# A Peek Under the Hood of iOS Malware

Laura García[†], **Ricardo J. Rodríguez**[‡]

laura@mlw.re, rjrodriguez@ieee.org



[†]MLW.RE NPO, Spain

[‡]Second University of Naples, Caserta, Italy
University of Zaragoza, Spain

September 02, 2016

**1st International Workshop on Malware Analysis**
Salzburg, Austria

# Introduction (I)

- Malicious software targeting mobile devices are increasing
  - Expected, given the increasing trend of these devices (7.2B in use)
  - Criminal though: *the more potential victims, the more revenue*
- Mobile market is mainly dominated by `Android` (82.8%) and `iOS` (13.9%)
- Same trend is followed by threats:
  - 5000 new Android malware files/day were found in 2015 (Kaspersky)
- Consequence: Android malware largely studied in the literature

Universidad Zaragoza 1542

# Introduction (I)

- Malicious software targeting mobile devices are increasing
  - Expected, given the increasing trend of these devices (7.2B in use)
  - Criminal though: *the more potential victims, the more revenue*
- Mobile market is mainly dominated by `Android` (82.8%) and `iOS` (13.9%)
- Same trend is followed by threats:
  - 5000 new Android malware files/day were found in 2015 (Kaspersky)
- Consequence: Android malware largely studied in the literature

**What about `iOS` malware?**

Universidad Zaragoza

# Introduction (II)

## Lack of attention

1. Market share: `Android` is preferred instead of `iOS` as malware deployment platform
2. Different security models, making Android more exposed
   - Both follow permission-based approaches (different granularity)
   - Both have native OS security mechanisms (as DEP, ASLR)
   - `Android` mainly relies on platform protection, while `iOS` relies on market protection

# Introduction (II)

## Lack of attention

1. Market share: Android is preferred instead of iOS as malware deployment platform
2. Different security models, making Android more exposed
   - Both follow permission-based approaches (different granularity)
   - Both have native OS security mechanisms (as DEP, ASLR)
   - Android mainly relies on platform protection, while iOS relies on market protection

## Some insights on iOS market protection

- Review/vetting process for any app to be published in App Store (official Apple market)
- Not totally effective
  - Example: `XCodeGhost`, a trojanized SDK, infected (at least) 39 apps
  - Other attack vectors also possible: private APIs, compromised iCloud accounts

# Introduction (III)

## Contributions

- Classification of 36 iOS malware families, from 2009 to 2015, according to:
  - Affected devices
  - Distribution channel
  - Infection method
  - Attack goal
  - Attack vector
- iOS malware analysis methodology
  - In-depth analysis of a selected sample (from `KeyRaider` family)

Universidad Zaragoza 1542

# Introduction (III)

## Contributions

- Classification of 36 iOS malware families, from 2009 to 2015, according to:
  - Affected devices
  - Distribution channel
  - Infection method
  - Attack goal
  - Attack vector
- iOS malware analysis methodology
  - In-depth analysis of a selected sample (from `KeyRaider` family)

## Take-home conclusions

- Most of iOS malware are distributed out of official markets

# Introduction (III)

## Contributions

- Classification of 36 iOS malware families, from 2009 to 2015, according to:
    - Affected devices
    - Distribution channel
    - Infection method
    - Attack goal
    - Attack vector
- iOS malware analysis methodology
    - In-depth analysis of a selected sample (from `KeyRaider` family)

## Take-home conclusions

- Most of iOS malware are distributed out of official markets
- Jailbreaking an iOS device makes it a potential target

# Introduction (III)

## Contributions

- Classification of 36 `iOS` malware families, from 2009 to 2015, according to:
  - Affected devices
  - Distribution channel
  - Infection method
  - Attack goal
  - Attack vector
- `iOS` malware analysis methodology
  - In-depth analysis of a selected sample (from `KeyRaider` family)

