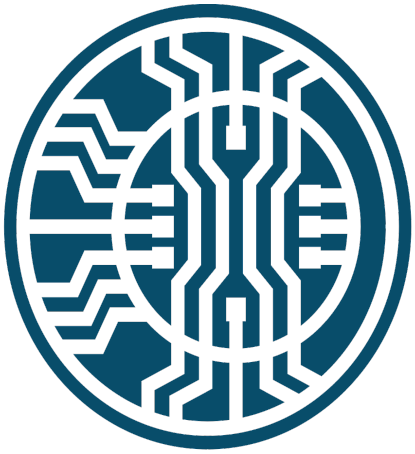

Ostorlab

Mobile Security Report

14 May 2020

Aphelion Labs

Table of Contents

Executive summary

Application summary

Platform: android

Package: ar.gob.coronavirus

Version: 3.0.7

SHA1 Hash: 3d014e8a04fe0fef6d7357a49e3ee47b4e6d2324

Size: 9 MB

Scan summary

Date: May 12, 2020, 3:51 p.m.

Findings list

Risk	CVSS v3 Score	Title	Short description
Potentially	6.4	Intent Spoofing	The application is vulnerable to intent spoofing which may lead to inappropriate access like data modification, information disclosure and data injection
Potentially	3.9	Backup mode enabled	Application is enabling backup mode
Hardening	4.0	Application code not obfuscated	Application's source code is not obfuscated and could be decompiled to retrieve the initial source code
Secure	-	Secure Network Configuration Settings	The application specifies a secure network security config
Secure	-	Debug mode disabled	Application is compiled with debug mode disabled

Potentially Intent Spoofing

Description

Intent Spoofing consists of sending an intent toward an application components (Exported Activity, Broadcast Receiver, Content Provider, Service) to achieve unauthorized access.

The access may to different objectives like unauthorized data modification and information leakage, untrusted input injection, etc.

Recommendation

To limit one's exposure to this type of attack, developers should avoid exporting components unless the component is specifically designed to handle requests from untrusted applications. Developers should be aware that declaring an intent filter will automatically export the component, exposing it to public access. Critical, state-changing actions should not be placed in exported components.

If a single component handles both inter- and intra-application requests, the developer should consider dividing that component into separate components. If a component must be exported (e.g., to receive system broadcasts), then the component should dynamically check the caller's identity prior to performing any operations. Requiring Signature or SignatureOrSystem permissions is an effective way of limiting a component's exposure to a set of trusted applications. Finally, the return values of exported components can also leak private data, so developers should check the caller's identity prior to returning sensitive values.

References

- Do not act on malicious intent (CERT Secure Coding)
- Improper Access Control (CWE-284)
- Intent Spoof (CAPEC-502)
- Analyzing Inter-Application Communication in Android

Technical details

Use of a string value `com.google.zxing.client.android.SCAN` to construct an Intent

Taint is traced from string `com.google.zxing.client.android.SCAN` to sink method `b.m.d.cstartActivityFromFragment()`

Taint trace:

```
at com.globant.pasaportesanitario.flujos.identificacion.IdentificacionDniManualFragment$7.onClick()
at c.d.c.t.a.a.a()
at android.content.Intent.setAction()
at c.d.c.t.a.a.a()
at androidx.fragment.app.Fragment.startActivityForResult()
at androidx.fragment.app.Fragment.startActivityForResult()
at b.m.d.c.startActivityFromFragment()
```

Method `com.globant.pasaportesanitario.flujos.identificacion.IdentificacionDniManualFragment$7.onClick()` :

```
public void onClick(android.view.View p3)
{
    com.globant.pasaportesanitario.flujos.identificacion.IdentificacionDniManualFragment.access$500(this.this$0).setVisibility(8);
    if (com.globant.pasaportesanitario.flujos.identificacion.IdentificacionDniManualFragment.access$1400(this.this$0).booleanValue()) {
        com.globant.pasaportesanitario.flujos.identificacion.IdentificacionDniManualFragment v3_1 = this.this$0;
        c.d.c.t.a.a v0_1 = new c.d.c.t.a.a(v3_1.getActivity());
        v0_1.b = v3_1;
        v0_1.a();
    }
    return;
}
```

Method `c.d.c.t.a.a.a()` :

```
public final void a()
{
    int v1_0 = this.a;
    if (this.d == null) {
        this.d = com.journeyapps.barcodescanner.CaptureActivity;
    }
    android.content.Intent v0_1 = new android.content.Intent(v1_0, this.d);
    v0_1.setAction(com.google.zxing.client.android.SCAN);
    v0_1.addFlags(67108864);
    v0_1.addFlags(524288);
    int v1_5 = this.c.entrySet().iterator();
    while (v1_5.hasNext()) {
        String v2_5 = ((java.util.Map$Entry) v1_5.next());
        String v3_1 = ((String) v2_5.getKey());
        String v2_6 = v2_5.getValue();
        if (!(v2_6 instanceof Integer)) {
            if (!(v2_6 instanceof Long)) {
                if (!(v2_6 instanceof Boolean)) {
                    if (!(v2_6 instanceof Double)) {
                        if (!(v2_6 instanceof Float)) {
                            if (!(v2_6 instanceof android.os.Bundle)) {
                                v0_1.putExtra(v3_1, v2_6.toString());
                            } else {
                                v0_1.putExtra(v3_1, ((android.os.Bundle) v2_6));
                            }
                        } else {
                            v0_1.putExtra(v3_1, ((Float) v2_6));
                        }
                    } else {
                        v0_1.putExtra(v3_1, ((Double) v2_6));
                    }
                } else {
                    v0_1.putExtra(v3_1, ((Boolean) v2_6));
                }
            } else {
                v0_1.putExtra(v3_1, ((Long) v2_6));
            }
        } else {
            v0_1.putExtra(v3_1, ((Integer) v2_6));
        }
    }
    String v2_2 = this.b;
    if (v2_2 == null) {
        this.a.startActivityForResult(v0_1, 49374);
    } else {
        v2_2.startActivityForResult(v0_1, 49374);
    }
    return;
}
```

