

Pentesting iPhone & iPad Apps

#Days 2011 – October 28



Who are we?

- Annika Meyer
 - President, co-founder of ADVTOOLS
- Sebastien Andrivet
 - Director, co-founder of ADVTOOLS

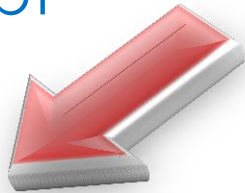
ADVTOOLS

- Swiss company founded in 2002 in Geneva
- Specialized in Information Security & Problems Diagnosis
 - Pentesting
 - Security Audits
 - Forensics
 - Training

Agenda

- Overviews
- Previous researches
- iPhone/iPad application pentest
 - [Our methodology](#)
- Live demonstrations
- Q&A

iOS Application Types

- Web Applications
 - HTML + CSS + Javascript
 - Run inside Safari
- Native Applications: 
 - Written in Objective-C (+ C/C++)
 - Compiled into CPU code: ARM for actual devices, x86 for iOS Simulator
- MonoTouch, Adobe Flash, ...
 - Written in high-level language
 - Compiled into CPU code

iOS Applications

- Distributed as “.ipa” files
 - in fact simply zip files
- Deployed as “.app” directories
 - like on Mac OS X
- Executable code is:
 - **encrypted** with FairPlay DRM (AES)
 - signed with Apple’s signature
 - decryption with GDB or Crackulous

Objective-C

- Objective-C = C + Smalltalk
- Object oriented language
- Created in early 1980s by Stepstone
- Objective-C 2.0 released with Leopard (Mac OS X 10.5)
- Can be mixed with C and C++

Reverse Engineering

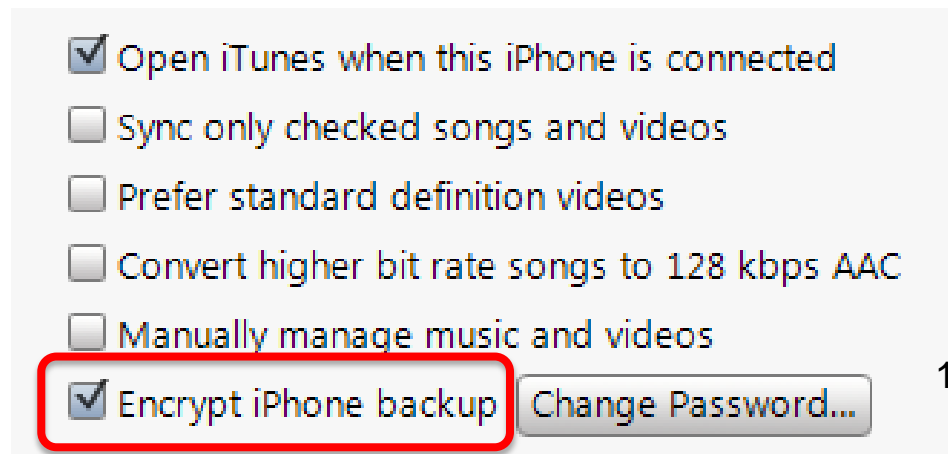
- Not so obvious at first:
 - ARM instruction set
 - Objective-C & objc_msgSend
 - Generated code sometimes strange
 - Few (working) scripts and tools
- Finally not so difficult
- Your best friend:
 - Hex-Rays IDA Pro (Win, Mac, Linux)

Data storage

- plist files (Property lists)
 - Used and **abused**
 - Binary (deprecated) or XML
- Sqlite 3
 - From time to time
- Keychain
- Binary data files (aka unknown)