## Take-home conclusions

- Most of `iOS` malware are distributed out of official markets
- Jailbreaking an `iOS` device makes it a potential target
- User interaction is (somehow) required as first infection stage

# Related Work (I)

## Mainly focused on Android

- Smartphone malware taxonomy (Amamra et al., 2012)
  - Reference behaviour, analysis approach, and malware behaviour
- ANDRUBIS tool (Lindorfer et al., 2014)
- 46 mobile malware samples, from 2009 to 2011 (Felt et al., 2011)
  - Current and future incentives
  - 4 iOS samples (also covered in our study)
- Set of 1200 Android samples (Zhou and Jiang, 2012)
  - Attack type and installation method

Universidad Zaragoza 1542

# Related Work (II)

## Survey of mobile malware (Suárez-Tangil et al., 2014)

- Attack goals, malware behaviour, distribution and infection, privilege acquisition
- 9 samples target iOS devices
- Our approach is similar, but with some differences:
  - Distribution and infection are separate features
  - Attack vectors also considered

Universidad Zaragoza 1542

# Related Work (II)

## Survey of mobile malware (Suárez-Tangil et al., 2014)

- Attack goals, malware behaviour, distribution and infection, privilege acquisition
- 9 samples target iOS devices
- Our approach is similar, but with some differences:
  - Distribution and infection are separate features
  - Attack vectors also considered

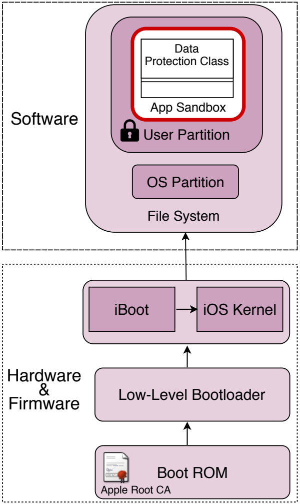
## iOS attack prevention

- PiOS: detect sensitive information exfiltration
- XiOS: prevents lazy bindings and abuse of private APIs
- iRiS: app vetting system that detect malicious behaviours
- Abuse of iOS sandboxing (Xing et al., CCS'15)

Universidad Zaragoza 1542

# Background (I)

## iOS security model




### Low level

- Hardware and firmware digitally signed and verified prior execution

### Application level

- Apple-issued certificate: all apps are signed
- Apps are isolated (sandbox)
- Data Protection: feature to protect data based on when it needs to be accessed
- Others: DEP, ASLR

# Background (II)

## Apple review/vetting process


Apple App Review Process (Simplified)

- Any app must comply with Apple Review Guidelines
  - Reliable, perform as expected, free of offensive material
- Set over 100 rules, covering different aspects (functionality, meta-data, advertising, etc.)

**Not 100% perfect**

- Trojanized SDK
- Obfuscation of private API calls
- Abuse of inter-app services

# Characterization of iOS malware (I)

- 36 malware families from 2009 to 2015
- Criteria: Who are targeting individuals?
  - On-sale malware: anyone
  - State-sponsored malware: governments, intelligent agencies, ...
  - Underground malware: criminals

# Characterization of iOS malware (I)

- 36 malware families from 2009 to 2015
- Criteria: Who are targeting individuals?
  - On-sale malware: anyone
  - State-sponsored malware: governments, intelligent agencies, …
  - Underground malware: criminals

| Malware family name(s) | Discovery date |
|---|---|
| ON-SALE MALWARE | |
| Trapsms | Jun 2009 |
| MobileSpy | Jul 2009 |
| OwnSpy | Feb 2010 |
| MobiStealth | Oct 2010 |
| FlexiSpy | Dec 2010 |
| iKeyGuard | April 2011 |
| Copy9 | Jul 2011 |
| StealthGenie | Nov 2011 |
| mSpy | Oct 2011 |
| iKeyMonitor | Mar 2012 |
| SpyKey | Apr 2012 |
| Copy10 | Aug 2012 |
| InnovaSPY | Sept 2012 |
| 1mole | Jan 2013 |
| Spy App | Oct 2014 |
| STATE-SPONSORED MALWARE | |
| FinSpy Mobile | Aug 2012 |
| Hacking Team tools | Jun 2014 |
| Inception/Cloud Atlas | Dec 2014 |
| XAgent/PawnStorm | Feb 2015 |

