Here is some other examples of our demystification report.

Some of them has more than one API misuse scenario, i.e., example-6

Some have more than one patch in an API misuse scenario, i.e., example-2 and example-3

Some have natural language patches, in which we highlight (using bold, italic and underline) the API(s) in the erroneous code that are mentioned in the natural language patch, such as patch-1 in example-2, patch-1 in example-3 and patch in example-4

For others, we detect and highlight (on purple) the code differences between the erroneous code example and the corresponding code patch if their similarity is > 0.7.

Example-1:

Concerned API: Manifest.permission.BIND_ACCESSIBILITY_SERVICE

Concerned API usage directive: Must be required by an AccessibilityService , to ensure that only the system can bind to it

API misuse scenario: Android AccessibilityService compatible to Kindle Fire? (#25351463)

Erroneous code example:

```
 <uses-permission android:name="android.permission.BIND_NOTIFICATION_LISTENER_SERVICE"/>
<uses-permission android:name="android.permission.BIND_ACCESSIBILITY_SERVICE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-sdk
      android:minSdkVersion="10"
      android:targetSdkVersion="18"
/>
<application
      android:icon="@drawable/ic_launcher"
      android:allowBackup="true">
      <activity
            android:name="de.test.notificationdistributor.SettingsActivity"
            android:icon="@drawable/ic_launcher"
            android:label="@string/app_name">
            <intent-filter>
                  <action android:name="android.intent.action.MAIN" />
                  <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
      </activity>
      <service              android:name="de.test.notificationdistributor.NotificationDistributorService">
            <intent-filter>
                  <action android:name="android.service.notification.NotificationListenerService" />
            </intent-filter>
      </service>
      <service          android:name="de.test.notificationdistributor.NotificationDistributionDeprService">
            <intent-filter>
                  <action android:name="android.accessibilityservice.AccessibilityService" />
            </intent-filter>
      </service>
</application>
```

Patch:

```
<service
 android:name="de.test.notificationdistributor.NotificationDistributorService"
 android:permission="android.permission.BIND_ACCESSIBILITY_SERVICE">
 <meta-data android:name="android.accessibilityservice" android:resource="@xml/accessibilityservice"
/>
 <intent-filter>
       <action android:name="android.service.notification.NotificationListenerService" />
 </intent-filter>
```

Related API: android.accessibilityservice, android.service.notification.NotificationListenerService, android.accessibilityservice.AccessibilityService

Confusing API: None

Example-2:

Concerned API: Intent.ACTION_CHOOSER

Concerned API usage directive: In this case the CHOOSER action should be used, to always present to the user a list of the things they can do, with a nice title given by the caller such as \"Send this photo with:\"

API misuse scenario: Android direct shared (#35760008)

Erroneous code example:

```
Intent shareIntent = ShareCompat.IntentBuilder
                        .from(getActivity())
                        .setType(\"text/plain\")
                        .setText(sTitle+ \"\\n\" + urlPost)
                        .getIntent();
    if shareIntent.resolveActivity(
                        getActivity().getPackageManager()) != null)
            startActivity(shareIntent);
```

Patch-1: You should use .createChooserIntent() instead of ***.getIntent()***

Patch-2:

```
Intent sharingIntent = new Intent(Intent.ACTION_SEND);
Uri screenshotUri = Uri.parse("file://" + filePath);
sharingIntent.setType("image/png");
sharingIntent.putExtra(Intent.EXTRA_STREAM, screenshotUri);
startActivity(Intent.createChooser(sharingIntent, "Share image using"));
```

Related API: Intent.ACTION_SEND, setType, putExtra, createChooser, EXTRA_TEXT

Confusing API: setAction

Example-3:

Concerned API: Intent.FLAG_ACTIVITY_NEW_DOCUMENT

Concerned API usage directive: whether the recents entry for it is kept after the activity is finished is different than the use of FLAG_ACTIVITY_NEW_TASK and R.attr.documentLaunchMode -- if this flag is being used to create a new recents entry, then by default that entry will be removed once the activity is finished

API misuse scenario: How to create the same Activity Multiple times to have an effect like Google Chrome Tabs? (#38119671)

Erroneous code example:

```
Intent intent=new Intent(this,    MainActivity.class);
intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_MULTIPLE_TASK);
startActivity(intent);
```

Patch-1:

Use this replace the flag new_task with new_document

```
Intent intent=new Intent(this, MainActivity.class);
intent.setFlags(Intent.FLAG_ACTIVITY_NEW_DOCUMENT |
Intent.FLAG_ACTIVITY_MULTIPLE_TASK);
startActivity(intent);
```

