HACKING AND SECURING IOS APPLICATIONS

http://www.securitylearn.net

-Satish B

Agenda

- iOS Security Concepts
- Loopholes in iOS
- Hacking & Securing iOS Applications
  - How does loophole in iOS affects the apps
  - How easy it's to steal data from the apps
  - How to protect these apps

Who Am I?

< 1 Development

- Framework for functional testing tools

5+ Information Security

- Web & Mobile application security

Other Interests

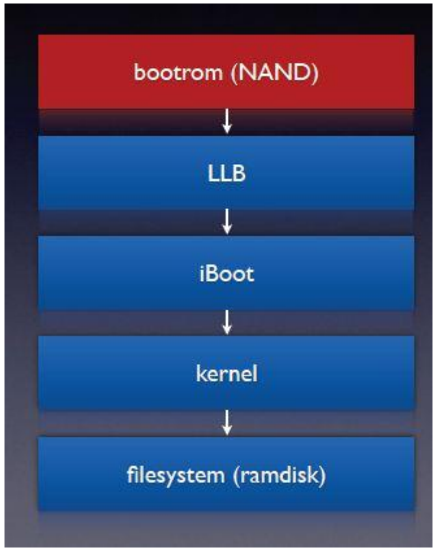
- iOS Forensics & hacking
- Tool development & Knowledge Sharing

iOS Basics

- iOS is the Operating System that run on Apple devices like iPhone, iPod, iPad & Apple TV
- Stripped down Mac OS X + XNU kernel
- Provides multi tasking
- Only allows to run Apple signed applications
- New features & Bug fixes with every release
  - Current version is iOS 6.0.1

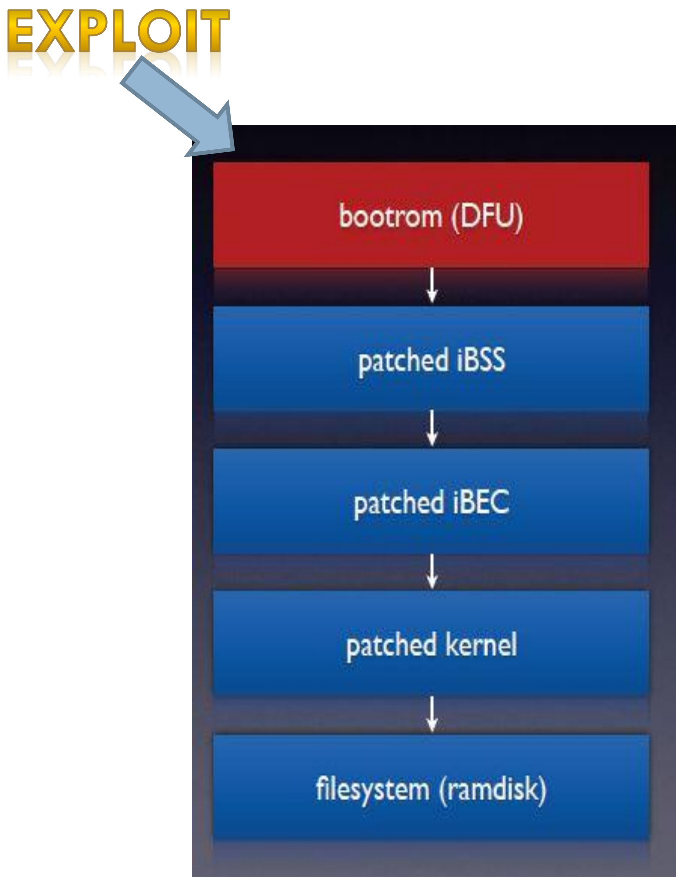
iOS Security Features

- Boot Chain
  - Chain of trust
  - Series of signature checks
  - BootRom->LLB->iBoot->kernel->file system
- Code Signing
  - Prevents running of unauthorized apps/code
  - Verifies the integrity of the app code at rest & runtime
  - Malware prevention
- Passcode
  - Prevents unauthorized access to the device
  - Default is 4-digit passcode & Support for complex passcode
  - Configurable data wipe after 10 failed attempts



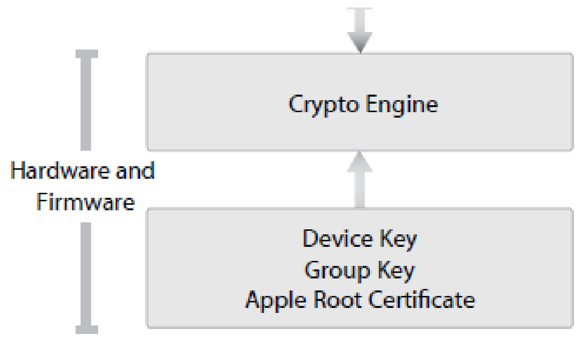
Access data without passcode

- Breaking Chain of trust
  - Bootrom exploit
  - Patch the series of signature checks
- Boot with custom ramdisk
  - Access file system
- No Bootrom exploit for latest devices
  - iPhone 4s & 5, iPad 2 &3, iPad Mini



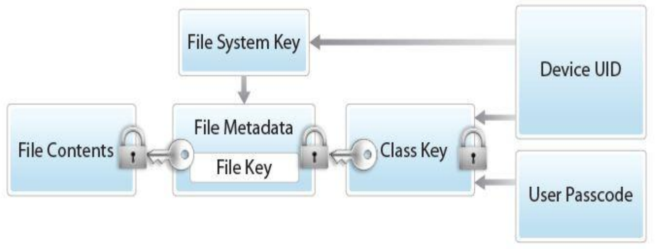
iOS Security Features

- Encryption
  - Dedicated crypto engine
  - Two hardcoded keys – UID & GID
  - Usage of UID & GID is limited



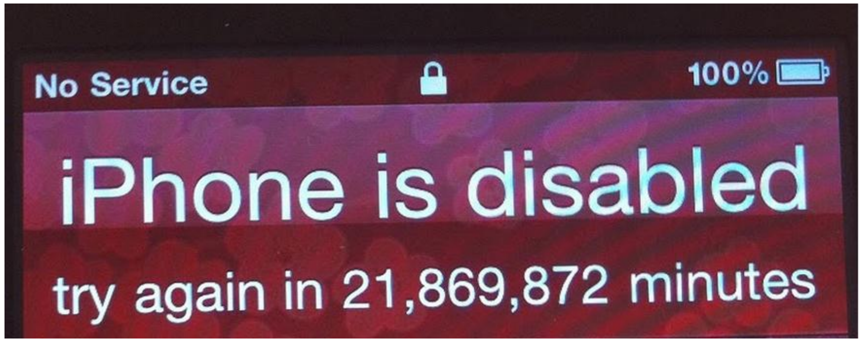
- Data Protection
  - Ties the data encryption to the user's passcode
  - Files are not accessible when the device is locked
  - No Passcode = No data protection

- File Encryption
  - Every File is encrypted with unique key
  - File key is stored in the file metadata
  - Metadata is encrypted with EMF Key



Bypassing the iPhone passcode

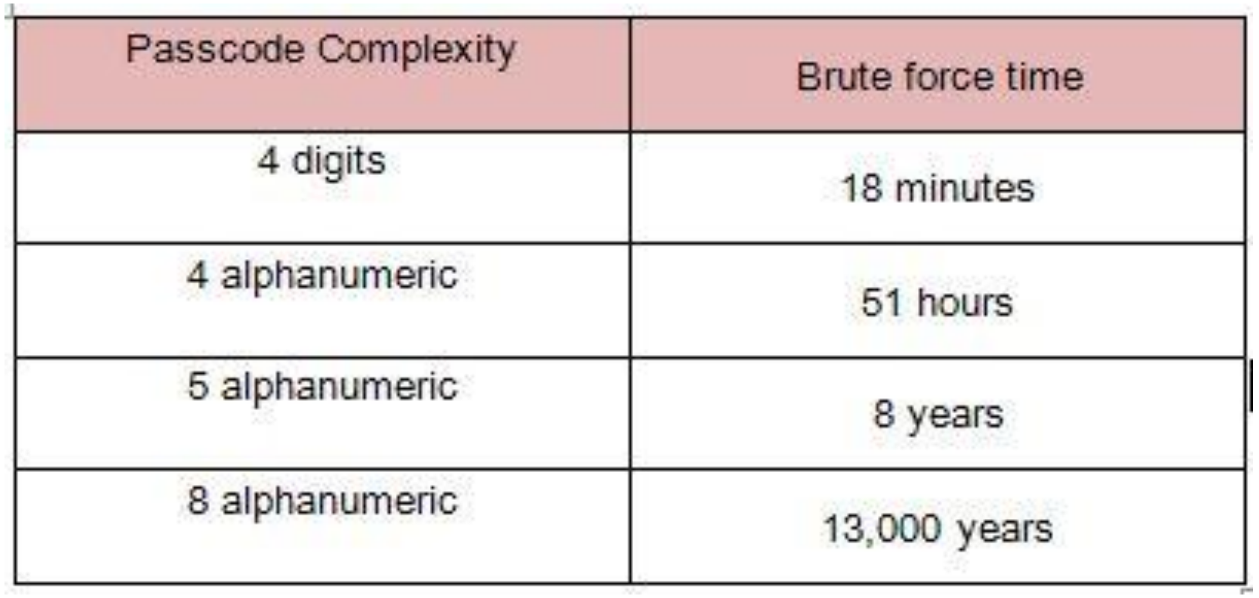
- Custom ram disk gives access to the file system
- Passcode is required to access those protected files
- Passcode is not stored on the device in any format
- Brute force is the only option