| Malware family name(s) | Discovery date |
|---|---|
| UNDERGROUND MALWARE | |
| Ikee/Eeki and Duh | Nov 2009 |
| LBTM | Sept 2010 |
| Find and Call | Jul 2012 |
| Nobitazzz (packages) | Aug 2012 |
| AdThief/Spad | Mar 2014 |
| SSLCreds/Unflod Baby Panda | Apr 2014 |
| AppBuyer | Sept 2014 |
| WireLurker | Nov 2014 |
| Xsser mRAT | Dec 2014 |
| Lock Saver Free | Jul 2015 |
| KeyRaider | Aug 2015 |
| XcodeGhost | Sept 2015 |
| YiSpecter | Oct 2015 |
| Muda/AdLord | Oct 2015 |
| Youmi Ad SDK | Oct 2015 |
| TinyV | Oct 2015 |
| SantaAPT | Dec 2015 |

# Characterization of iOS malware (II)

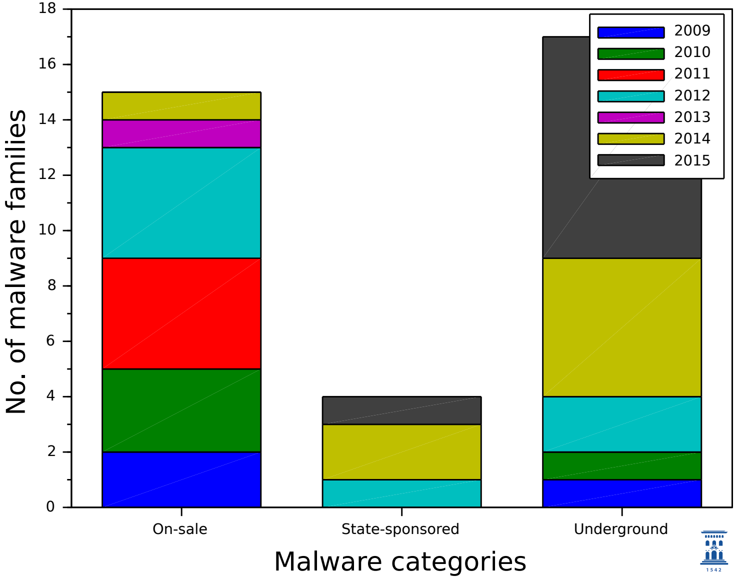
## Features of interest

- devices
  - non-jailbroken *(NJ)*
  - jailbroken *(JD)*
- distribution
  - official market *(OM)*
  - alternative market *(AM)*
  - unknown sources *(US)*
- infection
  - allowed by the user *(AS)*
  - exploit any vulnerability *(EV)*

- attack goals
  - spamming *(SM)*
  - ransom *(RS)*
  - data theft *(DT)*
  - fraud *(FR)*
  - spying *(SP)*
- attack vector
  - Apple-issued enterprise/developer certificates *(DC)*
  - bundle ID forged *(BF)*
  - private APIs *(PA)*
  - trojanized official SDK *(TO)*
  - Cydia Substrate *(CS)*
  - bypassed vetting process *(BV)*
  - compromised credentials *(CC)*

