

SECURING MOBILE APPLICATIONS

Ulf Larson



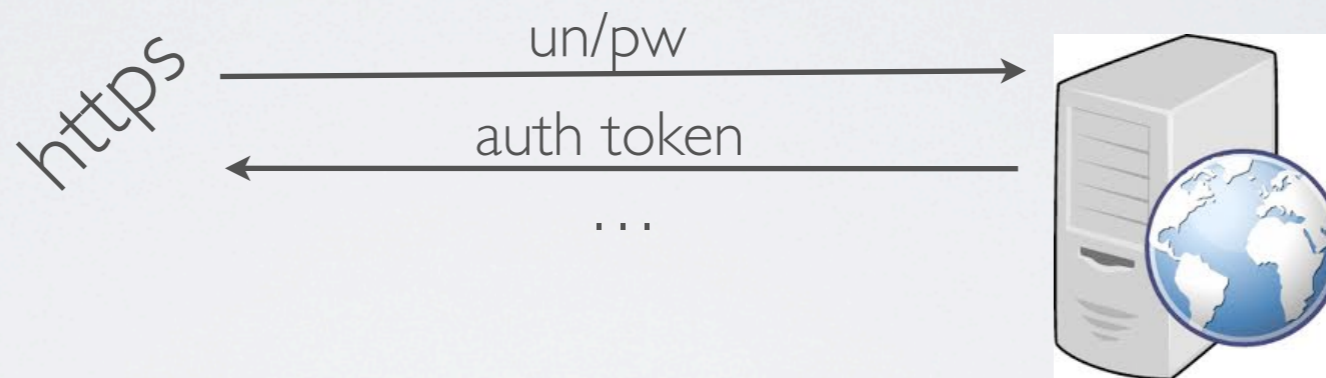
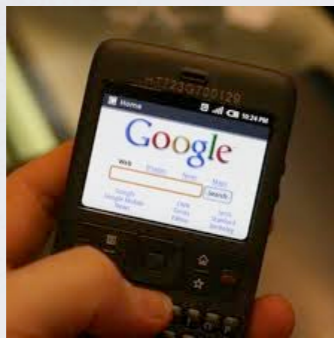
ulf.larson@owasp.org



ulf.larson@adecco.se

A REAL WORLD EXAMPLE TO GET US STARTED

Google ClientLogin Authentication Protocol

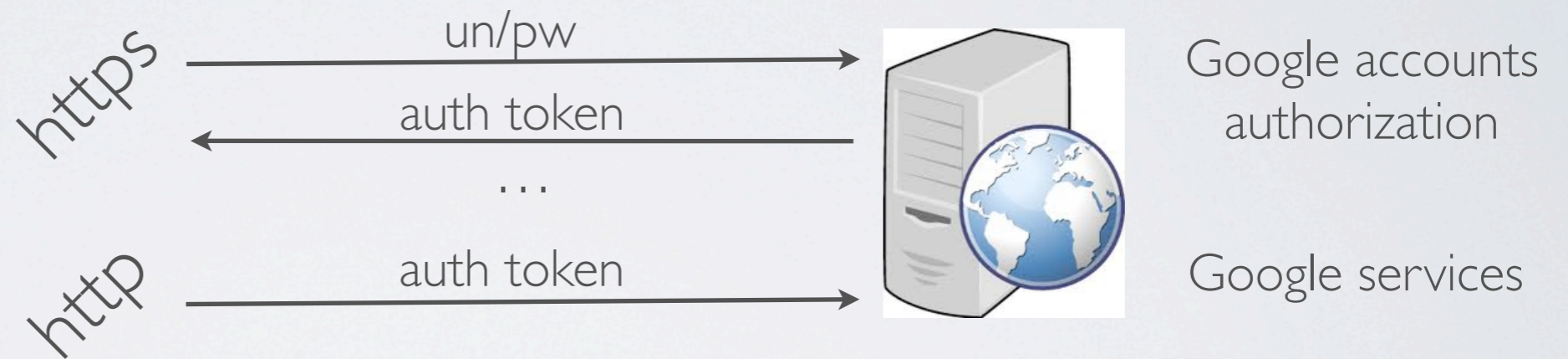
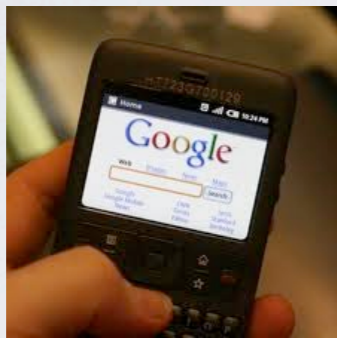


