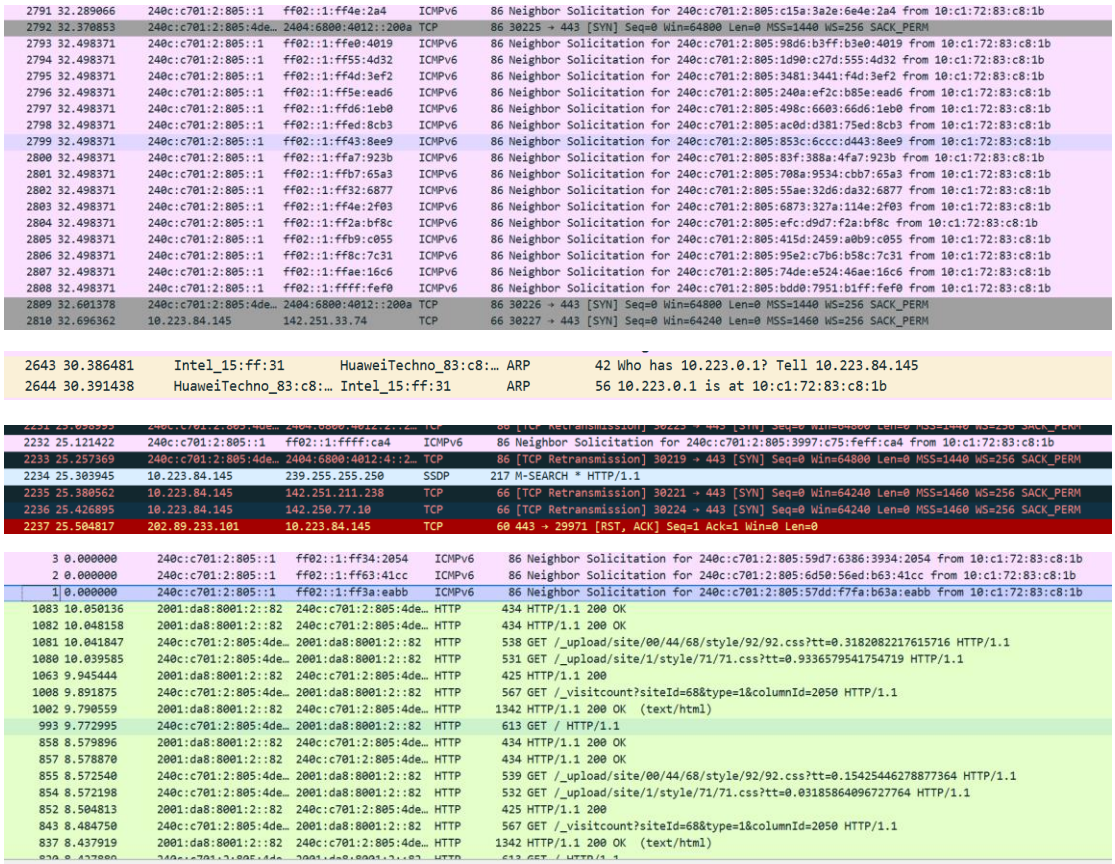
2024.9.13
计算机网络（H） Lab1
实验人：谢志康
学号：22307110187

