms

www.baidu.com. 1200 IN      CNAME   www.a.shifen.com.
;; Received 100 bytes from 153.3.238.93#53(ns3.baidu.com) in 9 ms
```

这里采用的是**迭代查询**。

递归查询:服务器必需回答目标 IP 与域名的映射关系。要求 DNS 服务器处理整个查询过程, 直到得到最终的结果。服务器需要进行多次查询, 但不是将每一步都返回给客户端。

迭代查询:服务器收到一次迭代查询回复一次结果, 这个结果不一定是目标 IP 与域名的映射关系, 也可以是其它 DNS 服务器的地址。客户端与每个 DNS 服务器逐步交互, 服务器只返回其知道的下一个 DNS 服务器地址, 客户端负责继续查询。

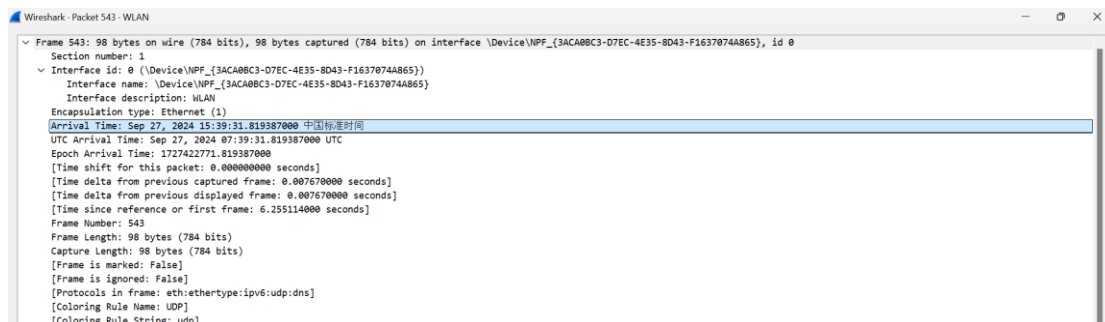
由截图可见, 查询首先向根 DNS 服务器发送请求, 根服务器返回负责 .com 域的顶级域名服务器的地址。由左边的. 变为 com. 然后查询继续向 .com 顶级域名服务器发送请求, 获取负责 baidu.com 的域名服务器的地址。最后查询到达 baidu.com 的权威 DNS 服务器, 获取到 www.baidu.com 的 CNAME 记录。

(同时, 据我在网上了解到的知识, dig 默认是递归方式查询, 而使用 +trace 可以禁用递归查询转变为迭代查询。命令将显示查询的每一步过程, 并提供每个服务器的响应。)

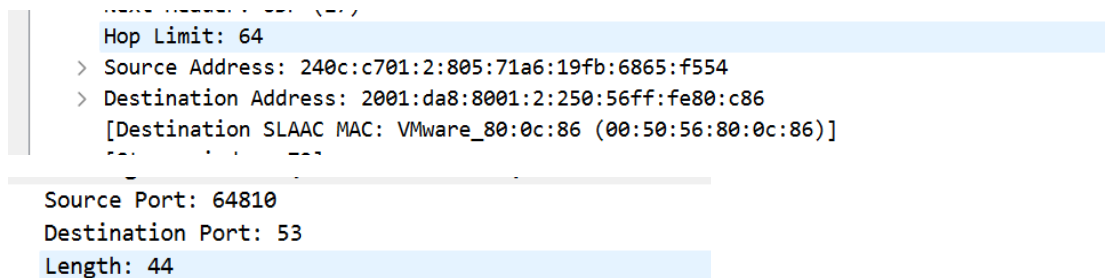
提交内容 6: 在 DNS 查询过程中使用 Wireshark 抓包, 任选一对 DNS 请求与应答分组, 解释 DNS 报文中各个字段的含义, 以及标志 (flag) 字段中各个 flag 位的含义。

542	6.247444	2001:da8:8001:2:250::240c:c701:2:805:71a::	DNS	207	Standard query response 0x4b4 AAAA k.gtld-servers.net AAAA 2001:503:d2d::30 NS av2.nstld.com NS av3.nstld.com NS...
543	6.255314	240c:c701:2:805:71a::2001:da8:8001:2:250::	DNS	98	Standard query 0xc862 AAAA g.gtld-servers.net
544	6.255270	240c:c701:2:805:71a::2001:da8:8001:2:250::	DNS	98	Standard query 0xc866 AAAA g.gtld-servers.net
545	6.257451	2001:da8:8001:2:250::240c:c701:2:805:71a::	DNS	195	Standard query response 0x6282 A g.gtld-servers.net A 192.42.93.30 NS av4.nstld.com NS av1.nstld.com NS av2.nstld...
546	6.257667	2001:da8:8001:2:250::240c:c701:2:805:71a::	DNS	207	Standard query response 0xc866 AAAA k.gtld-servers.net AAAA 2001:503:eea3::30 NS av1.nstld.com NS av2.nstld.com N...

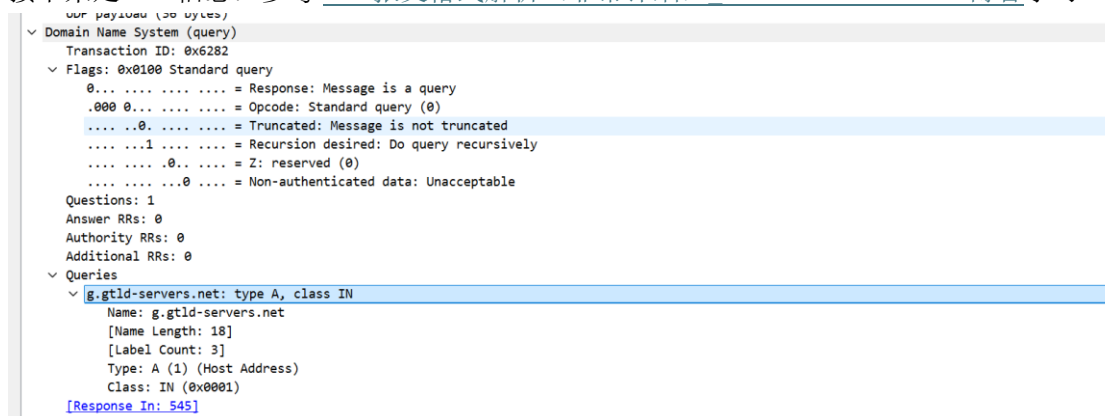
这是一对 DNS 请求与应答分组中的请求信息: 时间, frame 个数和大小都有标出



Source 和 destination 的 IP 也有标出，端口号也有标出：



接下来是 DNS 信息：参考 [DNS 报文格式解析（非常详细）](#) [dns class in-CSDN 博客](#) 学习



Transaction ID 是请求的 ID，用于将请求与对应的响应匹配。 0x6282

Flags 是 0x0100

QR (0): Message is a query——表示这是一个查询（请求包）。

Opcode (0): 标准查询。

AA (0): 非权威应答。

TC (0): Message is not truncated——报文未被截断。

RD (1): 期望递归查询。

RA (0): 响应中不允许递归（在这个请求信息中无关）。

Z (0): reserved 保留字段，必须为 0。

AD (0): unacceptable——非认证数据。

CD (0): 不检查 DNSSEC 签名。

Questions:1 问题计数，这里查询的问题数为 1

Answer RRs:0 没有答案资源记录（因为这是请求包）。

Authority RRs:0 权威名称服务器计数为 0

Additional RRs:0 附加资源记录数为 0

Name:g.tld-servers.net 请求查询的域名。

Type A (1) (Host Address): 查询记录类型为 A（即 IPv4 地址）。

Class IN (0x0001): 表示查询的是互联网地址。

再看其对应的应答分组:

前面时间啊、frame 大小等等信息, 还有 source 和 destination 的 IP 地址和端口号等等也都有, 不多赘述。