- Brute forcing at Springboard
  - 6 failed attempts introduces delay
  - Delay from 1 min to several days
- Brute forcing at kernel level
  - Passcode validity is verified by unlocking the System Keybag
  - Load brute force script in custom ramdisk and try to unlock Keybag

Bypassing the iPhone passcode

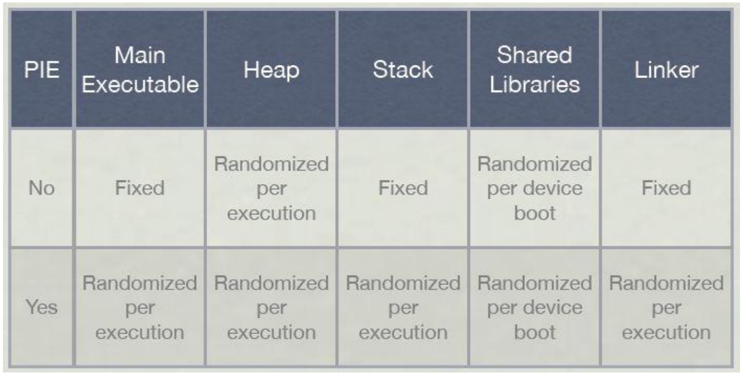
- Brute force time depends on the iPhone hardware
- On iPhone 4 –



Passcode Complexity	Brute force time
4 digits	18 minutes
4 alphanumeric	51 hours
5 alphanumeric	8 years
8 alphanumeric	13,000 years

iOS Security Features

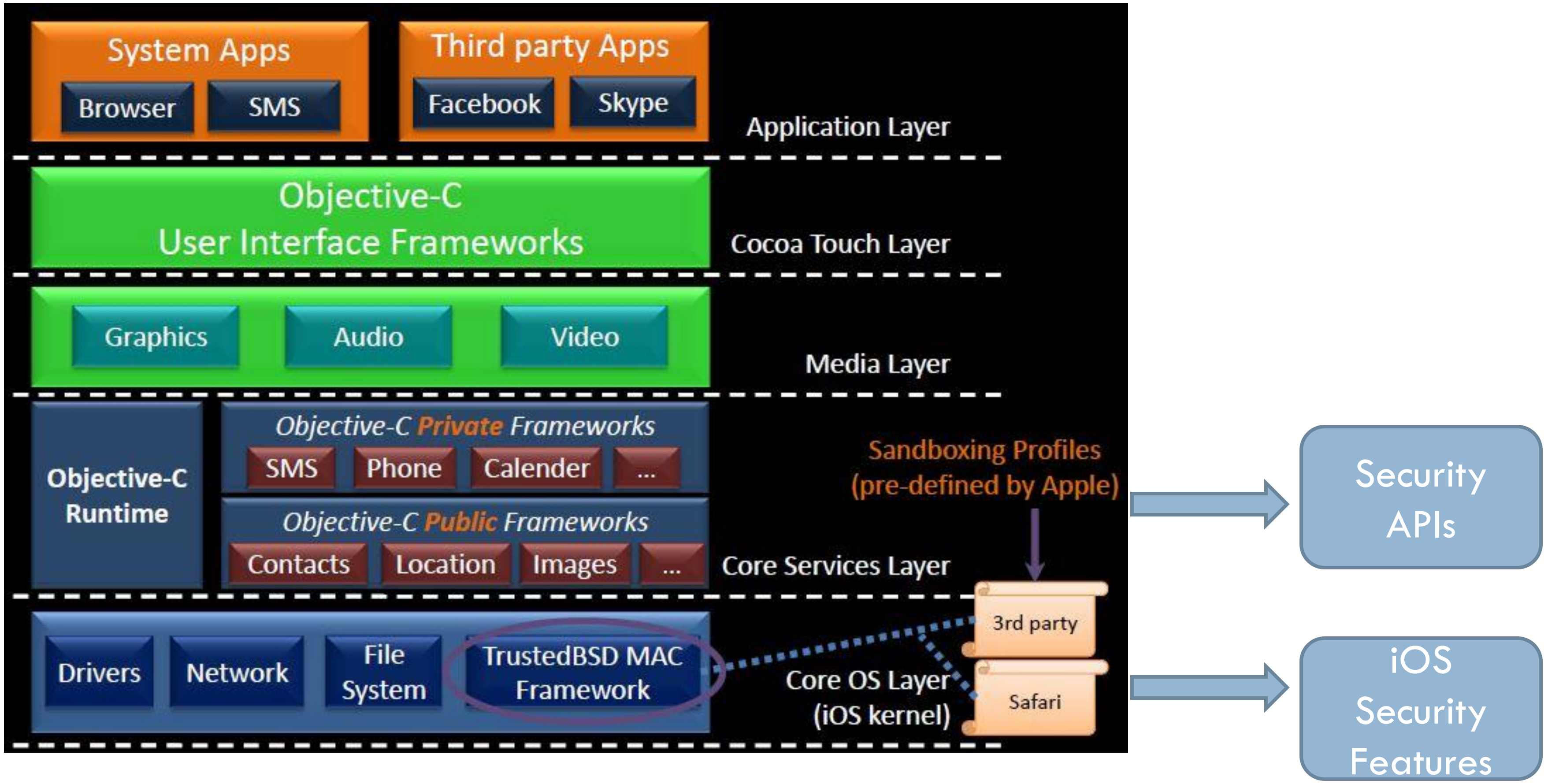
- ASLR - Address Space layout randomization
  - Randomizes the memory address
  - Apps can built with partial or full ASLR
    - Full - Compiled with PIE



PIE	Main Executable	Heap	Stack	Shared Libraries	Linker
No	Fixed	Randomized per execution	Fixed	Randomized per device boot	Fixed
Yes	Randomized per execution	Randomized per execution	Randomized per execution	Randomized per device boot	Randomized per execution

- DEP – Data Execution Prevention
  - Differentiates code and data
  - Prevents the execution of code from non executable memory pages
- Stack Canaries
  - Stack smashing protection
  - Canary is placed after the local variables
  - Protects from buffer overflows

iOS Software Stack



Ahmad-Reza Sadeghi et al, "Overview on apple iOS" paper

Types of iOS Applications

- Browser based
  - Run inside safari
  - Built with server side technology like PHP, .NET,…
  - HTML, CSS & JavaScript rendering styled to the device

- Native
  - Built with iOS SDK & ARM compiled
  - Written in Objective C & Cocoa Touch API

- Hybrid
  - Native container that leverage browser engine
  - Objective C, HTML 5 & JavaScript

Areas of focus for hacking

- Device storage
  - Plist
  - Sqlite
  - Cookies
  - Keychain
- Run time analysis
  - Breaking simple locks
- Sniffing Networks
  - MITM & Transport security

Local Storage

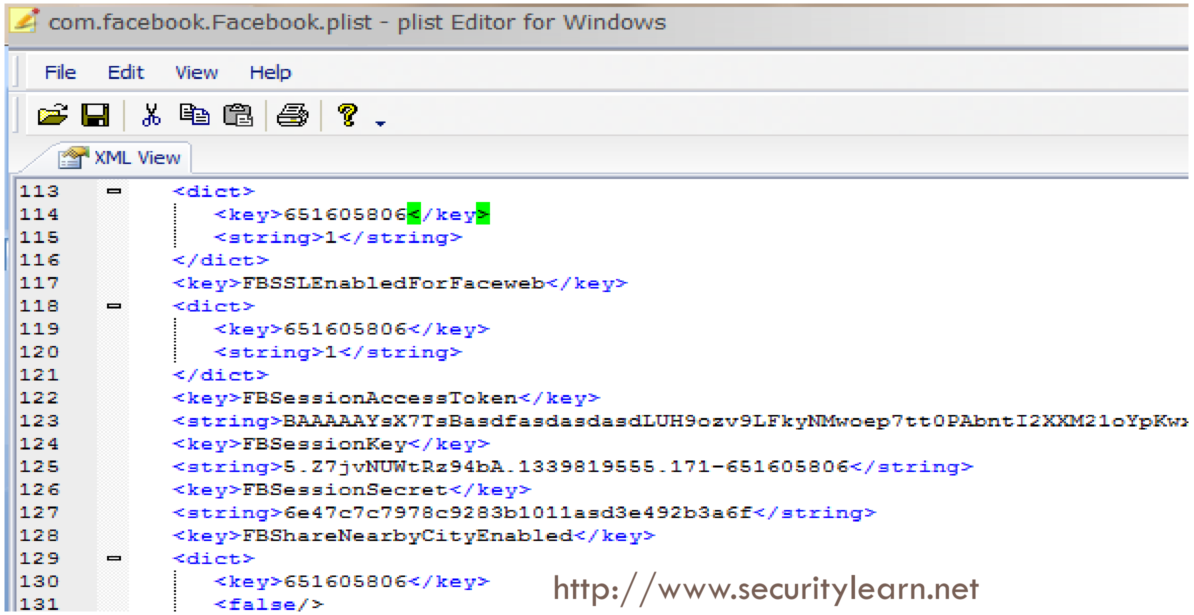
App Sandbox

- Apps run in a self-contained environment called Sandbox
- Apps can not access data from other apps
- All apps run as one user: mobile

