SECURING MOBILE APPLICATIONS

Ulf Larson

OWASP Göteborg

ulf.larson@owasp.org

Adecco
better work, better life

ulf.larson@adecco.se

A REAL WORLD EXAMPLE TO GET US STARTED

Google ClientLogin Authentication Protocol

https

un/pw

auth token

...

Google accounts authorization

Google services

A REAL WORLD EXAMPLE TO GET US STARTED

Google ClientLogin Authentication Protocol

https

un/pw

auth token

...

http

auth token

Google accounts authorization

Google services

A REAL WORLD EXAMPLE TO GET US STARTED

Google ClientLogin Authentication Protocol

https

un/pw

auth token

...

http

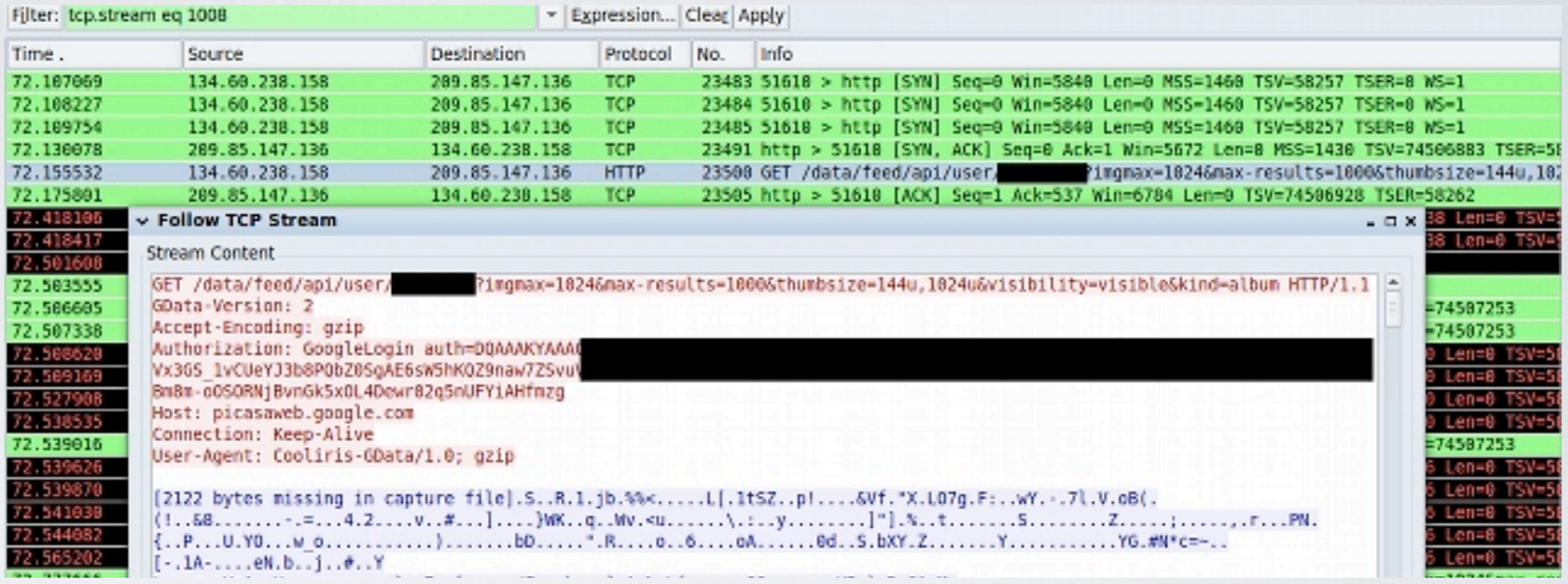
auth token

Google accounts authorization

Google services

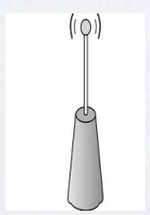
- Authorization header sent over HTTP
- When users connected via wifi, apps automatically sent the token in an attempt to automatically synchronize data from server
- Sniff this value, impersonate the user

A REAL WORLD EXAMPLE TO GET US STARTED



Evil twin

authToken over http



Free WiFi-hotspot

99% of all android phones were vulnerable

MOBILE DEVELOPMENT IS NO DIFFERENT FROM OTHER DEVELOPMENT

- Security (at the application layer) means being aware of how your code uses information and ensuring that it does so safely and responsibly

- Keep personal data safe from prying eyes
- Ensure that your software collects only the information that it requires.
- Prevent unauthorized access to or modification of the data while in transit.

