Mobile Security Attacks

A Glimpse From the Trenches

Yair Amit
CTO & Co-Founder
Skycure
@YairAmit

Adi Sharabani
CEO & Co-Founder
Skycure
@AdiSharabani
About the Presenters

Adi Sharabani
- CEO & co-founder of Skycure
- Watchfire's research and security group [Acquired by IBM]
- Led the security of much of IBM software
- Fellow at Yuval Ne’eman’s workshop
- Teacher at Ohel Shem high-school

Yair Amit
- CTO & co-founder of Skycure
- Former manager of the Application Security & Research group at IBM
- Web, network and mobile researcher
- Filed over 15 security patents

Seamless Mobile Security

Skycure

OWASP
Open Web Application Security Project
A Holistic Outlook on Mobile Security

- Physical Security
- Basic Threat
- Biggest Threat
- Emerging Threat
- Network
- Changing Threat
- Application Security & Privacy
- Malware
The Physical Layer
The Physical Layer

- Threat vector
  - Device lost / Device stolen / Temporary physical access
- Basic physical security needs:
  - Remote wipe
  - Locate device
  - Backup
  - Local storage
  - Passcode protection
- The above becomes OS responsibility
- MDM provides the above OS features together with management and policy enforcement
Network Based Attacks
Real World Incident Statistics

10.1% of scanned networks pose a threat

Based on over 40,000 scanned networks
Real World Incident Statistics

Affected Devices Over Time

Based on Skycure enabled devices worldwide
Implementation-Based Vulnerabilities Vs. Design-Based Vulnerabilities
Network Based Attacks
Implementation issues
Implementation-Based Vulnerabilities

iOS vs. Android
Example I:
gotofail
static OSStatus SSLVerifySignedServerKeyExchange(SSLContext *ctx, bool isRsa, SSLBuffer signedParams,
        uint8_t *signature, UInt16 signatureLen)
{
    ...
    if ((err = SSLHashSHA1.update(&hashCtx, &clientRandom)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
        goto fail;
    goto fail;
    if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
        goto fail;
    err = sslRawVerify(ctx,
            ctx->peerPubKey, /* plaintext */
            dataToSign,        /* plaintext length */
            dataToSignLen,
            signature,         
            signatureLen);
    ...
    fail:
    SSLFreeBuffer(&signedHashes);
    SSLFreeBuffer(&hashCtx);
    return err;
}
Example II:

Heartbleed
Network Based Attacks

Design issues
Design issues are much more interesting
   ... and much harder to fix
These are divided into two types:
   General “protocol” vulnerabilities
   Design issues affecting mobile OS
Mobile devices are more susceptible:
   Lack of adequate security solutions
   Excessive use of untrusted networks
Example I:

**sslstrip**

Attacker removes *redirections* and *links* to HTTPS

Server returns a redirection to HTTPS

Victim continues to interact via HTTP instead of HTTPs
Example II:

SSL decryption

92% of users click on “Continue” compromising their Exchange identity (username and password)
Example III:

Karma

Hak5’s WiFi Pineapple
Network Based Attacks

Mobile-specific design issues
iOS Security Model

### App Characteristics
- One Store
- Heavy Screening
- App Sandboxing

### Profile Characteristics
- No Store
- No Screening
- No Sandboxing

Source: Apple’s App Sandbox Design Guide
Where Do We Find Them?

- Mobile Device Management (MDM)
- Cellular carriers
  - Usually used for APN settings
- Mobile applications
- Service providers
Malicious Profiles

Configuration profiles can also be malicious

- Malicious “service providers” (apps/services/Wi-Fis/etc.)
- Vulnerable services
- Privacy violating services

Hacker gains access to your mail, business apps, cloud services, bank accounts and more, even if traffic is encrypted
Malicious Profiles

Going Viral

- Attacker hijacks victim’s key identities
  - Corporate Exchange
  - Facebook
  - LinkedIn
- Attacker sends mass messages to victim’s contacts, luring them to install the malicious profile
- Attack propagates
Profile listing could indicate suspicious profiles

Cat-and-mouse game: attackers can name their profile to look benign
Example II:

WiFiGate
App Level Security

- Physical Security
- Network
- Application Security & Privacy
- Malware
Mobile OS enforce additional security models
  - Sandbox
  - Better updates
  - Controlled application stores

App-level issues are now on the rise
App Vulnerabilities

Physical Security

Network

Malware

Application Security & Privacy
Example I:

Plain HTTP

Daaa!
Example II: Certificate Pinning
A Long Way to Go

- Almost all major apps today lack SSL Pinning
  - Susceptible to attacks such as malicious profiles by design
  - Also exploited when attacker gains access to a trusted CA
- Slow adoption should not come as a surprise
  - Implementation challenges
    - Less flexibility
    - Can become a nightmare if done wrong...
Example III:

HTTP Request Hijacking
Victim interacts with the malicious server.

A while later, victim opens the app.

Victim opens the app in an untrusted environment.

App logic has changed!

Attacker returns a 301 directive specifying a permanent change in URI.

App continues to connect to the malicious server!

Malicious server can return actual results from the target server.
Malicious Apps
Google’s Focus on Malware

- **2011**: The year of Android malware [1]


- **2013**: Malware is moving out of the Google Play [3]

- **2014**: Google adds full-time app scanning to address malware on external stores [4]

Android is becoming like iOS when it comes to malware
While OS anti-malware techniques advance, there are other similar problems (harder to address)
Malicious services sometimes try to justify their actions
- “I need all your key-strokes to provide you with a better service”

We are concerned about the Maliciously-Vulnerable app:
- App with semi-naive service is created
- App does not pose a privacy/security issue
- App is approved to go on AppStore/Google Play.
- App has a special crafted carefully thought vulnerability
- Vulnerability used as a backdoor to escalate app for malicious activity
Summary
Summary

- **The physical layer**
  - Becomes the OS responsibility

- **Network based attacks**
  - Implementation vulnerabilities
  - Design vulnerabilities
    - Generic vs. mobile specific

- **App level**
  - Vulnerabilities
    - HTTP/S, Certificate Pinning, HTTP Request Hijacking
  - The “maliciousness” axis
    - Malware ↔ Ad Networks ↔ Privacy Violations
Thank you!

- twitter: @YairAmit, @AdiSharabani
- email: {yair,adi} @skycure.com
- blog: http://www.skycure.com/blog