Google accounts
authorization

Google services

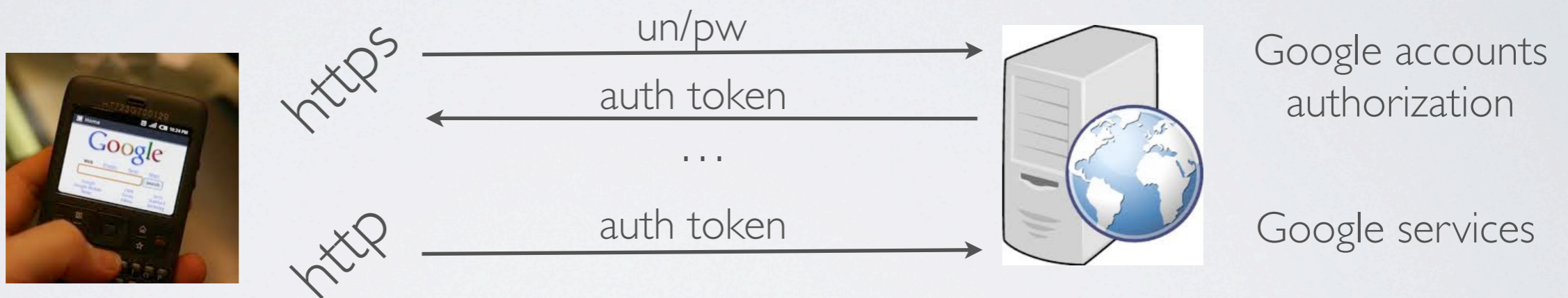
A REAL WORLD EXAMPLE TO GET US STARTED

Google ClientLogin Authentication Protocol



A REAL WORLD EXAMPLE TO GET US STARTED

Google ClientLogin Authentication Protocol



- 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

Filter: tcp.stream eq 1008

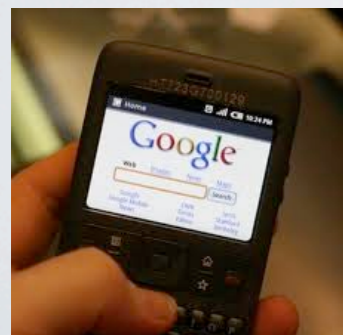
Time	Source	Destination	Protocol	No.	Info
72.187069	134.60.238.158	209.85.147.136	TCP	23483	51618 > http [SYN] Seq=0 Win=5840 Len=0 MSS=1460 TSV=58257 TSER=0 WS=1
72.108227	134.60.238.158	209.85.147.136	TCP	23484	51610 > http [SYN] Seq=0 Win=5840 Len=0 MSS=1460 TSV=58257 TSER=0 WS=1
72.109754	134.60.238.158	209.85.147.136	TCP	23485	51618 > http [SYN] Seq=0 Win=5840 Len=0 MSS=1460 TSV=58257 TSER=0 WS=1
72.130078	209.85.147.136	134.60.238.158	TCP	23491	http > 51618 [SYN, ACK] Seq=0 Ack=1 Win=5672 Len=8 MSS=1430 TSV=74506883 TSER=58257
72.155532	134.60.238.158	209.85.147.136	HTTP	23500	GET /data/feed/api/user/ [redacted] ?imgmax=1024&max-results=1000&thumbsize=144u,1024u&visibility=visible&kind=album HTTP/1.1
72.175001	209.85.147.136	134.60.238.158	TCP	23505	http > 51610 [ACK] Seq=1 Ack=537 Win=6784 Len=0 TSV=74506928 TSER=58262