# Discussion (I)

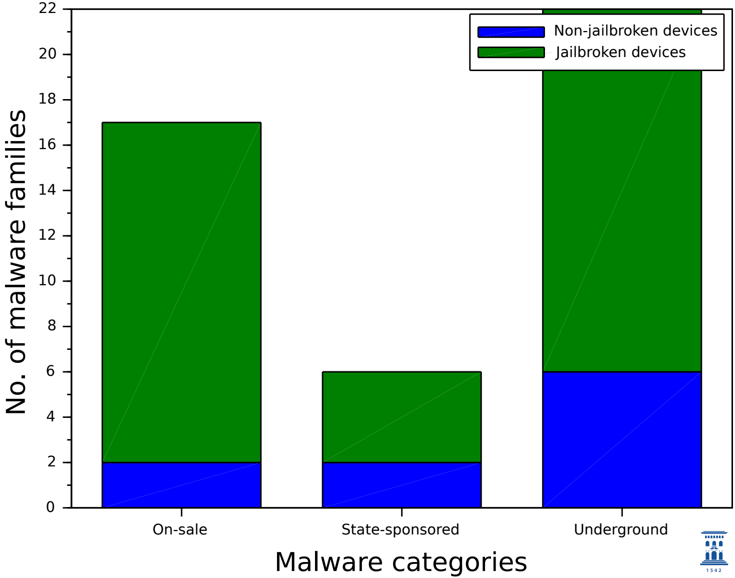
## On evolution




Universidad Zaragoza

# Discussion (II)

## On affected devices




Universidad Zaragoza

# Discussion (III)

## On distribution and infection

### Distribution

- On-sale & state-sponsored malware evenly distributed over *AM* & *US*
- $\approx$ 14% of underground malware distributed over *OM*
  - Only one family (`Ikee`, worm behaviour) comes from *US*, the rest come from *AM*
  - Recall that *AM* have no vetting process

Universidad Zaragoza

# Discussion (III)

## On distribution and infection

### Distribution

- On-sale & state-sponsored malware evenly distributed over *AM* & *US*
- $\approx$ 14% of underground malware distributed over *OM*
  - Only one family (`Ikee`, worm behaviour) comes from *US*, the rest come from *AM*
  - Recall that *AM* have no vetting process

### Infection

- Only 8.3% of malware families exploit any vulnerability (one state-sponsored, two underground)
  - Default passwords in jailbroken devices
  - Compromised enterprise/provisioning certificates
  - Masque attack (bundle ID forged)

# Discussion (IV)

## On attack goals and attack vectors

### Attack goals

- Spying and data theft: main goal of on-sale and state-sponsored malware (expected behaviour)
- Underground malware, more sparse ($\approx$ 65% present more than one goal):
    - $\approx$ 50% data thefts; $\approx$ 35% fraudsters
    - $\approx$ 24% spammers; $\approx$ 18% spying activities

# Discussion (IV)

## On attack goals and attack vectors

### Attack goals

- Spying and data theft: main goal of on-sale and state-sponsored malware (expected behaviour)
- Underground malware, more sparse ($\approx$ 65% present more than one goal):
  - $\approx$ 50% data thefts; $\approx$ 35% fraudsters
  - $\approx$ 24% spammers; $\approx$ 18% spying activities

### Attack vectors

- On-sale and state-sponsored malware, focused on Cydia Substrate
- Compromised credentials (two on-sale malware families)
- Other attack vectors used by state-sponsored malware:
  - Bundle ID forged
  - Misuse of enterprise/developer certificates
- Underground malware also use private APIs and trojanized SDK

# Discussion (V)

## Findings summary

- DO NOT jailbreak your iOS devices
- Keep them updated
- Install only from trusted sources
- Use native iOS mechanisms to grant/revoke permissions

Universidad Zaragoza

# Discussion (V)

## Findings summary

- DO NOT jailbreak your iOS devices
- Keep them updated
- Install only from trusted sources
- Use native iOS mechanisms to grant/revoke permissions

