# BUILDING A MOBILE APP PEN TESTING BLUEPRINT

NowSecure™

# AGENDA

WHO WE ARE

WHY MOBILE MATTERS

TOOLS OF THE TRADE

COMMON FINDINGS

SHARING WITH STAKEHOLDERS

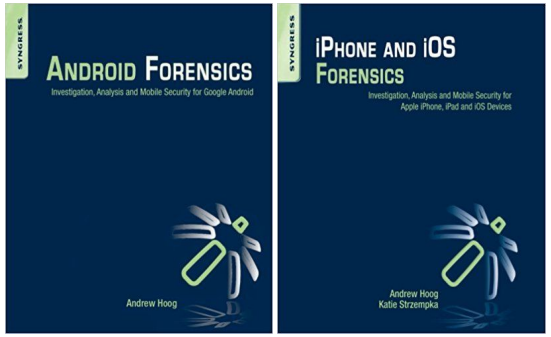
QUESTIONS

# SPEAKER

TONY RAMIREZ
MOBILE SECURITY ANALYST

# NOWSECURE DEEP MOBILE SECURITY EXPERTISE

## Books & Speaking




DEFCON

black hat

RSA CONFERENCE IEEE OWASP

## Open source

FRIDA

## MOBILE SECURITY RESEARCH IS IN OUR DNA

Dream team of security researchers

Discovering critical vulns
Identifying novel attack vectors
Creating/maintaining renowned open-source mobile security tools/projects

Expert team of security pen testers

Pen tested thousands of mobile apps
Comprehensive experience and testing blueprint
Certified some of the worlds most complex, high security apps

## THE NOWSECURE MISSION

Save the world from unsafe mobile apps

Educate enterprises on the latest mobile threats

Maximize the security of apps enterprises develop, purchase and use

**85%** of Mobile Apps Have Security Vulnerabilities

**49%** of Mobile Apps Leak Personal Data to Violate GDPR

# MOBILE APP RISKS ARE REAL AND PAINFULLY EXPOSED

## Under Armour says data breach affected about 150 million MyFitnessPal accounts

- The breach affected an estimated 150 million users of its food and nutrition application, MyFitnessPal.
- The investigation indicates that affected information may include usernames, email addresses, and hashed passwords.

Chloe Aiello | @chlobo_ilo

Published 4:38 PM ET Thu, 29 March 2018 | Updated 8:20 PM ET Thu, 29 March 2018

CNBC

## Air Canada mobile app breach affects 20,000 people

1.7 million use the app, but only about 1% may have been compromised

Pete Evans · CBC News · Posted: Aug 29, 2018 8:54 AM ET | Last Updated: August 29, 2018