Follow TCP Stream

Stream Content

```
GET /data/feed/api/user/ [redacted] ?imgmax=1024&max-results=1000&thumbsize=144u,1024u&visibility=visible&kind=album HTTP/1.1
GData-Version: 2
Accept-Encoding: gzip
Authorization: GoogleLogin auth=DQAAAKYAAA [redacted]
Vx36S 1vCUeYJ3b8P0bZ0SgAE6sW5hK0Z9naw7Z5vU [redacted]
Bn8x-q0S0RNjBvnGk5x0L4Dewr82q5nUFY1AHfnzg
Host: picasaweb.google.com
Connection: Keep-Alive
User-Agent: Cooliris-GData/1.0; gzip

[2122 bytes missing in capture file].S..R.1.jb.44<.....L[.ltSZ..pl....&Vf.*X.L07g.F:..wY.-.7l.V.oB(.
{!.6B.....-...4.2....v..#...]}WK..q..Wv.<u.....\.:.y.....]"%.t.....S.....Z.....;.....r...PN.
{..P...U.YU...w o.....)}.....bD.....".R....o...6....oA.....0d..S.bXY.Z.....Y.....Y.G.#N*c=-...
[-.1A-....eN.b...].#.Y
```

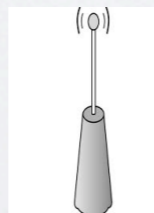


authToken over http



Evil twin

99% of all android phones were vulnerable



Free WiFi-hotspot

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?



OWASP
The Open Web Application Security Project

[Log in / create account](#)

Page **Discussion**

[Read](#)

[View source](#)

[View history](#)

Navigation

- [Home](#)
- [News](#)
- [OWASP Projects](#)
- [Downloads](#)
- [Local Chapters](#)
- [Global Committees](#)
- [AppSec Job Board](#)
- [AppSec Conferences](#)
- [Cheat Sheets](#)
- [OWASP Training](#)
- [Presentations](#)
- [Video](#)
- [Press](#)
- [Get OWASP Books](#)
- [Get OWASP Gear](#)
- [Mailing Lists](#)
- [About OWASP](#)
- [Membership](#)

[Reference](#)

[How To...](#)

OWASP Mobile Security Project

[Project Overview](#) [For Mobile Security Testers](#) [Mobile Secure Development Guidelines](#) [Top Ten Mobile Risks](#) [Top Ten Mobile Controls](#) [OWASP GoatDroid Project](#) [OWASP Mobile Threat Model Project](#)

[OWASP MobiSec Project](#)

The OWASP Mobile Security Project is a centralized resource intended to give developers and security teams the resources they need to build and maintain secure mobile applications. Through the project, our goal is to classify mobile security risks and provide developmental controls to reduce their impact or likelihood of exploitation.

We have a Google Doc up where anyone that wants to be involved with the project can add their thoughts, suggestions, and take ownership of initiatives. <https://docs.google.com/document/d/1bScrvrLJLOHcSbztjBxYoN-jN3kR8bViy9tF8Nx0c0B/edit> There are various tasks that people have started over the past 6 months with varying levels of quality and completeness.

PROJECT INFO

What does this OWASP project offer you?

what

is this project?

Name: OWASP Mobile Security Project ([home page](#))

Purpose: Our primary focus is at the application layer. While we take into consideration the underlying mobile platform and carrier inherent risks when threat modeling and building controls, we are targeting the areas that the average developer can make a difference. Additionally, we focus not only on the mobile applications deployed to end user devices, but also on the broader server-side infrastructure which the mobile apps communicate with. We focus heavily on the integration between the mobile application, remote authentication services, and cloud platform-specific features.