SECURITY IS NOT A BULLET POINT ITEM

- The most important thing to understand about security is that **it is not a bullet point item**.
- Security cannot be bolted on at the end of the development process.
- You must **consciously design security into your app or service from the very beginning**, and make it a conscious part of the entire process from design through implementation, testing, and release.

SECURITY IS NOT A BULLET POINT ITEM

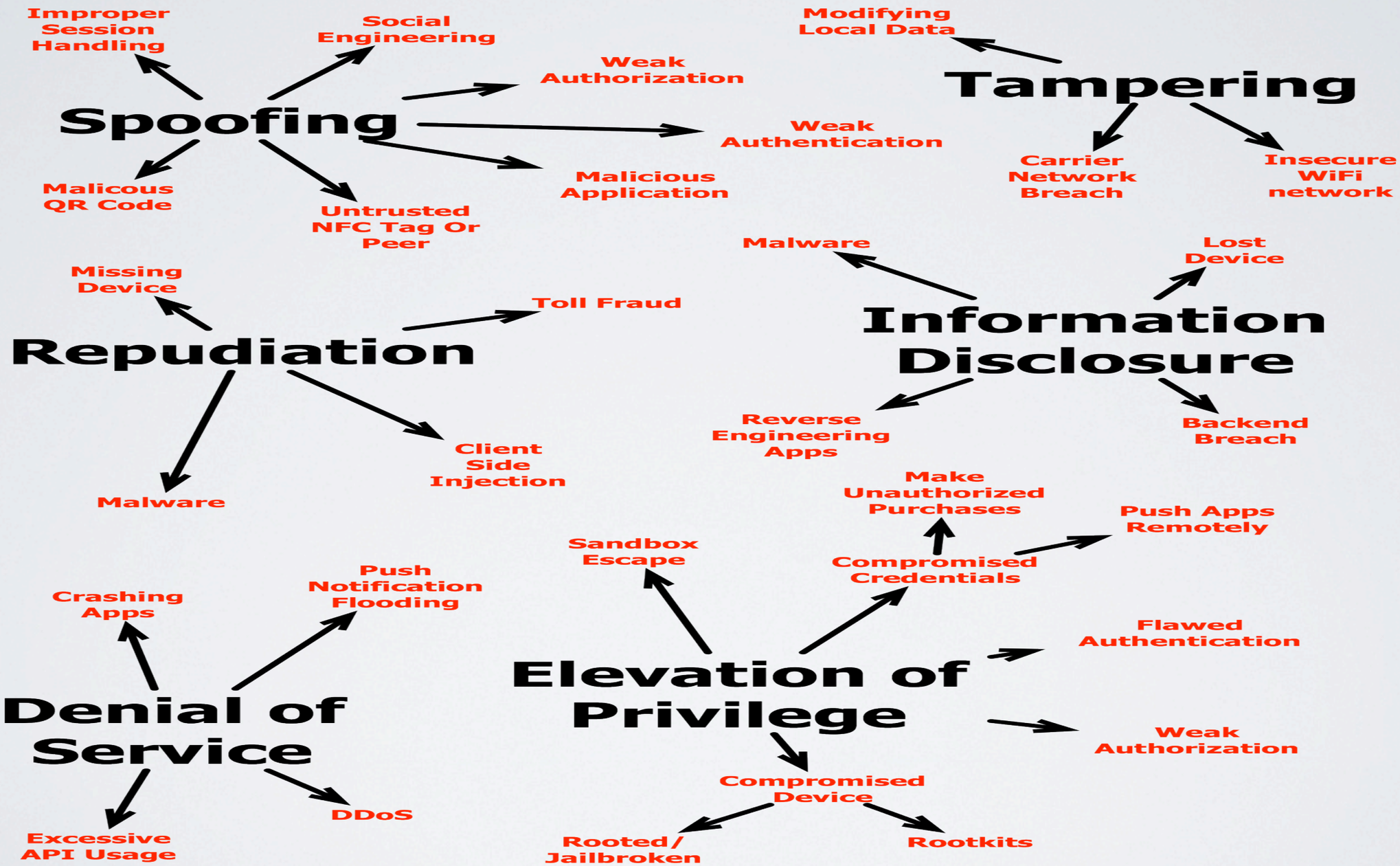
- The most important thing to understand about security is that **it is not a bullet point item**.
- Security cannot be bolted on at the end of the development process.
- You must **consciously design security into your app or service from the very beginning**, and make it a conscious part of the entire process from design through implementation, testing, and release.

