iPhone TCP Connection Through USB - Documentation

Project
iPhone Reversing Toolkit

Author
Nicolas Economou (neconomou)

Description

The development of these tools was based on ideas from iPHUC, a tool to establish a ftp-like connection via USB to the iPhone.

The `iphone_tunnel` patches in runtime library `iTunesMobileDevice.dll` and hooks functions `send` and `recv` used by the library redirecting the connection to the port indicated as a parameter.

Application 'iphone_tunnel.exe' accepts multiple connections and several instances (using different ports on the PC) can be runned simultaneously.

Setup

The usage is very simple, only has 2 parameters, the port where you want to connect inside the iPhone (i.e. the port on iPhone's `localhost`) and the port where the tool will stay listening for connections on the PC. For example on your PC run:

```
iphone_tunnel.exe 22 22
```

In this example, if you connect using a `ssh` client to the machine serving the tunnel and inside iPhone your are running the OpenSSH server, your will open a ssh session. A step by step recipe for the `ssh` session follows:

- Install the OpenSSH server on your iPhone.
- Install iTunes on your PC.
- Download the binary or compile it from scratch using Borland C++ compiler.
- Move `iphone_tunnel.exe` to where `iTunesMobileDevice.dll` is located (`C:Program FilesCommon FilesAppleMobile Device Supportbin`) or move the DLL to the `iphone_tunnel.exe` folder.
- Plug the USB to your PC and your iPhone.
- Run on your PC `iphone_tunnel.exe` with parameters `22` and `22`.
- Start a ssh from a third computer to your PC.

Detailed Description

Using ideas from `iphuc` and debugging a little, we will describe how we can connect via USB to the `sshd` running inside iPhone on the `localhost`. For further details take a look at the source code released.

1. The first step is installing iTunes, because this application provides a service to speak directly with the iPhone via USB and a DLL that talks directly to that service.

2. Once you have iTunes installed on your Windows, a service called `Apple Mobile Device` must be running. It's important that this service is up to establish the connection with the iPhone.

3. At the same time, in folder `C:Program FilesCommon FilesAppleMobile Device Supportbin` should be library `iTunesMobileDevice.dll`, necessary to establish the connection against service `Apple Mobile Device`. This DLL could be omitted and we could establish directly the connection with the service, that listens on address `127.0.0.1` port `27015`, if we understand better the information flowing forward and backward from the DLL and the service.

4. Sniffing the connections between `iphuc.exe` and the service, we notice that in the third connection we can see inside the first packet (sent by the DLL to the service) the port (in Big Endian) that it connects to. The port travelling in the packet is the port to which service `afcd`, running inside iPhone, is listening. This iPhone service is started in the last steps of the second connection. All this work is done by function `AMDeviceStartService` included in the DLL.

5. In the third parameter of function `AMDeviceStartService`, is returned a handle to the connection established against the service, which we can use directly with `send` and `recv`.

6. Well, once we knew all this, the next step was to to build an application in `C` to make the same calls to the DLL as `iPhuc.exe` including the possibility of modifying the port. We also include the functionality to redirect the call to the function `send` inside the DLL's IAT so it points to a function on the our program. When the program detects the third connection it patches the packet to make it connect to our preferred port.

7. The final result is an application that listens for connections on your PC's LAN and tunnels them to the iPhone localhost through the USB cable (AWESOME!).