iTunes & Backups

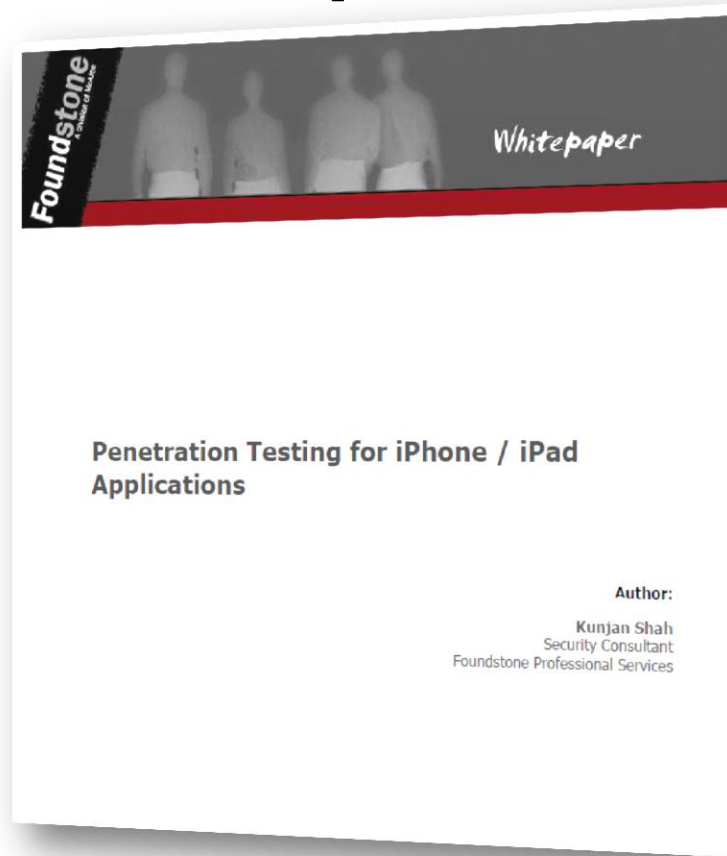
- Every time you connect your device to your computer, a backup is made
- Contains almost all data
- By default, **not encrypted**.
- To mitigate security problems:



Previous researches

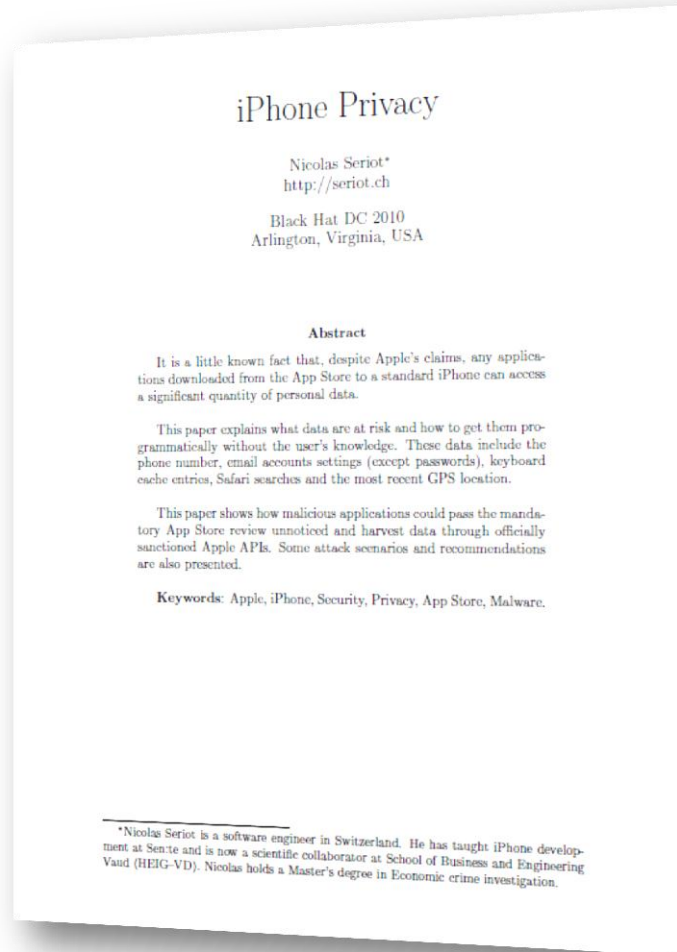
- In general, out of date
- Often inaccurate
- But contain interesting information
- We will give here only some examples

Foundstone (McAfee / Intel)



<http://www.mcafee.com/us/resources/white-papers/foundstone/wp-pen-testing-iphone-ipad-apps.pdf>

Nicolas Seriot



TippingPoint (now HP)

TippingPoint Digital Vaccine Laboratories

DVLabs

- ABOUT
- TEAM
- BLOG
- DVLABS ADVISORIES
- UPCOMING
- PUBLISHED
- APPEARANCES
- RESOURCES
- ZERO DAY INITIATIVE
- RSS FEEDS

DID YOU KNOW...
We release at least two Digital Vaccine updates a week to our IPS customers; on average each has about 10 new security filters, many of which are turned on by default.