So: Where do I start?

HOW CAN OWASP ASSIST MOBILE DEVELOPERS?

https://www.owasp.org/index.php/OWASP_Mobile_Security_Project

OWASP MOBILE THREAT MODEL



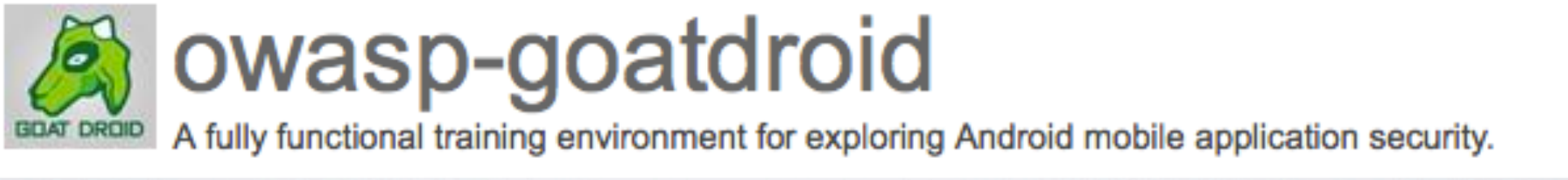
OWASP MOBILE TOP 10 RISKS

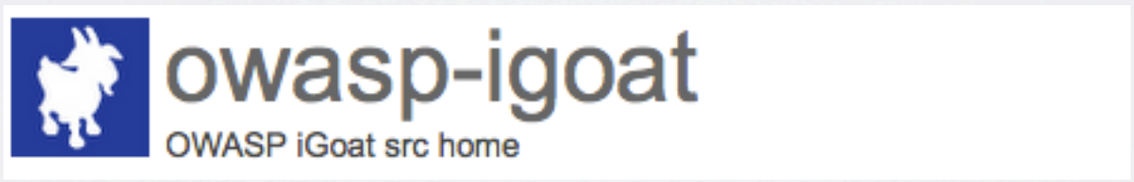
OWASP Mobile Top 10 Risks	
M1- Insecure Data Storage	M6- Improper Session Handling
M2- Weak Server Side Controls	M7- Security Decisions Via Untrusted Inputs
M3- Insufficient Transport Layer Protection	M8- Side Channel Data Leakage
M4- Client Side Injection	M9- Broken Cryptography
M5- Poor Authorization and Authentication	M10- Sensitive Information Disclosure

OWASP MOBILE DESIGN GUIDELINES

1. Identify and protect sensitive data on the mobile device
2. Handle password credentials securely on the device
3. Ensure sensitive data is protected in transit
4. Implement user authentication/authorization and session management correctly
5. Keep the backend APIs (services) and the platform (server) secure
6. Perform data integration with third party services/applications securely
7. Pay specific attention to the collection and storage of consent for the collection and use of the user's data
8. Implement controls to prevent unauthorized access to paid-for resources (wallet, SMS, phone calls etc...)
9. Ensure secure distribution/provisioning of mobile applications
10. Carefully check any runtime interpretation of code for errors

