Static Analysis for Android Applications to Detect Protected Health Information (PHI) and Analyze Intent Relationship

Ashfak Md Shibli
Tahiatul Islam

Everything secure?

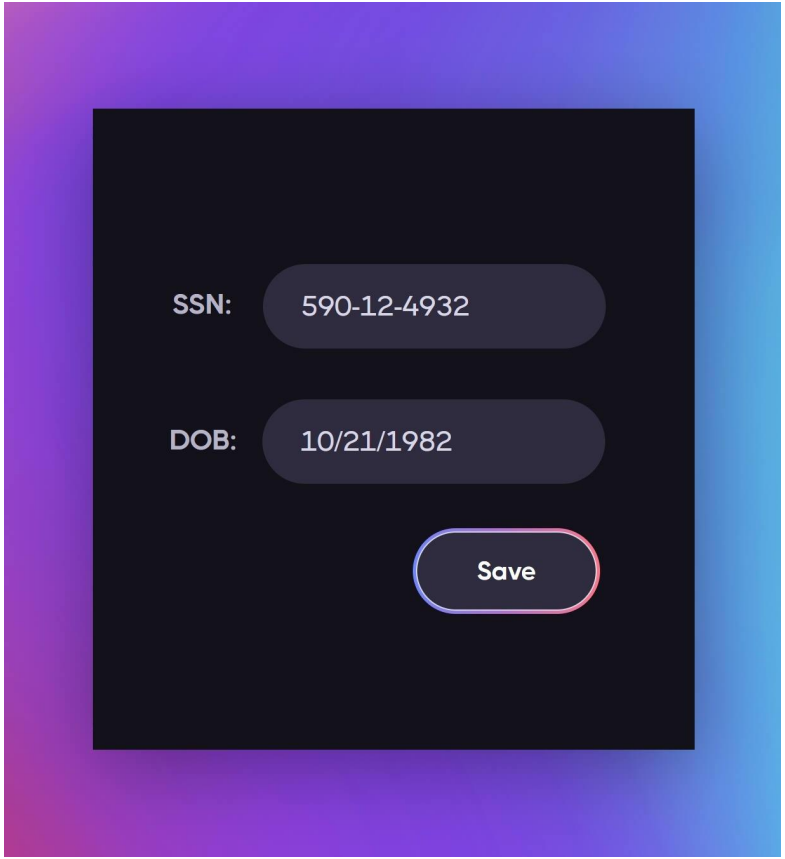
We think we are.

But there can be potential leak.

This makes things vulnerable.

Like giving data to attacker.

Same goes for data in the Apps.