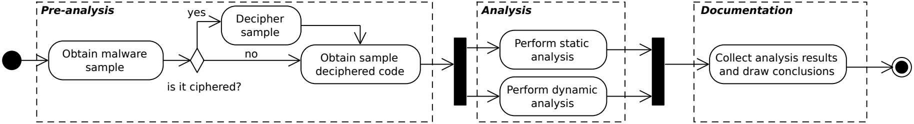

www.techrepublic.com/article/update-all-ios-devices-to-9-3-5-immediately-or-risk-a-remote-jailbreak/

Kindly reminder ⌣̈ (**update now!**)

Universidad Zaragoza

# Case Study: Analyzing a malware sample (I)

## Analysis methodology




- `KeyRaider` sample
  - MD5 hash: `8985ecbc80d257e02c1e30b0268d91e7`

**Samples available on the web (spread the love ♥):**

`http://webdiis.unizar.es/~ricardo/software-tools/supplementary-research-material/ios-malware-samples/`

Universidad Zaragoza 1542

# Case Study: Analyzing a malware sample (II)

## Pre-analysis stage

```
...
 Load command 10
           cmd LC_ENCRYPTION_INFO
       cmdsize 20
      cryptoff 16384
     cryptsize 835584
       cryptid 0
...
```

- otool command (option -l)
- *cryptid* $= 0 \rightarrow$ it's uncrypted

Universidad Zaragoza 1542

# Case Study: Analyzing a malware sample (II)

## Analysis stage

```
000b2c10  db "", 0
000b2c11  db "iPhone5,1", 0
000b2c1b  db "\xE4\xB8\xAD\xE5\x9B\xBD\xE8\x81\x94\xE9\x80\x9A",
000b2c28  db "8.1.2", 0
000b2c2e  db "11A470a", 0
000b2c36  db "GET /data.php?table=other&game=(game) HTTP/1.1\r\n"
000b2c67  db "Host: www.wushidou.cn\r\n\r\n", 0
000b2c81  db "(game)", 0
000b2c88  db "iappstore", 0
000b2c92  db "www.wushidou.cn", 0
000b2ca2  db "name", 0
000b2ca7  db "pass", 0
000b2cac  db "pod", 0
```

### Static analysis

- `strings` reveals a web domain
  - `www.wushidou.cn`
  - Resolves to localhost!

Universidad Zaragoza 1542

# Case Study: Analyzing a malware sample (III)

## Analysis stage

```
bl      __Z14getProcessNamev             ; getProcessName()
movw    r1, #0xf754                      ; "itunesstored", :lower16:(0xb405e - 0x1490a)
mov     r0, r4
movt    r1, #0x9                         ; "itunesstored", :upper16:(0xb405e - 0x1490a)
movs    r2, #0x0
add     r1, pc                           ; "itunesstored"
---------------------------------------------------------------------------
movw    r1, #0xf51d                      ; "/System/Library/Frameworks/Security.framewo
movs    r0, #0x4
movt    r1, #0x9                         ; "/System/Library/Frameworks/Security.framewo
str     r0, [sp, #0x78 + var_48]
---------------------------------------------------------------------------
movw    r1, #0xed9d                      ; :lower16:(0x1378d - 0x149f0)
movs    r0, #0x9
movt    r1, #0xffff                      ; :upper16:(0x1378d - 0x149f0)
movw    r2, #0xf2a6                      ; :lower16:(_o_SSLWrite - 0x149f2)
movt    r2, #0xc                         ; :upper16:(_o_SSLWrite - 0x149f2)
add     r1, pc                           ; 0x1378d
add     r2, pc                           ; _o_SSLWrite
str     r0, [sp, #0x78 + var_48]
ldr     r0, [sp, #0x78 + var_74]
blx     imp___symbolstub1__MSHookFunction
```

### Dynamic analysis

- Uses Mobile Substrate Framework to hook *SSLRead*, *SSLWrite* functions in `itunesstored`
  - Enables it to get username, password, and device GUID very easily
  - Once retrieved, exfiltrated to the C&C server

