Attacking iPhone & iPad Applications

Soft-Shake 2011 – October 3

Who am I?

- Sebastien Andrivet
  - Director & co-founder of ADVTOOLS

Agenda

- iOS & iOS Applications
- Pentesting iOS Applications
- Demos
- Common vulnerabilities
  - and defense
- Agility and Security
- Conclusion and Q&A

iPhone / iPad (iOS)

- Very similar to Macintoshes
- Major differences
  - Based on ARM processors, not on Intel ones
  - Closed system
  - More secure than Mac OS X (mandatory code signing…)
  - Many OS X APIs not available (CSSM, Secure Transport…)

iOS Application Types

- Web Applications
  - HTML + CSS + Javascript
  - Run inside Safari
- Native Applications
  - Written in Objective-C (+ C++).
  - Compiled into CPU code: ARM for actual devices, i386 for iOS Simulator
- MonoTouch, Adobe Flash, …

Pentesting iOS Applications

- Our methodology
  - Preparing a device
  - Preparing a workstation
  - Preparing a network
  - Actual Pentesting

Preparing a device

- Take (buy) a dedicated iPhone or iPad
- Choose to Jailbreak it or not
  - Forbidden by Apple (developer agreement)
  - It is dangerous (disable security features)
  - Be sure to change root and mobile passwords!
    - Password = alpine
- Install tools on the device

Preparing a workstation

- Windows
  - OK
- Macintosh
  - Better
- Linux, FreeBSD, …
  - Good luck!
  - Possible but you will need a Windows to run some tools (virtual machine…)