Method `android.content.Intent.setAction()` not found.

Method `c.d.c.t.a.a.a()`:

```
public final void a()
{
    int v1_0 = this.a;
    if (this.d == null) {
        this.d = com.journeyapps.barcodescanner.CaptureActivity;
    }
    android.content.Intent v0_1 = new android.content.Intent(v1_0, this.d);
    v0_1.setAction(com.google.zxing.client.android.SCAN);
    v0_1.addFlags(67108864);
    v0_1.addFlags(524288);
    int v1_5 = this.c.entrySet().iterator();
    while (v1_5.hasNext()) {
        String v2_5 = ((java.util.Map$Entry) v1_5.next());
        String v3_1 = ((String) v2_5.getKey());
        String v2_6 = v2_5.getValue();
        if (!(v2_6 instanceof Integer)) {
            if (!(v2_6 instanceof Long)) {
                if (!(v2_6 instanceof Boolean)) {
                    if (!(v2_6 instanceof Double)) {
                        if (!(v2_6 instanceof Float)) {
                            if (!(v2_6 instanceof android.os.Bundle)) {
                                v0_1.putExtra(v3_1, v2_6.toString());
                            } else {
                                v0_1.putExtra(v3_1, ((android.os.Bundle) v2_6));
                            }
                        } else {
                            v0_1.putExtra(v3_1, ((Float) v2_6));
                        }
                    } else {
                        v0_1.putExtra(v3_1, ((Double) v2_6));
                    }
                } else {
                    v0_1.putExtra(v3_1, ((Boolean) v2_6));
                }
            } else {
                v0_1.putExtra(v3_1, ((Long) v2_6));
            }
        } else {
            v0_1.putExtra(v3_1, ((Integer) v2_6));
        }
    }
    String v2_2 = this.b;
    if (v2_2 == null) {
        this.a.startActivityForResult(v0_1, 49374);
    } else {
        v2_2.startActivityForResult(v0_1, 49374);
    }
    return;
}
```

Method `androidx.fragment.app.Fragment.startActivityForResult()`:

```
public void startActivityForResult(android.content.Intent p2, int p3)
{
    this.startActivityForResult(p2, p3, 0);
    return;
}
```

Method `androidx.fragment.app.Fragment.startActivityForResult()`:

```
public void startActivityForResult(android.content.Intent p2, int p3, android.os.Bundle p4)
{
    b.m.d.c v0_0 = this.mHost;
    if (v0_0 == null) {
        throw new IllegalStateException(c.a.a.a.a.a(Fragment , this,  not attached to Activity));
    } else {
        ((b.m.d.c$a) v0_0).f.startActivityFromFragment(this, p2, p3, p4);
        return;
    }
}
```

Method `b.m.d.c.startActivityFromFragment()` :

```
public void startActivityFromFragment(androidx.fragment.app.Fragment p4, android.content.Intent p5, int p6, andr
oid.os.Bundle p7)
{
    this.mStartedActivityFromFragment = 1;
    try {
        if (p6 != -1) {
            b.m.d.c.checkForValidRequestCode(p6);
            b.h.e.a.a(this, p5, (((this.allocateRequestIndex(p4) + 1) << 16) + (p6 & 65535)), p7);
            this.mStartedActivityFromFragment = 0;
            return;
        } else {
            b.h.e.a.a(this, p5, -1, p7);
            this.mStartedActivityFromFragment = 0;
            return;
        }
    } catch (Throwable v4_2) {
        this.mStartedActivityFromFragment = 0;
        throw v4_2;
    }
}
```

Use of a string value `com.google.zxing.client.android.SCAN` to construct an Intent

Taint is traced from string `com.google.zxing.client.android.SCAN` to sink method `android.app.ActivitystartActivityForResult()`

Taint trace:

```
at com.globant.pasaportesanitario.flujos.identificacion.IdentificacionDniManualFragment$7.onClick()
at c.d.c.t.a.a.a()
at android.content.Intent.setAction()
at c.d.c.t.a.a.a()
at android.app.Activity.startActivityForResult()
```

