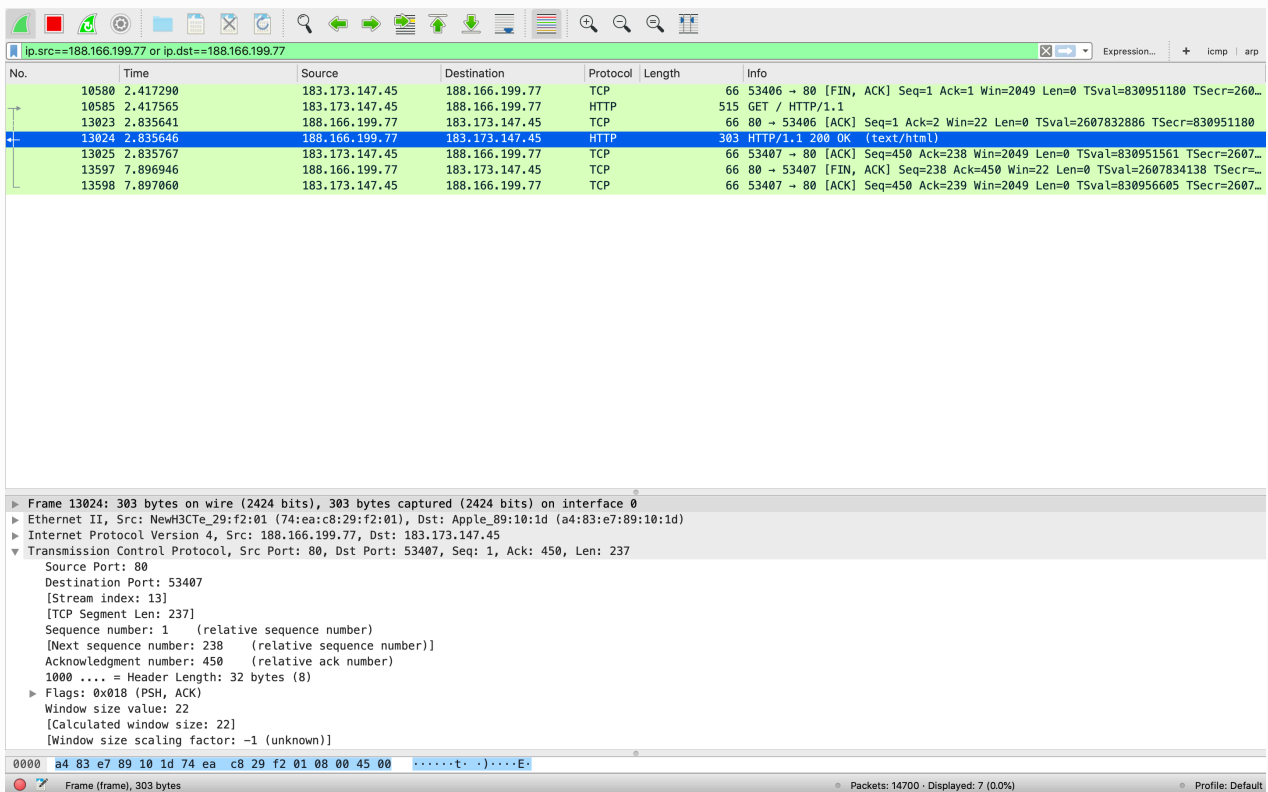


计65 赵鋈峰 2016011373

计65 王展鹏 2016011356

实验准备

- 当我们访问目标网站的**首页**时，wireshark抓包如图：



，同时网页端网站能正常访问。

- 而当我们访问**目标网站**时，wireshark抓包如图：

No.	Time	Source	Destination	Protocol	Length	Info
24742	192.897251	183.173.147.45	188.166.199.77	TCP	78	52841 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=828517979 TSecr=
24743	192.897298	183.173.147.45	188.166.199.77	TCP	78	52842 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=828517979 TSecr=
24783	193.147460	183.173.147.45	188.166.199.77	TCP	78	52843 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=828518228 TSecr=
24786	193.302131	188.166.199.77	183.173.147.45	TCP	74	80 → 52842 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1300 SACK_PERM=1 TSv
24787	193.302138	183.173.147.45	188.166.199.77	TCP	66	52842 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0 TSval=828518380 TSecr=260722
24788	193.302342	183.173.147.45	188.166.199.77	HTTP	527	GET /tibetalk.php HTTP/1.1
24789	193.313319	188.166.199.77	183.173.147.45	TCP	74	80 → 52841 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1300 SACK_PERM=1 TSv
24790	193.313377	183.173.147.45	188.166.199.77	TCP	66	52841 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0 TSval=828518391 TSecr=260722
24791	193.315413	188.166.199.77	183.173.147.45	TCP	60	80 → 52842 [RST, ACK] Seq=1 Ack=462 Win=7616512 Len=0
24792	193.315416	188.166.199.77	183.173.147.45	TCP	60	80 → 52842 [RST, ACK] Seq=1 Ack=462 Win=7616512 Len=0
24793	193.315416	188.166.199.77	183.173.147.45	TCP	60	80 → 52842 [RST, ACK] Seq=1 Ack=462 Win=7616512 Len=0
24794	193.315417	188.166.199.77	183.173.147.45	TCP	60	80 → 52842 [RST] Seq=1 Win=339968 Len=0
24801	193.593877	188.166.199.77	183.173.147.45	TCP	74	80 → 52843 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1300 SACK_PERM=1 TSv
24802	193.594016	183.173.147.45	188.166.199.77	TCP	66	52843 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0 TSval=828518670 TSecr=260722
24808	193.718022	188.166.199.77	183.173.147.45	TCP	66	80 → 52842 [ACK] Seq=1 Ack=462 Win=45056 Len=0 TSval=2607221285 TSecr=8285