## Equifax, Western Union, Priceline settle with New York attorney general over insecure mobile apps

Zack Whittaker @zackwhittaker / 1 month ago

TechCrunch

Comment

## British Airways Website, Mobile App Breach Compromises 380k

threatpost

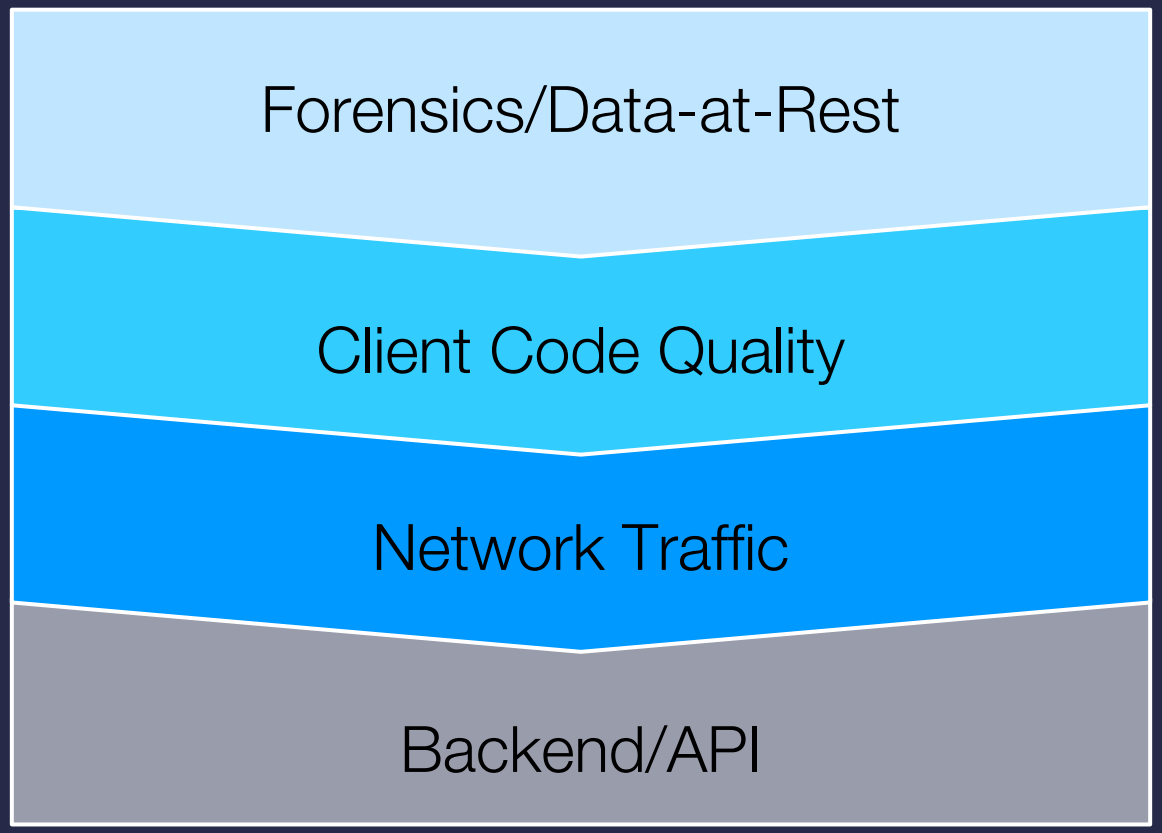
Author:
Lindsey O'Donnell

September 7, 2018 / 11:36 am

# TOOLS OF THE TRADE

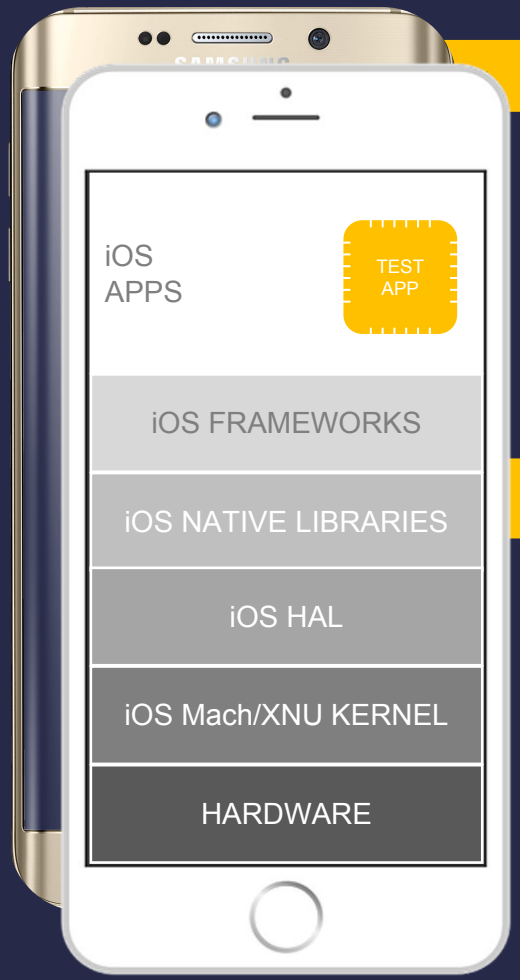
1. Terminal of choice
2. Jailbroken/rooted iOS and Android devices
3. Network interception tools
4. Developer tools
5. Reverse engineering tools
6. Patience, creativity, and attention to detail

# MOBILE VULNERABILITY AREAS – THE ATTACK SURFACE



# INSIDE THE MOBILE ATTACK SURFACE

NowSecure™




## CODE FUNCTIONALITY

- GPS spoofing
- Buffer overflow
- allowBackup Flag
- allowDebug Flag
- Code Obfuscation
- Configuration manipulation
- Escalated privileges
- URL schemes
- GPS Leaking
- Integrity/tampering/repacking
- Side channel attacks
- App signing key unprotected
- JSON-RPC
- Automatic Reference Counting
- Android rooting/iOS jailbreak
- User-initiated code
- Confused deputy attack
- Media/file format parsers
- Insecure 3rd party libraries
- World Writable Files
- World Writable Executables
- Dynamic runtime injection
- Unintended permissions
- UI overlay/pin stealing
- Intent hijacking
- Zip directory traversal
- Clipboard data
- World Readable Files

## DATA AT REST

- Data caching
- Data stored in application directory
- Decryption of keychain
- Data stored in log files
- Data cached in memory/RAM
- Data stored in SD card
- OS data caching
- Passwords & data accessible
- No/Weak encryption
- TEE/Secure Enclave Processor
- Side channel leak
- SQLite database
- Emulator variance

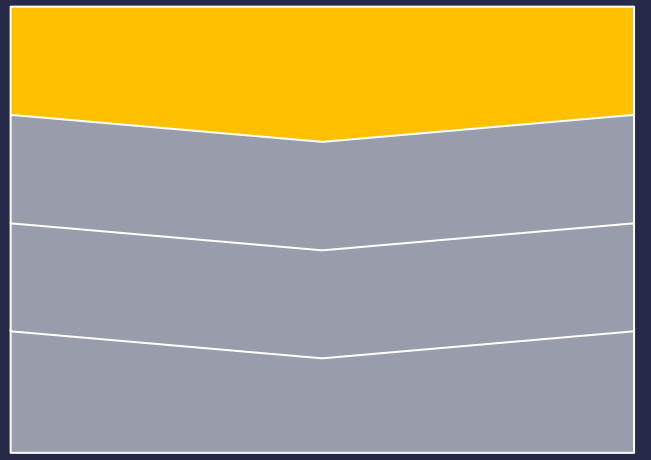
## DATA IN MOTION

- Wi-Fi (no/weak encryption)
- Rogue access point
- Packet sniffing
- Man-in-the-middle
- Session hijacking
- DNS poisoning
- TLS Downgrade
- Fake TLS certificate
- Improper TLS validation
- HTTP Proxies
- VPNs
- Weak/No Local authentication
- App transport security
- Transmitted to insecure server
- Zip files in transit
- Cookie “httpOnly” flag
- Cookie “secure” flag

Network &
Cloud Services

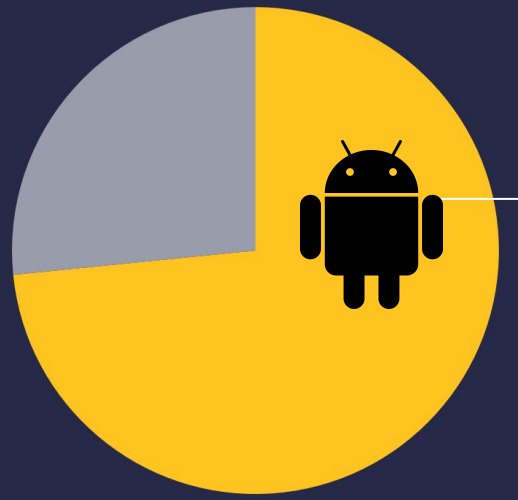
Data Center
& App Backend

# FORENSICS/DATA-AT-REST



# COMMON FORENSICS ISSUES

- Sensitive data on the device
  - Credentials
  - PII (SSNs, addresses, phone numbers)
  - Session tokens
  - Cryptographic keys and IVs
- Data in insecure locations
  - System Logs
  - Emulated Storage / SD Card
- Arbitrary Code on SD Card

