

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。

2791	32.289066	240c:c701:2:805::1	ff02::1:ff4e:2a4	ICMPv6	86 Neighbor Solicitation for 240c:c701:2:805:c15a:3a2e:6e4e:2a4 from 10:c1:72:83:c8:1b
2792	32.370853	240c:c701:2:805:4de...	2404:6800:4012::200a	TCP	86 30225 → 443 [SYN] Seq=0 Win=64800 Len=0 MSS=1440 WS=256 SACK_PERM
2793	32.498371	240c:c701:2:805::1	ff02::1:ffe0:4019	ICMPv6	86 Neighbor Solicitation for 240c:c701:2:805:98d6:b3ff:b3e0:4019 from 10:c1:72:83:c8:1b
2794	32.498371	240c:c701:2:805::1	ff02::1:ff55:4d32	ICMPv6	86 Neighbor Solicitation for 240c:c701:2:805:1d90:c27d:555:4d32 from 10:c1:72:83:c8:1b
2795	32.498371	240c:c701:2:805::1	ff02::1:ff4d:3ef2	ICMPv6	86 Neighbor Solicitation for 240c:c701:2:805:3481:3441:f4d:3ef2 from 10:c1:72:83:c8:1b
2796	32.498371	240c:c701:2:805::1	ff02::1:ff5e:ead6	ICMPv6	86 Neighbor Solicitation for 240c:c701:2:805:240a:ef2c:b85e:ead6 from 10:c1:72:83:c8:1b
2797	32.498371	240c:c701:2:805::1	ff02::1:ff6d:1eb0	ICMPv6	86 Neighbor Solicitation for 240c:c701:2:805:498c:6603:66d6:1eb0 from 10:c1:72:83:c8:1b
2798	32.498371	240c:c701:2:805::1	ff02::1:ffed:8cb3	ICMPv6	86 Neighbor Solicitation for 240c:c701:2:805:ac0d:d381:75ed:8cb3 from 10:c1:72:83:c8:1b
2799	32.498371	240c:c701:2:805::1	ff02::1:ff43:8ee9	ICMPv6	86 Neighbor Solicitation for 240c:c701:2:805:853c:6ccc:d443:8ee9 from 10:c1:72:83:c8:1b
2800	32.498371	240c:c701:2:805::1	ff02::1:ffa7:923b	ICMPv6	86 Neighbor Solicitation for 240c:c701:2:805:83f:388a:4fa7:923b from 10:c1:72:83:c8:1b
2801	32.498371	240c:c701:2:805::1	ff02::1:ffb7:65a3	ICMPv6	86 Neighbor Solicitation for 240c:c701:2:805:708a:9534:cb7:65a3 from 10:c1:72:83:c8:1b
2802	32.498371	240c:c701:2:805::1	ff02::1:ff32:6877	ICMPv6	86 Neighbor Solicitation for 240c:c701:2:805:55ae:32d6:da32:6877 from 10:c1:72:83:c8:1b
2803	32.498371	240c:c701:2:805::1	ff02::1:ff4e:2f03	ICMPv6	86 Neighbor Solicitation for 240c:c701:2:805:6873:327a:114e:2f03 from 10:c1:72:83:c8:1b
2804	32.498371	240c:c701:2:805::1	ff02::1:ffa2:bfc8	ICMPv6	86 Neighbor Solicitation for 240c:c701:2:805:efcd:d9d7:f2a:bfc8 from 10:c1:72:83:c8:1b
2805	32.498371	240c:c701:2:805::1	ff02::1:ffb9:c955	ICMPv6	86 Neighbor Solicitation for 240c:c701:2:805:415d:2459:a0b9:c955 from 10:c1:72:83:c8:1b
2806	32.498371	240c:c701:2:805::1	ff02::1:ff6c:7c31	ICMPv6	86 Neighbor Solicitation for 240c:c701:2:805:95e2:c7b6:b58c:7c31 from 10:c1:72:83:c8:1b
2807	32.498371	240c:c701:2:805::1	ff02::1:ffae:16c6	ICMPv6	86 Neighbor Solicitation for 240c:c701:2:805:74de:e524:46ae:16c6 from 10:c1:72:83:c8:1b
2808	32.498371	240c:c701:2:805::1	ff02::1:ffff:fe90	ICMPv6	86 Neighbor Solicitation for 240c:c701:2:805:bdd0:7951:bfff:fe90 from 10:c1:72:83:c8:1b
2809	32.601378	240c:c701:2:805:4de...	2404:6800:4012::200a	TCP	86 30226 → 443 [SYN] Seq=0 Win=64800 Len=0 MSS=1440 WS=256 SACK_PERM
2810	32.696362	10.223.84.145	142.251.33.74	TCP	66 30227 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM

2643	30.386481	Intel_15:ff:31	HuaweiTechno_83:c8:...	ARP	42 Who has 10.223.0.1? Tell 10.223.84.145
2644	30.391438	HuaweiTechno_83:c8:...	Intel_15:ff:31	ARP	56 10.223.0.1 is at 10:c1:72:83:c8:1b

2232	25.121422	240c:c701:2:805::1	ff02::1:ffff:ca4	ICMPv6	86 Neighbor Solicitation for 240c:c701:2:805:3997:c75:feff:ca4 from 10:c1:72:83:c8:1b
2233	25.257369	240c:c701:2:805:4de...	2404:6800:4012::12	TCP	86 [TCP Retransmission] 30219 → 443 [SYN] Seq=0 Win=64800 Len=0 MSS=1440 WS=256 SACK_PERM
2234	25.303945	10.223.84.145	239.255.255.250	SSDP	217 M-SEARCH * HTTP/1.1
2235	25.308562	10.223.84.145	142.251.211.238	TCP	66 [TCP Retransmission] 30221 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
2236	25.426895	10.223.84.145	142.250.77.10	TCP	66 [TCP Retransmission] 30224 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
2237	25.504817	202.89.233.101	10.223.84.145	TCP	60 443 → 29971 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

3	0.000000	240c:c701:2:805::1	ff02::1:ff34:2054	ICMPv6	86 Neighbor Solicitation for 240c:c701:2:805:59d7:6386:3934:2054 from 10:c1:72:83:c8:1b
2	0.000000	240c:c701:2:805::1	ff02::1:ff63:41cc	ICMPv6	86 Neighbor Solicitation for 240c:c701:2:805:6d50:56ed:b63:41cc from 10:c1:72:83:c8:1b
1	0.000000	240c:c701:2:805::1	ff02::1:ff3a:eabb	ICMPv6	86 Neighbor Solicitation for 240c:c701:2:805:57d0:7f7a:b63a:eabb from 10:c1:72:83:c8:1b
1083	10.050136	2001:da8:8001:2::82	240c:c701:2:805:4de...	HTTP	434 HTTP/1.1 200 OK
1082	10.048158	2001:da8:8001:2::82	240c:c701:2:805:4de...	HTTP	434 HTTP/1.1 200 OK
1081	10.041847	240c:c701:2:805:4de...	2001:da8:8001:2::82	HTTP	538 GET /_upload/site/00/44/68/style/92/92.css?tt=0.3182082217615716 HTTP/1.1
1080	10.039585	240c:c701:2:805:4de...	2001:da8:8001:2::82	HTTP	531 GET /_upload/site/1/style/71/71.css?tt=0.9336579541754719 HTTP/1.1
1063	9.945444	2001:da8:8001:2::82	240c:c701:2:805:4de...	HTTP	425 HTTP/1.1 200
1008	9.891875	240c:c701:2:805:4de...	2001:da8:8001:2::82	HTTP	567 GET /_visitcount?siteId=68&type=1&columnId=2050 HTTP/1.1
1002	9.790559	2001:da8:8001:2::82	240c:c701:2:805:4de...	HTTP	1342 HTTP/1.1 200 OK (text/html)
993	9.772995	240c:c701:2:805:4de...	2001:da8:8001:2::82	HTTP	613 GET / HTTP/1.1
858	8.579896	2001:da8:8001:2::82	240c:c701:2:805:4de...	HTTP	434 HTTP/1.1 200 OK
857	8.578870	2001:da8:8001:2::82	240c:c701:2:805:4de...	HTTP	434 HTTP/1.1 200 OK
855	8.572540	240c:c701:2:805:4de...	2001:da8:8001:2::82	HTTP	539 GET /_upload/site/00/44/68/style/92/92.css?tt=0.15425446278877364 HTTP/1.1
854	8.572198	240c:c701:2:805:4de...	2001:da8:8001:2::82	HTTP	532 GET /_upload/site/1/style/71/71.css?tt=0.03185864096727764 HTTP/1.1
852	8.504813	2001:da8:8001:2::82	240c:c701:2:805:4de...	HTTP	425 HTTP/1.1 200
843	8.484750	240c:c701:2:805:4de...	2001:da8:8001:2::82	HTTP	567 GET /_visitcount?siteId=68&type=1&columnId=2050 HTTP/1.1
837	8.437919	2001:da8:8001:2::82	240c:c701:2:805:4de...	HTTP	1342 HTTP/1.1 200 OK (text/html)
830	8.437880	240c:c701:2:805:4de...	2001:da8:8001:2::82	HTTP	613 GET / HTTP/1.1