实验内容：

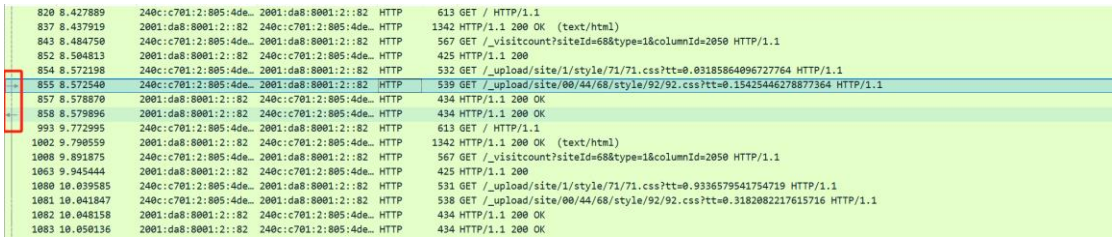
1. 请列出捕获到的 5 种不同类型的协议。



有 TCP ICMPv6 ARP SSDP HTTP 协议。

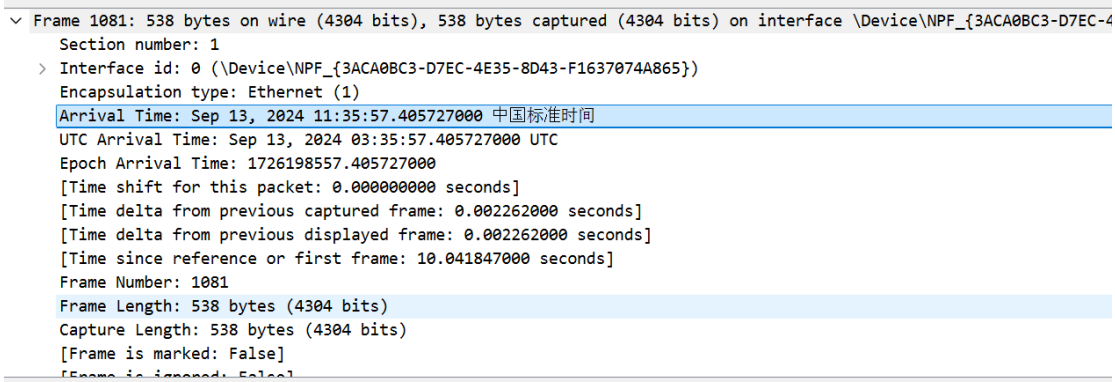
2. 用显示过滤器过滤出所有 http 消息，从发送第一条 HTTP GET 请求到收到对应的 HTTP OK 回复用了多长时间？默认情况下，数据包列表窗口中 “Time”列的值是自 Wireshark 捕获开始经过的秒数，也可以点击菜单栏“视图-时间显示格式”切换到其他格式。