Method `com.globant.pasaportesanitario.flujos.identificacion.IdentificacionDniManualFragment$7.onClick()` :

```
public void onClick(android.view.View p3)
{
    com.globant.pasaportesanitario.flujos.identificacion.IdentificacionDniManualFragment.access$500(this.this$0).setVisi
bility(8);
    if (com.globant.pasaportesanitario.flujos.identificacion.IdentificacionDniManualFragment.access$1400(this.this$0).bo
oleanValue()) {
        com.globant.pasaportesanitario.flujos.identificacion.IdentificacionDniManualFragment v3_1 = this.this$0;
        c.d.c.t.a.a v0_1 = new c.d.c.t.a.a(v3_1.getActivity());
        v0_1.b = v3_1;
        v0_1.a();
    }
    return;
}
```

Method `c.d.c.t.a.a.a()` :

```
public final void a()
{
    int v1_0 = this.a;
    if (this.d == null) {
        this.d = com.journeyapps.barcodescanner.CaptureActivity;
    }
    android.content.Intent v0_1 = new android.content.Intent(v1_0, this.d);
    v0_1.setAction(com.google.zxing.client.android.SCAN);
    v0_1.addFlags(67108864);
    v0_1.addFlags(524288);
    int v1_5 = this.c.entrySet().iterator();
    while (v1_5.hasNext()) {
        String v2_5 = ((java.util.Map$Entry) v1_5.next());
        String v3_1 = ((String) v2_5.getKey());
        String v2_6 = v2_5.getValue();
        if (!(v2_6 instanceof Integer)) {
            if (!(v2_6 instanceof Long)) {
                if (!(v2_6 instanceof Boolean)) {
                    if (!(v2_6 instanceof Double)) {
                        if (!(v2_6 instanceof Float)) {
                            if (!(v2_6 instanceof android.os.Bundle)) {
                                v0_1.putExtra(v3_1, v2_6.toString());
                            } else {
                                v0_1.putExtra(v3_1, ((android.os.Bundle) v2_6));
                            }
                        } else {
                            v0_1.putExtra(v3_1, ((Float) v2_6));
                        }
                    } else {
                        v0_1.putExtra(v3_1, ((Double) v2_6));
                    }
                } else {
                    v0_1.putExtra(v3_1, ((Boolean) v2_6));
                }
            } else {
                v0_1.putExtra(v3_1, ((Long) v2_6));
            }
        } else {
            v0_1.putExtra(v3_1, ((Integer) v2_6));
        }
    }
    String v2_2 = this.b;
    if (v2_2 == null) {
        this.a.startActivityForResult(v0_1, 49374);
    } else {
        v2_2.startActivityForResult(v0_1, 49374);
    }
    return;
}
```

Method `android.content.Intent.setAction()` not found.

Method `c.d.c.t.a.a.a()` :

```
public final void a()
{
    int v1_0 = this.a;
    if (this.d == null) {
        this.d = com.journeyapps.barcodescanner.CaptureActivity;
    }
    android.content.Intent v0_1 = new android.content.Intent(v1_0, this.d);
    v0_1.setAction(com.google.zxing.client.android.SCAN);
    v0_1.addFlags(67108864);
    v0_1.addFlags(524288);
    int v1_5 = this.c.entrySet().iterator();
    while (v1_5.hasNext()) {
        String v2_5 = ((java.util.Map$Entry) v1_5.next());
        String v3_1 = ((String) v2_5.getKey());
        String v2_6 = v2_5.getValue();
        if (!(v2_6 instanceof Integer)) {
            if (!(v2_6 instanceof Long)) {
                if (!(v2_6 instanceof Boolean)) {
                    if (!(v2_6 instanceof Double)) {
                        if (!(v2_6 instanceof Float)) {
                            if (!(v2_6 instanceof android.os.Bundle)) {
                                v0_1.putExtra(v3_1, v2_6.toString());
                            } else {
                                v0_1.putExtra(v3_1, ((android.os.Bundle) v2_6));
                            }
                        } else {
                            v0_1.putExtra(v3_1, ((Float) v2_6));
                        }
                    } else {
                        v0_1.putExtra(v3_1, ((Double) v2_6));
                    }
                } else {
                    v0_1.putExtra(v3_1, ((Boolean) v2_6));
                }
            } else {
                v0_1.putExtra(v3_1, ((Long) v2_6));
            }
        } else {
            v0_1.putExtra(v3_1, ((Integer) v2_6));
        }
    }
    String v2_2 = this.b;
    if (v2_2 == null) {
        this.a.startActivityForResult(v0_1, 49374);
    } else {
        v2_2.startActivityForResult(v0_1, 49374);
    }
    return;
}
```

Method `android.app.Activity.startActivityForResult()` not found.

Use of a string value `com.google.zxing.client.android.SCAN` to construct an Intent

Taint is traced from string `com.google.zxing.client.android.SCAN` to sink method `android.content.Intent<init>()`