License: N/A

RELEASE(S) INFO

What releases are available for this project?

current release

Not Yet Published

last reviewed release

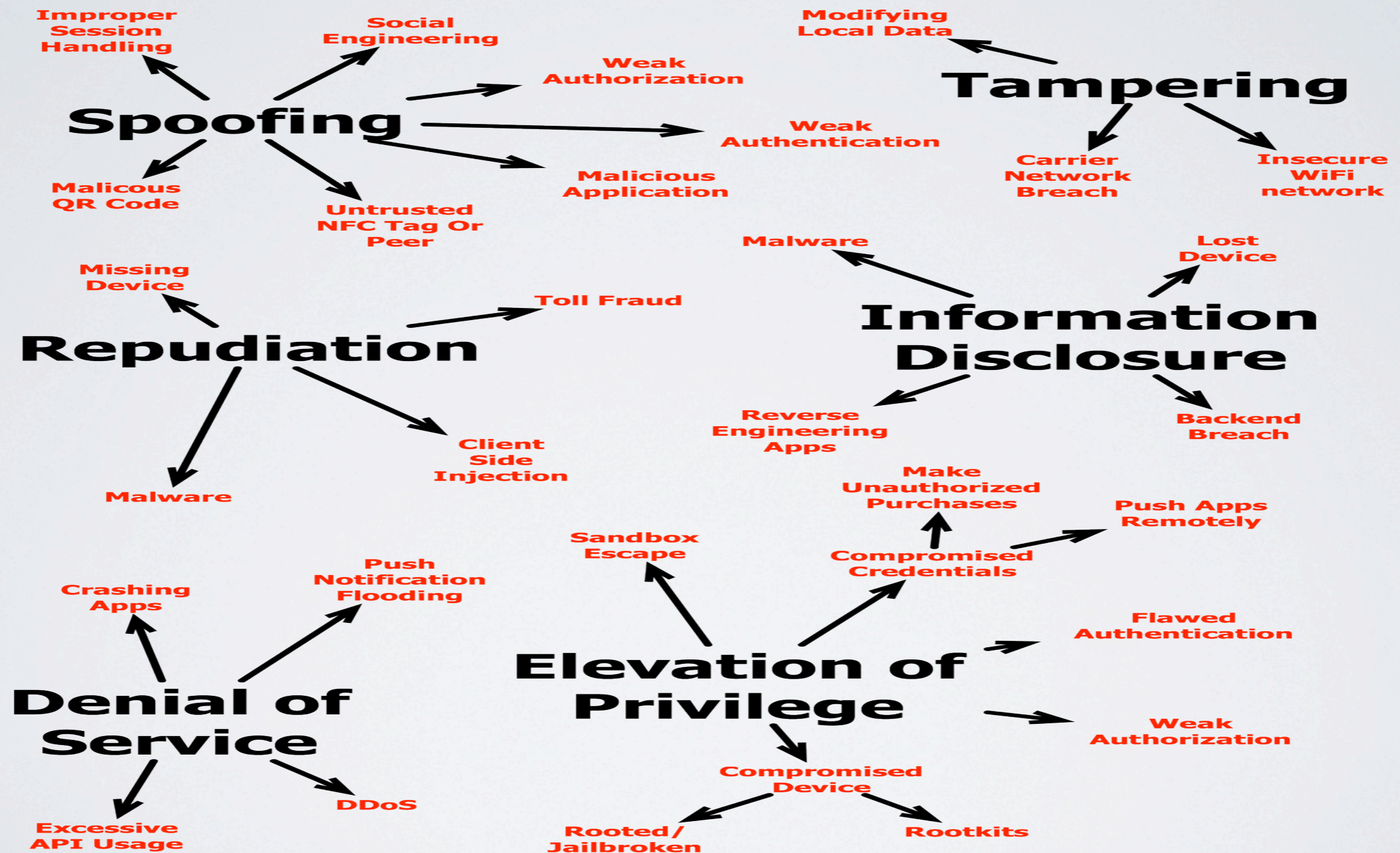
Not Yet Reviewed

all releases

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



owasp-goatdroid

A fully functional training environment for exploring Android mobile application security.