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

820	8.427889	240c:c701:2:805:4de...	2001:da8:8001:2::82	HTTP	613 GET / HTTP/1.1
837	8.437919	2001:da8:8001:2::82	240c:c701:2:805:4de...	HTTP	1342 HTTP/1.1 200 OK (text/html)
843	8.484750	240c:c701:2:805:4de...	2001:da8:8001:2::82	HTTP	567 GET /_visitcount?siteId=68&type=1&columnId=2050 HTTP/1.1
852	8.504813	2001:da8:8001:2::82	240c:c701:2:805:4de...	HTTP	425 HTTP/1.1 200
854	8.572198	240c:c701:2:805:4de...	2001:da8:8001:2::82	HTTP	532 GET /_upload/site/1/style/71/71.css?tt=0.03185864096727764 HTTP/1.1
855	8.572540	240c:c701:2:805:4de...	2001:da8:8001:2::82	HTTP	539 GET /_upload/site/00/44/68/style/92/92.css?tt=0.15425446278877364 HTTP/1.1
857	8.578870	2001:da8:8001:2::82	240c:c701:2:805:4de...	HTTP	434 HTTP/1.1 200 OK
858	8.579896	2001:da8:8001:2::82	240c:c701:2:805:4de...	HTTP	434 HTTP/1.1 200 OK
993	9.772995	240c:c701:2:805:4de...	2001:da8:8001:2::82	HTTP	613 GET / HTTP/1.1
1002	9.790559	2001:da8:8001:2::82	240c:c701:2:805:4de...	HTTP	1342 HTTP/1.1 200 OK (text/html)
1008	9.891875	240c:c701:2:805:4de...	2001:da8:8001:2::82	HTTP	567 GET /_visitcount?siteId=68&type=1&columnId=2050 HTTP/1.1
1063	9.945444	2001:da8:8001:2::82	240c:c701:2:805:4de...	HTTP	425 HTTP/1.1 200
1080	10.039585	240c:c701:2:805:4de...	2001:da8:8001:2::82	HTTP	531 GET /_upload/site/1/style/71/71.css?tt=0.9336579541754719 HTTP/1.1
1081	10.041847	240c:c701:2:805:4de...	2001:da8:8001:2::82	HTTP	538 GET /_upload/site/00/44/68/style/92/92.css?tt=0.3182082217615716 HTTP/1.1
1082	10.048158	2001:da8:8001:2::82	240c:c701:2:805:4de...	HTTP	434 HTTP/1.1 200 OK
1083	10.050136	2001:da8:8001:2::82	240c:c701:2:805:4de...	HTTP	434 HTTP/1.1 200 OK

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：

```

✓ Hypertext Transfer Protocol
  > GET /_upload/site/00/44/68/style/92/92.css?tt=0.3182082217615716 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/128.0.0.0 Safari/537.36\r\n
    Accept: text/css,*/*;q=0.1\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
\r\n
[Response in frame: 1083]
[Full request URI: http://ecampus.fudan.edu.cn/_upload/site/00/44/68/style/92/92.css?tt=0.3182082217615716]