Taint trace:

```
at c.e.a.f$a$a.run()
at android.content.Intent.<init>()
```

Method `c.e.a.f$a$a.run()`:

```
public void run()
{
    String v2_7;
    android.app.Activity v0_1 = this.c.a;
    java.util.Iterator v1_7 = this.b;
    int v3 = 0;
    if (!v0_1.d) {
```

```
            v2_7 = 0;
        } else {
            byte[] v6_8;
            String v2_6 = v1_7.b;
            String v5_1 = v2_6.f;
            if ((v2_6.e % 180) == 0) {
                v6_8 = 0;
            } else {
                v6_8 = 1;
            }
            if (v6_8 != null) {
                v5_1 = new android.graphics.Rect(v5_1.top, v5_1.left, v5_1.bottom, v5_1.right);
            }
            int v12_0 = new android.graphics.YuvImage;
            v12_0(v2_6.a, v2_6.d, v2_6.b, v2_6.c, 0);
            byte[] v6_24 = new java.io.ByteArrayOutputStream();
            v12_0.compressToJpeg(v5_1, 90, v6_24);
            String v5_2 = v6_24.toByteArray();
            byte[] v6_26 = new android.graphics.BitmapFactory$Options();
            v6_26.inSampleSize = 2;
            String v8_3 = com.newrelic.agent.android.instrumentation.BitmapFactoryInstrumentation.decodeByteArray(v5_2, 0,
v5_2.length, v6_26);
            if (v2_6.e != 0) {
                android.graphics.Matrix v13_1 = new android.graphics.Matrix();
                v13_1.postRotate(((float) v2_6.e));
                v8_3 = android.graphics.Bitmap.createBitmap(v8_3, 0, 0, v8_3.getWidth(), v8_3.getHeight(), v13_1, 0);
            }
            try {
                String v2_5 = java.io.File.createTempFile(barcodeimage, .jpg, v0_1.a.getCacheDir());
                String v5_6 = new java.io.FileOutputStream(v2_5);
                v8_3.compress(android.graphics.Bitmap$CompressFormat.JPEG, 100, v5_6);
                v5_6.close();
                v2_7 = v2_5.getAbsolutePath();
            } catch (String v2_8) {
                byte[] v6_31 = new StringBuilder();
                v6_31.append(Unable to create temporary file and store bitmap! );
                v6_31.append(v2_8);
                android.util.Log.w(f, v6_31.toString());
            }
        }
        String v5_9 = new android.content.Intent(com.google.zxing.client.android.SCAN);
        v5_9.addFlags(524288);
        v5_9.putExtra(SCAN_RESULT, v1_7.a.a);
        v5_9.putExtra(SCAN_RESULT_FORMAT, v1_7.a.d.toString());
        byte[] v6_2 = v1_7.a.b;
        if ((v6_2 != null) && (v6_2.length > 0)) {
            v5_9.putExtra(SCAN_RESULT_BYTES, v6_2);
        }
        java.util.Iterator v1_1 = v1_7.a.e;
        if (v1_1 != null) {
            if (v1_1.containsKey(c.d.c.m.i)) {
                v5_9.putExtra(SCAN_RESULT_UPC_EAN_EXTENSION, v1_1.get(c.d.c.m.i).toString());
            }
            byte[] v6_11 = ((Number) v1_1.get(c.d.c.m.c));
            if (v6_11 != null) {
                v5_9.putExtra(SCAN_RESULT_ORIENTATION, v6_11.intValue());
            }
            byte[] v6_15 = ((String) v1_1.get(c.d.c.m.e));
            if (v6_15 != null) {
                v5_9.putExtra(SCAN_RESULT_ERROR_CORRECTION_LEVEL, v6_15);
            }
            java.util.Iterator v1_3 = ((Iterable) v1_1.get(c.d.c.m.d));
            if (v1_3 != null) {
                java.util.Iterator v1_4 = v1_3.iterator();
                while (v1_4.hasNext()) {
                    byte[] v6_20 = ((byte[]) v1_4.next());
                    String v7_7 = new StringBuilder();
                    v7_7.append(SCAN_RESULT_BYTE_SEGMENTS_);
                    v7_7.append(v3);
                    v5_9.putExtra(v7_7.toString(), v6_20);
                    v3++;
                }
            }
```

```
        }
    }
    if (v2_7 != null) {
        v5_9.putExtra(SCAN_RESULT_IMAGE_PATH, v2_7);
    }
    v0_1.a.setResult(-1, v5_9);
    v0_1.a.finish();
    return;
}
```

Method `android.content.Intent.<init>()` not found.

Use of a string value `com.google.zxing.client.android.SCAN` to construct an Intent

Taint is traced from string `com.google.zxing.client.android.SCAN` to sink method `b.m.d.cstartActivityFromFragment()`

Taint trace:

```
at c.d.c.t.a.a.a()
at android.content.Intent.setAction()
at c.d.c.t.a.a.a()
at androidx.fragment.app.Fragment.startActivityForResult()
at androidx.fragment.app.Fragment.startActivityForResult()
at b.m.d.c.startActivityFromFragment()
```

