GitHub - itaymigdal/Nimbo-C2: Nimbo-C2 is yet another (simple and lightweight) C2 framework

By itaymigdal

Archived: 2026-04-05 15:40:07 UTC

About

Nimbo-C2 is yet another (simple and lightweight) C2 framework.

Nimbo-C2 agent supports x64 Windows & Linux. It's written in Nim, with some usage of .NET on Windows (by dynamically loading the CLR to the process). Nim is powerful, but interacting with Windows is much easier and robust using Powershell, hence this combination is made. The Linux agent is slimier and capable only of basic commands, including ELF loading using the `memfd` technique.

All server components are written in Python:

- HTTP listener that manages the agents.
- Builder that generates the agent payloads.
- Nimbo-C2 is the interactive C2 component that rule'em all!

My work wouldn't be possible without the previous great work done by others, listed under credits.

Features

- Build EXE, DLL, ELF payloads.
- Encrypted implant configuration and strings using NimProtect.
- Packing payloads using UPX and obfuscate the PE section names (`UPX0` , `UPX1`) to make detection and unpacking harder.
- Encrypted HTTP communication (AES in CBC mode, key hardcoded in the agent and configurable by the `config.jsonc`).
- Auto-completion in the C2 Console for convenient interaction.
- File & Registry commands.
- In-memory Powershell commands execution.
- File download and upload commands.
- Built-in discovery commands.

- Screenshot taking, clipboard stealing, audio recording, and keylogger.
- ETW & AMSI patching using indirect syscalls.
- LSASS and SAM hashes dumping.
- Shellcode injection using indirect syscalls.
- Inline .NET assemblies execution.
- Persistence capabilities.
- UAC bypass methods.
- Token impersonation and getsystem.
- Setting implant process as critical (BSOD on termination).
- (Linux) ELF loading using `memfd` in 2 modes.
- And more !

Installation

Warning: Nimbo-C2 is meant to be run only within the provided Docker container

Easy Way

> Note that installing this way may cause problems or incompatibility in the future as the Docker image now doesn't enforces languages and libraries versions, so consider skipping to the next method.

1. Clone the repository and `cd` in

```
 git clone https://github.com/itaymigdal/Nimbo-C2
cd Nimbo-C2
```

2. Build the docker image

```
 docker build -t nimbo-dependencies .
```

3. `cd` again into the source files and run the docker image interactively, expose port 80 and mount Nimbo-C2 directory to the container (so you can easily access all project files, modify `config.jsonc`, download and upload files from agents, etc.). For Linux replace `${pwd}` with `$(pwd)`.

```
 cd Nimbo-C2
docker run -it --rm -p 80:80 -v ${pwd}:/Nimbo-C2 -w /Nimbo-C2 nimbo-dependencies
```

Easier Way

> Here we're using the already built, tested and stored Docker image - **recommended**.

```
 git clone https://github.com/itaymigdal/Nimbo-C2
cd Nimbo-C2/Nimbo-C2
```

```
docker run -it --rm -p 80:80 -v ${pwd}:/Nimbo-C2 -w /Nimbo-C2 itaymigdal/nimbo-dependencies
```

For Linux replace `${pwd}` with `$(pwd)`.

Usage

First, edit `config.jsonc` for your needs.

Then run with: `python3 Nimbo-C2.py`

Use the `help` command for each screen, and tab completion.

Limitations, Warnings, Notes

- Even though the HTTP communication is encrypted, the 'user-agent' header is in plain text and it carries the real agent id, which some products may flag it suspicious.
- Wrap paths or arguments with spaces with double quoutes.
- CLR works with primary access token so `impersonate` / `getsys` don't affect `iex` / `assembly`.
- `audio`, `lsass` (except the Evil Lsass Twin method) commands temporarily save artifacts to disk before exfiltrate and delete them.
- If you tunnel the Nimbo server (e.g. if you expose it via Pinggy), use TCP, not HTTP.

Contribution

This software may be buggy or unstable in some use cases as it not being fully and constantly tested. Feel free to open issues, PR's, and contact me for any reason at (Gmail | Linkedin)
