FIELD NOTE / IOS

Effectively Obtaining .ipa Files for iOS Pentesting from the App Store

Petruknisme · 24 Dec, 2022 · 4 min to read



Image from HackerOne — iOS Hacking — Application Basics https://www.youtube.com/watch?v=VQTQ0VaIXF0 (https://www.youtube.com/watch?v=VQTQ0VaIXF0)

During pentesting or bug bounty hunting, it is sometimes necessary to analyze an iOS app that is available in the App Store. However, in order to do this, it is necessary to have access to the app's `.ipa` file. If the `.ipa` file is not readily available, it can be challenging to perform a thorough analysis of the app's security. In these situations, it may be necessary to find ways to obtain the `.ipa` file or to explore other methods for analyzing the app's security.

Obtaining an `.ipa` file for an iOS app that is available on the App Store can be a challenge, especially if you don't have access to the file directly. Fortunately, the `ipatool` tool offers a solution. With `ipatool`, you can search for and download a copy of the app package from the App Store, giving you the ability to perform a thorough analysis of the app's security. Whether you are an iOS pentester, a bug bounty hunter, or a developer looking to ensure the security of your app, `ipatool` can be a valuable tool in your toolkit.

Licensed under CC BY-SA 4.0

Prerequisites

Before you can use `ipatool`, you'll need to have the following software installed on your computer:

- Apple ID set up to use the App Store.
- Homebrew (https://brew.sh/) (optional, but recommended for installing `ipatool`)

If you are using Linux or Windows, please follow installation instructions from https://github.com/majd/ipatool (https://github.com/majd/ipatool)

Package Manager (macOS)

You can install `ipatool` using Homebrew (https://brew.sh/).

```
❯ brew tap majd/repo
Running `brew update --auto-update`...
==> Auto-updated Homebrew!
==> Updated Homebrew from 3.6.14 (11cdffb4f) to 3.6.16 (895d32212).
The 3.6.16 changelog can be found at:
  https://github.com/Homebrew/brew/releases/tag/3.6.16
==> Tapping majd/repo
Cloning into '/usr/local/Homebrew/Library/Taps/majd/homebrew-repo'...
remote: Enumerating objects: 79, done.
remote: Counting objects: 100% (79/79), done.
remote: Compressing objects: 100% (38/38), done.
remote: Total 79 (delta 28), reused 73 (delta 23), pack-reused 0
Receiving objects: 100% (79/79), 10.58 KiB | 1.32 MiB/s, done.
Resolving deltas: 100% (28/28), done.
Tapped 1 cask and 1 formula (16 files, 21.0KB).
```

```
❯ brew install ipatool
==> Downloading https://github.com/majd/ipatool/releases/download/v2.0.0/i
==> Downloading from https://objects.githubusercontent.com/github-producti
######################################################################## 1
==> Installing Cask ipatool
```

Licensed under CC BY-SA 4.0

```
==> Linking Binary 'ipatool-2.0.0-macos-amd64' to '/usr/local/bin/ipatool'
   ipatool was successfully installed!
```

Here is a list of useful commands and options for using `ipatool` to download and search app package from app store.

IPATool

```
❯ ipatool --help
A cli tool for interacting with Apple's ipa files

Usage:
  ipatool [command]

Available Commands:
  auth        Authenticate with the App Store
  completion  Generate the autocompletion script for the specified shell
  download    Download (encrypted) iOS app packages from the App Store
  help        Help about any command
  purchase    Obtain a license for the app from the App Store
  search      Search for iOS apps available on the App Store

Flags:
      --format format     sets output format for command; can be 'text', '
  -h, --help              help for ipatool
      --non-interactive   run in non-interactive session
      --verbose           enables verbose logs
  -v, --version           version for ipatool

Use "ipatool [command] --help" for more information about a command.
```

Before utilizing the `ipatool` tool to download an `.ipa` file from the App Store, you will need to set up an Apple ID and authenticate your device. This process will allow you to access the App Store and download the app package for analysis and testing.

```
❯ ipatool auth login -h
Login to the App Store