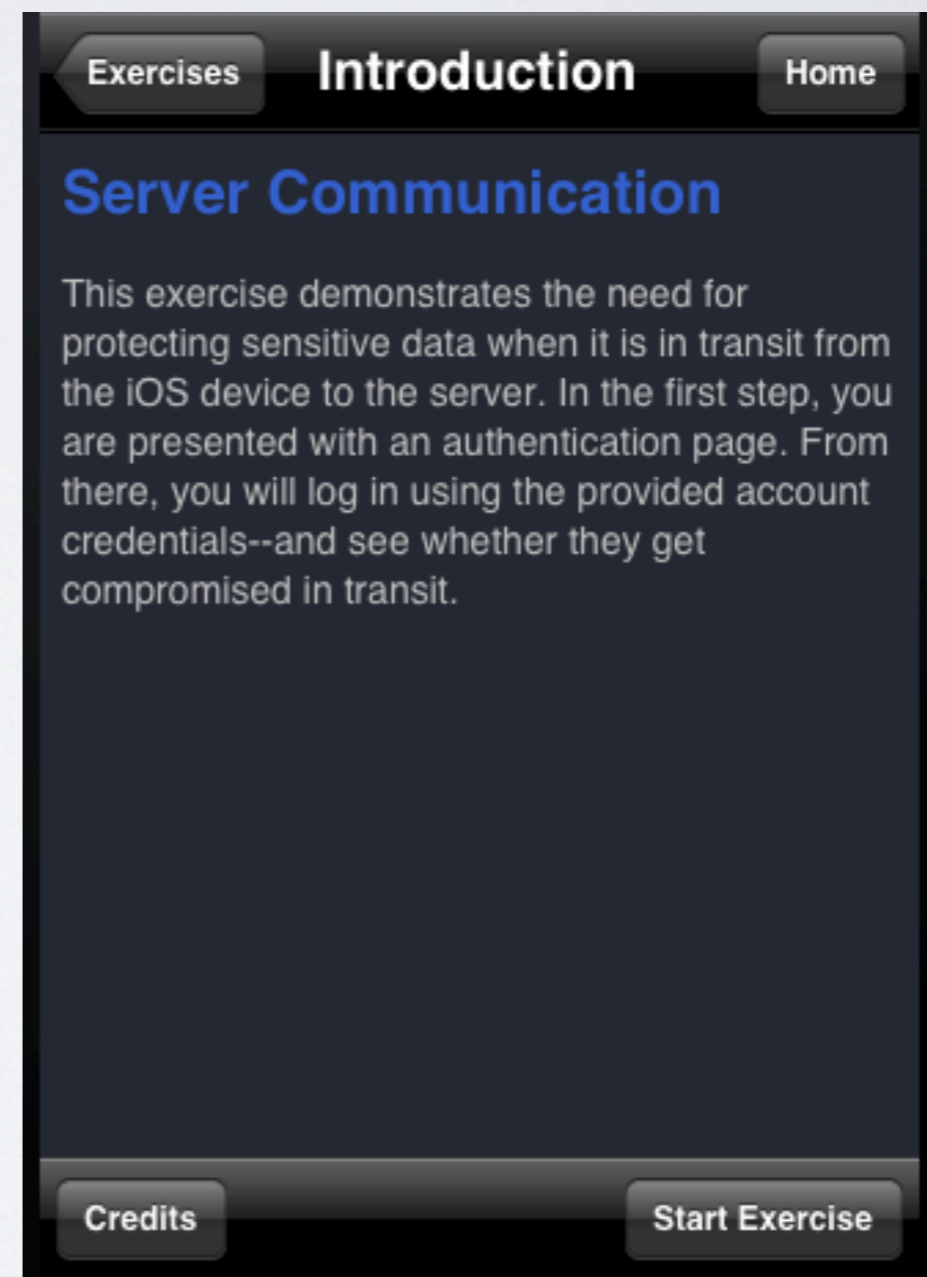
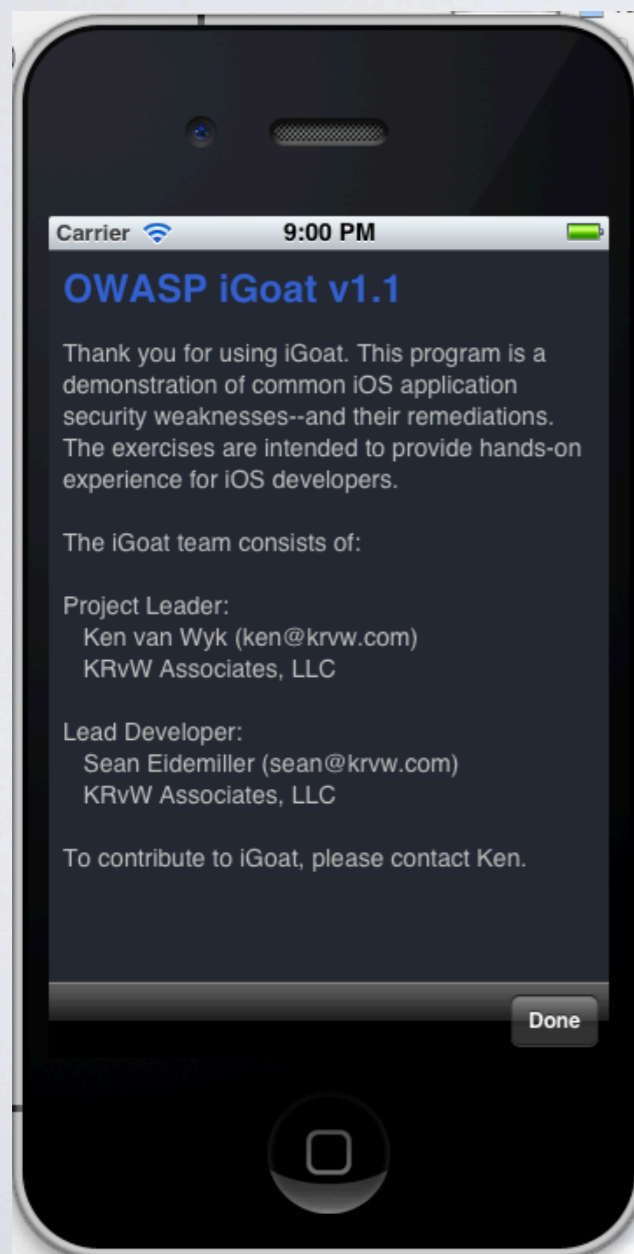
owasp-igoat