24809	193.718066	183.173.147.45	188.166.199.77	TCP	54	52842 → 80 [RST] Seq=462 Win=0 Len=0
27933	224.257352	188.166.199.77	183.173.147.45	TCP	74	[TCP Retransmission] 80 → 52841 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS
27934	224.257440	183.173.147.45	188.166.199.77	TCP	66	[TCP Dup ACK 24790#1] 52841 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0 TSval=
27939	224.537817	188.166.199.77	183.173.147.45	TCP	74	[TCP Retransmission] 80 → 52843 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS
27940	224.537923	183.173.147.45	188.166.199.77	TCP	66	[TCP Dup ACK 24802#1] 52843 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0 TSval=
32601	269.388543	188.166.199.77	183.173.147.45	TCP	54	[TCP Keep-Alive] 52841 → 80 [ACK] Seq=0 Ack=1 Win=131328 Len=0
32609	269.668113	183.173.147.45	188.166.199.77	TCP	54	[TCP Keep-Alive] 52843 → 80 [ACK] Seq=0 Ack=1 Win=131328 Len=0
32613	269.797952	188.166.199.77	183.173.147.45	TCP	66	[TCP Window Update] 80 → 52841 [ACK] Seq=1 Ack=1 Win=45056 Len=0 TSval=266
32706	270.090251	188.166.199.77	183.173.147.45	TCP	66	[TCP Window Update] 80 → 52843 [ACK] Seq=1 Ack=1 Win=45056 Len=0 TSval=266
32798	270.922809	183.173.147.45	188.166.199.77	TCP	66	52841 → 80 [FIN, ACK] Seq=1 Ack=1 Win=131328 Len=0 TSval=828595764 TSecr=2
32799	270.922834	183.173.147.45	188.166.199.77	TCP	66	52843 → 80 [FIN, ACK] Seq=1 Ack=1 Win=131328 Len=0 TSval=828595764 TSecr=2

Frame 24742: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface 0
 Ethernet II, Src: Apple_89:10:1d (a4:83:e7:89:10:1d), Dst: NewH3CFe_29:f2:01 (74:ea:c8:29:f2:01)
 Internet Protocol Version 4, Src: 183.173.147.45, Dst: 188.166.199.77
 Transmission Control Protocol, Src Port: 52841, Dst Port: 80, Seq: 0, Len: 0
 Source Port: 52841
 Destination Port: 80
 [Stream index: 62]
 [TCP Segment Len: 0]
 Sequence number: 0 (relative sequence number)
 [Next sequence number: 0 (relative sequence number)]
 Acknowledgment number: 0
 1011 ... = Header Length: 44 bytes (11)
 Flags: 0x002 (SYN)
 Window size value: 65535
 [Calculated window size: 65535]
 Checksum: 0x3439 [unverified]