```

Licensed under CC BY-SA 4.0

```
Usage:
  ipatool auth login [flags]

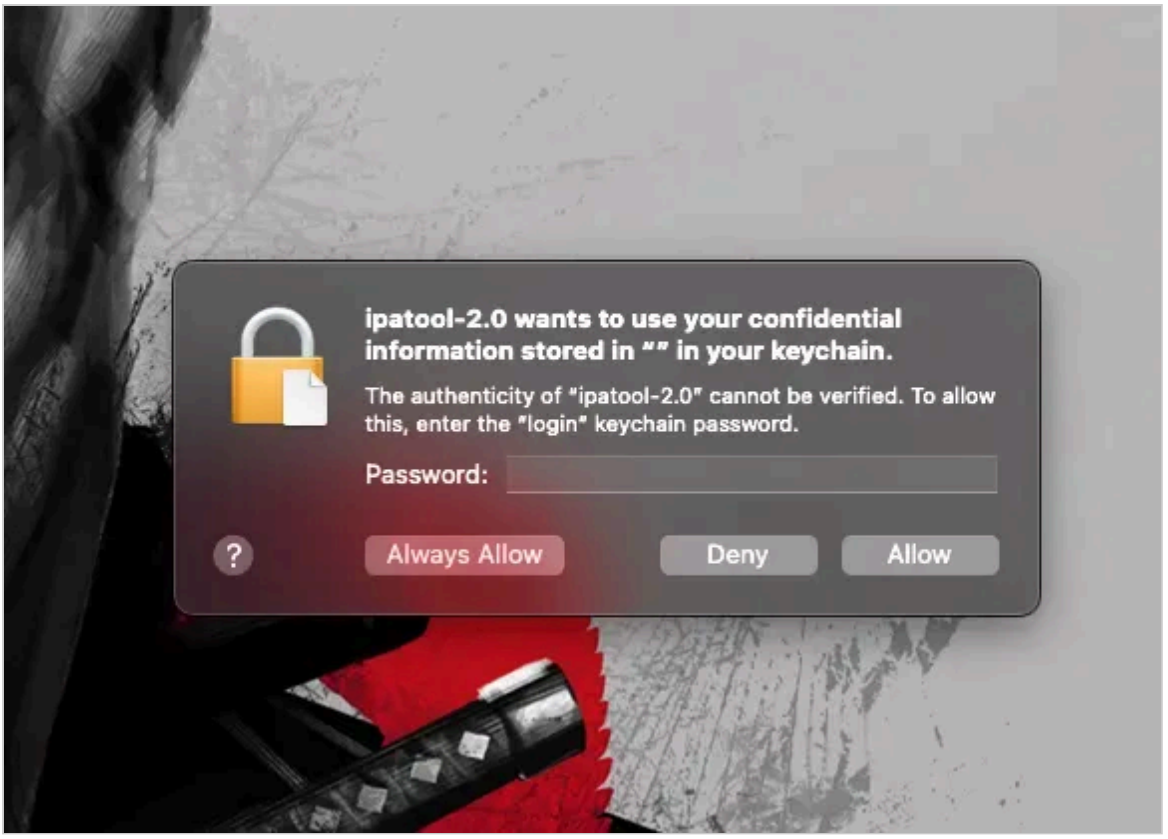
Flags:
      --auth-code string   2FA code for the Apple ID
  -e, --email string       email address for the Apple ID (required)
  -h, --help               help for login
  -p, --password string    password for the Apple ID (required

Global Flags:
      --format format     sets output format for command; can be 'text', '
      --non-interactive   run in non-interactive session
      --verbose           enables verbose logs
```

In order to use **`ipatool`** to download **`.ipa`** files from the App Store, you will need to authenticate the tool using your Apple ID. Once you have entered your Apple ID, you will receive a one-time passcode (OTP) from Apple, which you will need to enter in order to complete the authentication process and gain access to the App Store through **`ipatool`**.

```
❯ ipatool auth login -e example@email.com
2:28PM INF enter password:
2:28PM INF enter 2FA code:
OTP_NUMBER_HERE
2:29PM INF email=example@email.com name="PERSON NAME" success=true
```

Licensed under CC BY-SA 4.0



Keychain prompt when running ipatool

Before we can use `ipatool` to download an `.ipa` file from the App Store, we need to search for the app package that we want to analyze. This can be done by using the `ipatool` search command, which allows us to search for apps by name or by the developer's name. In this example, we will download youtube ipa.

```
❯ ipatool search --limit 1 Youtube
2:33PM INF apps=[{"bundleID":"com.google.ios.youtube","id":544007664,"name
```

Once you have located the desired app in the App Store using `ipatool`, you can use the app's bundle ID as an identifier to download the `.ipa` file. The bundle ID is a unique identifier that is assigned to every iOS app, and it can be used to locate and download the app package from the App Store.

To download the `.ipa` file for an iOS app from the App Store using `ipatool`, you can use the `--bundle-identifier` parameter followed by the app's bundle ID. You can also use the `--output` flag to specify the location and name of the file that you want to save the `.ipa` file as. For example:

Licensed under CC BY-SA 4.0

```
❯ ipatool --bundle-identifier com.example.app --output /path/to/save/file.
```

This command will download the `.ipa` file for the app with the bundle ID `com.example.app` and save it to the specified location as `file.ipa`.

```
❯ ipatool download --bundle-identifier com.google.ios.youtube --output you
downloading   28% |████████████████████████████████| (50/172 MB, 14 MB/s
2:36PM INF output=youtube.ipa success=true
❯ ls -la
total 187444
drwxr-xr-x    3 user staff        96 Dec 24 14:36 ./
drwx------+ 490 user staff     15680 Dec 24 14:18 ../
-rw-r--r--    1 user staff 179954362 Dec 24 14:36 youtube.ipa
```

As demonstrated above, using `ipatool` we were able to successfully download the `.ipa` file for the YouTube app from the App Store. With this file in hand, we can now proceed with our iOS security testing and analysis. I hope that this guide has been helpful, happy hacking~

Note: In creating this article, I utilized ChatGPT (https://chat.openai.com/), a large language model trained by OpenAI, to assist with writing and editing. ChatGPT was instrumental in ensuring that the content was clear, concise, and easy to understand. Its advanced capabilities allowed me to quickly and efficiently produce high-quality content that is both informative and engaging.

Originally published on Medium (https://medium.com/@petruknisme/effectively-obtaining-ipa-files-for-ios-pentesting-from-the-app-store-2c1551b230).

iOS # Security # Tutorial

Licensed under CC BY-SA 4.0