Reverse Engineering iPhone AppStore Binaries

BY PEDRAM AMINI
FRI 06 MAR 2009 13:09PM 21431 VIEWS 5 COMMENTS LINK

I recently had the need to peek under the hood of an iPhone application I purchased through the AppStore and quickly came to discover that getting started takes a bit more effort than simply dragging and dropping into IDA. I'm certainly not the first person to have done this, but when faced with a new challenge I like to figure it out the hard way at first, to better understand the fine details. This blog entry details how to get an application into a reversible state.

iPhone apps purchased through the AppStore live in your iTunes library under the folder "Mobile Applications". Each app is stored in a zip archive with a .IPA extension. You can simply rename the file to .ZIP and decompress to view the contents. I'll use the game *Fieldrunners* as the example in this blog, which is in my opinion, the best iPhone game available. Decompressing and loading `Payload/Fieldrunners.app/Fieldrunners` into IDA 5.4 will properly parse the Mach-O binary, list some symbols and provide you with very little and very odd looking disassembled code. Examining the string table reveals next to nothing. This is because the binary is encrypted, the app is in an unacceptable state for reverse engineering. The iPhone loader is responsible for decryption at run-time so I figured my best bet would be to jailbreak my phone and get on the actual device. Jailbreaking is an impressively easy operation these days, requiring only a few minutes with *QuickPWN* and installing some basic necessities like *OpenSSH* and *GDB*. Once on the device, you have to find your target applications directory and make a working copy of it:

```
# cd /private/var/mobile/Applications/  
# find ./ -iname '*.app' | grep Field  
C838FFC-8D74-4DB3-AB99-9410A7E860B7/Fieldrunners.app
```

The executable is a 32-bit Mach-O file which consists of 3 main regions. A header, followed by load commands, followed by segments/sections. Here is an illustration (not my own, found it on Google):

The diagram illustrates the structure of a Mach-O file. It is divided into three main regions: Header, Load commands, and Data. The Header region contains a box for 'Load commands'. The Load commands region contains two boxes labeled 'Segment command 1' and 'Segment command 2'. The Data region contains a box labeled 'Segment 1' which is further divided into three sections: 'Section 1 data', 'Section 2 data', and 'Section 3 data'. Arrows on the right side of the diagram indicate the flow of data between these sections.

<http://dvlabs.tippingpoint.com/blog/2009/03/06/reverse-engineering-iphone-14-appstore-binaries>

ARTeam



2008

Primer on Reversing Jailbroken
iPhone Native Applications



PATCHING APPLICATIONS FROM
APPLE'S APPSTORE WITH
ADDITIONAL PROTECTION



<http://www.accessroot.com/arteam/site/download.php?view.222>
<http://www.accessroot.com/arteam/site/download.php?view.308>

Pentesting iOS Applications

- **Step 1:** Preparing a device
- **Step 2:** Preparing a workstation
- **Step 3:** Preparing a network
- **Step 4:** Pentesting
- **Step 5:** Report

Step 1: Preparing a device

- Dedicated iPhone or iPad
- Jailbreaking
 - Easier if you jailbreak
 - Forbidden by Apple if you are a developer
 - Dangerous: jailbreaking is disabling most of the security features of iOS
- Install tools

Tools

- APT 0.7 Strict
- adv-cmds
- Darwin CC Tools
- GNU Debugger
- inetutils
- lsof
- MobileTerminal
- netcat
- network-cmds
- nmap
- OpenSSH
- tcpdump
- top
- wget

iOS Default Passwords

- By default, there are two users:
 - root
 - mobile
- Passwords = alpine
- **Be sure to change them:**
 - passwd
 - passwd mobile

Step 2 : Workstation

- Windows:
 - OK
- Mac OS X (Lion or Snow Leopard)
 - Better
- Linux, FreeBSD, ...
 - Good luck!
 - Possible but you will need a Windows to run some tools (virtual machine...)