Patch-2:

```
final Intent newDocumentIntent = new Intent(this, NewDocumentActivity.class);
newDocumentIntent.addFlags(Intent.FLAG_ACTIVITY_NEW_DOCUMENT);
newDocumentIntent.addFlags(Intent.FLAG_ACTIVITY_MULTIPLE_TASK);
startActivity(newDocumentIntent);
```

Related API: Intent.FLAG_ACTIVITY_MULTIPLE_TASK

Confusing API: Intent. FLAG_ACTIVITY_NEW_TASK

Example-4:

Concerned API: r.attr.scheme

Concerned API usage directive: As a result, schemes here should always use lower case letters

API misuse scenario:Start browser via intent, url with schema “HTTP” (uppercase) error (#27251456)

Erroneous code example:

startActivity(new Intent(Intent.ACTION_VIEW, ***Uri***.parse(url)));

Patch:

RFC 2396, sec. 3.1 says:

Scheme names consist of a sequence of characters beginning with a lower case letter and followed by any combination of lower case letters, digits, plus ("+"), period ("."), or hyphen ("-"). For resiliency, programs interpreting ***URI*** should treat upper case letters as equivalent to lower case in scheme names (e.g., allow "HTTP" as well as "http").

I.e. uppercase "HTTP" is incorrect according to the spec. Although programs should treat uppercase as equivalent to lower case, it does not have to.

It's also trivial for you to make sure that the scheme part of your URI is in lowercase, so it's easy to avoid.

Related API: None

Confusing API: None

Example-5：

Concerned API: String.replaceAll

Concerned API usage directive: replaceAll (repl) Note that backslashes (\ ) and dollar signs ($) in the replacement string may cause the results to be different than if it were being treated as a literal replacement string; see Matcher.replaceAll

API misuse scenario: Android Pattern having problem with $ sign(#7251834)

Erroneous code example:

```
Pattern pattern = Pattern.compile("(\\{#\\})");
  String raw_q = "Pete has {#}. He bought {#}.";
  String[] var = ["$70","6 pens at $5 each"];
  String act_q = "";
  for (int m = 0; m < var.length(); m++) {
       try{
            act_q = raw_q.replaceFirst(pattern.pattern(),var[m]);
       }catch(Exception e){
            e.printStackTrace();
}
```

Patch:

```
Matcher m = pattern.matcher(raw_q);
StringBuffer sb = new StringBuffer();
int i = 0;
while (m.find()) {
        m.appendReplacement(sb, var[i].replaceAll("\\$", "\\\\$"));
        i++;
}
m.appendTail(sb);
act_q = sb.toString();
```

Related API: matcher, appendReplacement

Confusing API compile, replaceFirst

Example-6：

Concerned API: R.attr.fillAfter

Concerned API usage directive: When set to true, the animation transformation is applied after the animation is over

API misuse scenario-1: onAnimationEnd is not getting called, onAnimationStart works fine (#5474923)

Erroneous code example:

```
mToolbar = mPopupContents.findViewById( R.web.toolbar );
TranslateAnimation anim = new TranslateAnimation(0, 0, -60, 0);
anim.setDuration(1000);
anim.setAnimationListener(new Animation.AnimationListener() {
    public void onAnimationStart(Animation a) {
        Log.d(LOGTAG, "---- animation start listener called"    );
    }
    public void onAnimationRepeat(Animation a) {}
    public void onAnimationEnd(Animation a) {
        Log.d(LOGTAG, "---- animation end listener called"    );
    }
});
mToolbar.startAnimation(anim);
```

Patch:

```
final FadeUpAnimation anim = new FadeUpAnimation(v);
    anim.setInterpolator(new AccelerateInterpolator());
    anim.setDuration(1000);
    anim.setFillAfter(true);
    new Handler().postDelayed(new Runnable() {
        public void run() {
            v.clearAnimation();
           //Extra work goes here
        }
    }, anim.getDuration());
    v.startAnimation(anim);
```

Related API: None

Confusing API: None

API misuse scenario-2: Android: Rotate animation get back to its real state after finishing the animation? (#10203577)

Erroneous code example:

```
<?xml version="1.0" encoding="utf-8"?>
<rotate xmlns:android="http://schemas.android.com/apk/res/android"
    android:pivotX="50%"
    android:pivotY="50%"
    android:fromDegrees="0"
    android:toDegrees="30"
    android:duration="2000">
</rotate>
```

Patch:

```
android:fillAfter="true"
    android:fillEnabled="true"
```

Related API: None

Confusing API: None