SubDirectory	Description
Appname.app	Contains the application code and static data
Documents	Data that may be shared with desktop through iTunes
Library	Application support files
Library/Preferences/	App specific preferences
Library/Caches/	Data that should persist across successive launches of the application but not needed to be backed up
tmp	Temporary files that do not need to persist across successive launches of the application

Plist files

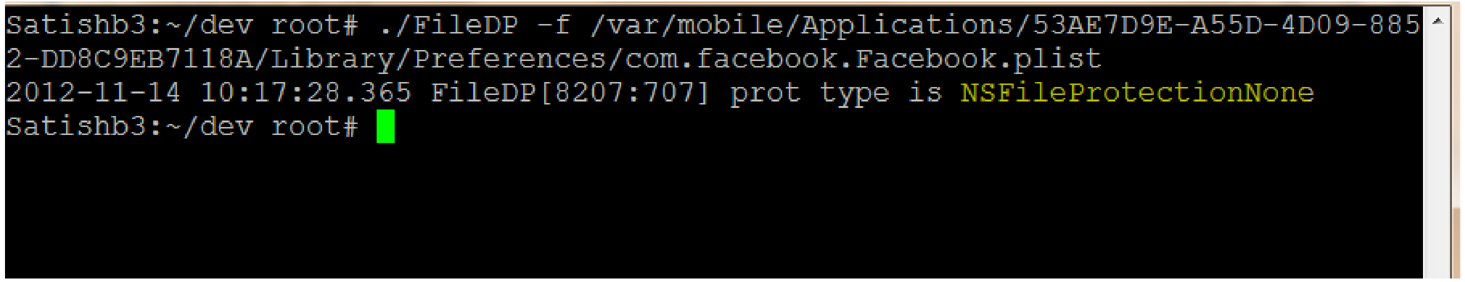
- Property list files - Key value pairs stored in binary or XML format
- Easily viewed and modified using property list editors (plutil)
- Designed to store user's properties and configuration information
- But Apps store usernames, passwords, email ids and session info
- Ex: Facebook stores the authentication tokens



```
<dict>
    <key>651605806</key>
    <string>1</string>
</dict>
<key>FBSSLEnabledForFaceweb</key>
<dict>
    <key>651605806</key>
    <string>1</string>
</dict>
<key>FBSessionAccessToken</key>
<string>BAAAAAYsX7TsBasdfasdasdasdLUH9ozv9LFkyNMwoep7tt0PAbntI2XXM21oYpKw
<key>FBSessionKey</key>
<string>5.Z7jvNUWtRz94bA.1339819555.171-651605806</string>
<key>FBSessionSecret</key>
<string>6e47c7c7978c9283b1011asd3e492b3a6f</string>
<key>FBShareNearbyCityEnabled</key>
<dict>
    <key>651605806</key>
    <false/>
```

Plist files

- Apps create plist files with any or without a file extension
- Plists are identified by a file header – bplist
- Plist files are not protected by Data protection



```
Satishb3:~/dev root# ./FileDP -f /var/mobile/Applications/53AE7D9E-A55D-4D09-885
2-DD8C9EB7118A/Library/Preferences/com.facebook.Facebook.plist
2012-11-14 10:17:28.365 FileDP[8207:707] prot type is NSFileProtectionNone
Satishb3:~/dev root#
```

- Plists are stored un-encrypted in the iOS normal backups (iTunes).
- Apps may delete the plist files upon logout
- File system changes are recorded in HFS Journal
- Deleted files can be recovered by carving the HFS Journal

Facebook Session Hijacking

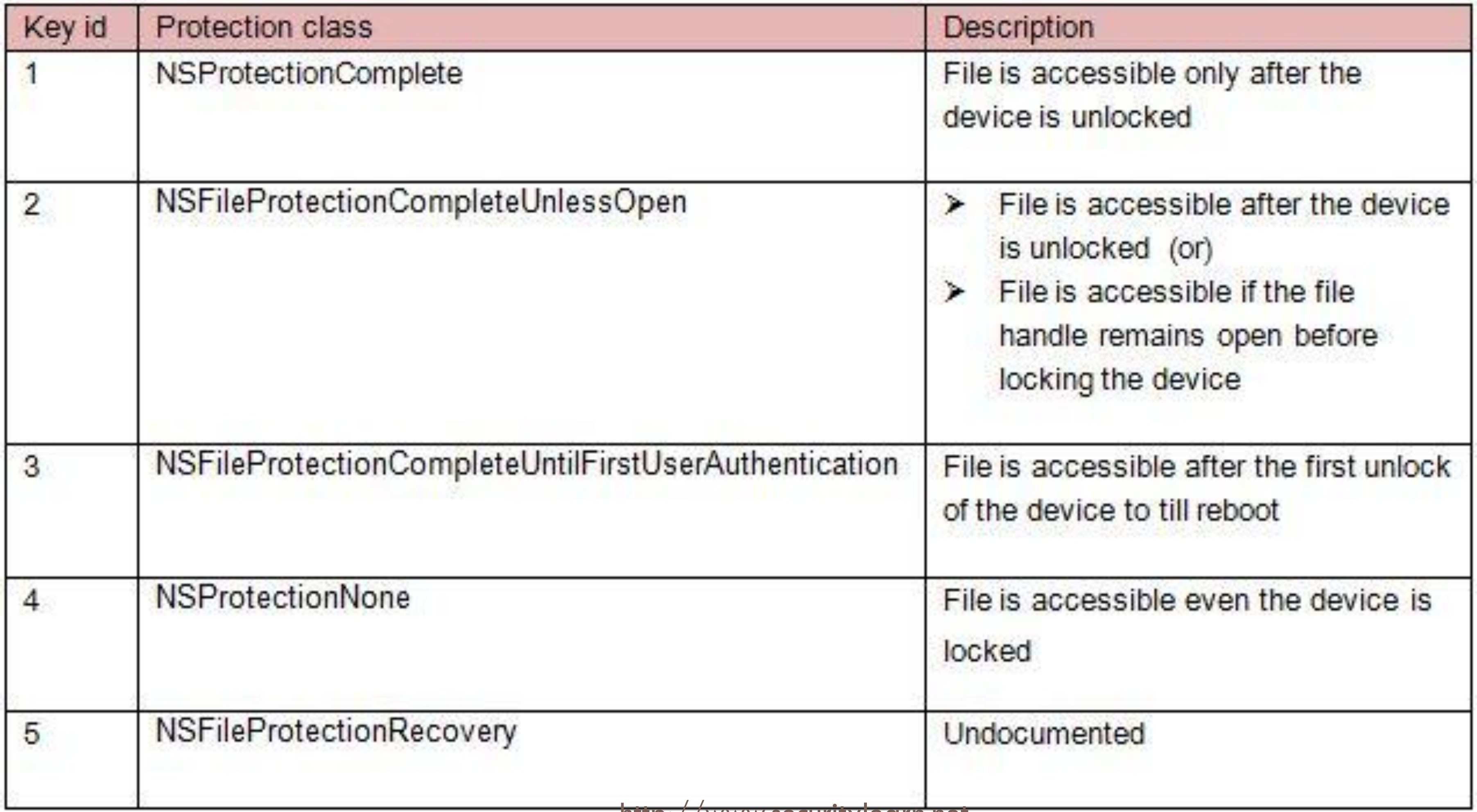
- Facebook stores authentication tokens in plist file
- Gaining access to the plist allows to log into the app
- Plist files can be stolen
  - Upon physical access to the device
  - From backups : Metasploit post exploitation script to read iOS backup
- In addition to that, Tokens never expired even on Logout

FIXED

Plist files

- Do not store sensitive data in Plist files
- If required, use custom encryption
- Protect plist files with data protection API
- Create plist files Library/Caches folder
  - iTunes does not backup caches directory
- For better security, Implement classes for secure file wipe
  - Before deleting the file overwrite the file bytes with junk values