```

其次查看时间节点：11:35:57.405727000

```

✓ Frame 1081: 538 bytes on wire (4304 bits), 538 bytes captured (4304 bits) on interface \Device\NPF_{3ACA0BC3-D7EC-4
  Section number: 1
  > Interface id: 0 (\Device\NPF_{3ACA0BC3-D7EC-4E35-8D43-F1637074A865})
    Encapsulation type: Ethernet (1)
    Arrival Time: Sep 13, 2024 11:35:57.405727000 中国标准时间
    UTC Arrival Time: Sep 13, 2024 03:35:57.405727000 UTC
    Epoch Arrival Time: 1726198557.405727000
    [Time shift for this packet: 0.000000000 seconds]
    [Time delta from previous captured frame: 0.002262000 seconds]
    [Time delta from previous displayed frame: 0.002262000 seconds]
    [Time since reference or first frame: 10.041847000 seconds]
    Frame Number: 1081
    Frame Length: 538 bytes (4304 bits)
    Capture Length: 538 bytes (4304 bits)
    [Frame is marked: False]
    [Frame is ignored: False]

```

再看第一个 OK 信号：时间节点：11:35:57.154439000

```

✓ Frame 1002: 1342 bytes on wire (10736 bits), 1342 bytes captured (10736 bits) on interface \Device\NPF_{3ACA0BC3-D7EC-4E35-8D43-F1637074A865},
  Section number: 1
  > Interface id: 0 (\Device\NPF_{3ACA0BC3-D7EC-4E35-8D43-F1637074A865})
    Encapsulation type: Ethernet (1)
    Arrival Time: Sep 13, 2024 11:35:57.154439000 中国标准时间
    UTC Arrival Time: Sep 13, 2024 03:35:57.154439000 UTC
    Epoch Arrival Time: 1726198557.154439000
    [Time shift for this packet: 0.000000000 seconds]
    [Time delta from previous captured frame: 0.000000000 seconds]
    [Time delta from previous displayed frame: 0.017564000 seconds]
    [Time since reference or first frame: 9.790559000 seconds]
    Frame Number: 1002
    Frame Length: 1342 bytes (10736 bits)
    Capture Length: 1342 bytes (10736 bits)
    [Frame is marked: False]
    [Frame is ignored: False]

```

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？

Time	Source	Destination	Protocol	Length	Info
10 0.004847	192.168.31.146	202.120.224.82	HTTP	593	GET / HTTP/1.1
39 0.019956	202.120.224.82	192.168.31.146	HTTP	794	HTTP/1.1 200 OK (text/html)
44 0.031236	192.168.31.146	202.120.224.82	HTTP	626	GET /_visitcount?siteId=68&type=1&columnId=2050 HTTP/1.1
45 0.044088	202.120.224.82	192.168.31.146	HTTP	542	HTTP/1.1 200
46 0.060374	192.168.31.146	202.120.224.82	HTTP	591	GET /_upload/site/1/style/71/71.css?tt=0.06016655217499833 HTTP/1.1
47 0.060535	192.168.31.146	202.120.224.82	HTTP	597	GET /_upload/site/00/44/68/style/92/92.css?tt=0.8793090876558189 HTTP/1.1
49 0.069216	202.120.224.82	192.168.31.146	HTTP	551	HTTP/1.1 200 OK
50 0.069216	202.120.224.82	192.168.31.146	HTTP	551	HTTP/1.1 200 OK

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
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
[Timestamps]
[Time since first frame in this TCP stream: 0.018349000 seconds]
[Time since previous frame in this TCP stream: 0.004475000 seconds]
[SEQ/ACK analysis]
[RTT: 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)]
Hypertext Transfer Protocol
> 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。