OWASP iGoat src home

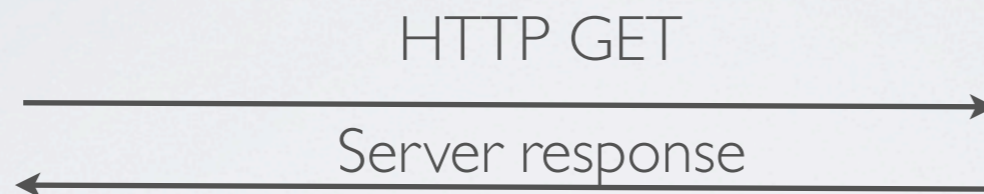
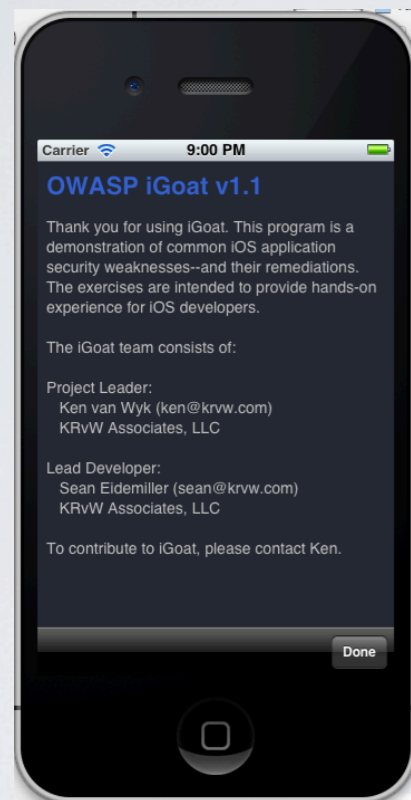
Secure coding training environments for Android and iOS developers