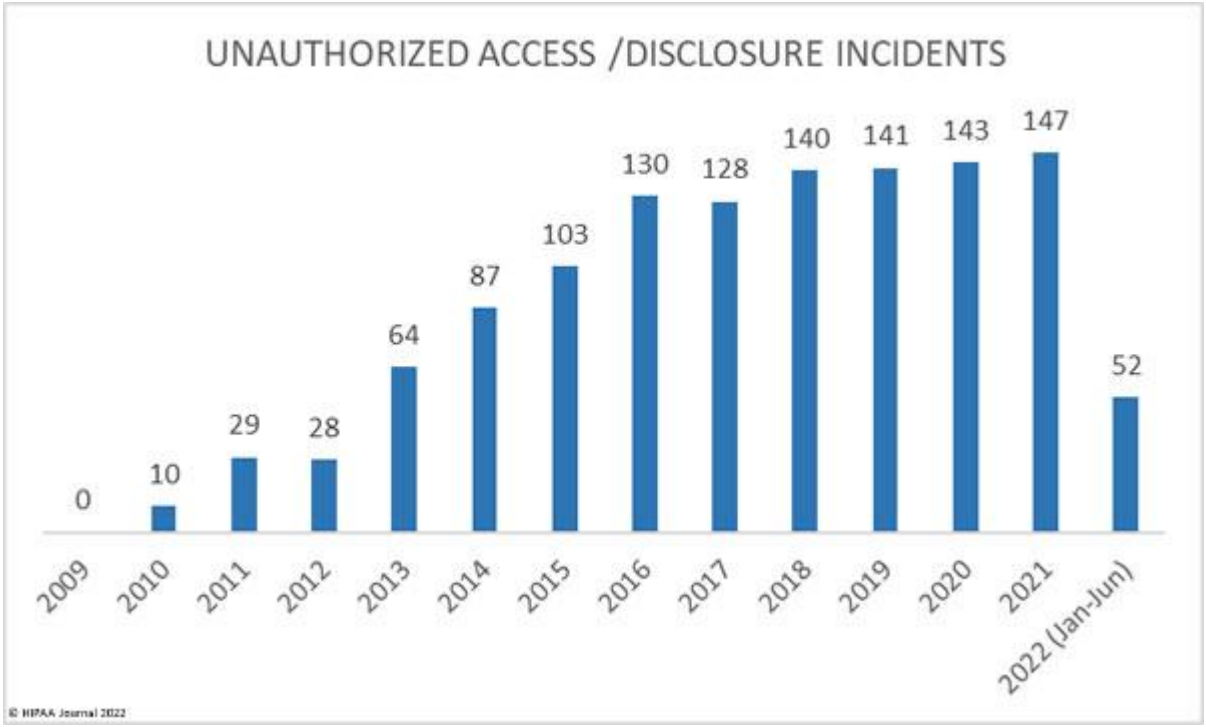


Source: Pinterest

Why Bother PHI Breach?

- Lead to stolen personal identity
- Face billing & treatment issue
- User trust lost
- Companies face penalties

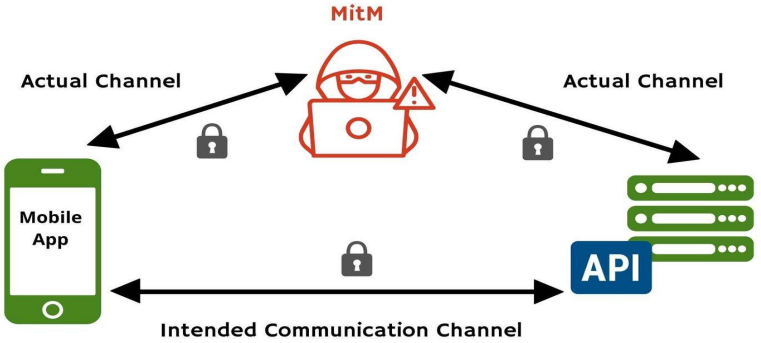
Recent Damage



Source: HIPAA Journal

How data leaks

- Uploading data insecurely to cloud
- Using insecure web clients
- Insecure ports



Source: approov.io

- Logging Secure Data
- Phishing attempt by Apps

```
# Beautiful code above

user_ssn = getSsn()
user_dob = getDOB()
Logger.Log("This is ssn: " + user_ssn + "This is birthdate: "+
user_dob)
# Beautiful code beyond
```

We have done something unexpected

Maybe photoshop it?

Can we do this when an app compromises data?

Before you get hit

Static Analysis - Taint - FlowDroid

→ Decompile the APK (Application Package)
→ Analyze the flow from Manifest+XML
→ Build Call graph using IFDS inter-procedural, finite, distributive, subset
→ Taint Analysis from Source to Sinks
→ Out of the box tool FlowDroid

PHI Mapping With Source/Sinks

PHI Data Point	Source/Sink Function
name	getAccounts| android.util.Log | sendTextMessage
address	android.util.Log | sendTextMessage
dates	android.util.Log | sendTextMessage|
phone number	getLine1Number|getVoiceMailNumber
email address	sendTextMessage|android.util.Log|
Social Security number	android.util.Log
medical record number	android.util.Log
health plan beneficiary number	android.util.Log
account number	android.util.Log
certificate or license number	android.util.Log
vehicle identifiers, such as serial numbers, license plate numbers	android.util.Log
device identifiers and serial numbers;	getDeviceId
web URL	getHost | android.util.Log | sendTextMessage
Internet Protocol (IP) address	getHost | getPort
biometric IDs, such as a fingerprint or voice print	getKey
full-face photographs and other photos of identifying characteristics	getKey
any other unique identifying characteristic	android.util.Log

Our Study

→ Analyze the application leakage
→ Empirically check the False Positives
→ Check the intent of developers
→ Analyze the possible exploit of Attacker
→ Determine relationships of Intents
→ Propose the countermeasure

What are we doing now

- Prepared A dataset of 50 Apps and counting
- We are analyzing using FlowDroid on every app
- Store leakage and processed flow data
- Determine relationship with leaks
- What is the developer intent?
- What is the Attacker Intent?

Example

- “Universal Remote Control” - 10M Downloads
- News says it has leakage
- We found leakage in Location data
- Wait what? Why it is leaking that data?
- May be a mistake/Disguised Malware Developer Intent
- An attacker exploit users location Attacker Exploit

Finally

→ Dataset with application insights and patterns

→ Empirically defined intent specifications

→ Data for future research

→ Data will help users and regulatory bodies too.