

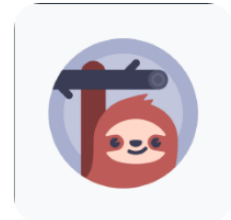
itaymigdal/Poshito

 github.com/itaymigdal/Poshito

itaymigdal

itaymigdal/**Poshito**

Poshito is a Windows C2 over Telegram



 1 Contributor
 0 Issues
 8 Stars
 1 Fork



Repository files navigation

 [README](#)

Poshito is a Windows C2 over Telegram

Really? Another C2? Why???



I'm not addicted to building C2s — I can stop whenever I want!

Just kidding. I've come to terms with the fact that I enjoy building offensive tooling as part of my learning journey. Plus, I like having full control to customize tools my way.

Ok... nonetheless, what's new here?



Nothing too fancy, really. But some of Poshito's features are:

- Obfuscated Go build using [Garble](#). On top of that, UPX with section name obfuscation is performed (same as [Nimbo-C2](#) does).

- C2 over a Telegram channel with password-protected access (no hardcoded chat ID), secured by a hashed password and resistant to tracking tools like [TeleTracker](#) that require both bot token and chat ID.
- Customizable agents by selectively removing features with Go build tags.
- (Quite) smart and (quite) safe inline .NET assembly execution, where assemblies are allocated once and reused, while patching `Environment.Exit()` to prevent irresponsible assemblies from terminating the agent.
- DRM protection: The agent is restricted to execution on a single Windows machine.
- Easy installation via Docker.

How to use



1. Build and run the provided Docker image:

```
docker build -t poshito .  
docker run -it --rm -v ${pwd}:/Poshito -w /Poshito/Poshito poshito
```

(In Linux replace `${pwd}` with `$(pwd)`)

2. For each new agent, generate a new Telegram bot using the [Bot Father](#). Grab the bot token, and bot URL.

3. Build the agent:

```
/Poshito/Poshito # python3 build.py -h
usage: build [-h] [-nx] [-ns] [-ng] [-en <name>] [-st SLEEP_TIME] [-sj <percent (%)>] [-dd] [-dr] [-dc] [-ds] [-da] bot_token password {exe,dll}
```

Poshito-C2 agent builder

positional arguments:

bot_token

Bot token

password

Operator password

{exe,dll}

Payload format

options:

-h, --help

show this help message and exit

-nx, --no-upx

don't UPX

-ns, --no-upx-sec-obf

don't obfuscate UPX section

names

-ng, --no-garble

don't use Garble (use standard

Go compiler)

-en <name>, --export-name <name>

dll export name (default:

DllRegisterServer)

-st <seconds>, --sleep-time <seconds>

time to sleep between callbacks

(default: 5)

-sj <percent (%)>, --sleep-jitter <percent (%)>

sleep time jitter in percent

(default: 0)

-dd, --disable-drm

disable DRM feature

-dr, --disable-dir

disable directory view feature

(/dir)

-dc, --disable-clip

disable clipboard feature

(/clip)

-ds, --disable-screen

disable screenshot feature

(/screen)

-da, --disable-asm

disable assemblies loading

feature (/asm + /iex)

4. After the agent execution, send the password to the Telegram bot.

Commands



/info	Send agent information	
/cmd	Execute a command	< command >
/iex	Execute a Powershell command	< powershell command >
/dir	Show directory content	< directory path >
/down	Download a file	< file path >
/up	Upload a file	< file to upload > < path to save
(optional) >		
/clip	Get clipboard content	
/screen	Get screenshot	
/asm	Execute .NET assembly	< (assembly file assembly hash) +
assembly arguments >		
/die	Kill agent	
/sleep	Change sleep time	< seconds > < jitter % >

/asm will send you the assembly hash on the first execution. use it for further executions of that assembly, see the [example](#).

Credits