Data Protection for files

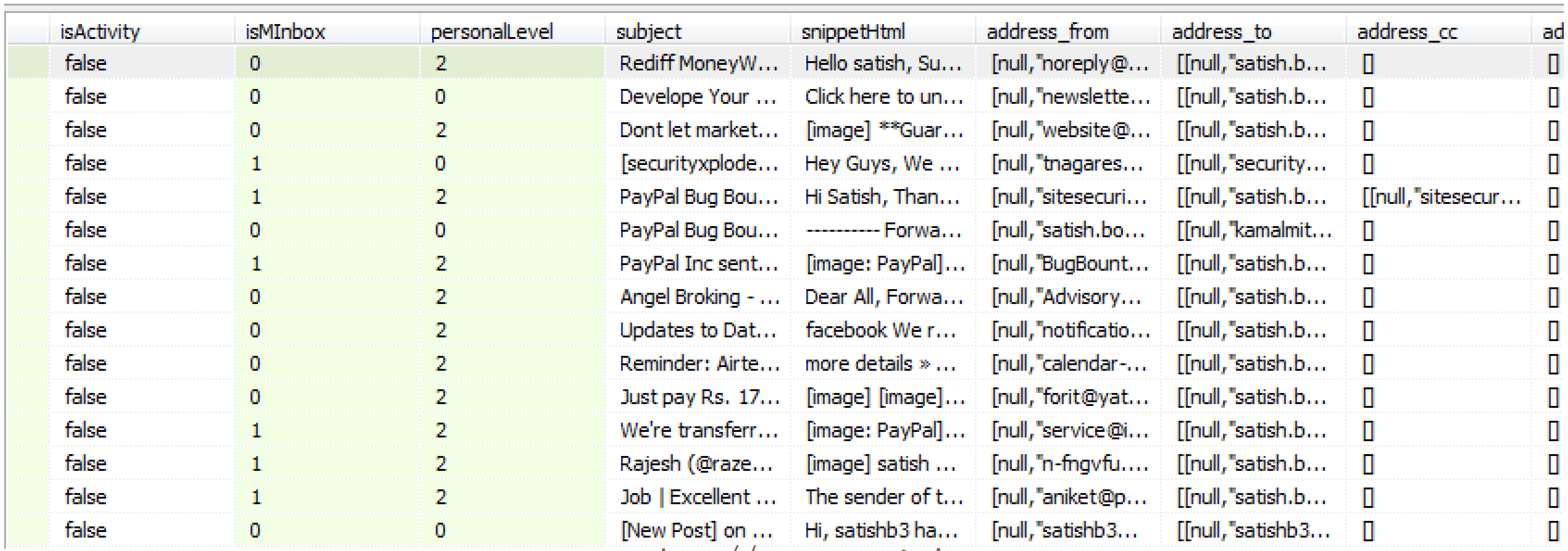


Key id	Protection class	Description
1	NSProtectionComplete	File is accessible only after the device is unlocked
2	NSFileProtectionCompleteUnlessOpen	➢ File is accessible after the device is unlocked (or) ➢ File is accessible if the file handle remains open before locking the device
3	NSFileProtectionCompleteUntilFirstUserAuthentication	File is accessible after the first unlock of the device to till reboot
4	NSProtectionNone	File is accessible even the device is locked
5	NSFileProtectionRecovery	Undocumented

Sqlite files

- Lightweight database for structured data storage
- Sqlite is portable, reliable, small and available as a single flat file
- Sqlite is preferred as it gives good memory usage and speed
- Apps store usernames, passwords, emails and sensitive data

  Ex: Gmail stores the emails in Sqlite db file for offline access



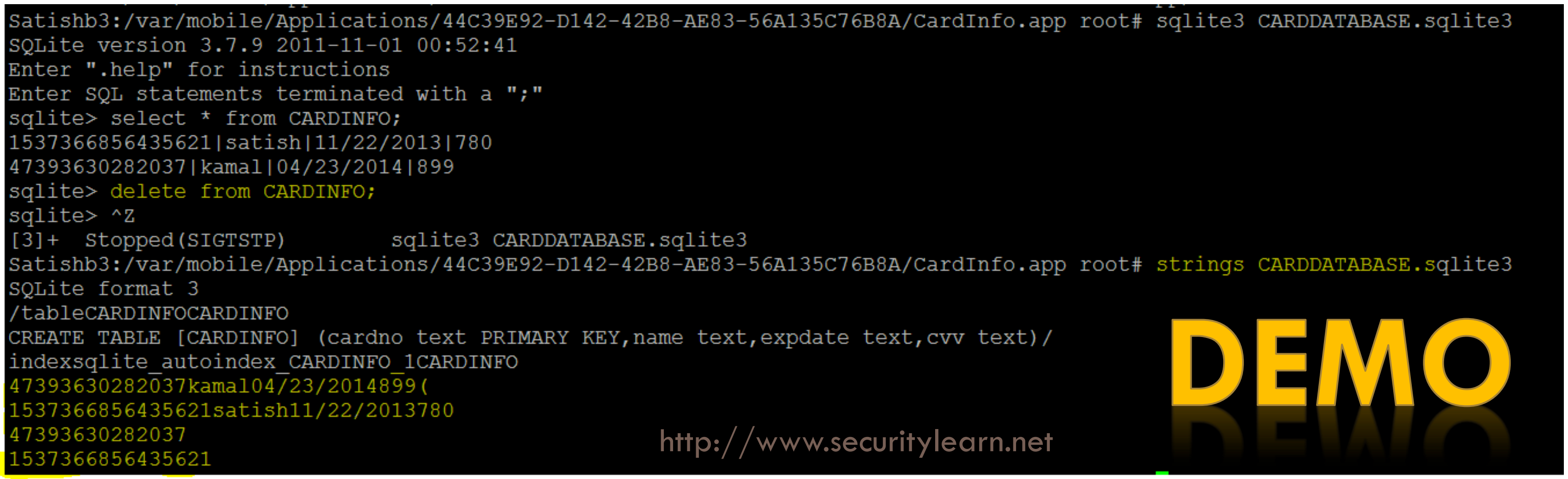
isActivity	isMInbox	personalLevel	subject	snippetHtml	address_from	address_to	address_cc	ad
false	0	2	Rediff MoneyW…	Hello satish, Su…	[null,"noreply@…	[[null,"satish.b…	[]	[]
false	0	0	Develope Your …	Click here to un…	[null,"newslette…	[[null,"satish.b…	[]	[]
false	0	2	Dont let market…	[image] **Guar…	[null,"website@…	[[null,"satish.b…	[]	[]
false	1	0	[securityxplode…	Hey Guys, We …	[null,"tnagares…	[[null,"security…	[]	[]
false	1	2	PayPal Bug Bou…	Hi Satish, Than…	[null,"sitesecuri…	[[null,"satish.b…	[[null,"sitesecur…	[]
false	0	0	PayPal Bug Bou…	---------- Forwa…	[null,"satish.bo…	[[null,"kamalmit…	[]	[]
false	1	2	PayPal Inc sent…	[image: PayPal]…	[null,"BugBount…	[[null,"satish.b…	[]	[]
false	0	2	Angel Broking - …	Dear All, Forwa…	[null,"Advisory…	[[null,"satish.b…	[]	[]
false	0	2	Updates to Dat…	facebook We r…	[null,"notificatio…	[[null,"satish.b…	[]	[]
false	0	2	Reminder: Airte…	more details » …	[null,"calendar-…	[[null,"satish.b…	[]	[]
false	0	2	Just pay Rs. 17…	[image] [image]…	[null,"forit@yat…	[[null,"satish.b…	[]	[]
false	1	2	We're transferr…	[image: PayPal]…	[null,"service@i…	[[null,"satish.b…	[]	[]
false	1	2	Rajesh (@raze…	[image] satish …	[null,"n-fngvfu.…	[[null,"satish.b…	[]	[]
false	1	2	Job | Excellent …	The sender of t…	[null,"aniket@p…	[[null,"satish.b…	[]	[]
false	0	0	[New Post] on …	Hi, satishb3 ha…	[null,"satishb3…	[[null,"satishb3…	[]	[]

Sqlite files

- Sqlite can be created with any or without a file extension
- Sqlite files can be viewed using Sqlite Spy or sqlite3
- Data stored in the Sqlite is un-encrypted
- Sqlite files are stored un-encrypted in the iOS backups (iTunes)
- Apps may delete Sqlite files upon logout
- Delete files can be recovered by carving the HFS Journal

Sqlite files

- Apps may delete the Sqlite records
- Sqlite – tags the records as deleted but not purge them
- Records which are marked as deleted can be recovered by reading the WAL (Write Ahead Log)
- Recovering Sqlite records is easy compare to recovering the files
  - **Strings** command can be used to print the deleted records



```
Satishb3:/var/mobile/Applications/44C39E92-D142-42B8-AE83-56A135C76B8A/CardInfo.app root# sqlite3 CARDDATABASE.sqlite3
SQLite version 3.7.9 2011-11-01 00:52:41
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite> select * from CARDINFO;
1537366856435621|satish|11/22/2013|780
47393630282037|kamal|04/23/2014|899
sqlite> delete from CARDINFO;
sqlite> ^Z
[3]+  Stopped(SIGTSTP)        sqlite3 CARDDATABASE.sqlite3
Satishb3:/var/mobile/Applications/44C39E92-D142-42B8-AE83-56A135C76B8A/CardInfo.app root# strings CARDDATABASE.sqlite3
SQLite format 3
/tableCARDINFOCARDINFO
CREATE TABLE [CARDINFO] (cardno text PRIMARY KEY,name text,expdate text,cvv text)/
indexsqlite_autoindex_CARDINFO_1CARDINFO
47393630282037kamal04/23/2014899(
1537366856435621satish11/22/2013780
47393630282037
1537366856435621
```

Sqlite files

- Do not store sensitive data in clear text
- Use custom encryption
- Protect Sqlite files with data protection API
- Implement classes for secure file wipe
- Purge the data upon deletion with VACUUM SQL command.
  - VACUUM rebuilds the database
  - Doing it for every delete consumes time
- Before deleting the Sqlite record, replace the data with junk values
  - Data and Junk value length has to be same