OWASP TRAINING ENVIRONMENTS



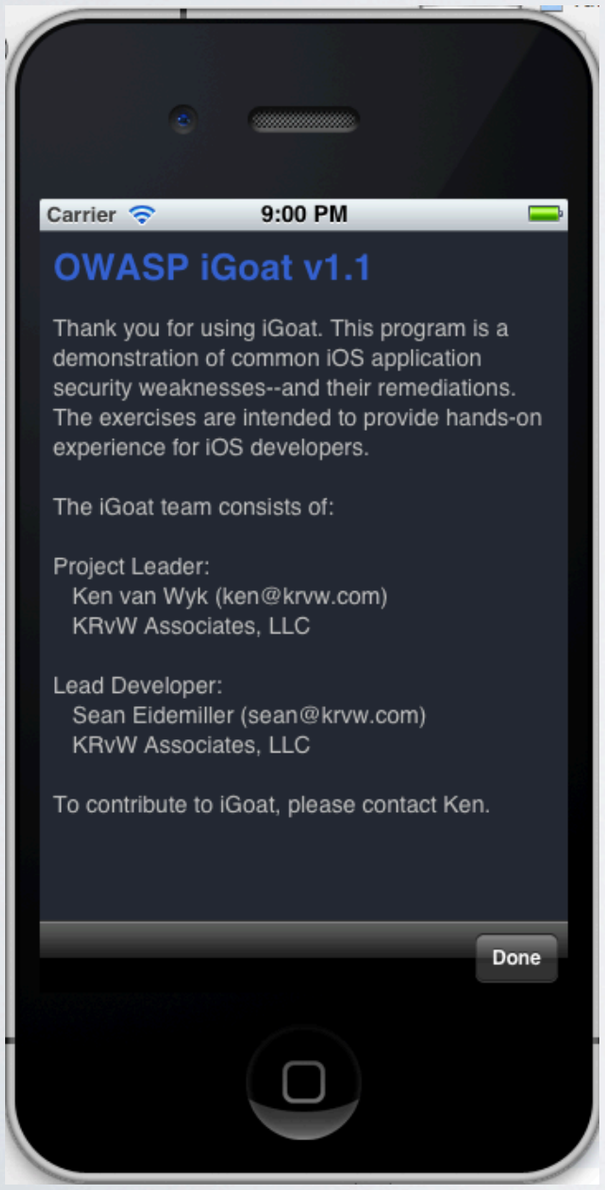


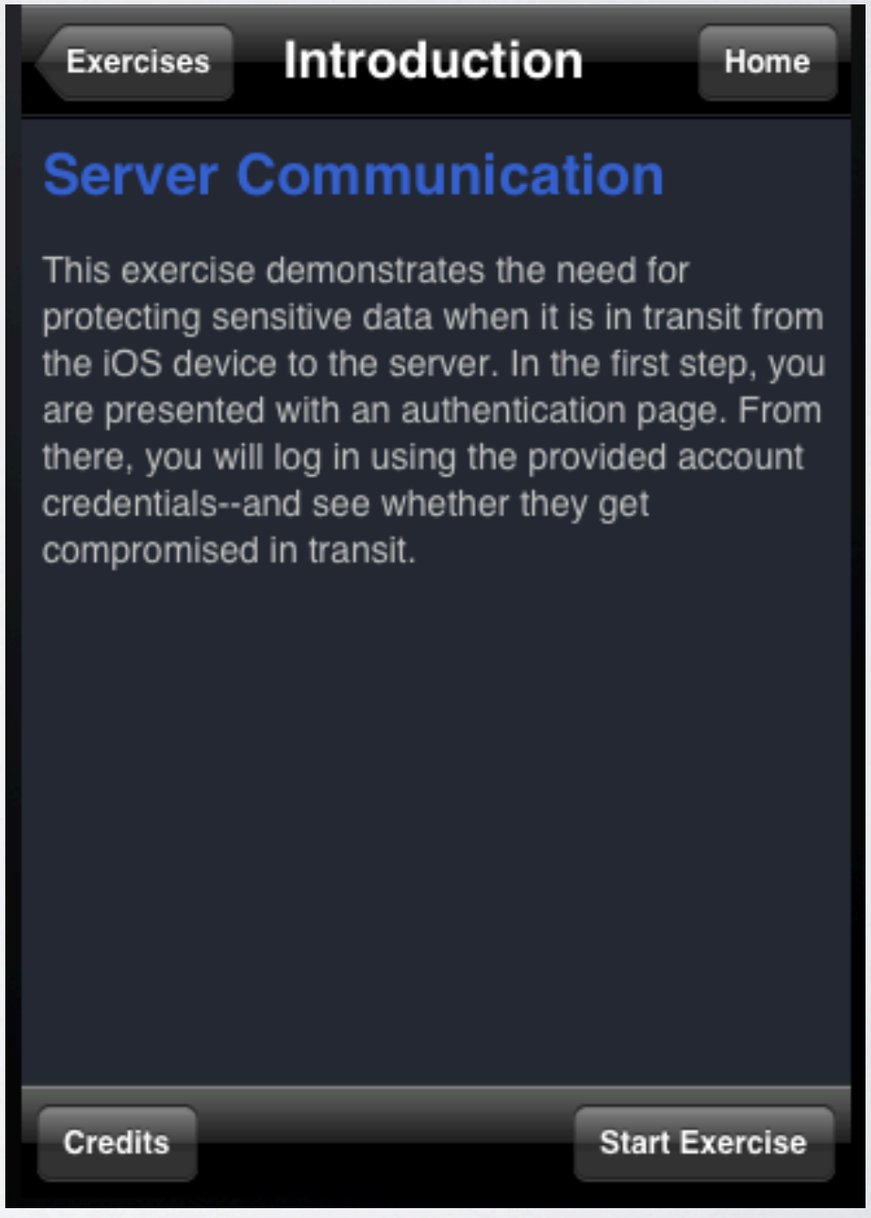
Secure coding training environments for Android and iOS developers

owasp-igoat

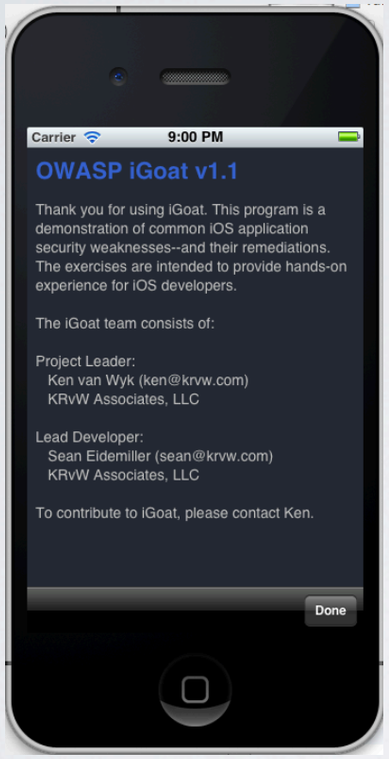
OWASP iGoat src home

Developed by Kenneth R. van Wyk





SAMPLE IGOAT EXERCISE: PROTECT DATA IN TRANSIT



HTTP GET

Server response

BURP
web-proxy

Ruby
A Programmer's

WHAT DID WE LEARN FROM THIS EXERCISE?

WHAT DID WE LEARN FROM THIS EXERCISE?

Protect data in transit!