Method `c.d.c.t.a.a.a()`:

```
public final void a()
{
    int v1_0 = this.a;
    if (this.d == null) {
        this.d = com.journeyapps.barcodescanner.CaptureActivity;
    }
    android.content.Intent v0_1 = new android.content.Intent(v1_0, this.d);
    v0_1.setAction(com.google.zxing.client.android.SCAN);
    v0_1.addFlags(67108864);
    v0_1.addFlags(524288);
    int v1_5 = this.c.entrySet().iterator();
    while (v1_5.hasNext()) {
        String v2_5 = ((java.util.Map$Entry) v1_5.next());
        String v3_1 = ((String) v2_5.getKey());
        String v2_6 = v2_5.getValue();
        if (!(v2_6 instanceof Integer)) {
            if (!(v2_6 instanceof Long)) {
                if (!(v2_6 instanceof Boolean)) {
                    if (!(v2_6 instanceof Double)) {
                        if (!(v2_6 instanceof Float)) {
                            if (!(v2_6 instanceof android.os.Bundle)) {
                                v0_1.putExtra(v3_1, v2_6.toString());
                            } else {
                                v0_1.putExtra(v3_1, ((android.os.Bundle) v2_6));
                            }
                        } else {
                            v0_1.putExtra(v3_1, ((Float) v2_6));
                        }
                    } else {
                        v0_1.putExtra(v3_1, ((Double) v2_6));
                    }
                } else {
                    v0_1.putExtra(v3_1, ((Boolean) v2_6));
                }
            } else {
                v0_1.putExtra(v3_1, ((Long) v2_6));
            }
        } else {
            v0_1.putExtra(v3_1, ((Integer) v2_6));
        }
    }
    String v2_2 = this.b;
    if (v2_2 == null) {
        this.a.startActivityForResult(v0_1, 49374);
    } else {
        v2_2.startActivityForResult(v0_1, 49374);
    }
    return;
}
```

Method `android.content.Intent.setAction()` not found.

Method `c.d.c.t.a.a.a()` :

```
public final void a()
{
    int v1_0 = this.a;
    if (this.d == null) {
        this.d = com.journeyapps.barcodescanner.CaptureActivity;
    }
    android.content.Intent v0_1 = new android.content.Intent(v1_0, this.d);
    v0_1.setAction(com.google.zxing.client.android.SCAN);
    v0_1.addFlags(67108864);
    v0_1.addFlags(524288);
    int v1_5 = this.c.entrySet().iterator();
    while (v1_5.hasNext()) {
        String v2_5 = ((java.util.Map$Entry) v1_5.next());
        String v3_1 = ((String) v2_5.getKey());
        String v2_6 = v2_5.getValue();
        if (!(v2_6 instanceof Integer)) {
            if (!(v2_6 instanceof Long)) {
                if (!(v2_6 instanceof Boolean)) {
                    if (!(v2_6 instanceof Double)) {
                        if (!(v2_6 instanceof Float)) {
                            if (!(v2_6 instanceof android.os.Bundle)) {
                                v0_1.putExtra(v3_1, v2_6.toString());
                            } else {
                                v0_1.putExtra(v3_1, ((android.os.Bundle) v2_6));
                            }
                        } else {
                            v0_1.putExtra(v3_1, ((Float) v2_6));
                        }
                    } else {
                        v0_1.putExtra(v3_1, ((Double) v2_6));
                    }
                } else {
                    v0_1.putExtra(v3_1, ((Boolean) v2_6));
                }
            } else {
                v0_1.putExtra(v3_1, ((Long) v2_6));
            }
        } else {
            v0_1.putExtra(v3_1, ((Integer) v2_6));
        }
    }
    String v2_2 = this.b;
    if (v2_2 == null) {
        this.a.startActivityForResult(v0_1, 49374);
    } else {
        v2_2.startActivityForResult(v0_1, 49374);
    }
    return;
}
```

Method `androidx.fragment.app.Fragment.startActivityForResult()` :

```
public void startActivityForResult(android.content.Intent p2, int p3)
{
    this.startActivityForResult(p2, p3, 0);
    return;
}
```

Method `androidx.fragment.app.Fragment.startActivityForResult()` :

```
public void startActivityForResult(android.content.Intent p2, int p3, android.os.Bundle p4)
{
    b.m.d.c v0_0 = this.mHost;
    if (v0_0 == null) {
        throw new IllegalStateException(c.a.a.a.a.a(Fragment , this,  not attached to Activity));
    } else {
        ((b.m.d.c$a) v0_0).f.startActivityFromFragment(this, p2, p3, p4);
        return;
    }
}
```

Method `b.m.d.c.startActivityFromFragment()` :

```
public void startActivityFromFragment(androidx.fragment.app.Fragment p4, android.content.Intent p5, int p6, andr
oid.os.Bundle p7)
{
    this.mStartedActivityFromFragment = 1;
    try {
        if (p6 != -1) {
            b.m.d.c.checkForValidRequestCode(p6);
            b.h.e.a.a(this, p5, (((this.allocateRequestIndex(p4) + 1) << 16) + (p6 & 65535)), p7);
            this.mStartedActivityFromFragment = 0;
            return;
        } else {
            b.h.e.a.a(this, p5, -1, p7);
            this.mStartedActivityFromFragment = 0;
            return;
        }
    } catch (Throwable v4_2) {
        this.mStartedActivityFromFragment = 0;
        throw v4_2;
    }
}
```

Use of a string value `com.google.zxing.client.android.SCAN` to construct an Intent

Taint is traced from string `com.google.zxing.client.android.SCAN` to sink method `android.app.ActivitystartActivityForResult()`

Taint trace:

```
at c.d.c.t.a.a.a()
at android.content.Intent.setAction()
at c.d.c.t.a.a.a()
at android.app.Activity.startActivityForResult()
```

