GAME of PWNS

Advanced iPhone pen-testing with iNalyzer framework

Chilik Tamir – Chief Scientist, appsec-labs
OWASP IL 2013
This presentation will demonstrate a new approach and tool to perform practical black box testing on any iOS application.

These demos will be illustrated using technical terms and tools of trade that relates to black-box effort on iOS applications.

If terms such as: ObjC, Class-Dump-z, Cycript, Clutch, Proxies, Scanners, etc. make you want to WTF it, please see the reference slides at the end of the presentation to upgrade your knowledge.

Or you can use the exit door to select a different track 😊
About me

◦ Security Researcher, Trainer, Speaker
  • Pervious Publications:
    • Lenovo privilege escalation WiFi driver
    • SOAP patch for Sqlmap
    • Belch – Burp suite plugin for binary protocols (AMF, Jser, etc.)
    • EvilQR open Research
    • AppUse - Android Application Uniform Security Evaluation Platform (Developed with Erez Metula)
    • Trainings: Black Hat USA, Intel, Cisco, HP, Amdocs and others

◦ B.Sc. Biomedical Engineering
Agenda

- “Pain Testing” iOS
- iNalyzer This?
- Advanced Pwnage
Recap: What’s an iOS App?

- ObjC/C/C++ Compiled (ARM) Executable
- Encrypted Executable (fairplay)
- Self contained under ~/Applications/GUID/AppName.app folder
- Installed by “mobile” user
- Executes under sandbox
- Under the radar can escape (SpyPhone, Storm8, etc.)
iOS App: Common Vulnerabilities

Source: www.owasp.org

OWASP Mobile Top 10 Risks

- M1 – Insecure Data Storage
- M2 – Weak Server Side Controls
- M3 – Insufficient Transport Layer Protection
- M4 – Client Side Injection
- M5 – Poor Authorization and Authentication
- M6 – Improper Session Handling
- M7 – Security Decisions Via Untrusted Inputs
- M8 – Side Channel Data Leakage
- M9 – Broken Cryptography
- M10 – Sensitive Information Disclosure

OWASP IL 2013
iOS “Pain-Testing”: typical approach

- **Binary:**
  - Decryption (Clutch)
  - Class identification (class-dump-z)
  - Reversing (IDA)
  - Patching (when needed)

- **Application Runtime:**
  - FileSystem, KeyChain, DB, Logs
  - Theos / Logos Tweaks
  - GDB
  - Cycript

- **Network (Proxy, Mallory)**
“Pain testing” iOS Apps: Cons

- No Code
- No Simulator
- Encrypted by iTunes
- Unknown end points
- % of Functionality Coverage
- Hidden vulnerabilities
Agony++ Binary Protocols

POST /aap.do HTTP/1.1
Host: data.flurry.com
Proxy-Connection: keep-alive
Accept-Encoding: gzip, deflate
Content-Type: application/octet-stream
Accept-Language: en-us
Accept: */*
Pragma: no-cache
Connection: keep-alive
User-Agent: QikSkype/6.7.125 CFNetwork/509.1.4 Darwin/13.0.0

I KNOW

BINARY CODE!

OWASP IL 2013
Agony++ unknown peers
Single Stepping

Everything drills down to ObjC_msgSend()
Not enough Time
 Doesn’t play well with other tools
The Business Axiom

ONE DOES NOT SIMPLY SUBMIT

AN EMPTY PEN-TEST REPORT TO A CUSTOMER

owasp-sec IL 2013

AppSec Labs
iNalyzer 5.6.0b

- **Static Analysis:**
  - Storyboard
  - SQL Queries
  - External Protocols
  - Embedded Strings
  - All Classes
  - All Objects
  - All Methods
  - All Parameters

- **Dynamic Runtime Manipulation:**
  - Variable Tampering
  - Constants Tampering
  - Methods Tampering
  - Live Attachment
  - Memory enumeration
  - Memory overwrite
  - Scriptable
  - Expandable
Objective C class interposing

What should be the result of running this code:

```objective-c
NSString* ErrorMsg = [NSString stringWithFormat:@"Access Denied" ];
NSString* ErrorMsg = [NSString stringWithFormat:@"Hello" ];
NSString* ErrorMsg = [NSString stringWithFormat:@"What Happened?" ];
```
Objective C class interposing