- Man-in-the-middle attacks
- Tampering w/ data in transit
- Confidentiality of data lost

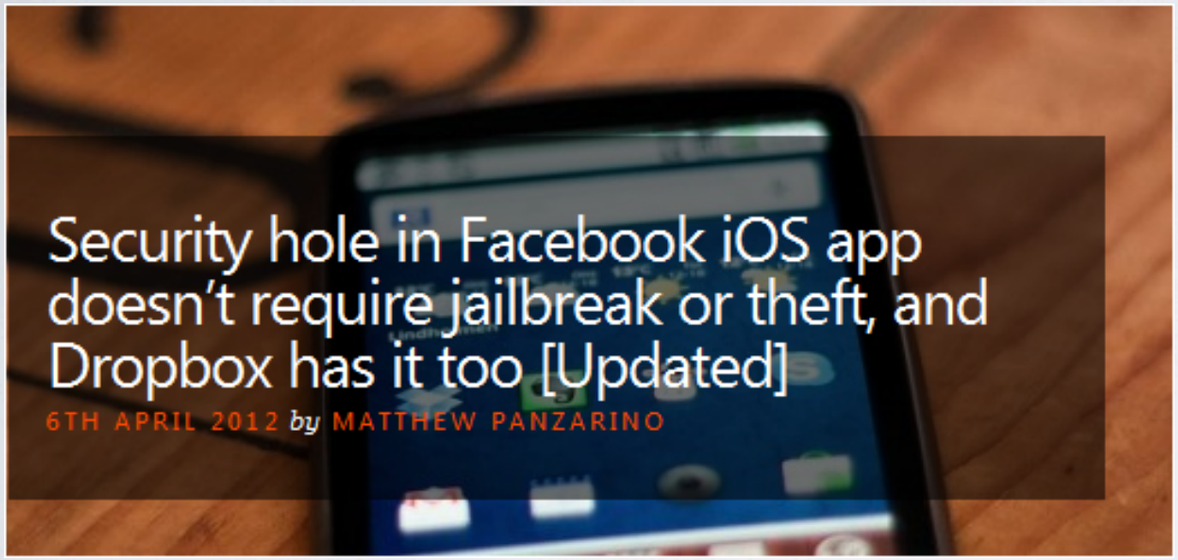
COMMON PITFALLS

- Complete lack of encryption for transmitted data
  - Or, only encrypting the login page (cf introductory example)
- Weakly encrypted data in transit
- Strong encryption, but ignoring security warnings
  - Ignoring certificate validation errors
  - Falling back to plain text after failures

MOBILE TOP TEN #1 - INSECURE DATA STORAGE

- Sensitive data left unprotected
- Applies to locally stored data + cloud synced
- Generally a result of:
  - Not encrypting data
  - Caching data not intended for long-term storage
  - Weak or global permissions
  - Not leveraging platform best-practices