清除浏览器缓存，重新运行，过滤后——按照时间排序

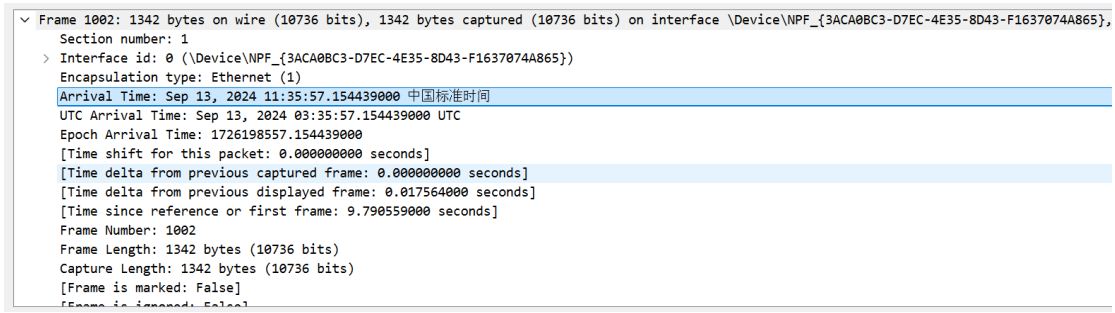


首先找到第一个确认确实是我们的目标网站（清除缓存且只打开该网页，这个应该不会出错）的 HTTP GET 信号，查看详细信息：

其次查看时间节点：11:35:57.405727000

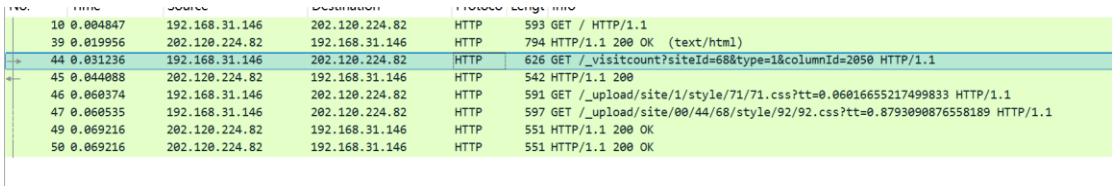


再看第一个 OK 信号：时间节点：11:35:57.154439000



所以，时间差为 0.405727-0.154439 = 0.251288

3. 复旦信息办的 IP 地址是什么？你的计算机发送 HTTP GET 请求时的 IP 地址是什么？



GET 情况下，本机发送，Source 即为本机 IPv4 地址：192.168.31.146
复旦信息办的 IPv4 地址为：202.120.224.82

4. 找到任意一个 HTTP 包，发出 HTTP 请求的网络浏览器是什么？你可以在分组详细信息中的 Hypertext Transfer Protocol（也即 HTTP ）部分，或者追踪流的页面中寻找“UserAgent”字段。

是 Chrome 浏览器

```
TCP payload (572 bytes)
Hypertext Transfer Protocol
> GET /_visitcount?siteId=68&type=1&columnId=2050 HTTP/1.1\r\n
  Host: ecampus.fudan.edu.cn\r\n
  Connection: keep-alive\r\n
  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36\r\n
  Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8\r\n
  Referer: http://ecampus.fudan.edu.cn/\r\n
  Accept-Encoding: gzip, deflate\r\n
  Accept-Language: en-HK,en;q=0.9,ja-CN;q=0.8,ja;q=0.7,zh-HK;q=0.6,zh-CN;q=0.5,zh;q=0.4,en-GB;q=0.3,en-US;q=0.2\r\n
> Cookie: NSC_Xfc-DpoufouTxjudi-80=ffffffff096ca61a45525d5f4f58455e445a4a423660\r\n
  \r\n
  [Response in frame: 45]
  [Full request URI: http://ecampus.fudan.edu.cn/_visitcount?siteId=68&type=1&columnId=2050]
```

5. 找到任意一个 TCP 包，源端口号和目的端口号各自是什么？你可以发现，分组详细信息中的 Transmission Control Protocol（也即 TCP）部分列出了 Src Port 和 Dst Port。

   源端口号：28397

   目的端口号：443

```
> Internet Protocol Version 4, Src: 192.168.31.146, Dst: 202.120.224.82
v Transmission Control Protocol, Src Port: 28397, Dst Port: 443, Seq: 1826, Ack: 4931, Len: 0
    Source Port: 28397
    Destination Port: 443
    [Stream index: 3]
    [Stream Packet Number: 13]
  > [Conversation completeness: Complete, WITH_DATA (63)]
    [TCP Segment Len: 0]
    Sequence Number: 1826    (relative sequence number)
    Sequence Number (raw): 535120628
    [Next Sequence Number: 1826    (relative sequence number)]
    Acknowledgment Number: 4931    (relative ack number)
```

6. 找到一个由多个 TCP 报文段组合而成的 HTTP 响应分组，这个分组是由多少个 TCP 报文段组成的？在分组详细信息中，会有“[x Reassembled TCP Segments]”提示。

   在详细信息的 TCP 下，有 5 个 TCP 报文段

```
    Urgent Pointer: 0
  v [Timestamps]
      [Time since first frame in this TCP stream: 0.018349000 seconds]
      [Time since previous frame in this TCP stream: 0.004475000 seconds]
  v [SEQ/ACK analysis]
      [iRTT: 0.002830000 seconds]
      [Bytes in flight: 740]
      [Bytes sent since last PSH flag: 740]
    TCP payload (740 bytes)
    TCP segment data (740 bytes)
> [5 Reassembled TCP Segments (9517 bytes): #26(477), #27(1100), #29(2880), #32(4320), #39(740)]
v Hypertext Transfer Protocol
  v HTTP/1.1 200 OK\r\n
      Response Version: HTTP/1.1
      Status Code: 200
      [Status Code Description: OK]
      Response Phrase: OK
    Server: nginx\r\n
```