Presenting a new implementation to a foundation class selector:

```objective-c
NSString->isa.messages["stringWithString:"
    function(a){
        return "HackedAccount" }
```

Running this code:

```objective-c
ErrorMsg = [NSString stringWithFormat:@"Access Denied"
        @"HackedAccount"
ErrorMsg = [NSString stringWithFormat:@"Hello"
        @"HackedAccount"
ErrorMsg = [NSString stringWithFormat:@"What Happend?"
        @"HackedAccount"
```
Cycrypt: Tampering tool
Solving Binary Protocols

```javascript
asd=[UIDevice currentDevice]
asd->isa.messages["uniqueIdentifier"] = function(){ return "OR '1'='1' --"; }
```
Storyboard Collection

Demo

OWASP IL 2013
Storyboard Failure

WhatsApp.app

Debug Menu Enabled
In the Production APP
iNalyzer Dashboard

Dropbox.app

Classes

Main Page Related Pages Classes Files

Class List Class Index Class Hierarchy Class Members

displaySystemPreferences correctConnector fileExceedsWWANLimitAlert handleCameraUploadTourDismissal handleLocalNotification handleRemoteNotification importControllerDidImportToPath importControllerDidSelectFolder importControllerWillCancel animated importFolderChooserDidCancel interfaceOrientation isPasscodeBackgroundDisplayed passcodeBackground passcodeParentController passcodeViewControllersDidReceiveCoder passcodeViewControllerPasscodeRet popoverControllerDidDismissPopover popoverControllerShouldDismissPopover presentAccountInfoViewIfNeeded presentBackgroundUploadingTourAt presentCameraUploadTour presentConnectControllerInitial presentCurrentSharingViewController presentLockoutAlertView presentModalNavController:animate presentModalNavController:fromController

<UIApplicationDelegate>

passcodeController

PasscodeBackgroundController passcodeBackground

setupView

DBSplitViewController

splitViewController

notAlertView

padSetupController

DropboxAppDelegate

OWASP IL 2013
Strings analysis

Analysis of Strings found in the executable

SQL Strings

1 11699 INSERT INTO asset_ids VALUES(?);
2 11700 INSERT INTO cache_index VALUES(?, ?, ?)
3 11701 INSERT INTO data_cache VALUES(‘%@’, ‘%@’)
4 11702 INSERT OR IGNORE INTO urls VALUES(?)
5 13642 SELECT SUM(file_size) FROM cache_index
6 13643 SELECT asset_id FROM asset_ids WHERE asset_id = ?;
7 13644 SELECT data FROM data_cache WHERE key = ‘%@’
8 13645 SELECT hash FROM hashes WHERE hash = ?;
9 13646 SELECT id, access_token FROM test_account WHERE app_id = ‘%a’
10 13647 SELECT key FROM data_cache WHERE key = ‘%@’
11 13648 SELECT name FROM sqlite_master WHERE type = ‘table’ AND name = ‘hashes’;
12 13649 SELECT uid, name FROM user WHERE uid IN (SELECT id FROM #test_accounts)
13 13650 SELECT url FROM urls WHERE url = ?
14 13651 SELECT uuid, key, access_time, file_size FROM cache_index WHERE key = ?
Command line Arguments?

waze://?%6f%70%65%6e%5f%75%72%6c%3d%68%74%74%70%3a%2f%2f%74%72%2e%69%6d%34%36%35%77%35
Demo: Expanding iNalyzer with Burp

GET /Dropbox/Invoke=?EdPasscodeViewController%20passcodeEquals:@221200%22%20%5D%5D
Host: coredumps-iphone.local:5544
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.6; rv:19.0) Gecko/20100101 Firefox/19.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Origin: null
Connection: keep-alive

OWASP IL 2013
Payload Position

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for details.