REAL LIFE EXAMPLE: FACEBOOK DATA LEAKAGE

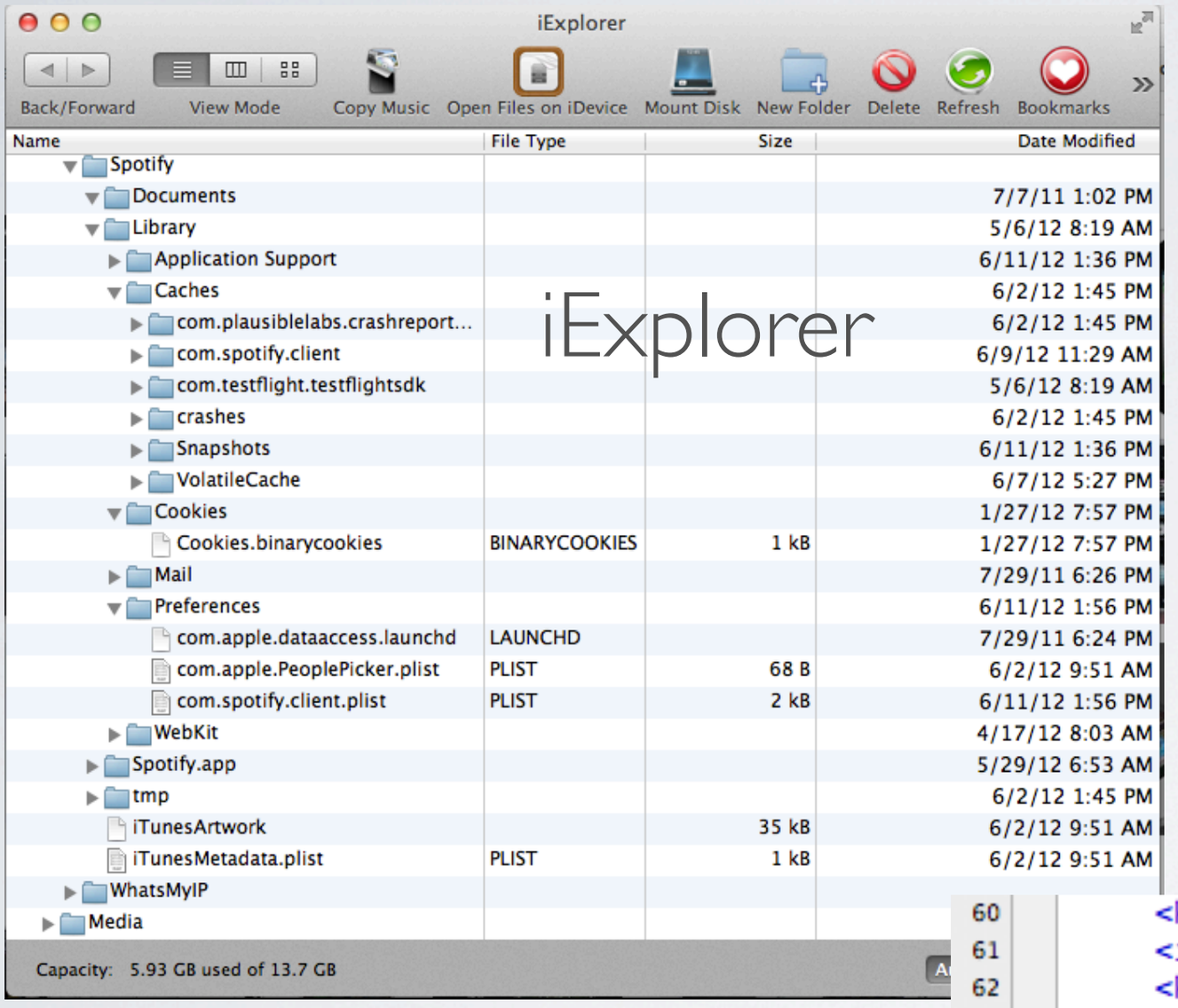


Works from any computer if PIN is not set, and if USB is inserted to charge phone, the .plist-file can be extracted.

.plist-file used for storing confidential data

- full oAuth key and secret in plain text.
- these are encrypted or salted with the device ID…only not.
- and they expired January 4001!