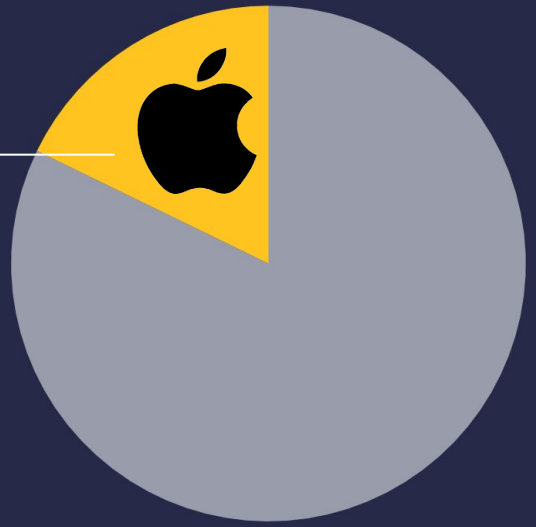


**73%**

of Android Apps
Leak Sensitive Data to
System Logs

**18%**

of iOS Apps
Leak Sensitive Data to
System Logs



# TESTING FORENSICS/DATA-AT-REST

## The search for sensitive values

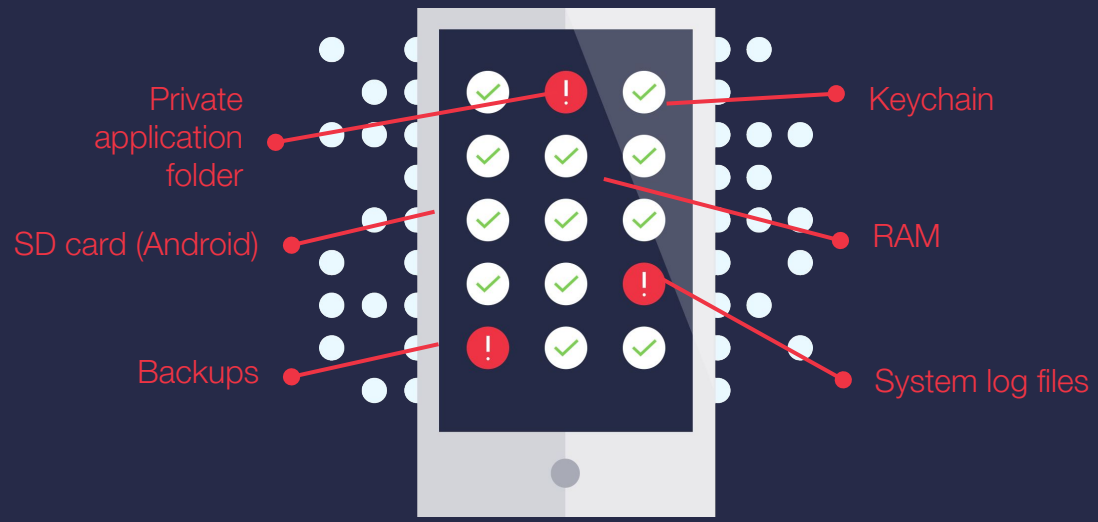
- Rainbow tables help
- regex and grep

## Jailbroken/Rooted devices

- Great for testing private folders and keychain
- Not necessary for backups, logs, and SDCard

## Exercise the app!

- Different data before and after log out



# REAL-WORLD EXAMPLE

## IoT app with wearable hardware

- Monitors users health
- Requires updates
- Syncs with mobile device via app

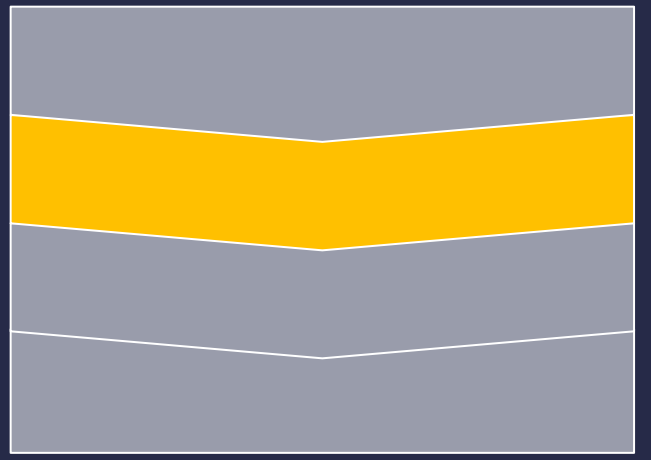
## Writing to insecure data storage

- Identifying health data on SD card
- Firmware binary stored on SD card

# REMEDIATIONS

1. Avoid writing sensitive data
   - Use encryption
     - No custom crypto
2. Avoid writing to SD card
3. Avoid writing to system logs

# CLIENT CODE QUALITY



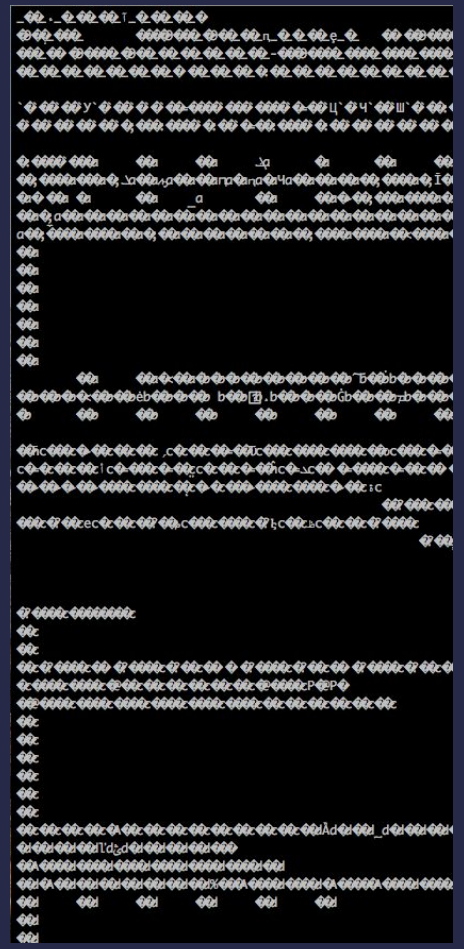
# COMMON CODE-LEVEL ISSUES

- Hardcoded crypto keys and IVs
- Hardcoded credentials
- Client side logic
- Vulnerable SDKs and libraries
- Free security features
- Backdoor methods

# TESTING CODE QUALITY

- Black-box testing approach
- Reverse Engineering:
  - Zip files -> AndroidManifest.xml, Info.plist
  - Disassemblers  / Decompilers (Radare2, apktool, procyon)
  - Dynamic binary instrumentation (Frida)
  - Developer tool (Android Studio, Xcode)
  - Source code analysis
    - Android - Java, Kotlin
    - iOS - Objective-C, Swift
- “strings” and “grep”, otool etc.

