Introducing the Sliver Framework written in Golang

By Joe DeMesy, Alumnus Principal

Archived: 2026-04-05 15:57:04 UTC

Cross-platform General Purpose Implant Framework Written in Golang

Senior Security Associate Joe DeMesy and Security Associate Ronan Kervella are the researchers behind the creation and maintenance of Sliver. They introduced Sliver in June at SummerCon 2019.

⚠️ **Warning:** Sliver is currently in **beta**, you've been warned :) and please consider contributing.

How Sliver Works

Sliver is designed to be an open source alternative to Cobalt Strike. Sliver supports asymmetrically encrypted C2 over DNS, HTTP, HTTPS, and Mutual TLS using per-binary X.509 certificates signed by a per-instance certificate authority and supports multiplayer mode for collaboration.

We will explore how to design stable, performant, and secure C2 channels as well as other design challenges when creating implants as they present.

Sliver is a general purpose cross-platform implant framework that supports C2 over Mutual-TLS, HTTP(S), and DNS. Implants are dynamically compiled with unique X.509 certificates signed by a per-instance certificate authority generated when you first run the binary.

The server, client, and implant all support MacOS, Windows, and Linux (and possibly every Golang compiler target but we've not tested them all).

Sliver's Features

- Dynamic code generation
- Compile-time obfuscation
- Local and remote process injection
- Anti-anti-anti-forensics
- Secure C2 over mTLS, HTTP(S), and DNS
- Windows process migration
- Windows user token manipulation
- Multiplayer-mode
- Procedurally generated C2 over HTTP *(work in progress)*
- Let's Encrypt integration
- In-memory .NET assembly execution

- DNS Canary Blue Team Detection

Getting Started

Download the latest release and see the Sliver wiki for a quick tutorial on basic setup and usage. To get the very latest and greatest compile from source.

Compile from Source

See the wiki.

Source Code

- `assets/` - Static assets that are embedded into the server binary, generated by `go-assets.sh`
- `client/` - Client code, the majority of this code is also used by the server
- `protobuf/` - - Protobuf code
- `server/` -Server-side code
- `sliver/` - Implant code, rendered by the server at runtime
- `util/` - Utility functions that may be shared by the server and client

License - GPLv3

Sliver is licensed under GPLv3, some subcomponents have separate licenses. See their respective subdirectories in this project for details.

Excerpt from GitHub:
Go to https://github.com/BishopFox/sliver for the complete tooling.

.github	Updated security policy, image location, and .github/
assets	Re-adding the hosting DLL
client	Add GPLv3
protobuf	Fix execute-assembly missing DLL and refactoring
server	Remove buildid from shared libs / shellcode
sliver	Place the comment where it should be
util	Added GPLv3 license
vendor	Updated vendor/
.dockerignore	Fixed unit tests