7. 找到一个带有明文图片的分组，通过“显示分组字节”在 wireshark 中显示图片，并在浏览器中找到对应图片。

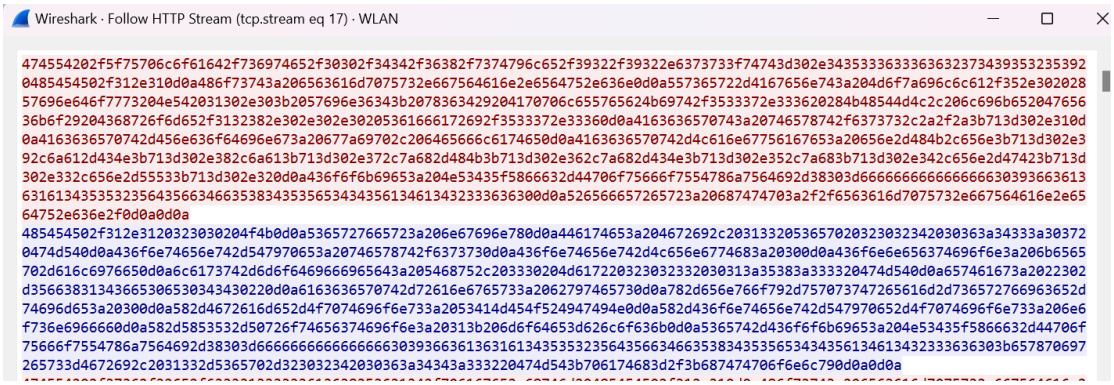
   点击图片后可以抓到 jpeg 格式的请求和响应

```
140 13.797179   192.168.31.146   202.120.224.82   HTTP   686 GET /_upload/article/images/7a/72/59c3d3494bef8b9aeeea06d50021/52bd077
142 13.797362   192.168.31.146   202.120.224.82   HTTP   633 GET /_visitcount?siteId=68&type=3&articleId=685614 HTTP/1.1
143 13.797461   192.168.31.146   36.141.40.58     HTTP   252 GET / HTTP/1.1
146 13.804834   202.120.224.82   192.168.31.146   HTTP   542 HTTP/1.1 200
162 13.820557   202.120.224.82   192.168.31.146   HTTP   810 HTTP/1.1 200 OK  (JPEG JFIF image)
```

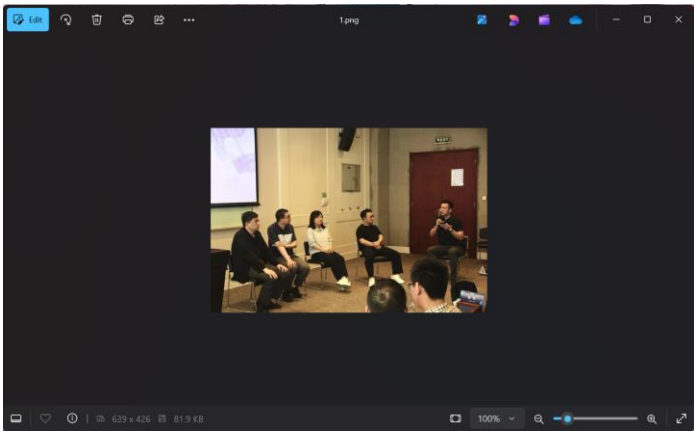
   核查信息是一张完整的图片：

```
[Request in frame: 140]
[Time since request: 0.023378000 seconds]
[Request URI: /_upload/article/images/7a/72/59c3d3494bef8b9aeeea06d50021/52bd0779-79fe-4ad6-94bc-616ec231bcc2.png]
[Full request URI: http://ecampus.fudan.edu.cn/_upload/article/images/7a/72/59c3d3494bef8b9aeeea06d50021/52bd0779-79fe-4ad6-94bc-616ec231bcc2.png]
File Data: 83907 bytes
```