# DEX



# -> smali

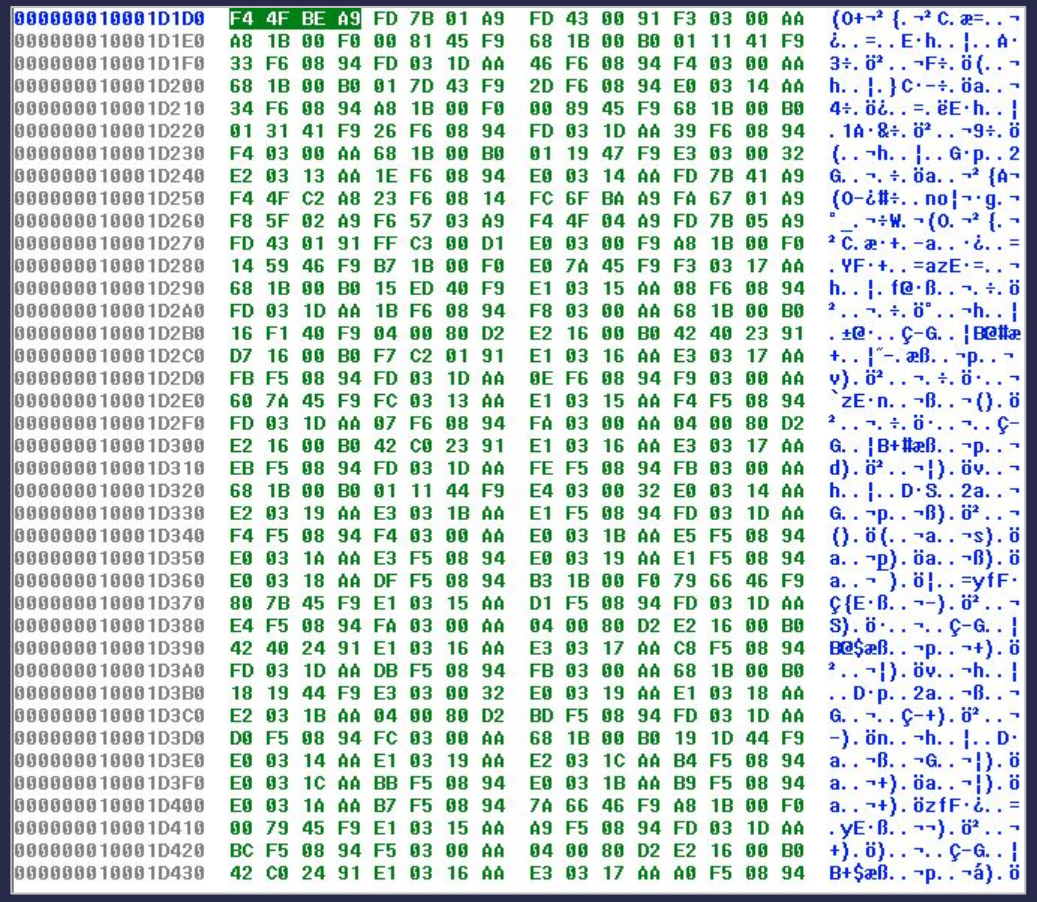
```
new-instance v1, Ljava/lang/StringBuilder
const-string v2, "the spice must flow"
invoke-direct {v1, v2}, Ljava/lang/String

invoke-virtual {v1, p1}, Ljava/lang/Strin
move-result-object v1

const-string v2, "some random string"
invoke-virtual {v1, v2}, Ljava/lang/Strin
move-result-object v1

invoke-virtual {v1}, Ljava/lang/StringBui
move-result-object v1

const-string v0, "Tag"
invoke-static {v0, v1}, Landroid/util/Log
move-result v0

invoke-static {}, Ljava/lang/System;->cu
move-result-wide v2
const-wide/16 v4, 0x300
div-long/2addr v2, v4
long-to-int v2, v2
```

# -> jar

```
package org.apache.commons.codec.binary;

import org.apache.commons.codec.a;
import org.apache.commons.codec.b;
import org.apache.commons.io.IOUtils;

public class Base64 {
    static final int BASELENGTH = 255;
    static final byte[] CHUNK_SEPARATOR = I
    static final int CHUNK_SIZE = 76;
    static final int EIGHTBIT = 8;
    static final int FOURBYTE = 4;
    static final int LOOKUPLENGTH = 64;
    static final byte PAD = (byte) 61;
    static final int SIGN = -128;
    static final int SIXTEENBIT = 16;
    static final int TWENTYFOURBITGROUP = 2
    private static byte[] base64Alphabet =
    private static byte[] lookUpBase64Alpha

    static {
        int i;
        int i2 = 0;
        for (i = 0; i < 255; i++) {
            base64Alphabet[i] = (byte) -1;
        }
        for (i = 90; i >= 65; i--) {
            base64Alphabet[i] = (byte) (i -
        }
```