0000 74 ea c8 29 f2 01 a4 83 e7 89 10 1d 08 00 45 00 t... ..E.
 Frame (frame), 78 bytes
 Packets: 258156 · Displayed: 30 (0.0%)
 Profile: Default

，同时网页端显示连接被重置。

- 所以我们可以得知：tibetalk为敏感词，被GFW识别之后向客户端发送了TCP RST包来重置连接，客户端接收之后又向目标服务器发送了RST请求。因此，第一个思路就是让客户端忽略所有的TCP RST请求，强行建立连接。

忽略TCP RST包

- 在MacOS上使用PF作为防火墙，配置规则如下

```
block in on en0 proto tcp all flags R/R
block out on en0 proto tcp all flags R/R
```

- （该规则也避免了在用户层实现TCP握手的时候内核主动发出RST来终止连接的这一行为）
- 再次访问得到

No.	Time	Source	Destination	Protocol	Length	Info
3466	28.295238	183.173.147.45	188.166.199.77	TCP	66	56241 → 80 [FIN, ACK] Seq=1 Ack=1 Win=2052 Len=0 TSval=894532158 TSecr=264936
3486	28.758715	188.166.199.77	183.173.147.45	TCP	60	80 → 56241 [RST] Seq=1 Win=0 Len=0
3491	28.799982	183.173.147.45	188.166.199.77	TCP	78	56295 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=894532660 TSecr=264936
3492	28.800132	183.173.147.45	188.166.199.77	TCP	78	56296 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=894532660 TSecr=264936
3517	29.050263	183.173.147.45	188.166.199.77	TCP	78	56297 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=894532810 TSecr=264936
3624	29.215907	188.166.199.77	183.173.147.45	TCP	74	80 → 56295 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1300 SACK_PERM=1 TSval=894533074 TSecr=264936
3625	29.215181	188.166.199.77	183.173.147.45	TCP	74	80 → 56296 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1300 SACK_PERM=1 TSval=894533074 TSecr=264936
3626	29.215188	183.173.147.45	188.166.199.77	TCP	66	56295 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0 TSval=894533074 TSecr=264936
3627	29.215188	183.173.147.45	188.166.199.77	TCP	66	56296 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0 TSval=894533074 TSecr=264936
3628	29.215473	183.173.147.45	188.166.199.77	HTTP	527	GET /tibetalk.php HTTP/1.1
3629	29.232948	188.166.199.77	183.173.147.45	TCP	60	80 → 56295 [RST, ACK] Seq=1 Ack=462 Win=3309568 Len=0
3630	29.232955	188.166.199.77	183.173.147.45	TCP	60	80 → 56295 [RST, ACK] Seq=1 Ack=462 Win=3309568 Len=0
3631	29.232956	188.166.199.77	183.173.147.45	TCP	60	80 → 56295 [RST, ACK] Seq=1 Ack=462 Win=3309568 Len=0
3632	29.232957	188.166.199.77	183.173.147.45	TCP	60	80 → 56295 [RST] Seq=1 Win=36485120 Len=0
3640	29.479929	183.173.147.45	188.166.199.77	TCP	66	[TCP Retransmission] 56241 → 80 [FIN, ACK] Seq=1 Ack=1 Win=2052 Len=0 TSval=894533353 TSecr=264936
3641	29.495752	188.166.199.77	183.173.147.45	TCP	74	80 → 56297 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1300 SACK_PERM=1 TSval=894533353 TSecr=264936
3642	29.495823	183.173.147.45	188.166.199.77	TCP	66	56297 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0 TSval=894533353 TSecr=264936
3645	29.629304	188.166.199.77	183.173.147.45	TCP	66	80 → 56295 [ACK] Seq=1 Ack=462 Win=45056 Len=0 TSval=2649385893 TSecr=894533353
3677	29.998372	188.166.199.77	183.173.147.45	TCP	60	80 → 56241 [RST] Seq=1 Win=0 Len=0
3902	31.646870	183.173.147.45	188.166.199.77	TCP	66	[TCP Retransmission] 56241 → 80 [FIN, ACK] Seq=1 Ack=1 Win=2052 Len=0 TSval=894533353 TSecr=264936
3961	32.037403	188.166.199.77	183.173.147.45	TCP	60	80 → 56241 [RST] Seq=1 Win=0 Len=0
4266	35.775836	183.173.147.45	188.166.199.77	TCP	66	[TCP Retransmission] 56241 → 80 [FIN, ACK] Seq=1 Ack=1 Win=2052 Len=0 TSval=894533353 TSecr=264936
4345	36.236671	188.166.199.77	183.173.147.45	TCP	60	80 → 56241 [RST] Seq=1 Win=0 Len=0
5397	43.832945	183.173.147.45	188.166.199.77	TCP	66	[TCP Retransmission] 56241 → 80 [FIN, ACK] Seq=1 Ack=1 Win=2052 Len=0 TSval=894533353 TSecr=264936
5522	44.247591	188.166.199.77	183.173.147.45	TCP	60	80 → 56241 [RST] Seq=1 Win=0 Len=0
7503	59.746986	183.173.147.45	188.166.199.77	TCP	66	[TCP Retransmission] 56241 → 80 [FIN, ACK] Seq=1 Ack=1 Win=2052 Len=0 TSval=894533353 TSecr=264936