<table>
<thead>
<tr>
<th>Request</th>
<th>Payload</th>
<th>Status</th>
<th>Error</th>
<th>Timeout</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>1224</td>
<td>200</td>
<td>✔️</td>
<td>✔️</td>
<td>153</td>
</tr>
<tr>
<td>26</td>
<td>1225</td>
<td>200</td>
<td>✔️</td>
<td>✔️</td>
<td>153</td>
</tr>
<tr>
<td>27</td>
<td>1226</td>
<td>200</td>
<td>✔️</td>
<td>✔️</td>
<td>153</td>
</tr>
<tr>
<td>28</td>
<td>1227</td>
<td>200</td>
<td>✔️</td>
<td>✔️</td>
<td>153</td>
</tr>
<tr>
<td>29</td>
<td>1228</td>
<td>200</td>
<td>✔️</td>
<td>✔️</td>
<td>153</td>
</tr>
<tr>
<td>30</td>
<td>1229</td>
<td>200</td>
<td>✔️</td>
<td>✔️</td>
<td>153</td>
</tr>
<tr>
<td>31</td>
<td>1230</td>
<td>200</td>
<td>✔️</td>
<td>✔️</td>
<td>153</td>
</tr>
<tr>
<td>32</td>
<td>1231</td>
<td>200</td>
<td>✔️</td>
<td>✔️</td>
<td>153</td>
</tr>
<tr>
<td>33</td>
<td>1232</td>
<td>200</td>
<td>✔️</td>
<td>✔️</td>
<td>153</td>
</tr>
<tr>
<td>34</td>
<td>1233</td>
<td>200</td>
<td>✔️</td>
<td>✔️</td>
<td>153</td>
</tr>
<tr>
<td>35</td>
<td>1234</td>
<td>200</td>
<td>✔️</td>
<td>✔️</td>
<td>153</td>
</tr>
<tr>
<td>36</td>
<td>1235</td>
<td>200</td>
<td>✔️</td>
<td>✔️</td>
<td>153</td>
</tr>
<tr>
<td>37</td>
<td>1236</td>
<td>200</td>
<td>✔️</td>
<td>✔️</td>
<td>153</td>
</tr>
<tr>
<td>38</td>
<td>1237</td>
<td>200</td>
<td>✔️</td>
<td>✔️</td>
<td>153</td>
</tr>
</tbody>
</table>

Request: /Dropbox/Invoke%5BPasscodeViewController%20passcodeEquals:0%22%5D%20EndInvoke HTTP/1.0
Host: coredumps-iphone.local:5544
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.6; rv:19.0) Gecko/20100101 Firefox/19.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Origin: null
Connection: keep-alive

OWASP IL 2013
iNalyzer & Burp Vs. Mailbox
Wishful thinking: IDA Remote Debugging Obsolete
Demo Single Stepping
iNalyzer 5.6.0b: The Recipe

1. Jail-borken device (@evad3rs)
2. Clutch to decrypt app (ttwj)
3. Class-dump-Z to app prototypes (@kennytm)
4. Doxygen engine to render a Dashboard (@doxygen)
5. FireFox to run the Dashboard (@firefox)
6. Cycript to modify the app behavior (@saurik)
7. Repeat step 6 until completed
8. SubjectiveC to log selectors (@kennytm)
Getting iNalyzer

Repository url: http://appsec-labs.com/cydia
Starting iNalyzer

After restart open browser to http://<you iDevice IP>:5544
Packaging an App

iNalyzer Packager

How to use:

1. Install GraphViz-Dot on PC/Laptop
2. Install DoxyGen on PC/Laptop
3. Choose Application from the list and click Package

Choose application to Pack: Dropbox

Package

Be patient as package creation can take a while

4. Save .zip to disk and extract
5. Run Setup.bat(Win) or Setup.sh(Other)
Download package
Dashboard Building

In the Payload/Appname.app/Doxygen/ folder:

Execute the doxMe.sh file (Mac)
Open dox.Template with DoxyGen (Win)
Open Live Demo (as time permits):
Summary

- iOS Pain-testing, just got easier 😊
- For exclusive alpha and beta releases please come to our booth.
- Mobile PT requires Mobile understanding
- Join our mobile application security hands-on training
  - iOS and Android Mobile Hacking (TBD, info@appsec-labs.com)
  - Mobile Secure Coding (TBD, info@appsec-labs.com)
  - Mobile Awareness (TBD, info@appsec-labs.com)
Questions ?
Thank You
References:

- ObjC interposing – http://culater.net/wiki/moin.cgi/CocoaReverseEngineering
- Clutch – https://github.com/ttwj/ClutchMod
- Class-dump-z – https://github.com/kennytm/Miscellaneous/downloads
- Cycript – http://www.cycript.org/
- Burp – http://www.portswigger.net/burp/download.html
Game of Pwns: Advanced iPhone testing with iNalyzer framework
https://appsec-labs.com/iNalyzer

Chilik Tamir
Chief Scientist

@_coreDump
chilik <at> appsec-labs.com
www.appsec-labs.com