# Executable code compared to disassembled code

## Executable code



```
000000010001D1D0  F4 4F BE A9 FD 7B 01 A9  FD 43 00 91 F3 03 00 AA  {O+¬² {. ¬² C. æ=. . ¬
000000010001D1E0  A8 1B 00 F0 00 81 45 F9  68 1B 00 B0 01 11 41 F9  ¿. . =. . E·h. . ¦. . A·
000000010001D1F0  33 F6 08 94 FD 03 1D AA  46 F6 08 94 F4 03 00 AA  3÷. ö² . . ¬F÷. ö(. . ¬
000000010001D200  68 1B 00 B0 01 7D 43 F9  2D F6 08 94 E0 03 14 AA  h. . ¦. }C·-÷. öa. . ¬
000000010001D210  34 F6 08 94 A8 1B 00 F0  00 89 45 F9 68 1B 00 B0  4÷. öi. . =. ëE·h. . ¦
000000010001D220  01 31 41 F9 26 F6 08 94  FD 03 1D AA 39 F6 08 94  . 1A·&÷. ö² . . ¬9÷. ö
000000010001D230  F4 03 00 AA 68 1B 00 B0  01 19 47 F9 E3 03 00 32  (. . ¬h. . ¦. . G·p. . 2
000000010001D240  E2 03 13 AA 1E F6 08 94  E0 03 14 AA FD 7B 41 A9  G. . ¬. ÷. öa. . ¬² {A¬
000000010001D250  F4 4F C2 A8 23 F6 08 14  FC 6F BA A9 FA 67 01 A9  (O-¿#÷. . no¦¬·g. ¬
000000010001D260  F8 5F 02 A9 F6 57 03 A9  F4 4F 04 A9 FD 7B 05 A9  °_. ¬÷W. ¬(O. ¬² {. ¬
000000010001D270  FD 43 01 91 FF C3 00 D1  E0 03 00 F9 A8 1B 00 F0  ² C. æ·+. -a. . ·¿. . =
000000010001D280  14 59 46 F9 B7 1B 00 F0  E0 7A 45 F9 F3 03 17 AA  . YF·+. . =azE·=. . ¬
000000010001D290  68 1B 00 B0 15 ED 40 F9  E1 03 15 AA 08 F6 08 94  h. . ¦. f@·ß. . ¬. ÷. ö
000000010001D2A0  FD 03 1D AA 1B F6 08 94  F8 03 00 AA 68 1B 00 B0  ². . ¬. ÷. öº . . ¬h. . ¦
000000010001D2B0  16 F1 40 F9 04 00 80 D2  E2 16 00 B0 42 40 23 91  . ±@·. . Ç-G. . ¦B@#æ
000000010001D2C0  D7 16 00 B0 F7 C2 01 91  E1 03 16 AA E3 03 17 AA  +. . ¦˜-. æß. . ¬p. . ¬
000000010001D2D0  FB F5 08 94 FD 03 1D AA  0E F6 08 94 F9 03 00 AA  v). ö² . . ¬. ÷. ö·. . ¬
000000010001D2E0  60 7A 45 F9 FC 03 13 AA  E1 03 15 AA F4 F5 08 94  `zE·n. . ¬ß. . ¬(). ö
000000010001D2F0  FD 03 1D AA 07 F6 08 94  FA 03 00 AA 04 00 80 D2  ². . ¬. ÷. ö·. . ¬. . Ç-
000000010001D300  E2 16 00 B0 42 C0 23 91  E1 03 16 AA E3 03 17 AA  G. . ¦B+#æß. . ¬p. . ¬
000000010001D310  EB F5 08 94 FD 03 1D AA  FE F5 08 94 FB 03 00 AA  d). ö² . . ¬¦). öv. . ¬
000000010001D320  68 1B 00 B0 01 11 44 F9  E4 03 00 32 E0 03 14 AA  h. . ¦. . D·S. . 2a. . ¬
000000010001D330  E2 03 19 AA E3 03 1B AA  E1 F5 08 94 FD 03 1D AA  G. . ¬p. . ¬ß). ö² . . ¬
000000010001D340  F4 F5 08 94 F4 03 00 AA  E0 03 1B AA E5 F5 08 94  (). ö(. . ¬a. . ¬s). ö
000000010001D350  E0 03 1A AA E3 F5 08 94  E0 03 19 AA E1 F5 08 94  a. . ¬p). öa. . ¬ß). ö
000000010001D360  E0 03 18 AA DF F5 08 94  B3 1B 00 F0 79 66 46 F9  a. . ¬ ). ö¦. . =yfF·
000000010001D370  80 7B 45 F9 E1 03 15 AA  D1 F5 08 94 FD 03 1D AA  Ç{E·ß. . ¬-). ö² . . ¬
000000010001D380  E4 F5 08 94 FA 03 00 AA  04 00 80 D2 E2 16 00 B0  S). ö·. . ¬. . Ç-G. . ¦
000000010001D390  42 40 24 91 E1 03 16 AA  E3 03 17 AA C8 F5 08 94  B@$æß. . ¬p. . ¬+). ö
000000010001D3A0  FD 03 1D AA DB F5 08 94  FB 03 00 AA 68 1B 00 B0  ². . ¬¦). öv. . ¬h. . ¦
000000010001D3B0  18 19 44 F9 E3 03 00 32  E0 03 19 AA E1 03 18 AA  . . D·p. . 2a. . ¬ß. . ¬
000000010001D3C0  E2 03 1B AA 04 00 80 D2  BD F5 08 94 FD 03 1D AA  G. . ¬. . Ç-+). ö² . . ¬
000000010001D3D0  D0 F5 08 94 FC 03 00 AA  68 1B 00 B0 19 1D 44 F9  -). ön. . ¬h. . ¦. . D·
000000010001D3E0  E0 03 14 AA E1 03 19 AA  E2 03 1C AA B4 F5 08 94  a. . ¬ß. . ¬G. . ¬¦). ö
000000010001D3F0  E0 03 1C AA BB F5 08 94  E0 03 1B AA B9 F5 08 94  a. . ¬+). öa. . ¬¦). ö
000000010001D400  E0 03 1A AA B7 F5 08 94  7A 66 46 F9 A8 1B 00 F0  a. . ¬+). özfF·¿. . =
000000010001D410  00 79 45 F9 E1 03 15 AA  A9 F5 08 94 FD 03 1D AA  . yE·ß. . ¬¬). ö² . . ¬
000000010001D420  BC F5 08 94 F5 03 00 AA  04 00 80 D2 E2 16 00 B0  +). ö). . ¬. . Ç-G. . ¦
000000010001D430  42 C0 24 91 E1 03 16 AA  E3 03 17 AA A0 F5 08 94  B+$æß. . ¬p. . ¬á). ö
```

## Disassembled code

```
; LoginDisclosureViewController - (void)accept

; void __cdecl -[LoginDisclosureViewController accept](struct LoginDisclosureViewController *self, SEL)
__LoginDisclosureViewController_accept_

var_20= -0x20
var_10= -0x10

STP             X20, X19, [SP,#var_20]!
STP             X29, X30, [SP,#0x20+var_10]
ADD             X29, SP, #0x20+var_10
MOV             X19, X0
ADRP            X8, #classRef_UIApplication@PAGE
LDR             X0, [X8,#classRef_UIApplication@PAGEOFF]
ADRP            X8, #selRef_sharedApplication@PAGE
LDR             X1, [X8,#selRef_sharedApplication@PAGEOFF]
BL              _objc_msgSend
MOV             X29, X29
BL              _objc_retainAutoreleasedReturnValue
MOV             X20, X0
ADRP            X8, #selRef_displayModalViewWorking@PAGE
LDR             X1, [X8,#selRef_displayModalViewWorking@PAGEOFF]
BL              _objc_msgSend
MOV             X0, X20
BL              _objc_release
ADRP            X8, #classRef_ServiceCall@PAGE
LDR             X0, [X8,#classRef_ServiceCall@PAGEOFF]
ADRP            X8, #selRef_instance@PAGE
LDR             X1, [X8,#selRef_instance@PAGEOFF]
BL              _objc_msgSend
MOV             X29, X29
BL              _objc_retainAutoreleasedReturnValue
MOV             X20, X0
ADRP            X8, #selRef_secondaryDisclosureWithDelegate_acceptDisclosure_@PAGE
LDR             X1, [X8,#selRef_secondaryDisclosureWithDelegate_acceptDisclosure_@PAGEOFF]
MOV             W3, #1
MOV             X2, X19
BL              _objc_msgSend
MOV             X0, X20
LDP             X29, X30, [SP,#0x20+var_10]
LDP             X20, X19, [SP+0x20+var_20],#0x20
```

