

# John Roark

telephone: 415-425-8489

email: [jroark@cs.usfca.edu](mailto:jroark@cs.usfca.edu)

[CD74 2F65](#)

---

## EDUCATION

### B.S., Computer Science

May 2002

[University of San Francisco \(USF\)](#), San Francisco, CA

### Programming Languages:

C/C++/Objective-C

## PROFESSIONAL EXPERIENCE

Principal Software Engineer

June 2015 - Present

[Bluebox Security](#)

Integrated F5 VPN sdk for application level VPN policy (iOS/Android)

Wrote ELF interposition library for Android

Designed/Wrote Application lifecycle proxy to delay application startup (iOS)

Added MITM detection policy (iOS/Android)

Modified SSL/TLS pinning policy to use SPKI pins (iOS/Android)

Refactored policy parser to use shared code and JSON schema (iOS/Android)

Researched and created POC hooking mechanism to overcome new iOS9 dyld shared cache binding

Principal Engineer/Team Lead

Sept. 2011 - May 2015

[Mocana Corporation](#)

Participated in the development of Mocana [Mobile App Protection](#)

Designed and implemented "Data at Rest" (DAR) injection policy for MAP (Android & iOS)

Implemented "Copy and Paste Protection" injection policy for MAP (Android)

Developed function interposing library for (Android & iOS)

Primarily responsible for the development of the [Mocana Secure Enterprise Browser \(Compass\)](#)

Senior Software Engineer

Jun. 2003 - Sept. 2011

[PGP Corporation](#) (Now part of [Symantec](#))

Led the design, development, and test planning for several major features of PGP Whole Disk Encryption

Updated and maintained tablet PC support in the WDE pre-boot environment

Participated in the development and maintenance of an EFI preboot environment

Wrote interrupt 13h handler for whole disk encryption using MASM and MSVC 1.52c

Participated in the design and development of multiple major releases

Primary developer and maintainer of several major components of PGP Desktop

Designed and implemented AOL Instant Messenger encryption proxy

Rewrote preference system to be XML based (using libxml2)

Participated in update and maintenance cycle of existing PGP products

Designed custom themeable win32 controls

## OPEN SOURCE/PERSONAL PROJECTS

[teensy-nextkb](#) (Teensy AVR code to interface/emulate a NeXT non-ADB keyboard/mouse) Jan. 2012 - Jan. 2013

Using a logic analyzer, reverse-engineered the NeXT kb/mouse protocol

Used existing open-source PS2 kb/mouse libraries to translate/emulate the NeXT kb/mouse protocol

Prototyped a PS2 to NeXT non-ADB kb/mouse adapter using a [Teensy-2.0](#)

[rideSF.com](#) (San Francisco bicycle trip planner \*Currently inactive\*)

Aug. 2009 - Mar. 2010

Combined publicly available data to form a database of linestrings and meta-data

Wrote several small utilities to sort, combine, and pre-process data set

Designed and implemented web front-end in PHP, Javascript, and CSS

Implemented iPhone/iPod touch compatible [front-end](#)

Wrote flexible back-end that returns route data in multiple standard formats (GPX, KML, JML, WKT, and JSON)

Provided public web service API for external use (Used by [Baytripper](#) iPhone app)

[Solitaire](#) (an iOS solitaire game) Jan. 2016

Game Center integration

[Scannerly](#) (an iOS app to scan the local network) Jan. 2016

Written in Swift

Scans local network using UDP and ARP requests

[Rokumote](#) (an iPhone/iTouch app that allowed remote control of the Roku DVP) Mar. 2009 - Jan. 2016

Rewritten in Swift

Wrote iOS app using Xcode and iPhone SDK

Published three versions to App store

Searches local network using SSDP to find Roku Devices

Controls Roku Device via TCP commands

[Pidgin Currenttrack](#) (Pidgin IM plugin that interfaces with multiple Media players) Aug. 2005 - Aug. 2007

Wrote GTK+ options UI

Builds in Windows, Linux, and OSX

Fetches Album art from iTunes or Amazon.com

Interfaces with 12 Media players in multiple operating systems

[Linux4.be](#) (Project to port Linux to the Casio BE-300) Jun. 2002 - Jun. 2003

Developed serial console driver and kgdb stub

Contributed to development of serial driver

Wrote non-interrupt driven button based scanning keyboard driver

Wrote several tutorials for creating [ramdisks](#)