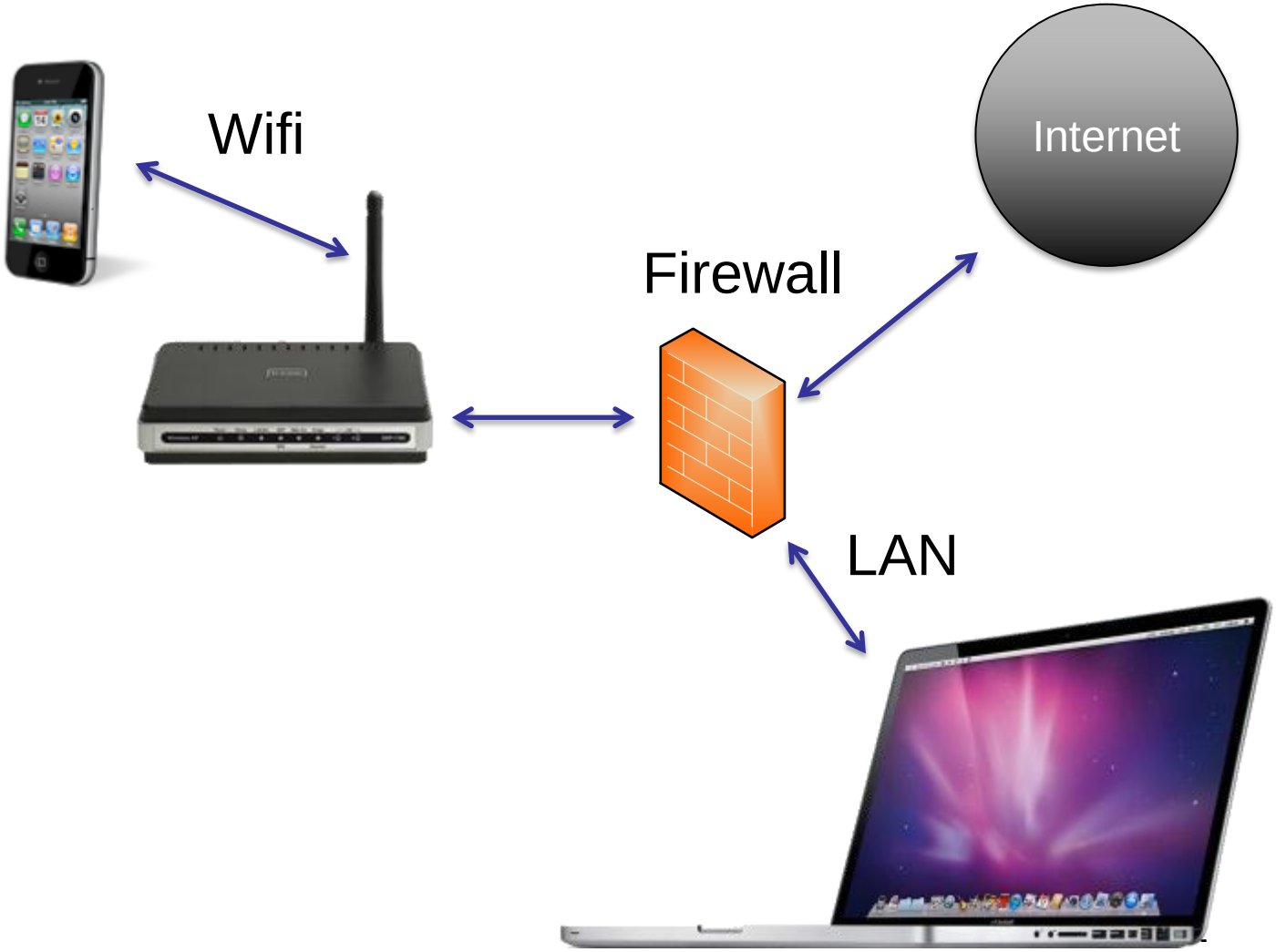
Some tools

- SecureCRT, Putty, WinSCP, Cyberduck
- iPhone Explorer, iPhone Extractor, iPhone Backup Extractor
- plist Editor for Windows
- SQLite Database Browser
- Apple iPhone Configuration Utility
- Apple XCode
- Wireshark, Burp Suite Pro, Webscarab
- IDA Pro

Our tools

- ADVsock2pipe
  - Remote network captures (Windows)
- ADVinterceptor 2.0
  - Communications interception
  - DNS & Web Servers
- Released on GitHub
  - https://github.com/ADVTOOLS
- Written in C# (Microsoft .NET 4.0/4.5)
- GPLv3

