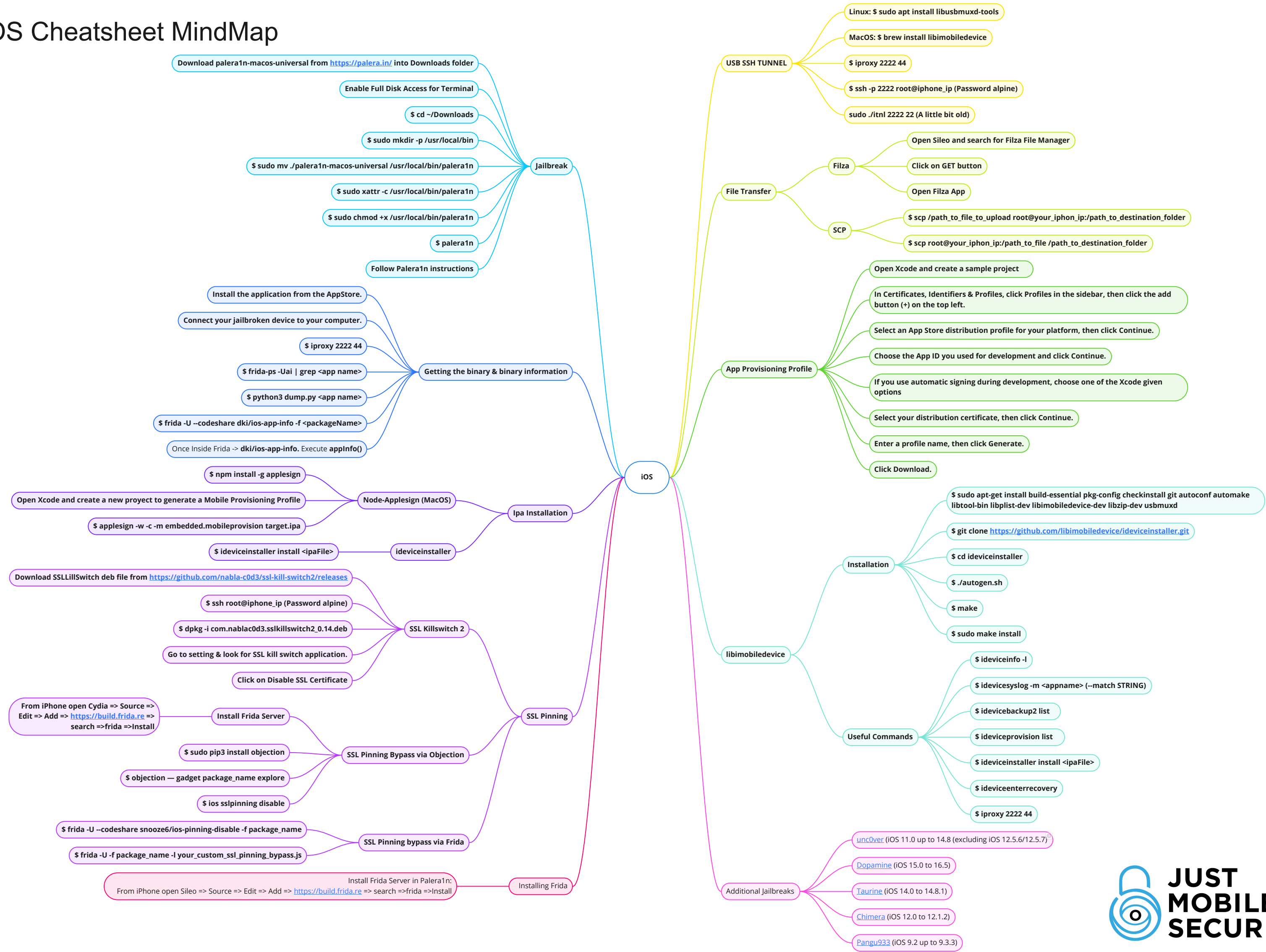


iOS Cheatsheet MindMap



Jailbreak

IMPORTANT! never setup the passcode!, if the phone had ever setted up passcode reset it from factory.

- 1. **Download palera1n-macos-universal** from <https://palera.in/> into Downloads folder
- 2. **Enable Full Disk Access for Terminal** (this only has to be done once) a. macOS Ventura and above: System Settings → Privacy & Security → Full Disk Access b. If Terminal does not show up in the list, click the plus icon and select it from Applications → Utilities. (this only has to be done once):

- 3. **\$ cd ~/Downloads**
- 4. **\$ sudo mkdir -p /usr/local/bin**
- 5. **\$ sudo mv ./palera1n-macos-universal /usr/local/bin/palera1n**
- 6. **\$ sudo xattr -c /usr/local/bin/palera1n**
- 7. **\$ sudo chmod +x /usr/local/bin/palera1n**
- 8. **\$ palera1n**
- 9. Follow Palera1n instructions

Additional Jailbreaks

Installing Frida

- [unc0ver](#) (iOS 11.0 up to 14.8 (excluding iOS 12.5.6/12.5.7)
- [Dopamine](#) (iOS 15.0 to 16.5)
- [Taurine](#) (iOS 14.0 to 14.8.1)
- [Chimera](#) (iOS 12.0 to 12.1.2)
- [Pangu933](#) (iOS 9.2 up to 9.3.3)