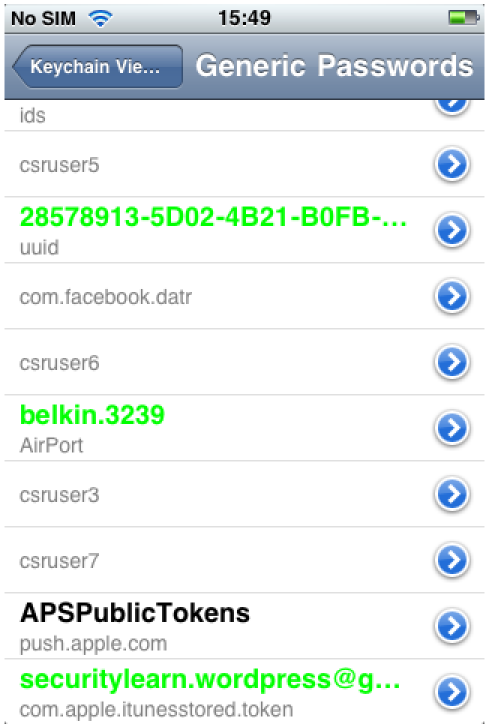
Keychain

- Sqlite database for sensitive data storage
- Apple says "keychain is a secure place to store keys and passwor
- Located at: /var/Keychains/keychain-2.db
- Four tables: genp, inet, cert, keys
- Keychain encryption is tied to the device
  - Protected entries are tied to the user's passcode
- Keychain file is accessible to all the applications
- Application can only access it's own key chain items
  - Based on app keychain access group

Keychain

- On a JailBroken device Keychain restrictions can be bypassed
- Design an app as a member of all keychain access groups (*)
  - Keychain Dumper Tool
- Design app with com.apple.keystore.access-keychain-keys permission
  - Keychain viewer – by Sogeti

DEMO


No SIM 15:49

Keychain Vie... Generic Passwords

ids

csruser5

28578913-5D02-4B21-B0FB-...
uuid

com.facebook.datr

csruser6

belkin.3239
AirPort

csruser3

csruser7

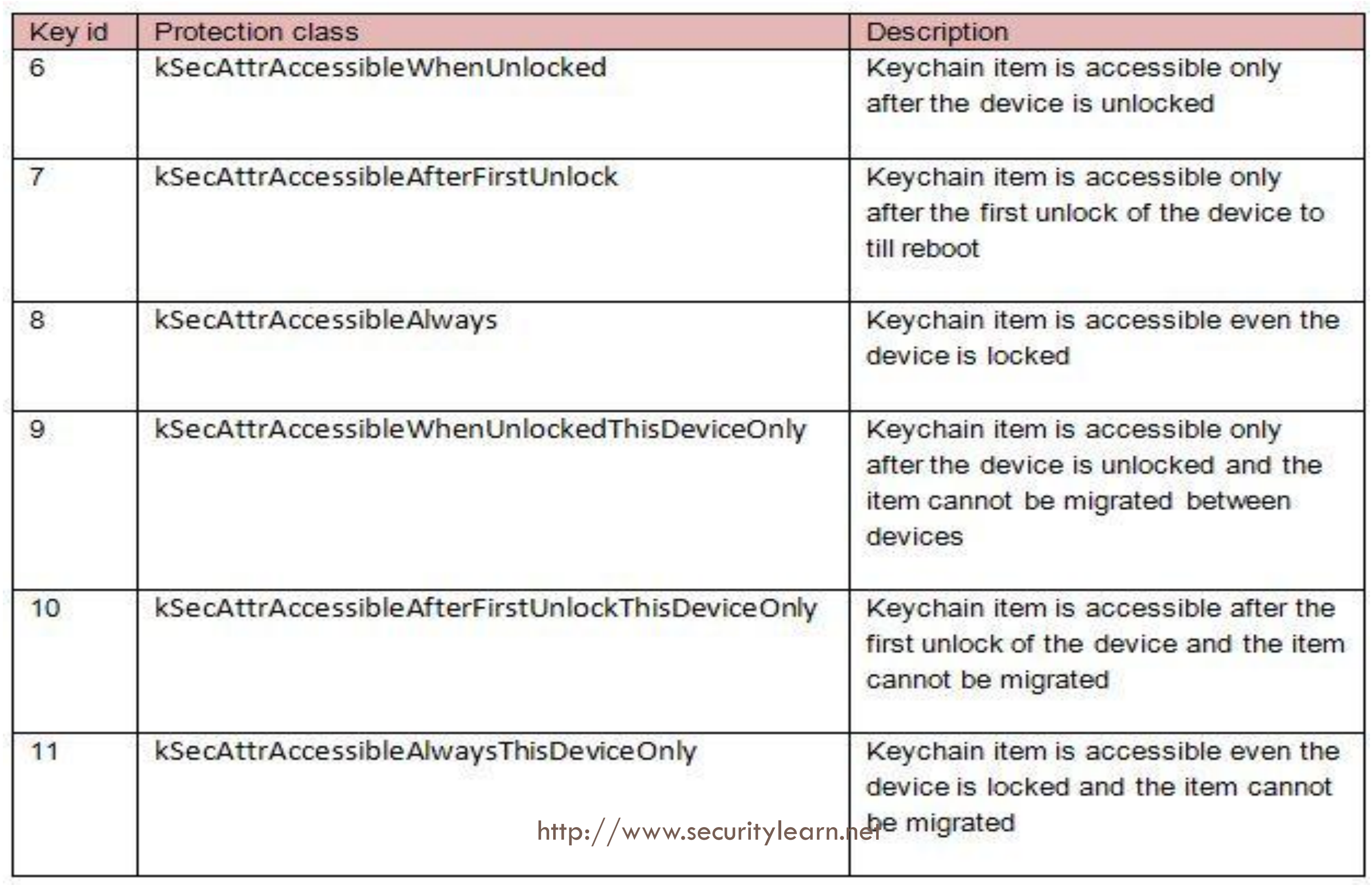
APSPublicTokens
push.apple.com

securitylearn.wordpress@g...
com.apple.itunesstored.token

Keychain

- Keychain is also not secure. Do not store sensitive data in clear text.
- Encrypt the data using custom encryption (CCCrypt)
- Use data protection API while storing data in keychain
- BY default entries are created with *kSecAttrAccessibleWhenUnlocked* data protection
- Apple may change the default protection any time
- Do not store the encryption keys in the binary

Data Protection for keychain



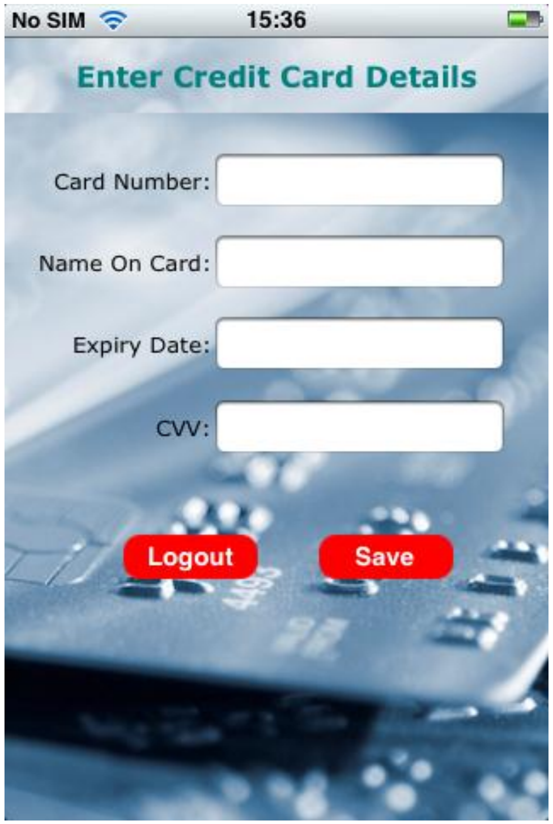
Key id	Protection class	Description
6	kSecAttrAccessibleWhenUnlocked	Keychain item is accessible only after the device is unlocked
7	kSecAttrAccessibleAfterFirstUnlock	Keychain item is accessible only after the first unlock of the device to till reboot
8	kSecAttrAccessibleAlways	Keychain item is accessible even the device is locked
9	kSecAttrAccessibleWhenUnlockedThisDeviceOnly	Keychain item is accessible only after the device is unlocked and the item cannot be migrated between devices
10	kSecAttrAccessibleAfterFirstUnlockThisDeviceOnly	Keychain item is accessible after the first unlock of the device and the item cannot be migrated
11	kSecAttrAccessibleAlwaysThisDeviceOnly	Keychain item is accessible even the device is locked and the item cannot be migrated

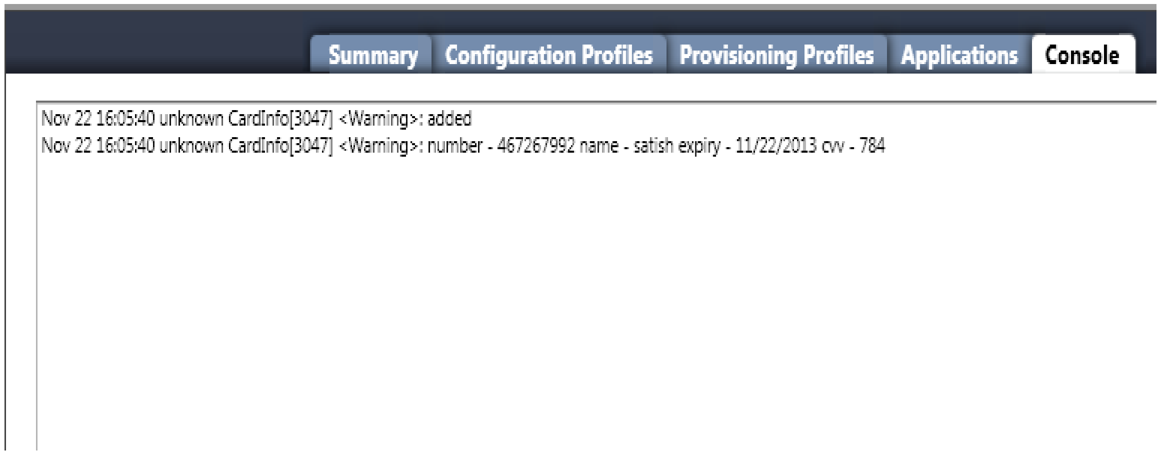
Error Logs

- Apps may write sensitive data in logs
  - Debugging (NSLog calls)
  - Trouble shooting
  - Requests & Responses
- Located at - /private/var/log/syslog
- To view iPhone logs
  - Console App (from AppStore)
  - iTunes Sync (CrashReporter folder)
  - iPhone configuration utility - Console