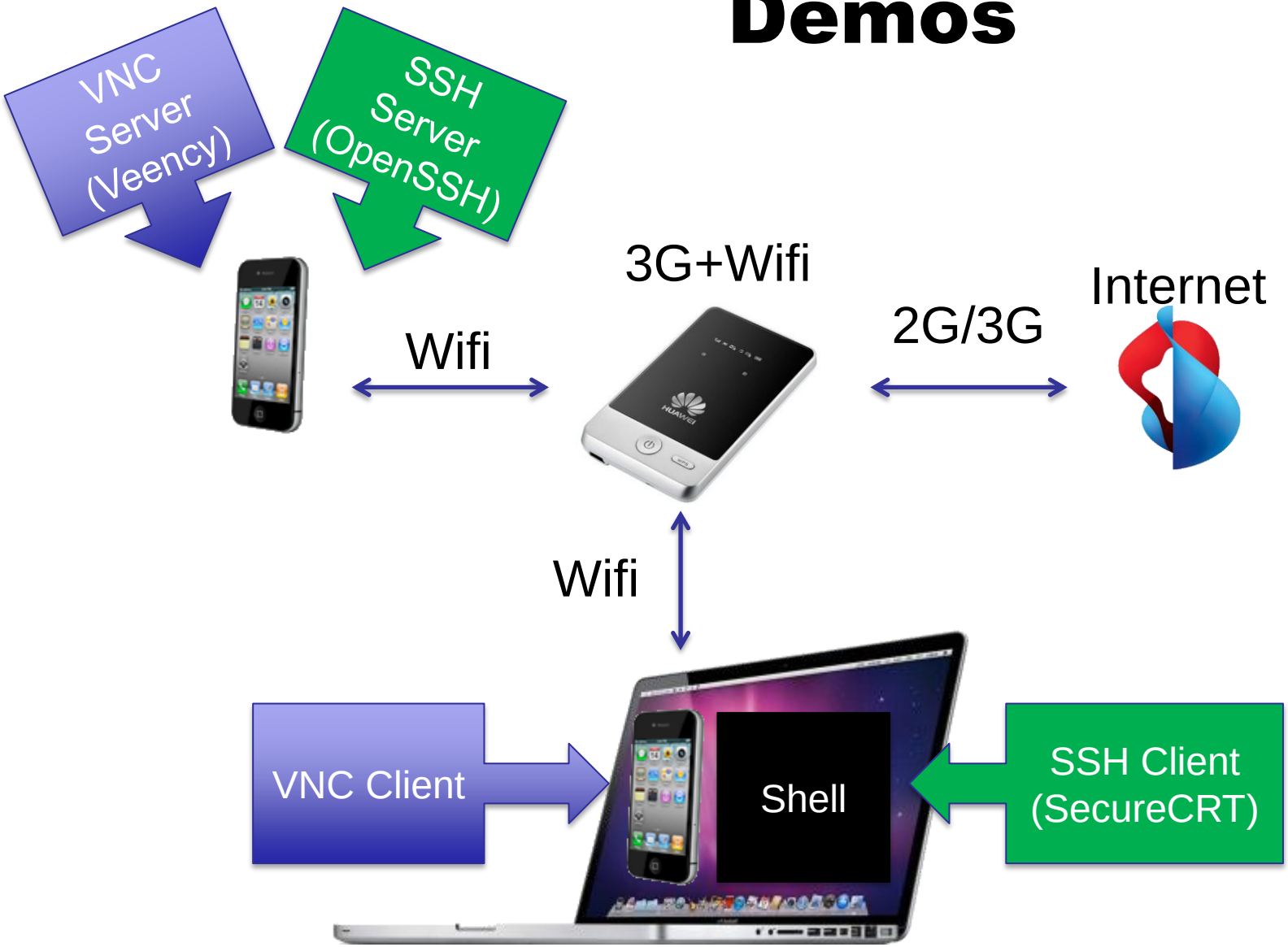
Preparing a network



Pentesting

- **Step A**: Install app. (from iTunes…)
- **Step B**: Reconnaissance (passive)
  - B.1: Capture network
  - B.2: Intercept communications
  - B.3: Study artifacts
  - B.4: Decrypt + Reverse engineering
- **Step C**: Attack (active)
  - C.1: Interception + tampering

Demos



Windows 7 on Mac Book

Demos

Some Common Security Problems

- Unencrypted communications
- Unencrypted storage
- Wrong usage of cryptography or pseudo-cryptography
- Wrong authentication mechanism
- Vulnerable web services
- Blindly trusting third-party web services

- Not an exhaustive list

Unencrypted communications

- Do not forget HTTPS
- Buy a certificate
- CFNetwork
- NSStream
  - NSStreamSocketSecurityLevelXxx
- URL Loading System
  - NSURLConnection, NSURLRequest
- By default, failed certificate verification not allowed (i.e. fail safe)

Unencrypted Storage

- Property Lists, SQLite database, files… are not encrypted
  - The iPhone disk is encrypted with AES but decrypted "on-the fly" by the device, don't rely on it
- Some locations
  - ./Documents
  - ./Library/Preferences
  - ./Library/Caches, ./Library/Caches/Snapshots
  - ./Library/Cookies
  - ./tmp

Protect Files

- Mark file as protected
  - Unreadable while device is locked
- writeToFile (NSData)
  - NSDataWritingFileProtectionComplete
- NSFileManager
  - NSFileProtectionComplete (iOS 4)
- Notification
  - UIApplicationProtectedDataWillBecomeUnavailable

Keychain

- Protected by iOS, encrypted in backups
- SecItemAdd, SecItemUpdate…
- kSecAttrAccessible
  - kSecAttrAccessibleWhenUnlocked
  - kSecAttrAccessibleAfterFirstUnlock
  - kSecAttrAccessibleAlways
  - Etc.
- Attackable, even with a PIN
  - For the moment, not attackable with strong passwords

SQLite

- Encrypt data manually
- SQLCipher
  - Extension to SQLite
  - AES in CBC mode (can be changed)
  - http://sqlcipher.net/

Cryptography

- Use the right algorithm
  - Prefer SHA1 to MD5
  - SHA2 is even better
- CC_SHA1…
- Better to hash passwords with salt than to encrypt passwords
- Encoding is not encrypting
- CRC, Base-64, … are not encryption algorithm

Key

- How and where to store the key?
  - General problem, not specific to mobiles
  - Keychain is a possibility
  - Derived from user credentials

Certificates

- Certificate, Key, Trust Services API
  - SecKeyEncrypt, SecKeyDecrypt

Wrong authentication

- Do not rely on a static password to authenticate the application
  - Even in SSL, very easy to intercept
- Do not rely on a static identifier to authenticate the user
  - DropBox…
- Do not rely on HTTP Agent or similar

Vulnerable Web Services

- Your mobile application is not the only potential client
- Classical vulnerabilities:
  - SQL injection
  - XML and XPath attacks
  - REST attacks
  - Etc.

Blindly trusting third-party web services

- Never trust foreign data
  - Validate
  - Prefer white lists to black lists

U.S. Exportation Regulations

- iTunes servers are in U.S.A.
- "Does your app use encryption?"
- If you use encryption, you fall under U.S. Government requirement for a CCATS review and approval
  - Even if you only use Apple APIs
  - Using "HTTPS" is enough
  - Exception: SSL for authentication
  - More simple if you only sell in the U.S. (and Canada)

Agile & Security

- "Agile hurts secure code development"
  - Adrian Lane, securosis, February 2010
- Main problem: sprint duration too short for meaningful security testing
- Example of solution
  - Microsoft SDL for Agile
  - Reorganize security practices into 3 categories

ADVTOOLS

- Swiss company founded in 2002 in Geneva
- Specialized in Information Security & Problems Diagnosis
  - Pentesting mobile apps (iOS…)
  - Trainings (secure development of iOS apps, pentesting iOS apps)
  - Forensics
  - Security Audits
  - Pentesting web apps and services

Thank you

To contact me:

sebastien@advtools.com

Twitter:

@AndrivetSeb

@ADVTOOLS

www.advtools.com