Method `c.d.c.t.a.a.a()` :

```
public final void a()
{
    int v1_0 = this.a;
    if (this.d == null) {
        this.d = com.journeyapps.barcodescanner.CaptureActivity;
    }
    android.content.Intent v0_1 = new android.content.Intent(v1_0, this.d);
    v0_1.setAction(com.google.zxing.client.android.SCAN);
    v0_1.addFlags(67108864);
    v0_1.addFlags(524288);
    int v1_5 = this.c.entrySet().iterator();
    while (v1_5.hasNext()) {
        String v2_5 = ((java.util.Map$Entry) v1_5.next());
        String v3_1 = ((String) v2_5.getKey());
        String v2_6 = v2_5.getValue();
        if (!(v2_6 instanceof Integer)) {
            if (!(v2_6 instanceof Long)) {
                if (!(v2_6 instanceof Boolean)) {
                    if (!(v2_6 instanceof Double)) {
                        if (!(v2_6 instanceof Float)) {
                            if (!(v2_6 instanceof android.os.Bundle)) {
                                v0_1.putExtra(v3_1, v2_6.toString());
                            } else {
                                v0_1.putExtra(v3_1, ((android.os.Bundle) v2_6));
                            }
                        } else {
                            v0_1.putExtra(v3_1, ((Float) v2_6));
                        }
                    } else {
                        v0_1.putExtra(v3_1, ((Double) v2_6));
                    }
                } else {
                    v0_1.putExtra(v3_1, ((Boolean) v2_6));
                }
            } else {
                v0_1.putExtra(v3_1, ((Long) v2_6));
            }
        } else {
            v0_1.putExtra(v3_1, ((Integer) v2_6));
        }
    }
    String v2_2 = this.b;
    if (v2_2 == null) {
        this.a.startActivityForResult(v0_1, 49374);
    } else {
        v2_2.startActivityForResult(v0_1, 49374);
    }
    return;
}
```

Method `android.content.Intent.setAction()` not found.

Method `c.d.c.t.a.a.a()`:

```
public final void a()
{
    int v1_0 = this.a;
    if (this.d == null) {
        this.d = com.journeyapps.barcodescanner.CaptureActivity;
    }
    android.content.Intent v0_1 = new android.content.Intent(v1_0, this.d);
    v0_1.setAction(com.google.zxing.client.android.SCAN);
    v0_1.addFlags(67108864);
    v0_1.addFlags(524288);
    int v1_5 = this.c.entrySet().iterator();
    while (v1_5.hasNext()) {
        String v2_5 = ((java.util.Map$Entry) v1_5.next());
        String v3_1 = ((String) v2_5.getKey());
        String v2_6 = v2_5.getValue();
        if (!(v2_6 instanceof Integer)) {
            if (!(v2_6 instanceof Long)) {
                if (!(v2_6 instanceof Boolean)) {
                    if (!(v2_6 instanceof Double)) {
                        if (!(v2_6 instanceof Float)) {
                            if (!(v2_6 instanceof android.os.Bundle)) {
                                v0_1.putExtra(v3_1, v2_6.toString());
                            } else {
                                v0_1.putExtra(v3_1, ((android.os.Bundle) v2_6));
                            }
                        } else {
                            v0_1.putExtra(v3_1, ((Float) v2_6));
                        }
                    } else {
                        v0_1.putExtra(v3_1, ((Double) v2_6));
                    }
                } else {
                    v0_1.putExtra(v3_1, ((Boolean) v2_6));
                }
            } else {
                v0_1.putExtra(v3_1, ((Long) v2_6));
            }
        } else {
            v0_1.putExtra(v3_1, ((Integer) v2_6));
        }
    }
    String v2_2 = this.b;
    if (v2_2 == null) {
        this.a.startActivityForResult(v0_1, 49374);
    } else {
        v2_2.startActivityForResult(v0_1, 49374);
    }
    return;
}
```

Method `android.app.Activity.startActivityForResult()` not found.

Use of a string value `com.android.vending` to construct an Intent

Taint is traced from string `com.android.vending` to sink method `android.content.IntentsetPackage()`

Taint trace:

```
at c.d.a.a.c.n.n.a()
at android.content.Intent.setPackage()
```

Method `c.d.a.a.c.n.n.a()` :

```
public static android.content.Intent a(String p3, String p4)
{
    android.content.Intent v0_1 = new android.content.Intent(android.intent.action.VIEW);
    int v3_1 = android.net.Uri.parse(market://details).buildUpon().appendQueryParameter(id, p3);
    if (!android.text.TextUtils.isEmpty(p4)) {
        v3_1.appendQueryParameter(pcampaignid, p4);
    }
    v0_1.setData(v3_1.build());
    v0_1.setPackage(com.android.vending);
    v0_1.addFlags(524288);
    return v0_1;
}
```

Method `android.content.Intent.setPackage()` not found.

Potentially Backup mode enabled

Description

Android performs by default a full backup of applications including the private files stored on /data partition. The Backup Manager service uploads those data to the user's Google Drive account.

Recommendation

if the application contains sensitive data that you don't want to be restored, you can disable backup mode by setting the attribute `android:allowBackup` to false in the application tag.