Error Logs

- Syslog is out of sandbox – Any app can access it
- Do not write sensitive data in the syslog file





```
Nov 22 16:05:40 unknown CardInfo[3047] <Warning>: added
Nov 22 16:05:40 unknown CardInfo[3047] <Warning>: number - 467267992 name - satish expiry - 11/22/2013 cvv - 784
```

DEMO

http://www.securitylearn.net

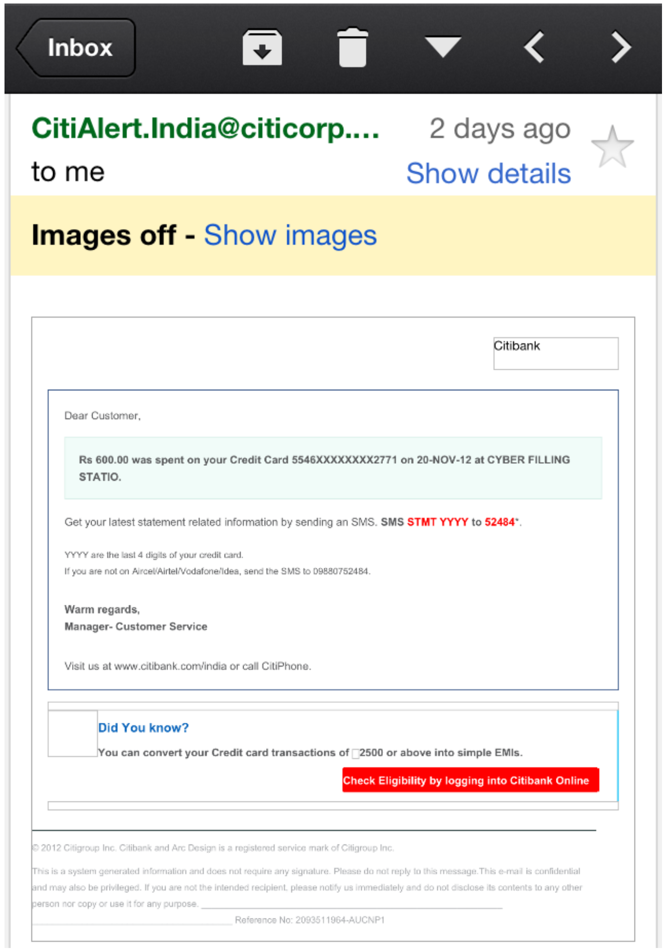
Screenshot

- Home button shrinks your application with a nice effect
- iOS takes screen shots of the application to create that effect
- Sensitive data may get cached
  - App directory/Library/Caches/Snapshots

- Remove sensitive data or change the screen before the applicationDidEnterBackground() function returns
- Instead of hiding or removing sensitive data you can also prevent back-grounding altogether by setting the "Application does not run in background" property in the application's Info.plist file

Screenshot

- Gmail Screenshot



CitiAlert.India@citicorp.... 2 days ago
to me Show details

Images off - Show images

Citibank

Dear Customer,

Rs 600.00 was spent on your Credit Card 5546XXXXXXXX2771 on 20-NOV-12 at CYBER FILLING STATIO.

Get your latest statement related information by sending an SMS. **SMS STMT YYYY to 52484***.

YYYY are the last 4 digits of your credit card.
If you are not on Aircel/Airtel/Vodafone/Idea, send the SMS to 09880752484.

Warm regards,
Manager- Customer Service

Visit us at www.citibank.com/india or call CitiPhone.

Did You know?

You can convert your Credit card transactions of 2500 or above into simple EMIs.

© 2012 Citigroup Inc. Citibank and Arc Design is a registered service mark of Citigroup Inc.

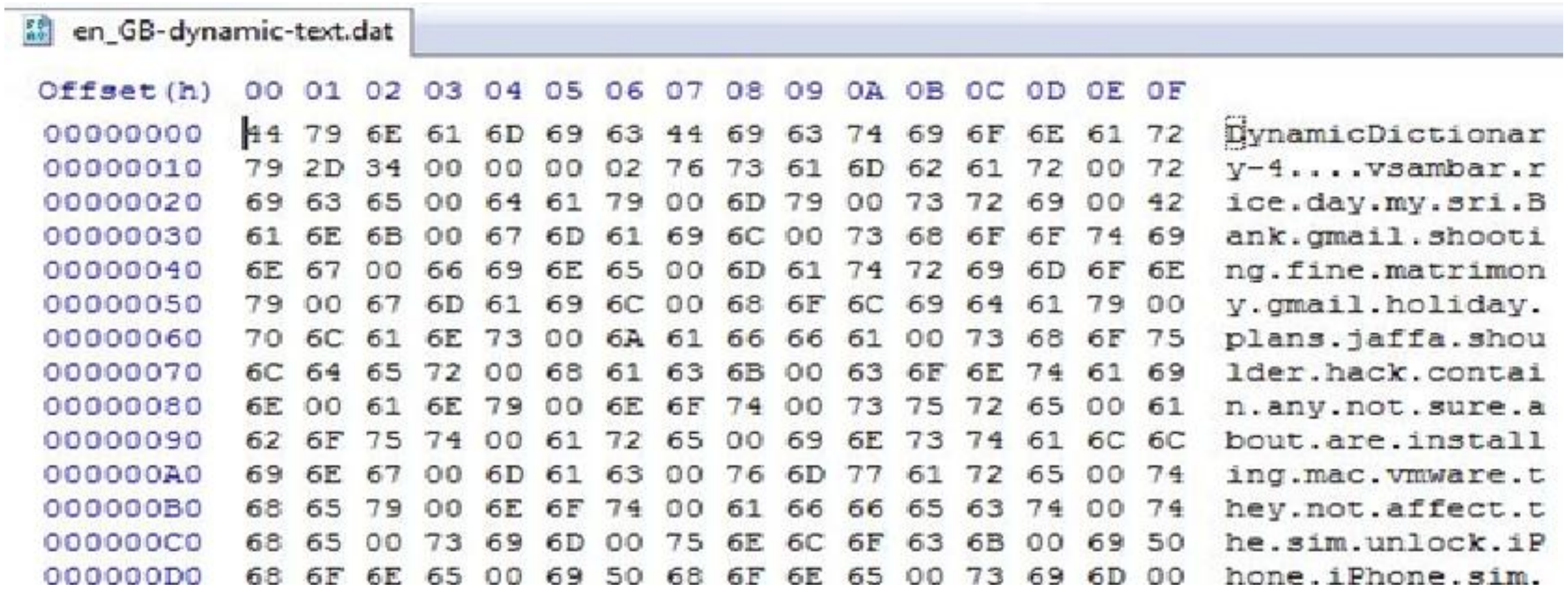
This is a system generated information and does not require any signature. Please do not reply to this message.This e-mail is confidential and may also be privileged. If you are not the intended recipient, please notify us immediately and do not disclose its contents to any other person nor copy or use it for any purpose.

Reference No: 2093511964-AUCNP1

http://www.securitylearn.net

Keyboard cache

- iPhone records everything that a user types in clear text
- Designed to auto complete the predictive common words
- Located at - Library/Keyboard/en_GB-dynamic-text.dat
- Viewed using a hex editor