# REAL-WORLD EXAMPLE

Crypto info hardcoded client-side

```
[LGE Nexus 5X::                    ]-> Java.perform(function() { var Decrypt = Java.use(                              ); var d
c = Decrypt.$new(); console.log(dc.decrypt('FWwoI8eLHcRaqtjZfIob8MZU53rtlIDFVdk9J1FObQgeFiHL9UbxGdY/Zbwu7NU/5X7+VWw
DpGto4GfTL96kb0gT7bjCc8+rQ1Pj6QolYmgfbjlcxpG6jg=='));})
6019180351902339
undefined
```

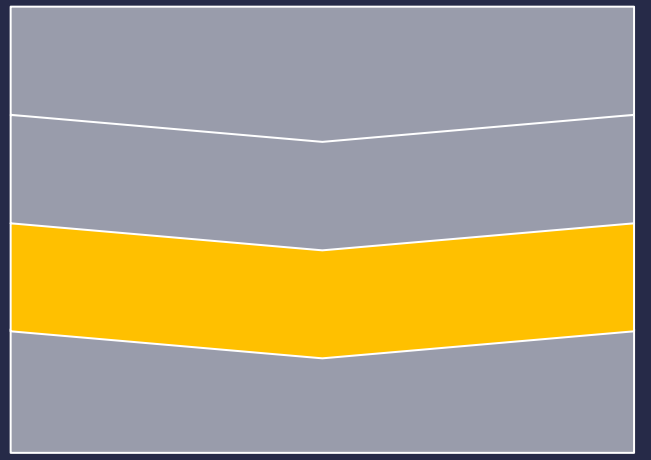
```
Utils.base64Key = "IgobAtWppFGw3O+dETgkjAec2ChdhaeaIl/ANJD8LnA=";
Utils.seliteBase64Key = "10cOen3cyJBcV7FJCYMcelB+CPVMJ+MTCywp3YCHg6I=";
Utils.dataEncryption = "AES/CBC/ISO10126Padding";
Utils.dataKeyType = "AES";
Utils.keyEncryption = "AESWRAP";
```

```
void * -[ EncryptionManager init](void * self, void * _cmd) {
    r7 = sp - 0x8;
    sp = sp - 0x34;
    r0 = [[sp + 0x1c super] init];
    stack[2045] = r0;
    objc_storeStrong(sp + 0x28, r0);
    if (stack[2045] != 0x0) {
            stack[2037] = [[NSData dataFromBase64String:@"10cOen3cyJBcV7FJCYMcelB+CPVMJ+MTCywp3YCHg6I=", _objc_msgSend,
            [stack[2045] setKekData:stack[2037], _objc_msgSend, stack[2035], _objc_msgSend];
            [stack[2037] release];
    }
    [stack[2045] retain];
    objc_storeStrong(sp + 0x28, 0x0);
    r0 = stack[2035];
    return r0;
}
```

# REMEDIATIONS

1. If you don't want it seen, don't hardcode it
   a. Keys, IVs, creds, and other sensitive data
2. Authentication and Authorization must be performed server side
3. Use free security
   a. Client side flags protect users
4. Remove extraneous functionality

# NETWORK TRAFFIC



# COMMON NETWORK SECURITY ISSUES

- HTTP Traffic
- MITM Issues
  - Certificate Validation
  - Hostname Verification
  - Certificate Pinning
- Third Party Endpoints
- Vulnerable Network Libraries

**1 in 5**

Android Apps use insecure HTTP

**1 in 7**

iOS Apps use insecure HTTP

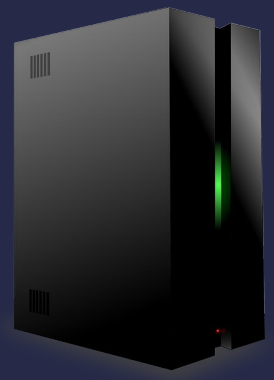
# TESTING NETWORK INTERACTIONS

- Use different MiTM environments
  - Different types of certs
- Test before and after login process
  - Be prepared to launch the proxy during different stages
- Exercise the entire app
  - Third party API or other content
- Look for sensitive data and interesting content types
  - Less work when testing the web API

# INTERCEPTION PROXY BASIC SETUP

Mitmproxy CA certificate
(optional)