Frame 3466: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0

Ethernet II, Src: Apple_89:10:1d (a4:83:e7:89:10:1d), Dst: NewH3CTe_29:f2:01 (74:ea:c8:29:f2:01)

Internet Protocol Version 4, Src: 183.173.147.45, Dst: 188.166.199.77

Transmission Control Protocol, Src Port: 56241, Dst Port: 80, Seq: 1, Ack: 1, Len: 0

Source Port: 56241

Destination Port: 80

(Stream index: 17)

[TCP Segment Len: 0]

Sequence number: 1 (relative sequence number)

[Next sequence number: 1 (relative sequence number)]

Acknowledgment number: 1 (relative ack number)

1000 = Header Length: 32 bytes (8)

Flags: 0x011 (FIN, ACK)

Window size value: 2052

[Calculated window size: 2052]

[Window size scaling factor: -1 (unknown)]

0000 74 ea c8 29 f2 01 a4 83 e7 89 10 1d 08 00 45 00 t...).....E

Frame (frame), 66 bytes

Packets: 10013 · Displayed: 35 (0.3%)

Profile: Default

- 可知客户端没有发送RST但服务器仍然向客户端发送了RST报文。这让我们怀疑GFW检测到可疑的访问时会向双方都发送RST，（这一点也在¹中得到了证实）如果要用这种方法绕过GFW，必须两边都忽略，而我们只能修改客户端的行为，因此该思路受阻。

分片

- 首先我们需要能用scapy模拟浏览器发起GET的HTTP请求，当用scapy直接在TCP的payload上写

```
GET / HTTP/1.0\n\n
```