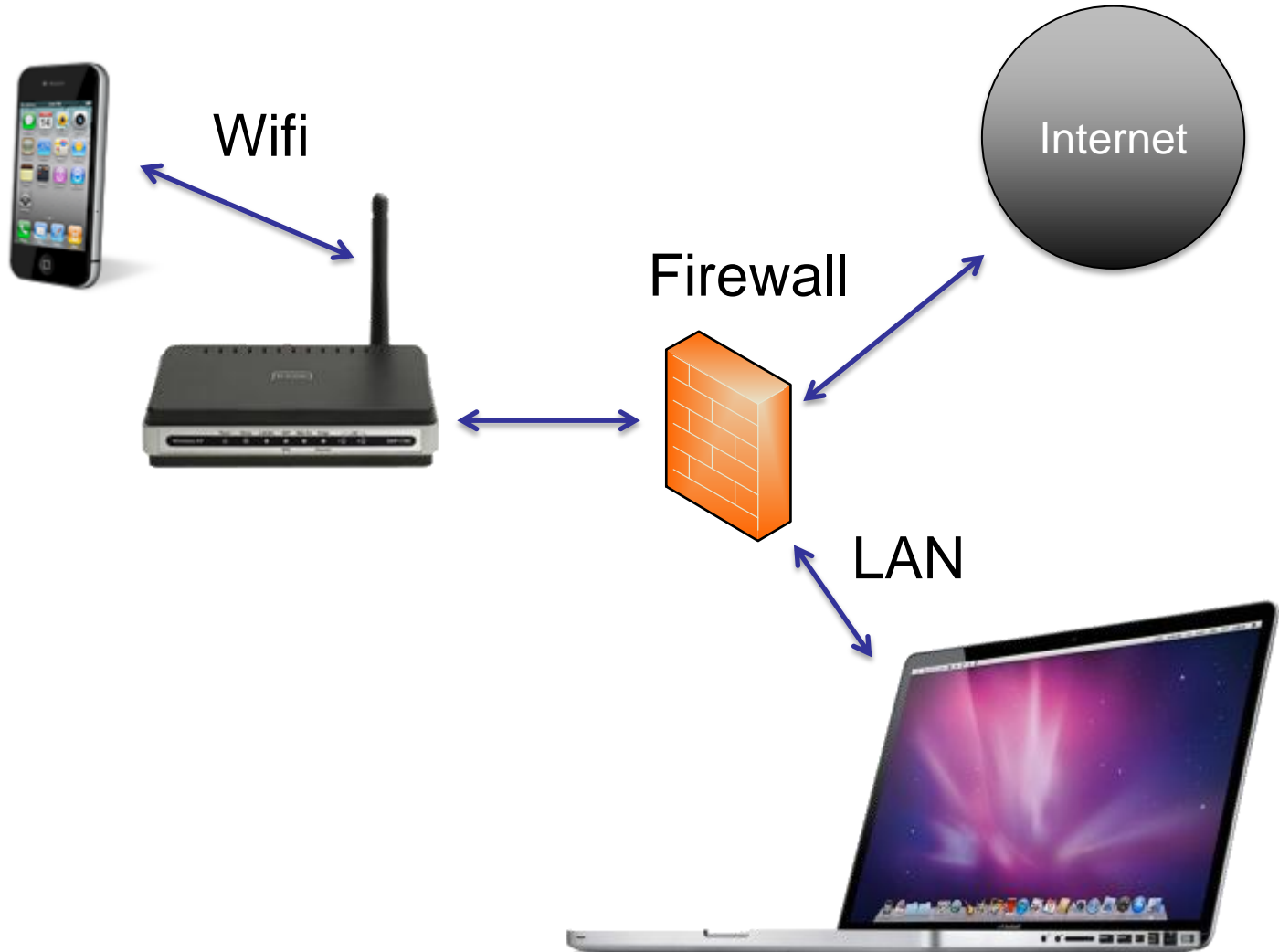
Some Tools

- Windows:
 - SecureCRT or Putty, WinSCP
 - plist Editor for Windows
- Mac OS X:
 - ssh, SecureCRT, Cyberduck
 - XCode
- Windows / Mac:
 - SQLite Database Browser
 - Apple iPhone Configuration Utility
 - Wireshark
 - Burp, Webscarab ...
 - IDA Pro (+ ARM decompiler)