Laptop w/ mitmproxy
Listening at ports 80 & 443
192.168.10.66



Device 192.168.10.15
Gateway set to 192.168.10.66

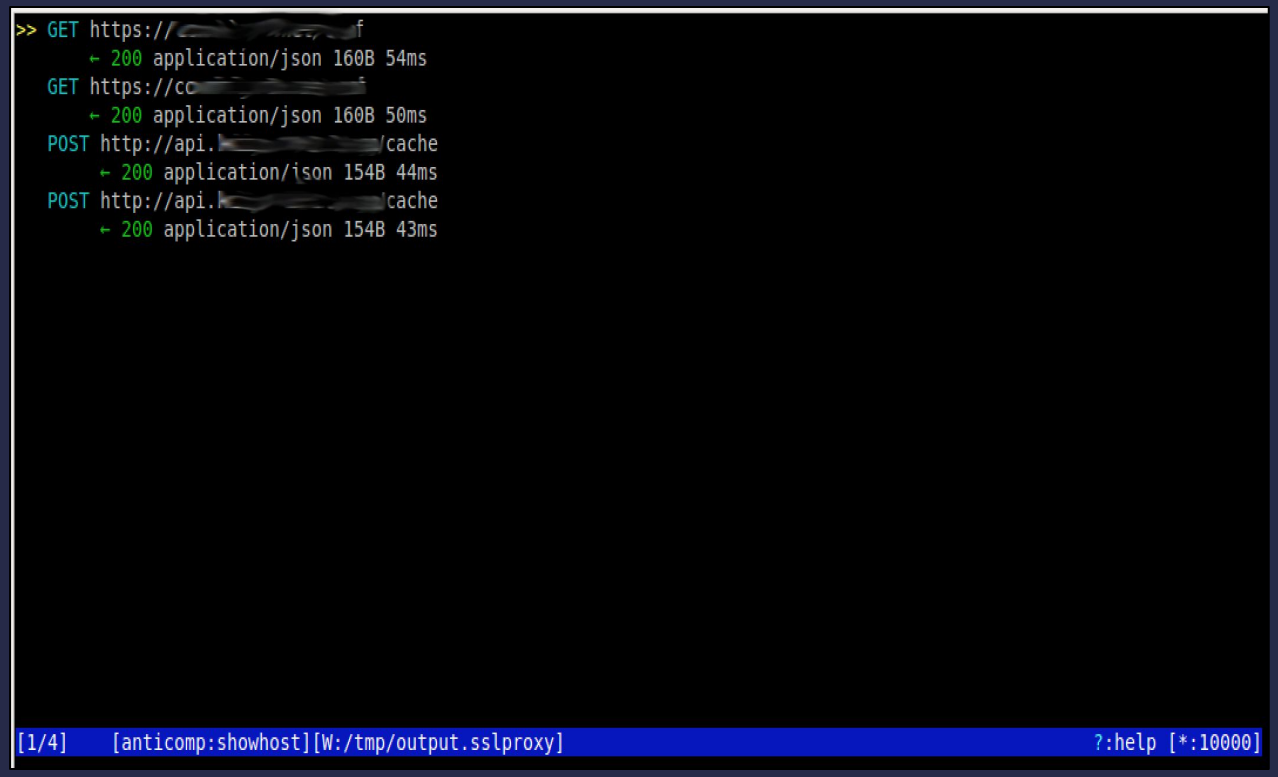
192.168.10.1

Backend Server

# REAL-WORLD EXAMPLE

## WebViews with MiTM issues

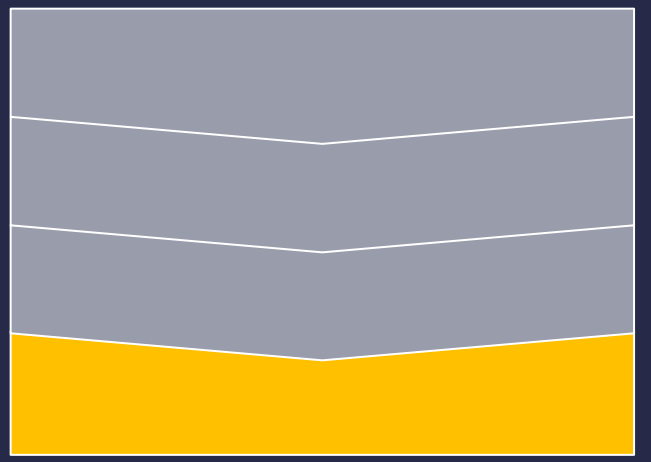
- In-app browser
- Arbitrary content
- JavaScript
- FAQ page can become a phishing vector



# REMEDIATIONS

1. Always use HTTPS because it's 2019
2. Ensure proper verification of certificates
3. Implement certificate pinning when transmitting highly sensitive PII over the network
4. Use up-to-date network libraries
5. Test your third party API requests

# BACKEND/API



# COMMON BACKEND/API ISSUES

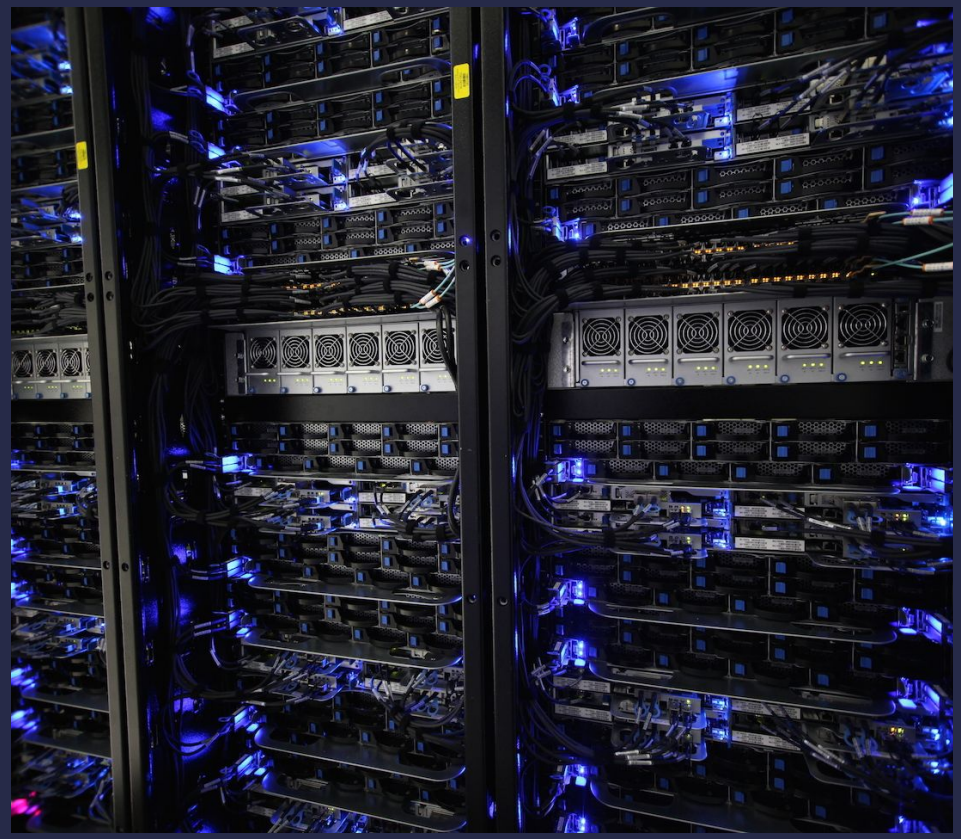
Insecure authentication

Insecure authorization

Session token issues

Lack of rate limiting

General web backend issues



# TESTING BACKEND/API

- Manipulate network traffic using interception proxy
  - Attempt to brute force important fields in the network request
  - Input arbitrary values and inspect server response (stack traces)
  - Fuzz URL for directory traversal etc.
- Exercise authentication and authorization functionalities
  - Test for session token replay, expiration
  - Try default admin credentials