时我们会得到如下结果

No.	Time	Source	Destination	Protocol	Length	Info
1641	40.719793	183.173.147.45	182.61.200.7	TCP	54	36697 → 80 [SYN] Seq=0 Win=8192 Len=0
1890	46.537190	183.173.147.45	182.61.200.7	TCP	54	36204 → 80 [SYN] Seq=0 Win=8192 Len=0
1891	46.549647	182.61.200.7	183.173.147.45	TCP	60	80 → 36204 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0
1892	46.571037	183.173.147.45	182.61.200.7	TCP	71	36204 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=17
1893	46.576883	182.61.200.7	183.173.147.45	TCP	60	80 → 36204 [ACK] Seq=1 Ack=18 Win=29200 Len=0
1894	46.576895	182.61.200.7	183.173.147.45	HTTP	82	HTTP/1.1 400 Bad Request
1895	46.576896	182.61.200.7	183.173.147.45	TCP	60	80 → 36204 [FIN, ACK] Seq=29 Ack=18 Win=29200 Len=0
1905	46.775623	182.61.200.7	183.173.147.45	TCP	82	[TCP Retransmission] 80 → 36204 [FIN, PSH, ACK] Seq=1 Ack=18 Win=29200 Len=0
1908	47.208457	182.61.200.7	183.173.147.45	TCP	82	[TCP Retransmission] 80 → 36204 [FIN, PSH, ACK] Seq=1 Ack=18 Win=29200 Len=0
1944	48.026804	182.61.200.7	183.173.147.45	TCP	82	[TCP Retransmission] 80 → 36204 [FIN, PSH, ACK] Seq=1 Ack=18 Win=29200 Len=0
2004	49.579680	182.61.200.7	183.173.147.45	TCP	82	[TCP Retransmission] 80 → 36204 [FIN, PSH, ACK] Seq=1 Ack=18 Win=29200 Len=0
2101	52.634912	182.61.200.7	183.173.147.45	TCP	60	80 → 36204 [RST] Seq=1 Win=0 Len=0
2930	74.055886	188.166.199.77	183.173.147.45	HTTP	537	HTTP/1.1 400 Bad Request (text/html)
15770	380.796810	183.173.147.45	182.61.200.7	TCP	78	58155 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=901520580 TSecr=0
15771	380.796938	183.173.147.45	182.61.200.7	TCP	78	58156 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=901520580 TSecr=0
15772	380.803254	182.61.200.7	183.173.147.45	TCP	78	443 → 58155 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1300 WS=32 SACK_PERM=1
15773	380.803257	182.61.200.7	183.173.147.45	TCP	78	443 → 58156 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1300 WS=32 SACK_PERM=1
15774	380.803340	183.173.147.45	182.61.200.7	TCP	54	58155 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0
15775	380.803340	183.173.147.45	182.61.200.7	TCP	54	58156 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0
15776	380.803529	183.173.147.45	182.61.200.7	TLSv1...	571	Client Hello
15777	380.803692	183.173.147.45	182.61.200.7	TLSv1...	571	Client Hello
15780	380.811478	182.61.200.7	183.173.147.45	TCP	60	443 → 58156 [ACK] Seq=1 Ack=518 Win=30336 Len=0
15781	380.811481	182.61.200.7	183.173.147.45	TCP	60	443 → 58155 [ACK] Seq=1 Ack=518 Win=30336 Len=0
15782	380.811482	182.61.200.7	183.173.147.45	TLSv1...	122	Server Hello
15783	380.811483	182.61.200.7	183.173.147.45	TLSv1...	122	Server Hello
15784	380.811483	182.61.200.7	183.173.147.45	TCP	1334	443 → 58155 [ACK] Seq=69 Ack=518 Win=30336 Len=1280 [TCP segment of a reassembled data segment]

Frame 2930: 537 bytes on wire (4296 bits), 537 bytes captured (4296 bits) on interface 0

Ethernet II, Src: NewH3CTe_29:f2:01 (74:ea:c8:29:f2:01), Dst: Apple_B9:10:1d (a4:83:e7:89:10:1d)

Internet Protocol Version 4, Src: 188.166.199.77, Dst: 183.173.147.45

Transmission Control Protocol, Src Port: 80, Dst Port: 13559, Seq: 1, Ack: 1, Len: 483

Hypertext Transfer Protocol

HTTP/1.1 400 Bad Request\r\n

[Expert Info (Chat/Sequence): HTTP/1.1 400 Bad Request\r\n]

Response Version: HTTP/1.1

Status Code: 400

[Status Code Description: Bad Request]

Response Phrase: Bad Request

Date: Fri, 08 Nov 2019 05:38:51 GMT\r\n

Server: Apache/2.4.18 (Ubuntu)\r\n

Content-Length: 301\r\n

Connection: close\r\n

Content-Type: text/html; charset=iso-8859-1\r\n

可见浏览器在发送时自动为payload添加了一些其他属性，单单这个payload是不够的，从浏览器对首页发起一次请求并且用wireshark抓包得到TCP的payload为

GET / HTTP/1.1
Host: lab3.jinzihao.me
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.87 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
HTTP/1.1 200 OK
Date: Fri, 08 Nov 2019 05:46:30 GMT
Server: Apache/2.4.18 (Ubuntu)
Access-Control-Allow-Origin: *
Content-Length: 1
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=gb2312
1

Packet 20046: 1 client pkt, 1 server pkt, 1 turn. Click to select.

Entire conversation (660 bytes)

Show and save data as ASCII

Find:

Find Next

Help Filter Out This Stream Print Save as... Back Close

- 得到访问首页应当发送的payload为

```
GET / HTTP/1.1\r\nHost: lab3.jinzihao.me\r\nConnection: keep-
alive\r\nUpgrade-Insecure-Requests: 1\r\nUser-Agent: Mozilla/5.0
(Macintosh; Intel Mac OS X 10_15_0) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/78.0.3904.87 Safari/537.36\r\nAccept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,
image/apng,*/*;q=0.8,application/signed-exchange;v=b3\r\nAccept-
Encoding: gzip, deflate\r\nAccept-Language: zh-
CN,zh;q=0.9\r\n\r\n
```