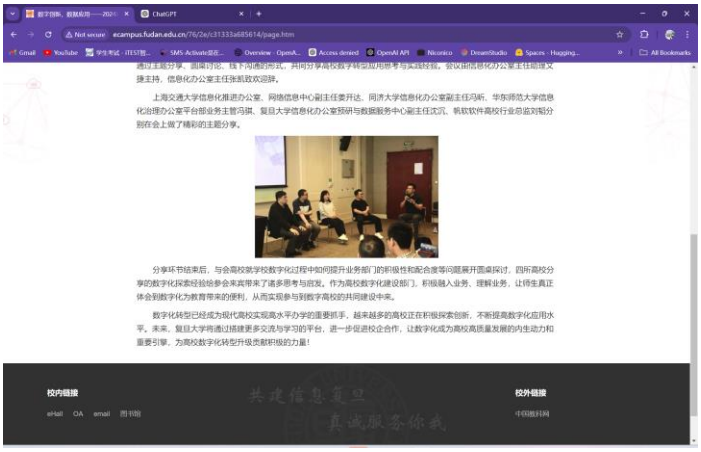
   我以 follow http stream 的方法得到 raw 数据：



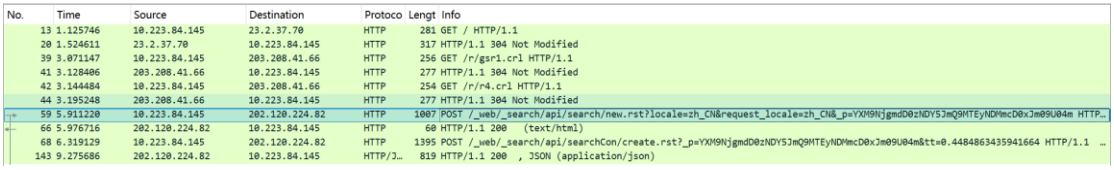
保存后即可打开：



在网站中也有这张图片：



8. （确保你已经为以上问题保存了必要的截图后）重新开启分组捕获，在复旦信息办网站右上角的导航栏搜索任意内容，在捕获到的分组里寻找你输入的内容，观察 HTTP 如何通过 POST 方法发送数据。



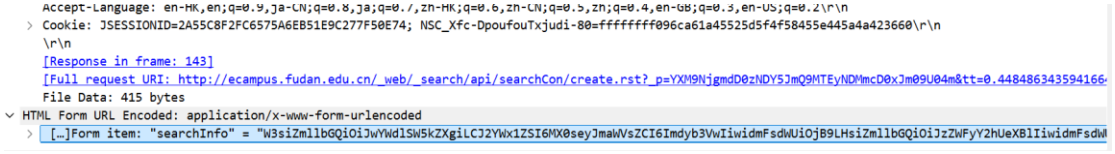
有两个 HTTP 的 POST 方法显示：

在分组详细信息中，可以在 keyword 那里找到自己输入的内容：

根据搜索资料，请求的内容通常会在 application/x-www-form-urlencoded 格式中发送，可以看到上图确实：HTML Form URL Encoded: application/x-www-form-urlencoded。

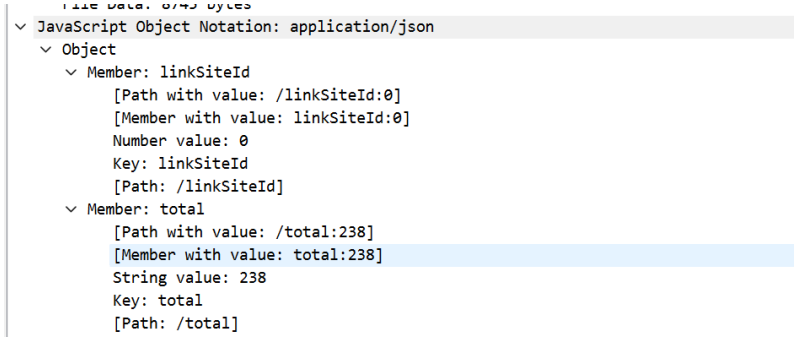
内容类型——Content-Type: application/x-www-form-urlencoded\r\n

在第二个 POST 中：SearchInfo 是 json 字符串



同时，有 encoded 字样，应该是加密过后。

在响应的详细信息中，可以看到传回来多少检索结果：238



检查后发现确实：

以下问题的现象可能会受浏览器缓存影响，如果你第一次访问指定网页时操作失误，请关闭该网页、清除浏览器缓存后再重新操作。重新开启分组捕获，访问 http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html ，输入用户名 wireshark-students，密码 network

This page is password protected! If you're seeing this, you've downloaded the page correctly
Congratulations!