References

- Random Musings on the M Developer Preview: the Ugly (Part Two)
- DRD22. Do not cache sensitive information

Technical details

```
<application xmlns:android="http://schemas.android.com/apk/res/android" android:theme="@7F12000C" android:label="
@7F11002A" android:icon="@7F0D0002" android:name="com.globant.pasaportesanitario.CovidApplication" android:allow
Backup="true" android:supportsRtl="true" android:networkSecurityConfig="@7F140000" android:roundIcon="@7F0D0004
" android:appComponentFactory="androidx.core.app.CoreComponentFactory">
  <activity android:theme="@7F12000D" android:name="com.globant.pasaportesanitario.flujos.ErrorGenericoActivity" and
roid:screenOrientation="1"/>
  <activity android:theme="@7F12000D" android:name="com.globant.pasaportesanitario.flujos.ActualizarForzadoActivity"
android:screenOrientation="1"/>
  <activity android:theme="@7F12000D" android:name="com.globant.pasaportesanitario.flujos.autodiagnostico.resultado.
ResultadoActivity" android:screenOrientation="1"/>
  <activity android:theme="@7F12000D" android:name="com.globant.pasaportesanitario.flujos.pantallaprincipal.PantallaP
rincipalActivity" android:screenOrientation="1"/>
  <activity android:theme="@7F12000D" android:name="com.globant.pasaportesanitario.flujos.inicio.InicioActivity" androi
d:screenOrientation="1">
    <intent-filter>
      <action android:name="android.intent.action.MAIN"/>
      <category android:name="android.intent.category.LAUNCHER"/>
    </intent-filter>
  </activity>
  <activity android:theme="@7F12000D" android:name="com.globant.pasaportesanitario.flujos.identificacion.Identificacio
nActivity" android:screenOrientation="1"/>
  <activity android:theme="@7F12000D" android:name="com.globant.pasaportesanitario.flujos.autodiagnostico.Autodiagn
osticoActivity" android:screenOrientation="1"/>
  <service android:name="com.globant.pasaportesanitario.locationtracker.ServicioDeRastreo" android:enabled="true" and
roid:exported="false" android:foregroundServiceType="0x00000008"/>
  <receiver android:name="com.globant.pasaportesanitario.locationtracker.ReceiverArranqueDeDispositivo">
    <intent-filter>
      <action android:name="android.intent.action.BOOT_COMPLETED"/>
      <action android:name="android.intent.action.QUICKBOOT_POWERON"/>
    </intent-filter>
  </receiver>
  <activity android:theme="@android:01030010" android:name="com.google.android.gms.common.api.GoogleApiActivity"
android:exported="false"/>
  <meta-data android:name="com.google.android.gms.version" android:value="@7F0A0009"/>
  <service android:name="androidx.room.MultiInstanceInvalidationService" android:exported="false" android:directBootA
ware="true"/>
  <activity android:theme="@7F1202DF" android:name="com.journeyapps.barcodescanner.CaptureActivity" android:clear
TaskOnLaunch="true" android:stateNotNeeded="true" android:screenOrientation="6" android:windowSoftInputMode="0x
00000003"/>
  <provider android:name="androidx.lifecycle.ProcessLifecycleOwnerInitializer" android:exported="false" android:multipro
cess="true" android:authorities="ar.gob.coronavirus.lifecycle-process"/>
</application>
```

Hardening Application code not obfuscated

Description

Obfuscation refers to methods to obscure code and make it hard to understand. Compiled Java classes can be decompiled if there is no obfuscation during compilation step.

Adversaries can steal code and repurpose it and sell it in a new application or create a malicious fake application based on the initial one.

Code obfuscation only slows the attacker from reverse engineering but does not make it impossible.

Recommendation

Design the application to add the following protections and slow reverse engineering of the application:

- Obfuscate Java source code with tools like Proguard or Dexguard

```
buildTypes {
      release {
         minifyEnabled true
         proguardFiles getDefaultProguardFile('proguard-android.txt'),
         'proguard-rules.pro'
      }
   }
```

- Verification application signing certificate during runtime by checking `context.getPackageManager().signature`
- Check application installer to ensure it matches the Android Market by calling `context.getPackageManager().getInstallerPackageName`
- Check running environment at runtime

```
private static String getSystemProperty(String name) throws Exception {
   Class systemPropertyClazz = Class.forName("android.os.SystemProperties");
   return (String) systemPropertyClazz.getMethod("get", new Class[] { String.class }).invoke(systemPropertyClazz,
new Object[] { name });
}

public static boolean checkEmulator() {

   try {
      boolean goldfish = getSystemProperty("ro.hardware").contains("goldfish");
      boolean qemu = getSystemProperty("ro.kernel.qemu").length() > 0;
      boolean sdk = getSystemProperty("ro.product.model").equals("sdk");

      if (qemu || goldfish || sdk) {
         return true;
      }

   } catch (Exception e) {
   }

   return false;
 }
```

- Check debug flag at runtime

```
context.getApplicationInfo().applicationInfo.flags & ApplicationInfo.FLAG_DEBUGGABLE;
```

References

- Lack of Binary Protections (OWASP Mobile Top 10)
- Proguard (Android developer)

Technical details

Package	Obfuscated
com.globant.pasaportesanitario	✖ False

Secure Secure Network Configuration Settings

Description

Android Network Security Configuration enables a declarative setting of the application network security.