# Case Study: Analyzing a malware sample (IV)

## Analysis stage

```
blx      imp___symbolstub1___Unwind_SjLj_Register
movw     r2, #0xba71                                    ; "POST /WebObjects/MZFinance.woa/wa/authenticate HTTP/
movt     r2, #0xa                                       ; "POST /WebObjects/MZFinance.woa/wa/authenticate HTTP/
add      r2, pc                                         ; "POST /WebObjects/MZFinance.woa/wa/authenticate HTTP/
addw     r0, sp, #0x624
add.w    r1, sp, #0x6d8
blx      imp___symbolstub1___ZNSt3__1plIcNS_11char_traitsIcEENS_9allocatorIcEEEENS_12basic_stringIT_T0_T1_EERKS9_PKS
movw     r2, #0xba8f                                    ; "Host: p(pod)-buy.itunes.apple.com\\r\\n", :lower16:(
movs     r0, #0x2
movt     r2, #0xa                                       ; "Host: p(pod)-buy.itunes.apple.com\\r\\n", :upper16:(
str.w    r0, [sp, #0x738 + var_48]
add      r2, pc                                         ; "Host: p(pod)-buy.itunes.apple.com\\r\\n"
add.w    r0, sp, #0x630
addw     r1, sp, #0x624
blx      imp___symbolstub1___ZNSt3__1plIcNS_11char_traitsIcEENS_9allocatorIcEEEENS_12basic_stringIT_T0_T1_EERKS9_PKS
movw     r2, #0xba97                                    ; "User-Agent: AppStore/2.0 iOS/(os) model/(phone) (4;
movs     r0, #0x3
movt     r2, #0xa                                       ; "User-Agent: AppStore/2.0 iOS/(os) model/(phone) (4;
str.w    r0, [sp, #0x738 + var_48]
add      r2, pc                                         ; "User-Agent: AppStore/2.0 iOS/(os) model/(phone) (4;
addw     r0, sp, #0x63c
add.w    r1, sp, #0x630
blx      imp___symbolstub1___ZNSt3__1plIcNS_11char_traitsIcEENS_9allocatorIcEEEENS_12basic_stringIT_T0_T1_EERKS9_PKS
```

### Dynamic analysis

- Emulate App Store login protocol with compromised accounts

Universidad Zaragoza

# Case Study: Analyzing a malware sample (V)

## Analysis stage

```
blx     imp___symbolstub1___Unwind_SjLj_Register
movw    r2, #0xa470                          ; "POST /WebObjects/MZBuy.woa/wa/buyProduct HTTP/1.1\\r
movt    r2, #0xa                             ; "POST /WebObjects/MZBuy.woa/wa/buyProduct HTTP/1.1\\r
add     r2, pc                               ; "POST /WebObjects/MZBuy.woa/wa/buyProduct HTTP/1.1\\r
add.w   r0, sp, #0x488
add.w   r1, sp, #0x578
blx     imp___symbolstub1___ZNSt3__1plIcNS_11char_traitsIcEENS_9allocatorIcEEEENS_12basic_stringIT_T0_T1_EERKS9_PKS
movw    r2, #0x9ef5                          ; "Host: p(pod)-buy.itunes.apple.com\\r\\n", :lower16:(
movs    r0, #0x2
movt    r2, #0xa                             ; "Host: p(pod)-buy.itunes.apple.com\\r\\n", :upper16:(
str.w   r0, [sp, #0x5d8 + var_48]
add     r2, pc                               ; "Host: p(pod)-buy.itunes.apple.com\\r\\n"
addw    r0, sp, #0x494
add.w   r1, sp, #0x488
blx     imp___symbolstub1___ZNSt3__1plIcNS_11char_traitsIcEENS_9allocatorIcEEEENS_12basic_stringIT_T0_T1_EERKS9_PKS
movw    r2, #0x9efd                          ; "User-Agent: AppStore/2.0 iOS/(os) model/(phone) (4;
movs    r0, #0x3
movt    r2, #0xa                             ; "User-Agent: AppStore/2.0 iOS/(os) model/(phone) (4;
str.w   r0, [sp, #0x5d8 + var_48]
add     r2, pc                               ; "User-Agent: AppStore/2.0 iOS/(os) model/(phone) (4;
add.w   r0, sp, #0x4a0
addw    r1, sp, #0x494
blx     imp___symbolstub1___ZNSt3__1plIcNS_11char_traitsIcEENS_9allocatorIcEEEENS_12basic_stringIT_T0_T1_EERKS9_PKS
movw    r2, #0x9f1e                          ; "Accept: */*\\r\\n", :lower16:(0xb2e90 - 0x8f72)
movs    r0, #0x4
movt    r2, #0xa                             ; "Accept: */*\\r\\n", :upper16:(0xb2e90 - 0x8f72)
str.w   r0, [sp, #0x5d8 + var_48]
add     r2, pc                               ; "Accept: */*\\r\\n"
addw    r0, sp, #0x4ac
add.w   r1, sp, #0x4a0
```