抓到的信息如下：

213	3.994088	10.223.84.145	128.119.245.12	HTTP	620	GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/1.1
660	10.435934	10.223.84.145	183.47.121.90	HTTP	791	POST /mmtls/000020f7 HTTP/1.1 (application/octet-stream)
663	10.484810	183.47.121.90	10.223.84.145	HTTP	401	HTTP/1.1 200 OK (application/octet-stream)
1238	20.196127	10.223.84.145	128.119.245.12	HTTP	620	GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/1.1
1279	20.583149	128.119.245.12	10.223.84.145	HTTP	771	HTTP/1.1 401 Unauthorized (text/html)
1462	24.023004	10.223.84.145	183.47.121.90	HTTP	790	POST /mmtls/00002125 HTTP/1.1 (application/octet-stream)
1465	24.042149	10.223.84.145	183.47.121.90	HTTP	808	POST /mmtls/00002125 HTTP/1.1 (application/octet-stream)
1481	24.255498	183.47.121.90	10.223.84.145	HTTP	1394	[TCP Previous segment not captured] Continuation
1483	24.258389	183.47.121.90	10.223.84.145	HTTP	1394	Continuation
1485	24.258389	183.47.121.90	10.223.84.145	HTTP	532	[TCP Previous segment not captured] Continuation
2277	39.810600	10.223.84.145	128.119.245.12	HTTP	705	GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/1.1
2286	40.127816	128.119.245.12	10.223.84.145	HTTP	544	HTTP/1.1 200 OK (text/html)

9. 在抓取的分组中找到输入的用户名和密码（提示：传输使用了 base64 编码）。将地址中的 http 改为 https（https://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTPwireshark-file5.html），还能否通过捕获分组获得密码？

在分组详细信息里面可以直接找到 credential 的信息：

```
    [Bytes sent since last PSH flag: 651]
  TCP payload (651 bytes)
▾ Hypertext Transfer Protocol
  ▾ GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/1.1\r\n
      Request Method: GET
      Request URI: /wireshark-labs/protected_pages/HTTP-wireshark-file5.html
      Request Version: HTTP/1.1
    Host: gaia.cs.umass.edu\r\n
    Connection: keep-alive\r\n
    Cache-Control: max-age=0\r\n
  ▾ Authorization: Basic d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=\r\n
      Credentials: wireshark-students:network
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.0.0 Safari/537.36\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-HK,en;q=0.9,ja-CN;q=0.8,ja;q=0.7,zh-HK;q=0.6,zh-CN;q=0.5,zh;q=0.4,en-GB;q=0.3,en-US;q=0.2\r\n
    \r\n
    [Response in frame: 2286]
    [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html]

0000  77 69 72 65 73 68 61 72  6b 2d 73 74 75 64 65 6e   wireshar k-studen
0010  74 73 3a 6e 65 74 77 6f  72 6b                    ts:netwo rk
```

完全没有加密，wireshark-students : network 账号密码都直接显示了

在使用 https 登录后：

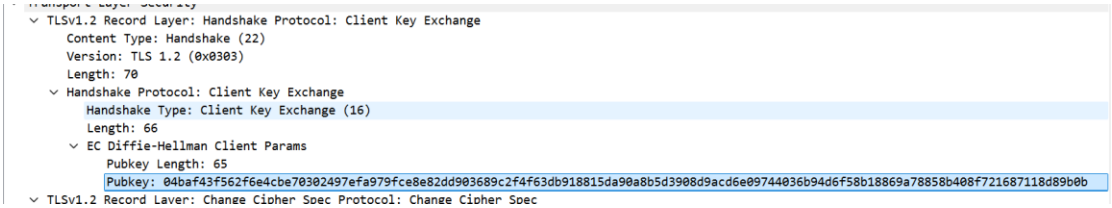
This page is password protected! If you're seeing this, you've downloaded the page correctly
Congratulations!