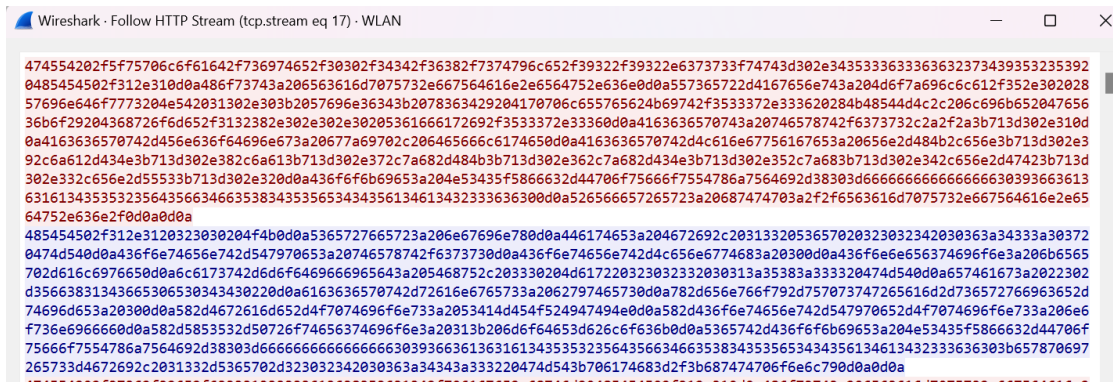
点击图片后可以抓到 jpeg 格式的请求和响应

140	13.797179	192.168.31.146	202.120.224.82	HTTP	686	GET /_upload/article/images/7a/72/59c3d3494bef8b9aeaea06d50021/52bd0779-79fe-4ad6-94bc-616ec231bcc2.png
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)

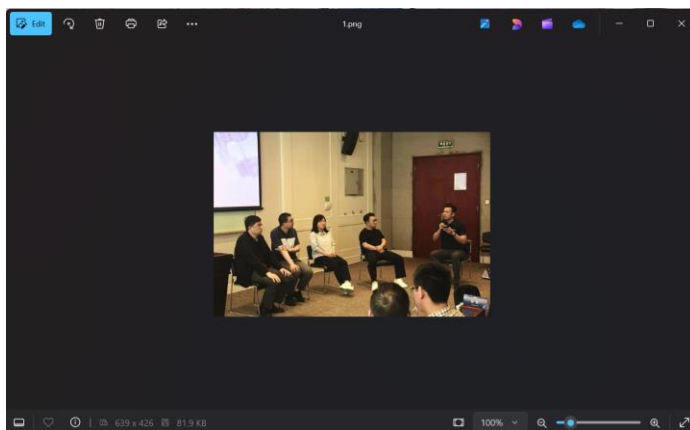
核查信息是一张完整的图片：

```
[Request in frame: 140]
[Time since request: 0.023378000 seconds]
[Request URI: /_upload/article/images/7a/72/59c3d3494bef8b9aeaea06d50021/52bd0779-79fe-4ad6-94bc-616ec231bcc2.png]
[Full request URI: http://ecampus.fudan.edu.cn/_upload/article/images/7a/72/59c3d3494bef8b9aeaea06d50021/52bd0779-79fe-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。

No.	Time	Source	Destination	Protocol	Length	Info
13	1.125746	10.223.84.145	23.2.37.70	HTTP	281	GET / HTTP/1.1
20	1.524611	23.2.37.70	10.223.84.145	HTTP	317	HTTP/1.1 304 Not Modified
39	3.071347	10.223.84.145	203.208.41.66	HTTP	256	GET /r/gsr1.crl HTTP/1.1
41	3.128486	203.208.41.66	10.223.84.145	HTTP	277	HTTP/1.1 304 Not Modified
42	3.144484	10.223.84.145	203.208.41.66	HTTP	254	GET /r/r4.crl HTTP/1.1
44	3.195248	203.208.41.66	10.223.84.145	HTTP	277	HTTP/1.1 304 Not Modified
59	5.911220	10.223.84.145	202.120.224.82	HTTP	1007	POST /_web_search/api/search/new.rst?locale=zh_CN&request_locale=zh_CN&_pa=YX09NjgmdD0zNDY5MzQ5HTeYNDmcD0x2m09U04m HTTP/1.1
66	5.976716	202.120.224.82	10.223.84.145	HTTP	60	HTTP/1.1 200 (text/html)
68	6.319129	10.223.84.145	202.120.224.82	HTTP	1395	POST /_web_search/api/searchCon/create.rst?_pa=YX09NjgmdD0zNDY5MzQ5HTeYNDmcD0x2m09U04m&tt=0.4484863435941664 HTTP/1.1
143	9.275686	202.120.224.82	10.223.84.145	HTTP/1.1	819	HTTP/1.1 200 , 350N (application/json)

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`

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 tag: 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 d2lyZXNoYXJrLXN0dWR1bnRzM0M5ldHdvcms=\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到 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	Protocol	Length 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_	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::2600:1417:8000::b81_	TLSv1.3	338 Server Hello, Change Cipher Spec, Application Data
989	20.627171	240c:c701:2:805:913::2600:1417:8000::b81_	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）