The features enable configuring:

- Custom Certificate Authority with support for debug only settings

```
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
   <debug-overrides>
      <trust-anchors>
         <certificates src="@raw/debug_cas"/>
      </trust-anchors>
   </debug-overrides>
</network-security-config>
```

- Declarative opt-out for cleartext traffic

```
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
   <domain-config cleartextTrafficPermitted="false">
      <domain includeSubdomains="true">secure.example.com</domain>
   </domain-config>
</network-security-config>
```

- Declartive setting of certificate pinning keys

```
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
   <domain-config>
      <domain includeSubdomains="true">example.com</domain>
      <pin-set expiration="2018-01-01">
         <pin digest="SHA-256">7HIpactkIAq2Y49orFOOQKurWxmmSFZhBCoQYcRhJ3Y=</pin>
         <!-- backup pin -->
         <pin digest="SHA-256">fwza0LRMXouZHRC8Ei+4PyuldPDcf3UKgO/04cDM1oE=</pin>
      </pin-set>
   </domain-config>
</network-security-config>
```

Recommendation

The implementation is secure, no recommendation apply.

References

- Network Security Configuration (Android developer)

Technical details

Application network security config do not authorize cleartext traffic.

```
<network-security-config>
  <debug-overrides>
    <trust-anchors>
      <certificates src="user"/>
    </trust-anchors>
  </debug-overrides>
</network-security-config>
```

Secure Debug mode disabled

Description

The application is compiled with debug mode disabled. Debug mode allows attackers to access the application filesystem and attach a debugger to access sensitive data or perform malicious actions.

For instance attach a Java (JDWP) debugger:

```
$adb forward tcp:7777 jdwp:$(adb shell ps | grep "package-id")
$jdb -attach localhost:7777
```

To access the applicatin filesystem:

```
$adb shell
$run-as package-id
$...insert malicious action...
```

Attacker can debug the application without access to source code and leverage it to perform malicious actions on behalf ot the user, modify the application behavior or access sensitive data like credentials and session cookies.

Recommendation

The implementation is secure, no recommendation apply.

References

- DRD10-J Do not release apps that are debuggable (CERT Secure Coding)

Technical details

```
<application xmlns:android="http://schemas.android.com/apk/res/android" android:theme="@7F12000C" android:label="
@7F11002A" android:icon="@7F0D0002" android:name="com.globant.pasaportesanitario.CovidApplication" android:allow
Backup="true" android:supportsRtl="true" android:networkSecurityConfig="@7F140000" android:roundIcon="@7F0D0004
" android:appComponentFactory="androidx.core.app.CoreComponentFactory">
  <activity android:theme="@7F12000D" android:name="com.globant.pasaportesanitario.flujos.ErrorGenericoActivity" and
roid:screenOrientation="1"/>
  <activity android:theme="@7F12000D" android:name="com.globant.pasaportesanitario.flujos.ActualizarForzadoActivity"
android:screenOrientation="1"/>
  <activity android:theme="@7F12000D" android:name="com.globant.pasaportesanitario.flujos.autodiagnostico.resultado.
ResultadoActivity" android:screenOrientation="1"/>
  <activity android:theme="@7F12000D" android:name="com.globant.pasaportesanitario.flujos.pantallaprincipal.PantallaP
rincipalActivity" android:screenOrientation="1"/>
  <activity android:theme="@7F12000D" android:name="com.globant.pasaportesanitario.flujos.inicio.InicioActivity" androi
d:screenOrientation="1">
    <intent-filter>
      <action android:name="android.intent.action.MAIN"/>
      <category android:name="android.intent.category.LAUNCHER"/>
    </intent-filter>
  </activity>
  <activity android:theme="@7F12000D" android:name="com.globant.pasaportesanitario.flujos.identificacion.Identificacio
nActivity" android:screenOrientation="1"/>
  <activity android:theme="@7F12000D" android:name="com.globant.pasaportesanitario.flujos.autodiagnostico.Autodiagn
osticoActivity" android:screenOrientation="1"/>
  <service android:name="com.globant.pasaportesanitario.locationtracker.ServicioDeRastreo" android:enabled="true" and
roid:exported="false" android:foregroundServiceType="0x00000008"/>
  <receiver android:name="com.globant.pasaportesanitario.locationtracker.ReceiverArranqueDeDispositivo">
    <intent-filter>
      <action android:name="android.intent.action.BOOT_COMPLETED"/>
      <action android:name="android.intent.action.QUICKBOOT_POWERON"/>
    </intent-filter>
  </receiver>
  <activity android:theme="@android:01030010" android:name="com.google.android.gms.common.api.GoogleApiActivity"
android:exported="false"/>
  <meta-data android:name="com.google.android.gms.version" android:value="@7F0A0009"/>
  <service android:name="androidx.room.MultiInstanceInvalidationService" android:exported="false" android:directBootA
ware="true"/>
  <activity android:theme="@7F1202DF" android:name="com.journeyapps.barcodescanner.CaptureActivity" android:clear
TaskOnLaunch="true" android:stateNotNeeded="true" android:screenOrientation="6" android:windowSoftInputMode="0x
00000003"/>
  <provider android:name="androidx.lifecycle.ProcessLifecycleOwnerInitializer" android:exported="false" android:multipro
cess="true" android:authorities="ar.gob.coronavirus.lifecycle-process"/>
</application>
```