Our Tools

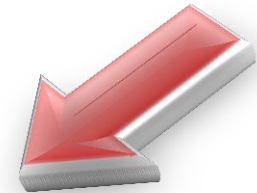
- **ADVsock2pipe**
 - Remote network captures (Windows)
- **ADVinterceptor 2.0**
 - Communications interception
 - DNS & Web Servers
- Available on GitHub under GPLv3
 - <https://github.com/ADVTOOLS>

Step 3: Network

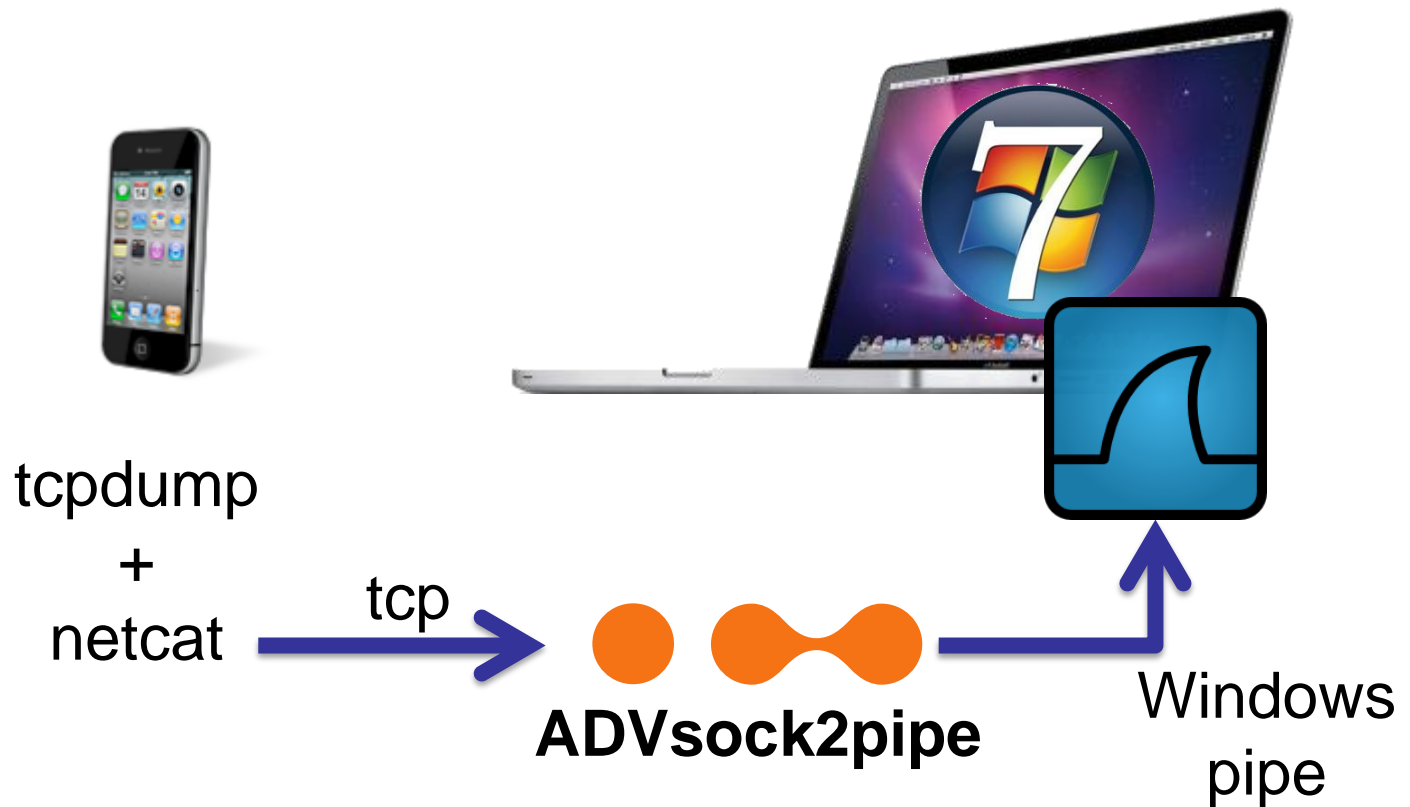


Step 4: Pentesting

- **Step A:** Install app. from iTunes
- **Step B:** Reconnaissance (passive)
 - B.1: Network capture
 - B.2: Interception
 - B.3: Artifacts
 - B.4: Decrypt + Reverse engineering
- **Step C:** Attack (active)
 - C.1: Interception + tampering

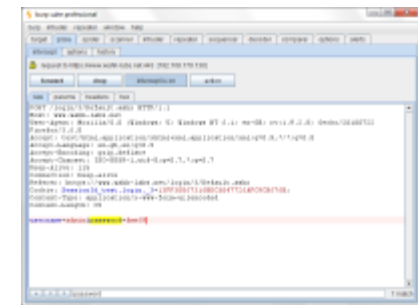


B.1: Network Capture



B.2: Interception

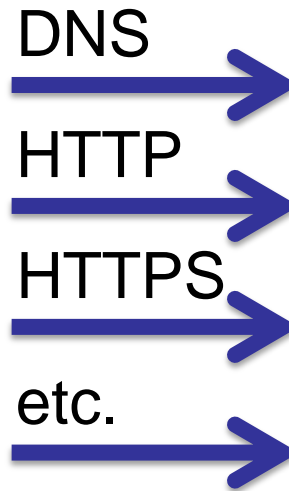
Proxy method



Burp Suite Pro
WebScarab

B.2: Interception

ADVinterceptor



ADVinterceptor 2
(DNS Server,
Web Server,...)