# REAL-WORLD EXAMPLES

Serialized IDs used to enroll IoT devices -> easy Denial of Service

```
POST [redacted]/api/DeviceV2 HTTP/1.1
Host: [redacted]
Content-Type: application/json
Cookie:
.AspNet.Cookies=69rmEyw2ZFwhwq3jaqW_wfQtU_JMHQ35kMlrU4EhWaiBCX6E6MnA011WyxCS0YxKhdzPD7-PRdHjQjE
CtxqRGDbqRWKJT4ertAQfqflFN9dYxZXZFEtak5rB2DWC7VDMvQUdcNh-56UyCNyTM-fhlG0zKPRmZwrkOllxTwvOJlbnq3
PzZqeuwacqqlyneSG3kJwv3_-Q6y2HPeCOKO2mH9p-ekt0uqRCNuzAGCSTaVqccF99KmPWX2HXyPI23UytuiqpmOI59U8xJ
KA1MFxtUFRscC6twKN7kTtjCGkI-fFL9bt6dnL-2CauWMYInoO3z_UeurlDtSkjuU03jIYPwsi4yuUSDTnxo6k-iu89txfF
wgq_77wc3K7mxIZjvWzyCBmfOHHoTHhNHmluvseEdK6t2b3YcLg3qZcfGQoifGqHVUT7UcnqU8W_aua66WEtxPKk5GkT0yN
f6Oid-dKfrkykopUnMWFDhTFL9AE687o
User-Agent: TestApp/1 CFNetwork/758.5.3 Darwin/15.6.0
Connection: close
Accept: */*
Accept-Language: en-us
Content-Length: 204
Authorization: Bearer
hTThSgGtnHNMMf2VfIA_kyWczJHUGqRTq_lcJ6wK_o-OhcdoZodeJNPRSyctkXI0SqlsKpYZOmbgBs5BDa5sbv4cUwsuo4A
UKWlyOYjsvK9MtXIt2Di6575kye3FM8Z5ybwFOEnFuxZpirY3oqmLksOGvc0QVBftGaLoMrex7gvokbgPcKtWAah2PhaCl9
F699JfLqIWVN3anXUAAv3aJPwcKvGWlrZukTjAg5vEz9SZXPcaaa_lPQrncVOGoqLL_WHDOM9gyL9R5vXXuEJTbjEinenk4
xQPZwLO8PV4y7eK4NZgxcsy_9BeP8c05QWCalt90mGu410XgBMcY2J-xEgIt-7rB-9iCsldkYyI-eSfjANNQay0Eykhbx-d
0lLSiU7o7hisrWlE5WSxRTbJq7NkzoVhcyfs3WprnjZCdjtFIXeXKu7Qu6HoTd-EFXIXletO5D4xNaNsmArpWHszRWjPsa3
mUvo9ddT8-vzzNHrqO9CqSPGKFWCpYxPPmwAs4d7l

{"LastSeenAddress":"10990 N Stelling Rd, Cupertino, CA
95014","DeviceBLEId":"99EA11AC-9B26-290A-8760-982DF6280B9D","Latitude":"37.337383","ItemName":"
Tracker","Longitude":"-122.040973","DeviceId":"TEST_12345_09"}
```

```
HTTP/1.1 200 OK
Content-Length: 81
Content-Type: application/json; charset=utf-8
Server: [redacted]
X-Powered-By: ASP.NET
Date: [redacted]
Connection: close

{"IsAvailable":"0","Message":"TEST_12345_09 This device is already enrolled!"}
```

# REAL-WORLD EXAMPLES

Improper access control on Facebook external access tokens -> Unauthorized access

```
POST [redacted]/v1/api/account/RegisterExternal HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Type: application/json; charset=utf-8
User-Agent: Dalvik/2.1.0 (Linux; U; Android 6.0.1; Nexus 5 Build/MMB29Q)
Host: api.testapp.com
Connection: close
Accept-Encoding: gzip, deflate
Content-Length: 321

{"email":"nstestemail@gmail.com","fullname":"Stevie
Thunder","provider":"Facebook","externalaccesstoken":"EAAXWW86NPzACAOviCxhgkHF9oCHW2zZBGwV..."}
```

```
HTTP/1.1 200 OK
Content-Length: 562
Content-Type: application/json; charset=utf-8
Server: [redacted]
X-Powered-By: ASP.NET
Date: Tue, 18 Dec 2018 18:08:05 GMT
Connection: close

{"userName":"nstestemail@gmail.com","access_token":"yy5RCjWckgi_5tDS4nAnR1oIhOVVOrW8pmofm6hbhiC
M6Sy_rPEFFkTjGcE_kXmWsDPFM4AGjSJHNav14BChYjYwxnKRCFjQbL9bmosbAvEnB5l13PvigshPq8m6HIntxqokEKYWS5
YgPZrShQdaKVIHgaE70sx4fToYacGzI5tA-lbV9J8UpYbAf0YACRSGnxuyOiDGxDBOO6wDqkAO6-i2PQN5ytnzRzVSpJNPB
4xUEXt7-74osJULGqPYVKAyb0aTcq0IIpsHu67glUI-0QbNrzgXEXLyy3jR7tyKgFDo3KVE4D7iagIDpEPIY_bn1hiTqKbH
YBJ-UvW_EXxgYwiltDrGk2OSsPRjeCYZObsUNo4dl09f1I35Hdbc0anF","token_type":"bearer","expires_in":"3
1536000",".issued":"12/18/2018 6:08:05 PM +00:00",".expires":"12/18/2019 6:08:05 PM +00:00"}
```

```
HTTP/1.1 200 OK
Content-Length: 451
Content-Type: application/json; charset=utf-8
Server: [redacted]
X-Powered-By: ASP.NET
Date: Tue, 18 Dec 2018 18:10:06 GMT
Connection: close

{"UserId":"3c0984b5-d430-5f24-a1db-d9e6cc59a37b","FullName":"John
Malkovich","Address":null,"UserEmail":"nstestemail@gmail.com","UnitOfMeasure":"Meter","QuickFin
dAlert":"On","FirstName":"John","LastName":"Malkovich","ProfileURl":"/api/Image/ProfileImage/51
e50114-522f-45de-c71e-24d250586f9d","TutorialReadStatus":1,"CrowdFindStatus":1,"CardDetectionEn
abled":1,"CardDetectionDelay":5,"RequestedAppRating":false,"IsTestUser":false,"User":null,"Devi
ces":null}
```

https://developers.facebook.com/docs/facebook-login/manually-build-a-login-flow#checktoken

# REMEDIATIONS

1. Perform rate limiting on endpoints
2. Ensure proper session handling
3. Follow web backend best practices: https://www.owasp.org/index.php/Category:OWASP_Backend_Security_Project

# COMMUNICATING THE ISSUES

- Report needs to make sense to everyone
  - Security, Developers, and Management
- Attack Scenario
- Remediation

# TOP 5 TAKEAWAYS

1. Unnecessary data storage on device (writing to external storage or logs)
2. HTTP network traffic
3. Lack of hostname verification /certificate validation
4. Client-side logic
5. Mobile API security

# TRUSTED BY THE WORLD'S HIGHEST SECURITY ORGANIZATIONS


NowSecure™

- Automated Mobile AppSec Testing Software
- Expert Pen Testing & Security Services
- Powers Security in Agile & DevOps Teams
- World-Class Security Research Team (builders of FRIDA & RADARE)
- Advanced Engineering & DevOps Teams from High Frequency Trading Companies
- Wrote the book on mobile forensics

# Questions

Tony Ramirez

Mobile Security Analyst

Email: aramirez@nowsecure.com