没有 http 的 item。首先通过查询资料了解道 https 通常使用 ssl 或 tls 加密信息，因此应该无法直接获取账号密码，首先先找一下加密信息——

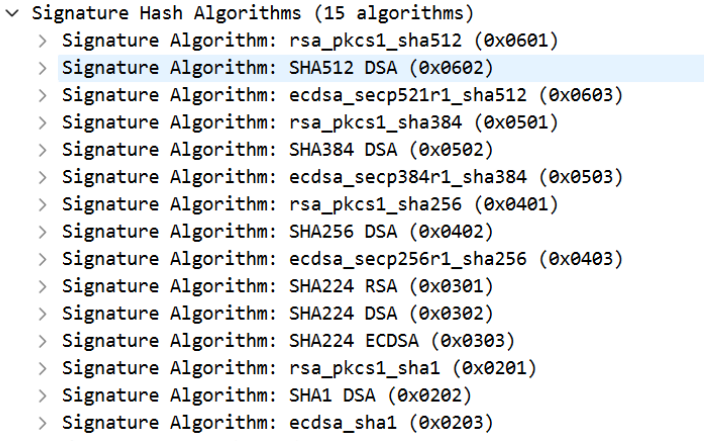
ssl || tls

No.	Time	Source	Destination	Protoco	Lengt	Info
734	18.750645	128.119.245.12	10.223.84.145	TLSv1.2	191	Server Hello, Change Cipher Spec, Encrypted Handshake Message
735	18.751084	10.223.84.145	128.119.245.12	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
787	19.095362	128.119.245.12	10.223.84.145	TLSv1.2	191	Server Hello, Change Cipher Spec, Encrypted Handshake Message
788	19.095362	128.119.245.12	10.223.84.145	TLSv1.2	602	Application Data, Application Data
790	19.095952	10.223.84.145	128.119.245.12	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
793	19.110151	10.223.84.145	204.79.197.203	TLSv1.2	5448	Application Data
794	19.110256	10.223.84.145	204.79.197.203	TLSv1.2	93	Application Data
821	19.300289	204.79.197.203	10.223.84.145	TLSv1.2	93	Application Data
829	19.453423	204.79.197.203	10.223.84.145	TLSv1.2	333	Application Data
911	20.098555	10.223.84.145	20.50.201.204	TLSv1.2	138	Application Data
912	20.098623	10.223.84.145	20.50.201.204	TLSv1.2	93	Application Data
913	20.098646	10.223.84.145	20.50.201.204	TLSv1.2	1337	Application Data
974	20.361776	20.50.201.204	10.223.84.145	TLSv1.2	93	Application Data
981	20.476825	240c:c701:2:805:913…	2600:1417:8000::b81…	TLSv1.3	2146	Client Hello (SNI=img-s-msn-com.akamaized.net)
988	20.626626	2600:1417:8000::b81…	240c:c701:2:805:913…	TLSv1.3	338	Server Hello, Change Cipher Spec, Application Data, Application Data
989	20.627171	240c:c701:2:805:913…	2600:1417:8000::b81…	TLSv1.3	154	Change Cipher Spec, Application Data
995	20.681153	20.50.201.204	10.223.84.145	TLSv1.2	148	Application Data
996	20.681938	10.223.84.145	20.50.201.204	TLSv1.2	89	Application Data
997	20.685570	10.223.84.145	20.50.201.204	TLSv1.2	139	Application Data
998	20.685638	10.223.84.145	20.50.201.204	TLSv1.2	865	Application Data

发现能获得加密的公钥：（在 client key exchange 中，在握手过程之后，即会话密钥协商阶段）



也可以看到加密算法：



在 application data 中，可以看到客户端和服务器之间传输的加密后的数据：例如：