```
> Source Address: 2001:da8:8001:2:250:56ff:fe80:c86
> Destination Address: 240c:c701:2:805:71a6:19fb:6865:f554
```

主要来看看 DNS 信息:

```
▼ Domain Name System (response)
  Transaction ID: 0x6282
  ▼ 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: 1
  Authority RRs: 4
  Additional RRs: 0
  ▼ Queries
    ▼ g.gtld-servers.net: type A, class IN
      Name: g.gtld-servers.net
      [Name Length: 18]
      [Label Count: 3]
      Type: A (1) (Host Address)
      Class: IN (0x0001)
  > Answers
  > Authoritative nameservers
  [Request In: 543]
  [Time: 0.002337000 seconds]
```

Transaction ID 是 0x6282, 与它对应的请求包相同, 以此来绑定。

Flags 字段——

QR(1) Message is a response, 与前面 QR(0) 对应, 这是一个响应包

后几个字段与请求包相同。Opcode (0): 标准查询响应。AA (0): 服务器不是权威服务器 (非权威应答)。TC (0): 报文未被截断。RD (1): 递归查询被请求。多了个 Recursion available: 1 表示服务器支持递归查询。Z (0): 保留字段, 必须为 0。AD (0): 响应未被认证。CD (0): 检查 DNSSEC 签名。Reply code (0): no error 没有错误, 表示查询成功。

Questions (1): 查询的问题数为 1。

Answer RRs (1): 1 条答案资源记录, 也就是找到一个回答。

Authority RRs (4): 有 4 条授权资源记录。

Additional RRs (0): 没有附加资源记录。

Queries: 这里与请求一致, 查询的是 g.gtld-servers.net 的 A 记录 (IPv4 地址)。Class IN(0x0001) 表示查询的是互联网地址。都与请求一致。

Answers: 响应包含 g.gtld-servers.net 的 A 记录。

Authoritative nameservers: 列出了授权名称服务器的记录, 用于处理后续查询。

Response time 表示响应时间为 0.002337000 秒。