所以访问目标网址的payload应当为

```
GET /tibetalk.php HTTP/1.1\r\nHost:
lab3.jinzihao.me\r\nConnection: keep-alive\r\nUpgrade-Insecure-
Requests: 1\r\nUser-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X
10_15_0) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/78.0.3904.87 Safari/537.36\r\nAccept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,
image/apng,*/*;q=0.8,application/signed-exchange;v=b3\r\nAccept-
Encoding: gzip, deflate\r\nAccept-Language: zh-
CN,zh;q=0.9\r\n\r\n
```

- 接下来我们需要将payload分片来让每次都不包含会被检测到的敏感词，而连接在一起还与原先的相同。分片方法是在握手后，多次发送ack序号相同，seq序号连续的ack报文。

尝试了多次分片结果后，我得出了一个能够稳定得到flag的分片方法：

```
[ 'GET /ti', 'be', 'ta', 'lk',
  '.php ',
  'HTTP/1.1\r\nHost: lab3.jinzihao.me',
  '\r\nConnection: keep-alive\r\nUpgrade-Insecure-Requests: 1\r\nUser-
Agent: Mozilla/5.0 (Macintosh; ',
  'Intel Mac OS X 10_15_0) AppleWebKit/537.36 (KHTML, ',
  'like Gecko) Chrome/78.0.3904.87 Safari/537.36\r\nAccept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/
apng,*/*;q=0.8,application/signed-exchange;v=b3\r\nAccept-Encoding:
gzip, deflate\r\nAccept-Language: zh-CN,zh;q=0.9\r\n\r\n' ]
```

wireshark抓包显示

ip.src==188.166.199.77 or ip.dst==188.166.199.77

No.	Time	Source	Destination	Protocol	Length	Info
3781	56.338492	183.173.147.45	188.166.199.77	TCP	54	25027 → 80 [SYN] Seq=0 Win=8192 Len=0
4178	61.565852	183.173.147.45	188.166.199.77	TCP	54	65012 → 80 [SYN] Seq=0 Win=8192 Len=0
4200	62.100144	188.166.199.77	183.173.147.45	TCP	60	80 → 65012 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1300
4201	62.109562	183.173.147.45	188.166.199.77	TCP	61	65012 → 80 [PSH, ACK] Seq=1 Ack=1 Win=8192 Len=7 [TCP segment of a reassem...
4252	62.553188	188.166.199.77	183.173.147.45	TCP	60	80 → 65012 [ACK] Seq=1 Ack=8 Win=29200 Len=0
4397	65.128742	183.173.147.45	188.166.199.77	TCP	56	65012 → 80 [PSH, ACK] Seq=8 Ack=1 Win=8192 Len=2 [TCP segment of a reassem...
4447	65.687030	188.166.199.77	183.173.147.45	TCP	60	80 → 65012 [ACK] Seq=1 Ack=10 Win=29200 Len=0
4605	68.148492	183.173.147.45	188.166.199.77	TCP	56	65012 → 80 [PSH, ACK] Seq=10 Ack=1 Win=8192 Len=2 [TCP segment of a reasse...
4831	71.164359	183.173.147.45	188.166.199.77	TCP	56	65012 → 80 [PSH, ACK] Seq=12 Ack=1 Win=8192 Len=2 [TCP segment of a reasse...
4888	71.641366	188.166.199.77	183.173.147.45	TCP	60	80 → 65012 [ACK] Seq=1 Ack=14 Win=29200 Len=0
5061	74.182124	183.173.147.45	188.166.199.77	TCP	59	65012 → 80 [PSH, ACK] Seq=14 Ack=1 Win=8192 Len=5 [TCP segment of a reasse...
5106	74.690568	188.166.199.77	183.173.147.45	TCP	60	80 → 65012 [ACK] Seq=1 Ack=19 Win=29200 Len=0
5267	77.201011	183.173.147.45	188.166.199.77	TCP	86	65012 → 80 [PSH, ACK] Seq=19 Ack=1 Win=8192 Len=32 [TCP segment of a reasse...
5327	77.652538	188.166.199.77	183.173.147.45	TCP	60	80 → 65012 [ACK] Seq=1 Ack=51 Win=29200 Len=0
5464	80.220150	183.173.147.45	188.166.199.77	TCP	146	65012 → 80 [PSH, ACK] Seq=51 Ack=1 Win=8192 Len=92 [TCP segment of a reasse...
5652	83.239899	183.173.147.45	188.166.199.77	TCP	105	65012 → 80 [PSH, ACK] Seq=143 Ack=1 Win=8192 Len=51 [TCP segment of a reas...
5843	86.257642	183.173.147.45	188.166.199.77	HTTP	296	GET /tibetalk.php HTTP/1.1
5886	86.756911	188.166.199.77	183.173.147.45	TCP	60	80 → 65012 [ACK] Seq=1 Ack=436 Win=30016 Len=0
6107	89.755661	188.166.199.77	183.173.147.45	HTTP	305	HTTP/1.1 200 OK (text/plain)