Developed by Kenneth R. van Wyk



SAMPLE IGOAT EXERCISE: PROTECT DATA IN TRANSIT



BURP
web-proxy

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

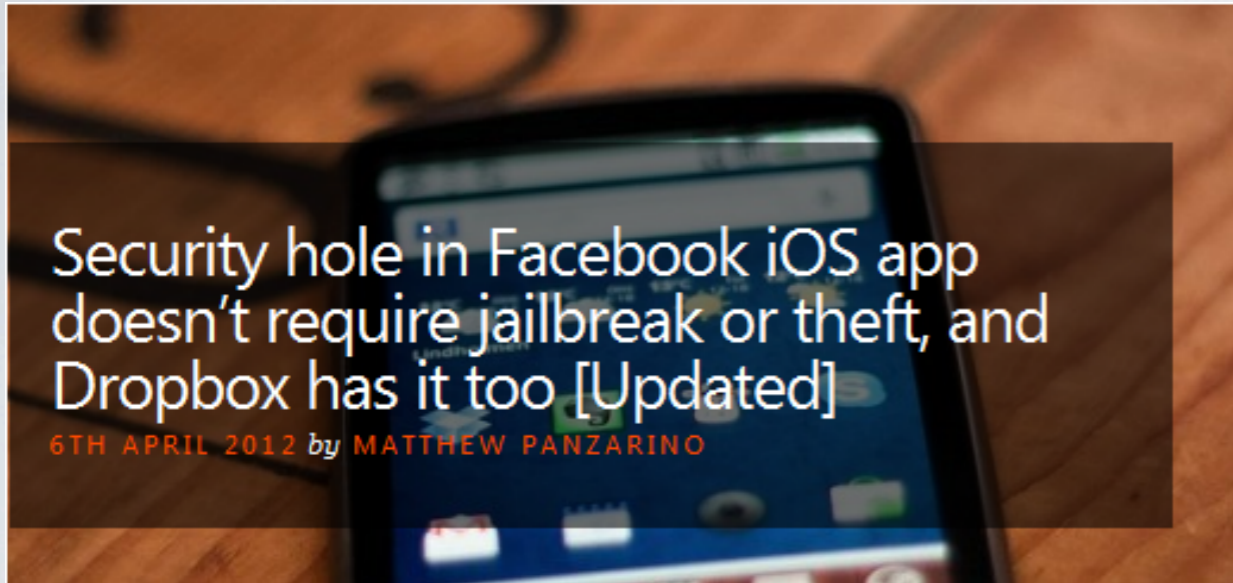
COMMON PITFALLS

- 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

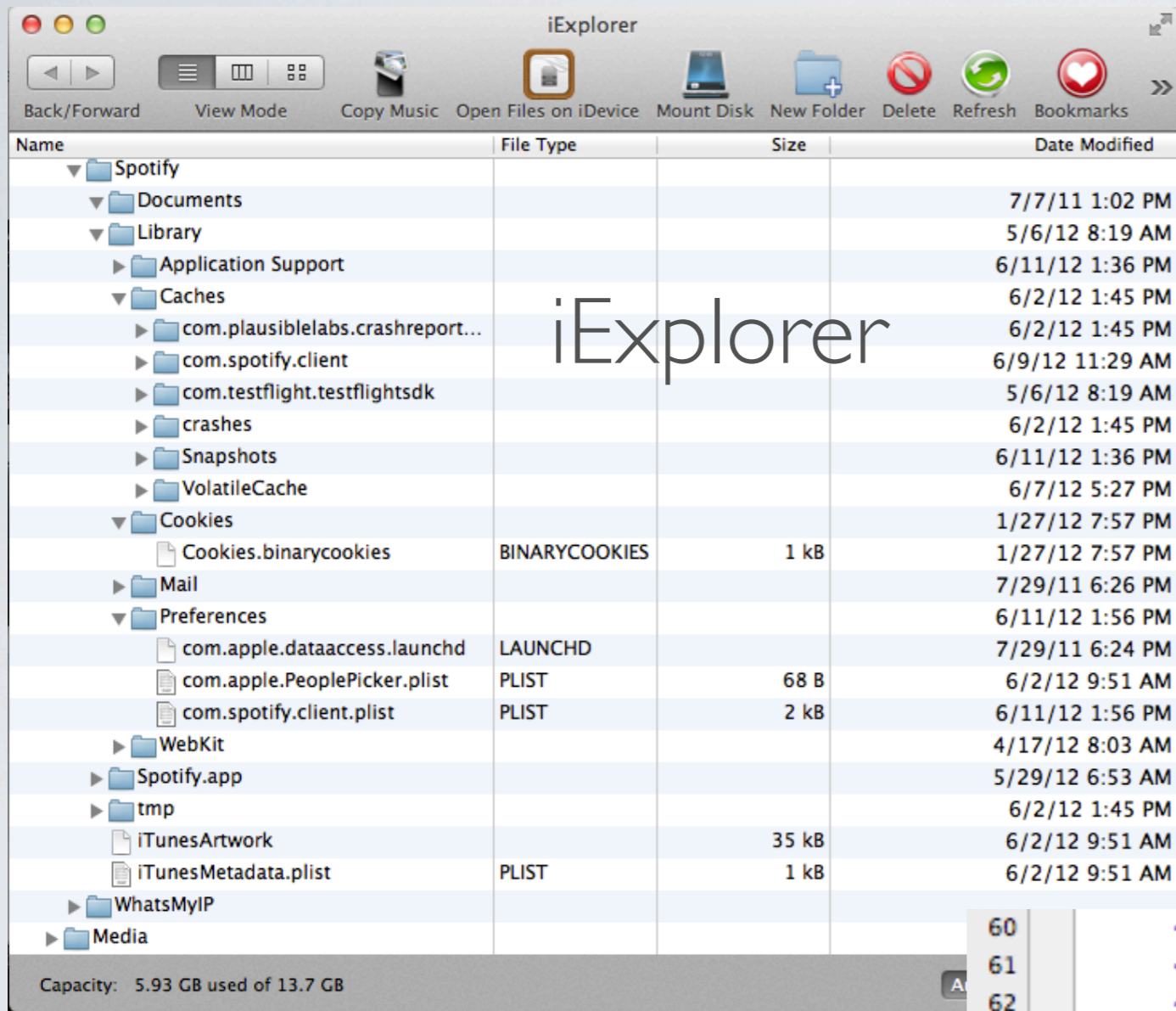


Works from any computer if PIN is not set, and if USB is inserted to charge phone, the .plist-file can be extracted.

.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 **400**!

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>ratePopupShown</key>
67 <true/>
68 <key>username</key>
69 <string>drlarson</string>
70 </dict>
71 </plist>
```

TextWrangler

What's up with vasttrafik's travel planner?

▶ Shazam			
▼ Reseplanerare			
▶ Documents			
▼ Library			
▶ Caches			
▼ Preferences			
com.apple.PeoplePicker.plist	PLIST	68 B	
com.vaesttrafik.reseplaneraren.plist	PLIST	37 kB	
▶ WebKit			
▶ tmp			
▶ Vaesttrafik.app			
iTunesArtwork		50 kB	
iTunesMetadata.plist	PLIST	1 kB	
▶ TrueCaller			
▶ FAIL Blog			

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