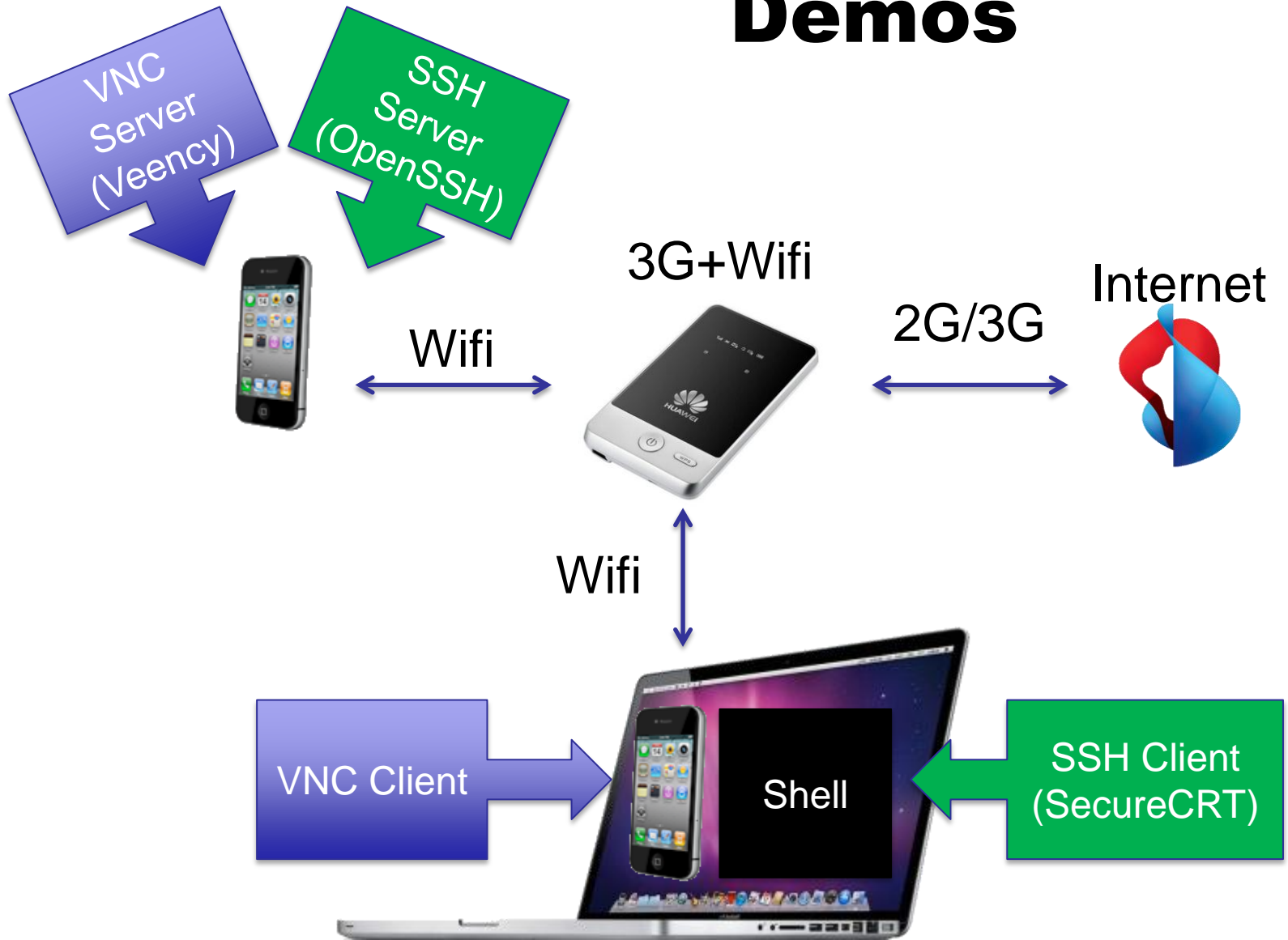
Inject SSL Certificates

- Root from Burp or ADVinterceptor
- Use Apple iPhone Configuration

The image shows two overlapping screenshots. The background is a screenshot of the 'iPhone Configuration Utility' application on a Mac. The window title is 'iPhone Configuration Utility'. It has a menu bar with 'File', 'Edit', 'View', 'Window', and 'Help'. Below the menu bar are icons for 'New', 'Share', and 'Export', and a search field. The main area is divided into a left sidebar and a main content area. The sidebar has sections for 'LIBRARY' (Devices, Applications, Provisioning Profiles, Configuration Profiles) and 'DEVICES' (iPhone). The main content area shows a table with columns 'Name', 'Identifier', and 'Created'. One entry is visible: 'iPhone Apps Interception' with identifier 'com.advttools.iphoneprofile.interception' and created date '5/10/2011 5:58:23 PM'. Below the table are configuration options for various services: LDAP, CalDAV, Subscribed Calendars, CardDAV, Web Clips, Credentials (2 Payloads Configured), SCEP, Mobile Device Management, and Advanced. The 'Credentials' section is expanded, showing two entries: 'ADVtools External Root CA' and 'ADVtools Computer External CA'. Each entry has fields for 'Credential Name' and 'Certificate or Identity Data', and a 'View Certificate' button.

The foreground is a screenshot of an iPhone's 'Profile' settings page. The status bar at the top shows 'No SIM', signal strength, Wi-Fi, and the time '10:51'. The page title is 'Profile' with a 'General' tab selected. The profile is named 'ADVtools Burp Ro...' and is associated with 'ADVTOOLS'. It features a gear icon, a green checkmark indicating it is 'Verified', and a red 'Remove' button. Below this, the 'Description' is 'Profile description.', 'Signed' is 'iPCU CA 907ca688-d134-482d-8371-cd097566e5e3', 'Received' is 'May 19, 2011', and 'Contains' is 'Certificate'. At the bottom right, there is a page number '28' and a 'More Details' link with a right-pointing arrow.

Demos



Windows 7 on Mac Book

Q&A



Thank you

To contact us:

annika@advtools.com

sebastien@advtools.com

Twitter:

[@AndrivetSeb](https://twitter.com/AndrivetSeb)

[@ADVTOOLS](https://twitter.com/ADVTOOLS)

www.advtools.com