en_GB-dynamic-text.dat

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000  44 79 6E 61 6D 69 63 44 69 63 74 69 6F 6E 61 72  DynamicDictionar
00000010  79 2D 34 00 00 00 02 76 73 61 6D 62 61 72 00 72  y-4....vsambar.r
00000020  69 63 65 00 64 61 79 00 6D 79 00 73 72 69 00 42  ice.day.my.sri.B
00000030  61 6E 6B 00 67 6D 61 69 6C 00 73 68 6F 6F 74 69  ank.gmail.shooti
00000040  6E 67 00 66 69 6E 65 00 6D 61 74 72 69 6D 6F 6E  ng.fine.matrimon
00000050  79 00 67 6D 61 69 6C 00 68 6F 6C 69 64 61 79 00  y.gmail.holiday.
00000060  70 6C 61 6E 73 00 6A 61 66 66 61 00 73 68 6F 75  plans.jaffa.shou
00000070  6C 64 65 72 00 68 61 63 6B 00 63 6F 6E 74 61 69  lder.hack.contai
00000080  6E 00 61 6E 79 00 6E 6F 74 00 73 75 72 65 00 61  n.any.not.sure.a
00000090  62 6F 75 74 00 61 72 65 00 69 6E 73 74 61 6C 6C  bout.are.install
000000A0  69 6E 67 00 6D 61 63 00 76 6D 77 61 72 65 00 74  ing.mac.vmware.t
000000B0  68 65 79 00 6E 6F 74 00 61 66 66 65 63 74 00 74  hey.not.affect.t
000000C0  68 65 00 73 69 6D 00 75 6E 6C 6F 63 6B 00 69 50  he.sim.unlock.iP
000000D0  68 6F 6E 65 00 69 50 68 6F 6E 65 00 73 69 6D 00  hone.iPhone.sim.
```

Keyboard cache

- Secure fields are not stored
  - Passwords are safe
- Strings with all digits are not stored
  - Pins and credit card numbers are safe
- Data typed into text fields are cached
  - Usernames and security question answers…

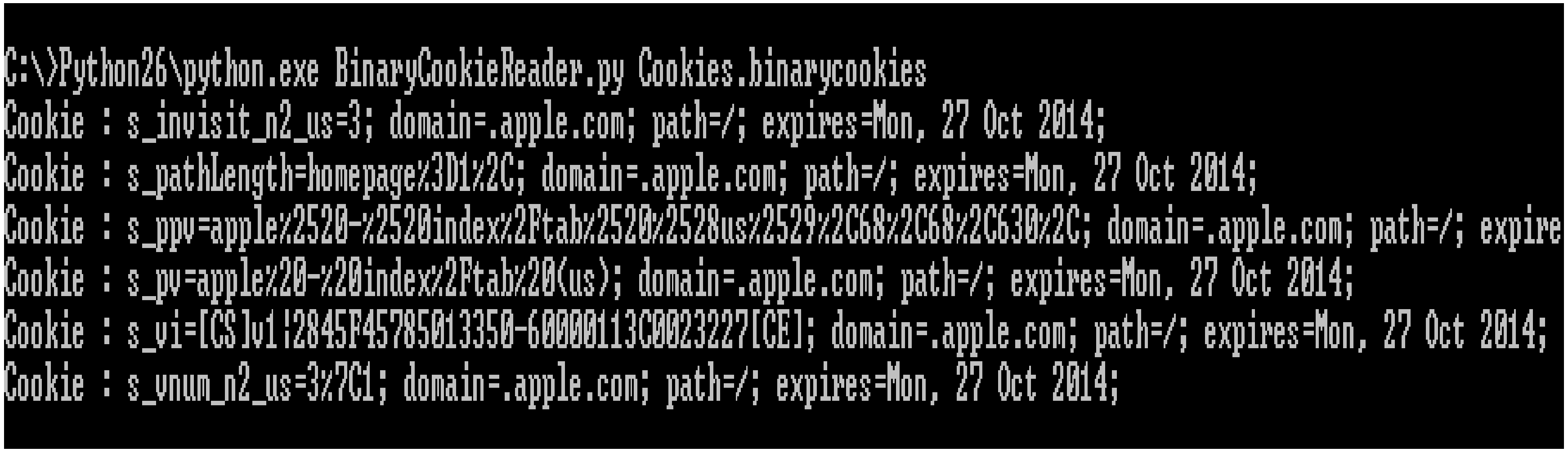
- To disable auto complete of a text field
  - Mark it as a secure field
    mytextField.secureTextEntry = YES
  - Disable auto correction
    mytextField.autocorrectionType = UITextAutocorrectionTypeNo;

Cookies.binarycookies

- Binary file to store the cookies
- Persistent cookies are stored along with the flags (Secure, HTTPOnly)
- Most iOS apps does not prompt the user for login every time and creates persistent cookies
- Apps store the session cookies locally
- Grabbing cookies allows to log into the user's account

Cookies.binarycookies

- **BinaryCookieReader.py** can be used to read the cookie files



```
C:\>Python26\python.exe BinaryCookieReader.py Cookies.binarycookies
Cookie : s_invisit_n2_us=3; domain=.apple.com; path=/; expires=Mon, 27 Oct 2014;
Cookie : s_pathLength=homepage%3D1%2C; domain=.apple.com; path=/; expires=Mon, 27 Oct 2014;
Cookie : s_ppv=apple%2520-%2520index%2Ftab%2520%2528us%2529%2C68%2C68%2C630%2C; domain=.apple.com; path=/; expire
Cookie : s_pv=apple%20-%20index%2Ftab%20(us); domain=.apple.com; path=/; expires=Mon, 27 Oct 2014;
Cookie : s_vi=[CS]v1|2845F45785013350-60000113C0023227[CE]; domain=.apple.com; path=/; expires=Mon, 27 Oct 2014;
Cookie : s_vnum_n2_us=3%7C1; domain=.apple.com; path=/; expires=Mon, 27 Oct 2014;
```

- For critical applications don't create persistent cookies

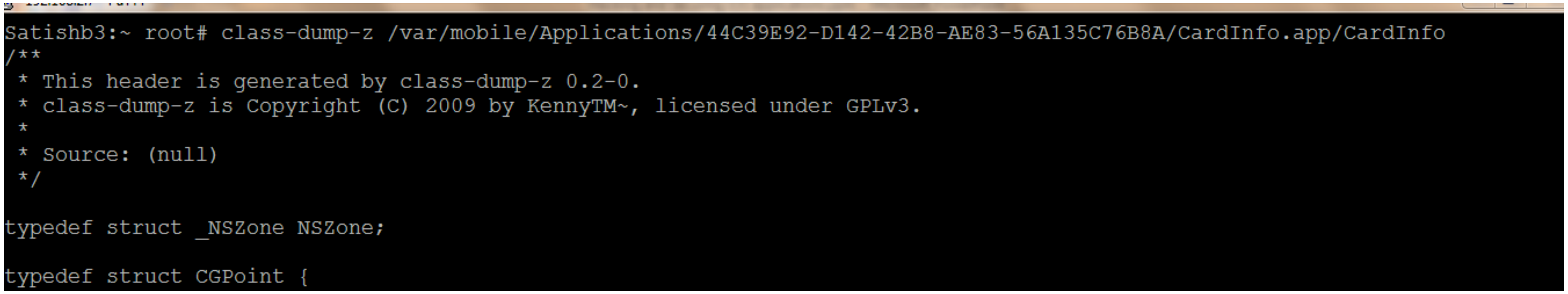
Run Time Analysis

Binary Analysis

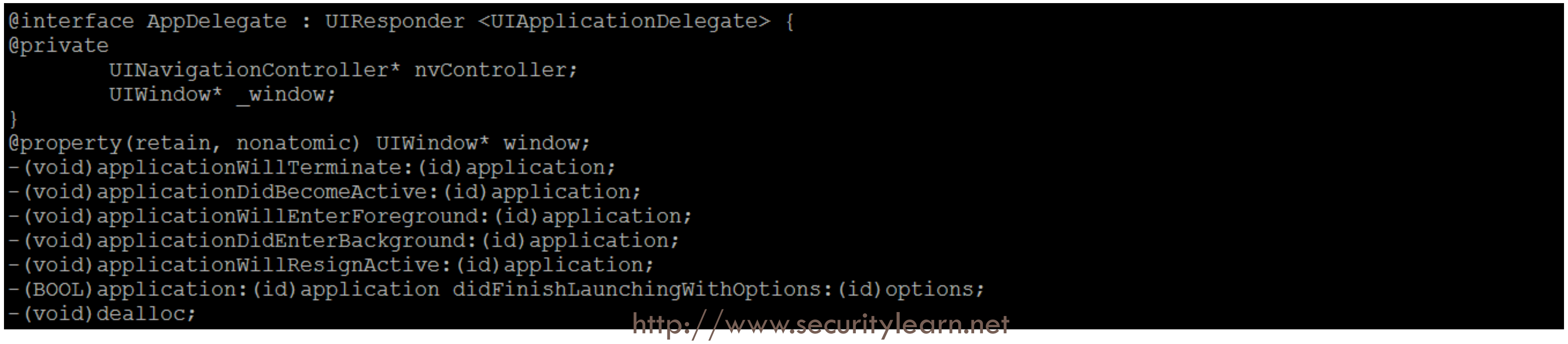
- Self distributed Apps are not encrypted
- AppStore binaries are encrypted
  - Similar to Fairplay DRM used on iTunes music
- Loader decrypts the apps when loaded into memory
- Debugger can be used to dump the decrypted app from memory into a file
- Tools are available: Craculous & Installous
- GNU Debugger or IDA Pro are used on decrypted binary for better analysis
- Look for Hard coded passwords, encryption keys, buffer over flows and format string attacks

Runtime Analysis

- Use class-dump-z on decrypted binary and map the application
- iOS app centralized point of control (MVC) – UIApplication class
- Analyze the class dump output and identify the interesting class



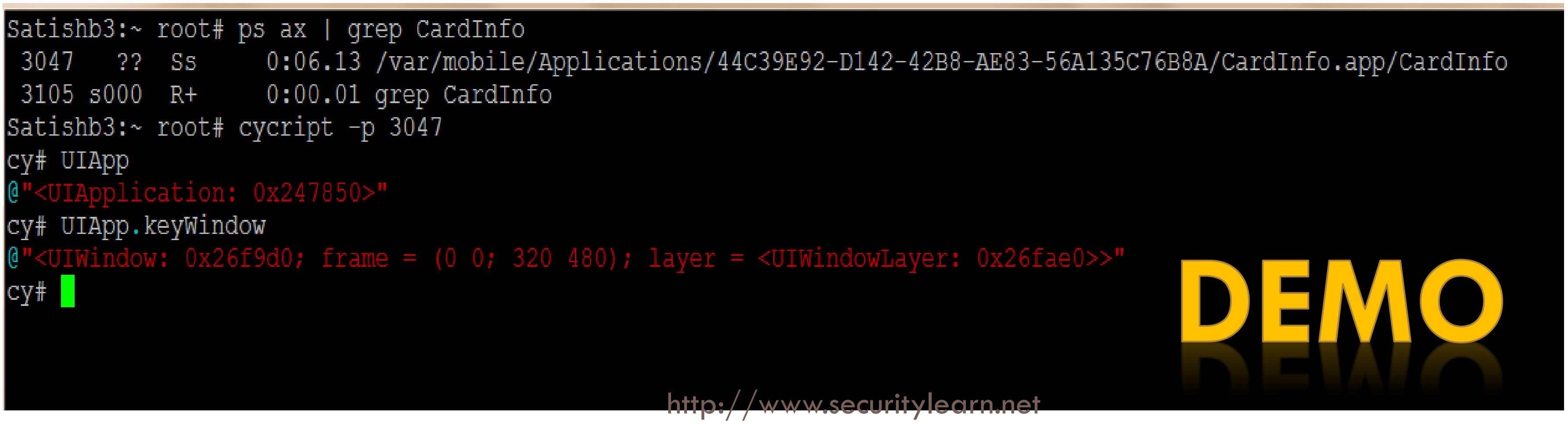
```
Satishb3:~ root# class-dump-z /var/mobile/Applications/44C39E92-D142-42B8-AE83-56A135C76B8A/CardInfo.app/CardInfo
/**
 * This header is generated by class-dump-z 0.2-0.
 * class-dump-z is Copyright (C) 2009 by KennyTM~, licensed under GPLv3.
 *
 * Source: (null)
 */