ATTACK VECTOR: LOST/STOLEN DEVICE



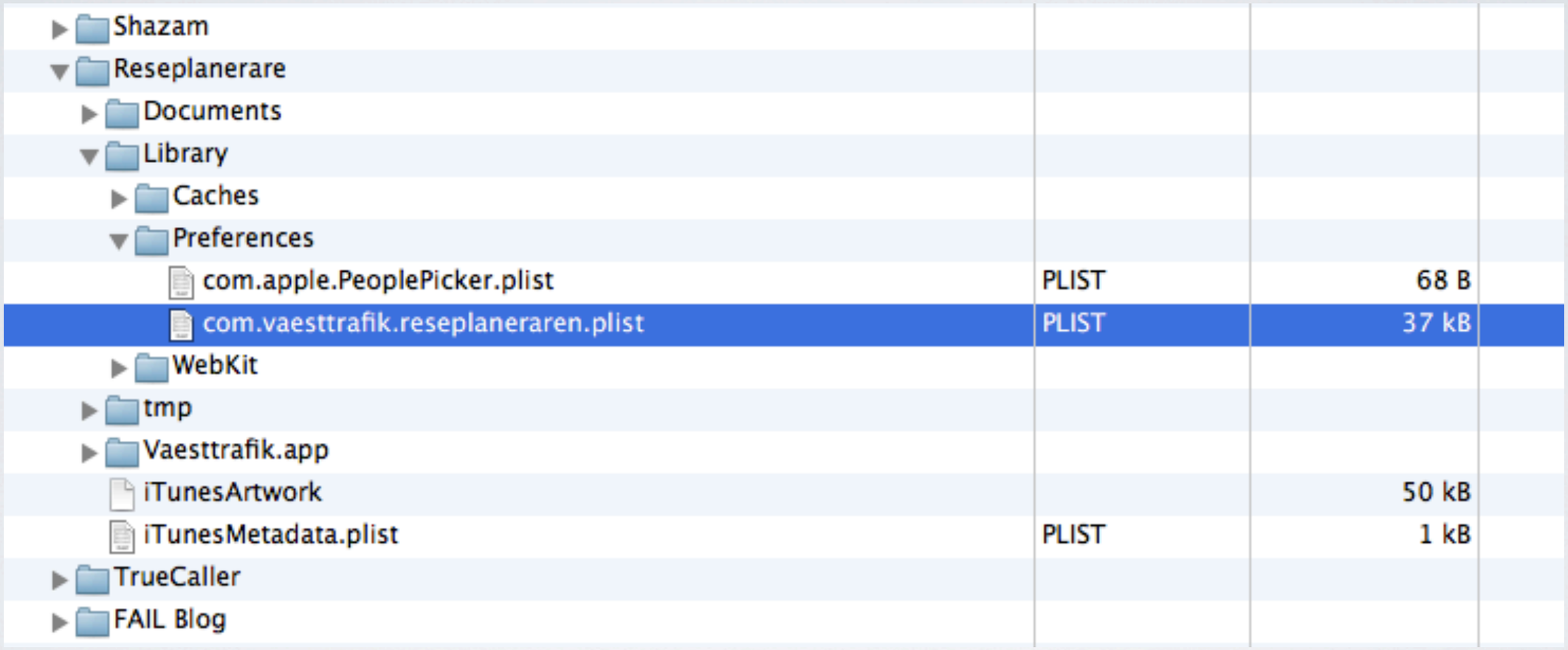
App bundle
- Hexdump of binary
- plist file

Explore folders
- ./Documents
- ./Library/Caches/*
- ./Library/Cookies
- ./Library/Preferences

```
60      <key>launchCount</key>
61      <integer>92</integer>
62      <key>password</key>
63      <string>4e6e1e6144431fcde96a0657729d66b5b1f48df9</string>
64      <key>proVersion</key>
65      <true/>
66      <key>ratePopupShowed</key>
67      <true/>
68      <key>username</key>
69      <string>drlarson</string>
70  </dict>
71  </plist>
```

TextWrangler

What's up with vasttrafik's travel planner?



Let's find out, shall we?

IMPACT

- Confidentiality of data lost
- Credentials disclosed
- Privacy violations
- Non-compliance
- Attack vectors?

PREVENTION TIPS

- Store ONLY what is absolutely required
- Leverage secure containers and platform provided file encryption APIs
- Do not grant files world readable or world writeable permissions

ROUND-UP

- Security cannot be bolted on at the end of the development process.
- OWASP has lots of resources: threat models, top lists, and training environments. Also, everything is FREE
- There may be serious concerns to brand and business if it turns out that your app discloses private information or breaks laws or regulations

- Overall good advise, models, lists, controls and training apps: https://www.owasp.org/index.php/OWASP_Mobile_Security_Project
  Owasp top ten mobile risks explained in detail with examples: http://www.slideshare.net/JackMannino/owasp-top-10-mobile-risks
  Mainly for developers (and testers to some extent): http://www.isecpartners.com/storage/docs/presentations/secure_development_on_ios.pdf
  OS security architecture (by Apple!): http://images.apple.com/ipad/business/docs/iOS_Security_May12.pdf