### Dynamic analysis

- Forge purchases requests

# Conclusions and Future Work

## Conclusions

- Classification of 36 iOS malware families, from 2009 to 2015
  - Affected devices, distribution channels, infection, attack goals, and attack vector
- Methodology for iOS malware analysis
  - Same as PC malware analysis

# Conclusions and Future Work

## Conclusions

- Classification of 36 iOS malware families, from 2009 to 2015
  - Affected devices, distribution channels, infection, attack goals, and attack vector
- Methodology for iOS malware analysis
  - Same as PC malware analysis

## Take-home conclusions

- Few of them target non-jailbroken devices

Universidad Zaragoza 1542

# Conclusions and Future Work

## Conclusions

- Classification of 36 iOS malware families, from 2009 to 2015
  - Affected devices, distribution channels, infection, attack goals, and attack vector
- Methodology for iOS malware analysis
  - Same as PC malware analysis

## Take-home conclusions

- Few of them target non-jailbroken devices
- Few of them exploit iOS vulnerabilities

Universidad Zaragoza

# Conclusions and Future Work

## Conclusions

- Classification of 36 `iOS` malware families, from 2009 to 2015
  - Affected devices, distribution channels, infection, attack goals, and attack vector
- Methodology for `iOS` malware analysis
  - Same as PC malware analysis

## Take-home conclusions

- Few of them target non-jailbroken devices
- Few of them exploit `iOS` vulnerabilities
- Data theft and spying are most common goals

# Conclusions and Future Work

## Conclusions

- Classification of 36 iOS malware families, from 2009 to 2015
  - Affected devices, distribution channels, infection, attack goals, and attack vector
- Methodology for iOS malware analysis
  - Same as PC malware analysis

## Take-home conclusions

- Few of them target non-jailbroken devices
- Few of them exploit iOS vulnerabilities
- Data theft and spying are most common goals

Universidad Zaragoza 1542

# Conclusions and Future Work

## Conclusions

- Classification of 36 `iOS` malware families, from 2009 to 2015
  - Affected devices, distribution channels, infection, attack goals, and attack vector
- Methodology for `iOS` malware analysis
  - Same as PC malware analysis

## Take-home conclusions

- Few of them target non-jailbroken devices
- Few of them exploit `iOS` vulnerabilities
- Data theft and spying are most common goals

## Future work

- Analyze (in-depth) more samples
  - Identify the underlying attack concepts (get fingerprints for detection)
  - Develop a `iOS` malware detection tool

# A Peek Under the Hood of iOS Malware


Laura García[†], **Ricardo J. Rodríguez**[‡]

laura@mlw.re, rjrodriguez@ieee.org





[†]MLW.RE NPO, Spain

[‡]Second University of Naples, Caserta, Italy
University of Zaragoza, Spain


September 02, 2016

**1st International Workshop on Malware Analysis**
Salzburg, Austria