typedef struct _NSZone NSZone;

typedef struct CGPoint {
```



```
@interface AppDelegate : UIResponder <UIApplicationDelegate> {
@private
        UINavigationController* nvController;
        UIWindow* _window;
}
@property(retain, nonatomic) UIWindow* window;
-(void)applicationWillTerminate:(id)application;
-(void)applicationDidBecomeActive:(id)application;
-(void)applicationWillEnterForeground:(id)application;
-(void)applicationDidEnterBackground:(id)application;
-(void)applicationWillResignActive:(id)application;
-(BOOL)application:(id)application didFinishLaunchingWithOptions:(id)options;
-(void)dealloc;
```

Runtime Analysis

- App runtime can be easily modified using Cycript (Cydia pkg)
- Combination of JavaScript and Objective-C interpreter
- Can be hooked to a running process (like GDB)
- Gives access to all classes and instance variables within the app
- Existing methods can be overwritten easily
- Create object for the class and directly access the instance variables and invoke methods



```
Satishb3:~ root# ps ax | grep CardInfo
 3047   ??  Ss     0:06.13 /var/mobile/Applications/44C39E92-D142-42B8-AE83-56A135C76B8A/CardInfo.app/CardInfo
 3105 s000  R+     0:00.01 grep CardInfo
Satishb3:~ root# cycript -p 3047
cy# UIApp
@"<UIApplication: 0x247850>"
cy# UIApp.keyWindow
@"<UIWindow: 0x26f9d0; frame = (0 0; 320 480); layer = <UIWindowLayer: 0x26fae0>>"
cy#
```

Runtime Analysis

- Possible attacks with Cycript
  - Authentication bypass
  - Breaking simple locks
  - Bypassing restrictions that stops apps from running on Jailbroken device
  - Extract hardcode encryption keys
  - Extract app passcodes
  - Malicious code injection

- Do not store encryption keys / passcode in memory
- Implement code that restricts debugger attachment

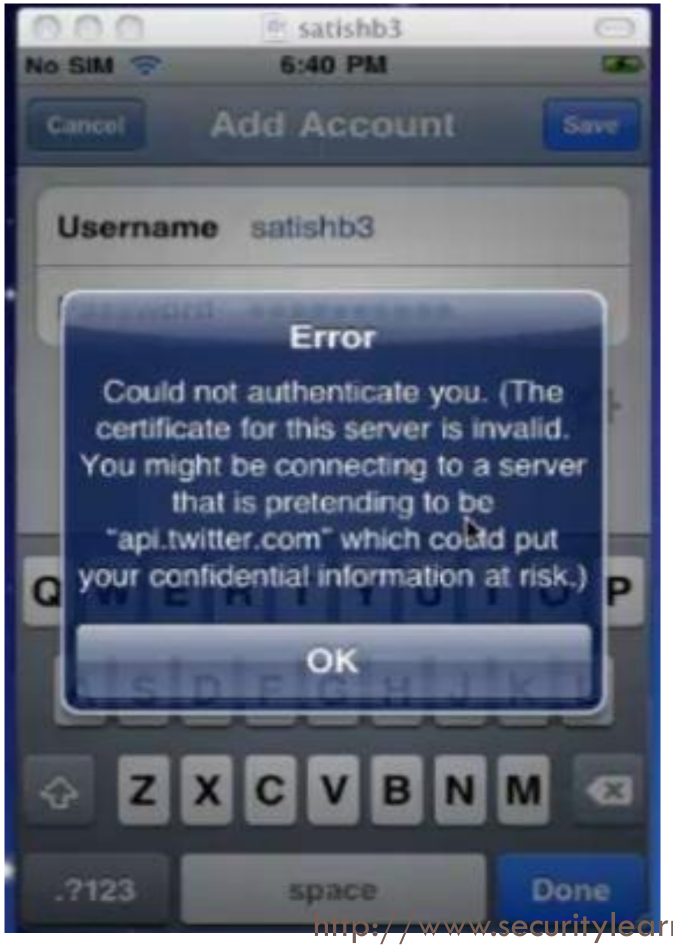
Transport Security

Transport Security

- iOS apps use SSL/https to do secure transactions
- NSURLRequest / NSURLConnection are commonly used
- CFNetwork – Alternate low level framework to implement SSL
- Frameworks by default rejects the self signed certificates to prevent MITM attacks
- Provides API to accept any un-trusted certificate
- NSURLRequest
  - setAllowsAnyHTTPSCertificate
- NSURLConnection delegate
  - continueWithoutCredentialForAuthenticationChallenge
- CFNetwork
  - kCFStreamSSLAllowsExpiredCertificates …

Transport Security

- DO not deploy iOS applications with cert validation bypass code

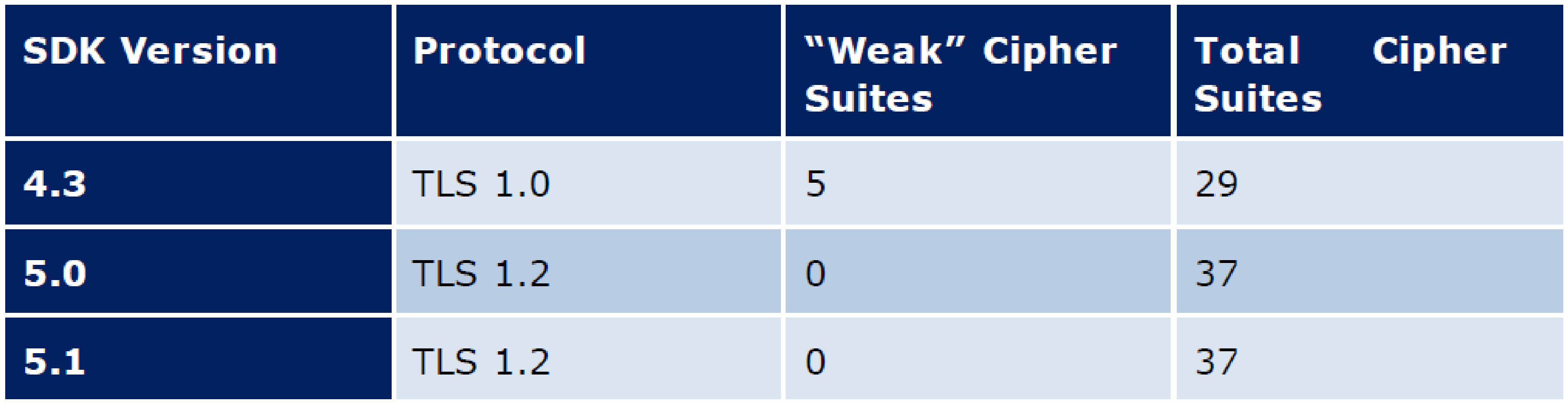


Error

Could not authenticate you. (The certificate for this server is invalid. You might be connecting to a server that is pretending to be "api.twitter.com" which could put your confidential information at risk.)

Transport Security

- API uses a default set of ciphers to setup a connection
- Does not provide an option to choose the cipher
- Apps can built with embedded SSL libraries
  - MatrixSSL, yaSSL
- Apps compiled with latest SDK (>5) does not support weak ciphers



SDK Version	Protocol	"Weak" Cipher Suites	Total Cipher Suites
4.3	TLS 1.0	5	29
5.0	TLS 1.2	0	37
5.1	TLS 1.2	0	37

Image from iOS Application In-Security paper by MDSec

Thank You

Email: Satishb3@hotmail.com

Twitter: @Satishb3