Content-Length: 46\r\n
Keep-Alive: timeout=5, max=100\r\n
Connection: Keep-Alive\r\n
Content-Type: text/plain;charset=gb2312\r\n
\r\n
[HTTP response 1/1]
[Time since request: 3.498019000 seconds]
[Request in frame: 5843]
[Request URI: http://lab3.jinzihao.me/tibetalk.php]
File Data: 46 bytes

Line-based text data: text/plain (1 lines)
flag{96c2474b07e55ad4da792c349e855350de448da0}

0090 65 6e 74 2d 4c 65 6e 67 74 68 3a 20 34 36 0d 0a ent-Leng th: 46..
00a0 4b 65 65 70 2d 41 6c 69 76 65 3a 20 74 69 6d 65 Keep-Ali ve: time
00b0 6f 75 74 3d 35 2c 20 6d 61 78 3d 31 30 30 0d 0a out=5, max=100..
00c0 43 6f 6e 6e 65 63 74 69 6f 6e 3a 20 4b 65 65 70 Connecti on: Keep
00d0 2d 41 6c 69 76 65 0d 0a 43 6f 6e 74 65 6e 74 2d -Alive: Content-
00e0 54 79 70 65 3a 20 74 65 78 74 2f 70 6c 61 69 6e Type: te xt/plain
00f0 3b 63 68 61 72 73 65 74 3d 67 62 32 33 31 32 0d ;charset =gb2312..
0100 0a 0d 0a 66 6c 61 67 7b 39 36 63 32 34 37 34 62 .. flag{ 96c2474b
0110 30 37 65 35 35 61 64 34 64 61 37 39 32 63 33 34 07e55ad4 da792c34
0120 39 65 38 35 35 33 35 30 64 65 34 34 38 64 61 30 9e855350 de448da0
0130 7d }

Text item (text), 46 bytes Packets: 7131 · Displayed: 19 (0.3%) Profile: Default

我们猜测之所以有的分片方法会被GFW gank的原因可能是分片过少、发送时间太快导致IDS能够利用缓存来迅速重组发现敏感词进而发出TCP RST报文，因此把tibetalk被重组起来的时间点拉的越久越不容易被发现。

- 完整源代码如下：

```
from scapy.layers.inet import *
from scapy.sendrecv import send

if __name__ == '__main__':
    ip = IP(dst='188.166.199.77')
    payloads = ['GET /ti', 'be', 'ta', 'lk',
                '.php ',
                'HTTP/1.1\r\nHost: lab3.jinzihao.me',
                '\r\nConnection: keep-alive\r\nUpgrade-Insecure-Requests: 1\r\nUser-Agent: Mozilla/5.0 (Macintosh; ',
                'Intel Mac OS X 10_15_0) AppleWebKit/537.36 (KHTML, ',
                'like Gecko) Chrome/78.0.3904.87',
                'Safari/537.36\r\nAccept: ',
                'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3\r\nAccept-Encoding: ',
                'gzip, deflate\r\nAccept-Language: zh-CN,zh;q=0.9\r\n\r\n']
    port = RandNum(1024, 65535)
    SYN = ip / TCP(sport=port, dport=80, flags='S', seq=42)
    ACK = sr1(SYN)
    ack_num = ACK.seq + 1
    send_length = 0
    for p in payloads:
        fragment = ip / TCP(sport=ACK.dport, dport=80, flags='A',
                             seq=ACK.ack + send_length, ack=ack_num) / p
        send(fragment)
```

```
send_length += len(p)
time.sleep(3)
```

1. <https://www.cl.cam.ac.uk/~rnc1/ignoring.pdf> 