Install Frida Server in Palera1n:
From iPhone open Sileo => Source => Edit => Add => <https://build.frida.re> => search =>frida =>Install

USB SSH TUNNEL

Getting the binary & binary information

Installing iproxy:
Linux: **\$ sudo apt install libusbmuxd-tools**
MacOS: **\$ brew install libimobiledevice**

- Connecting via SSH:
- 1. **\$ iproxy 2222 44**
 - Starting iproxy binding port 44 (Palera1n default SSH port) to 2222
 - 2. **\$ ssh -p 2222 root@iphone_ip** (Password alpine)
 - Connecting via ssh to device

- 1. Install the application from the AppStore.
- 2. Connect your jailbroken device to your computer.
- 3. **\$ iproxy 2222 44**
- Run iProxy from terminal
- 4. **\$ frida-ps -Uai | grep <app name>**
- Obtain app Package name
- 5. **\$ python3 dump.py <app name>**
- Pull a decrypted IPA from a jailbroken device using frida-ios-dump
- 6. **\$ frida -U --codeshare dki/ios-app-info -f <packageName>**
- Get additional information
- 7. Once Inside Frida -> **dki/ios-app-info**. Execute **appInfo()**

File Transfer

Ipa Installation

- Installing Filza:** (also useful to install .ipa files)
- 1. Open **Sileo** and a new source "<http://apt.thebigboss.org/>"
 - 2. Search for **Filza File Manager**
 - 3. Click on **GET** button
 - 4. **Open Filza App**
- Using scp:
- 1. **\$ scp /file_path_to_upload root@your_iphon_ip:/path_to_destination_folder**
 - Push file to device
 - 2. **\$ scp root@your_iphon_ip:/path_to_file /path_to_destination_folder**
 - Pull file from device

- ideviceinstaller:
- \$ ideviceinstaller install <ipaFile>**
- Node-Applesign (MacOS):
- 1. **\$ npm install -g applesign**
 - 2. Open Xcode and create a new project to **generate a Mobile Provisioning Profile**
 - 3. **\$ applesign -w -c -m embedded.mobileprovision target.ipa**

SSL Pinning

SSL Killswitch 2:

- 1. On the device **download SSLKillSwitch deb file** from <https://github.com/nabla-c0d3/ssl-kill-switch2/releases>
- 2. **\$ ssh root@iphone_ip** (Password alpine)
Connect via ssh to device.
- 3. **\$ dpkg -i com.nabla0d3.sslkillswitch2_0.14.deb**
Installing the Killswitch 2 package.
- 4. Go to **setting** & look for **SSL kill switch** application.
- 5. **Click on Disable SSL Certificate** and SSL pinning of all the applications will be bypassed.

SSL Pinning Bypass via Objection:

- 1. **Install Frida Server:** From iPhone open Cydia => Source => Edit => Add => <https://build.frida.re> => search => frida => Install
- 2. **\$ sudo pip3 install objection**
Installing objection in MacBook
- 3. **\$ objection — gadget package_name explore**
Running Objection
- 4. **\$ ios sslpinning disable**
Running bypass SSL pinning command

Useful Sileo Repositories

SSL Pinning bypass via Frida:

```
$ frida -U --codeshare snooze6/ios-pinning-disable -f package_name  
or  
$ frida -U -f package_name -l your_custom_ssl_pinning_bypass.js
```

- <https://opa334.github.io>
- <https://ios.jjolano.me>
- <https://build.frida.re>
- <https://apt.thebigboss.org>
- <https://repo.co.kr>

App Provisioning Profile

- 1. Open **Xcode** and create a sample project
- 2. In **Certificates, Identifiers & Profiles**, click **Profiles** in the sidebar, then click the **add button (+)** on the top left.
- 3. Under **Distribution**, select an App Store distribution profile for your platform, then **click Continue**.
- 4. Choose the **App ID** you used for development (the App ID that matches your bundle ID) from the App ID pop-up menu, then **click Continue**.
- 5. If you use **automatic signing** during development, choose one of the **Xcode given options**
- 6. **Select your distribution** certificate, then **click Continue**.
- 7. Enter a profile name, then **click Generate**.
- 8. **Click Download**.

libimobiledevice

Installation:

- 1. **\$ sudo apt-get install build-essential pkg-config checkinstall git autoconf automake libtool-bin libplist-dev libimobiledevice-dev libzip-dev usbmuxd**
- 2. **\$ git clone <https://github.com/libimobiledevice/ideviceinstaller.git>**
- 3. **\$ cd ideviceinstaller**
- 4. **\$./autogen.sh**
- 5. **\$ make**
- 6. **\$ sudo make install**

Useful Commands:

```
$ deviceinfo -l  
Show information about a connected device  
$ deviceprovision list  
Manage provisioning profiles on a device  
$ deviceinstaller install <ipaFile>  
Installing ipa files into the device  
$ devicecenterrecovery  
Make a device enter recovery mode  
$ iproxy 2222 44  
Starting iproxy binding port 44 (Palera1n default SSH port) to 2222  
$ devicesyslog -m <appname> (--match STRING)  
Relay syslog of a connected device  
$ devicebackup2 list  
Create or restore backups for devices running iOS 4 or later
```