

LeafPic

1. App Name: LeafPic

- Details: LeafPic is a fluid, material-designed alternative gallery, it also is ad-free and open source under GPLv3 license. It doesn't miss any of the main features of a stock gallery, and we also have plans to add more useful features.
- Permissions:

Version v0.5.9 (12) suggested Added on 2016-09-29

This version requires Android 4.4 or newer.

It is built and signed by F-Droid, and guaranteed to correspond to [this source tarball](#).

▼ Permissions

- **modify or delete the contents of your shared storage**
Allows the app to write the contents of your shared storage.
- **read the contents of your shared storage**
Allows the app to read the contents of your shared storage.
- **have full network access**
Allows the app to create network sockets and use custom network protocols. The browser and other applications provide means to send data to the internet, so this permission is not required to send data to the internet.
- **install shortcuts**
Allows an application to add Homescreen shortcuts without user intervention.
- **com.android.vending.BILLING**

[Download APK](#) 4.2 MiB [PGP Signature](#) | [Build Log](#)

- GitHub Repository: <https://github.com/HoraApps/LeafPic>
- F-droid Link (Apk download link):
<https://f-droid.org/packages/org.horaapps.leafpic/>

Observations:

- Detailed Reports:
<https://github.com/hritikchaudhary/SecurityAnalysisOfAndroidApplications>
- Reverse Engineering and Verification of vulnerabilities:

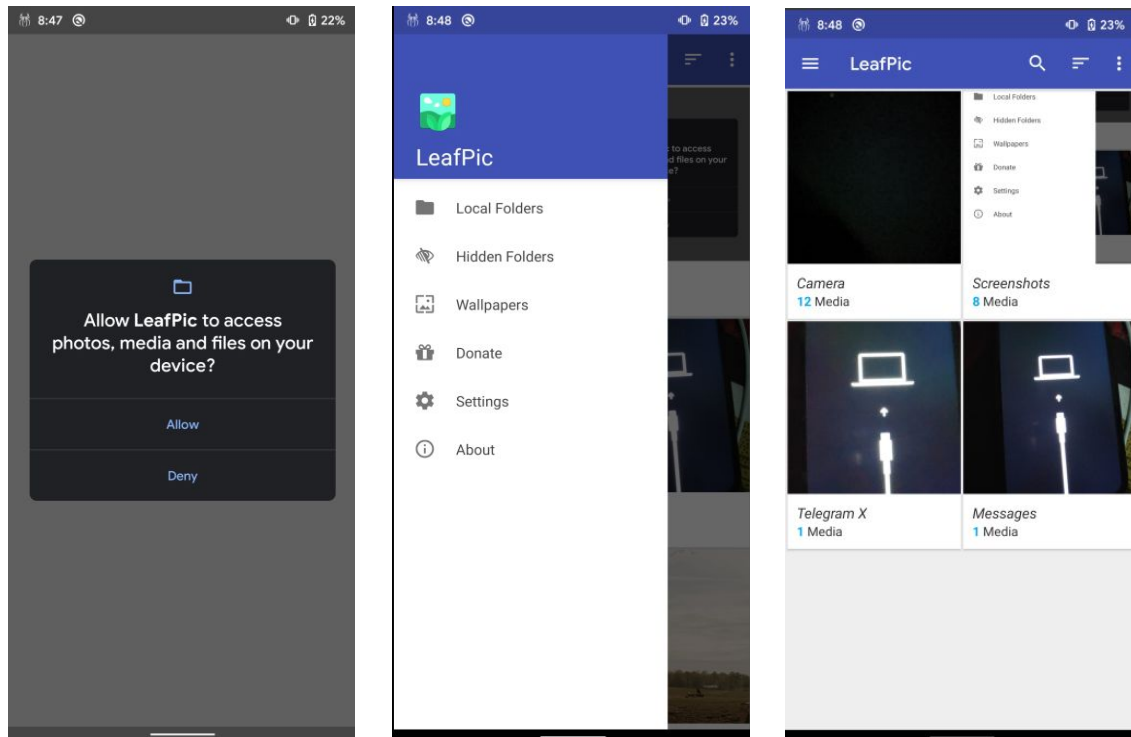
THREAT LEVELS:

1. Low
2. Medium
3. High
4. Dangerous

PERMISSION REPORT

PERMISSION	STATUS	INFO	DESCRIPTION	OBSERVATION	VERDICT (THREAT LEVEL)
android.permission.INTERNET	dangerous	full Internet access	Allows an application to create network sockets.	LeafPic is a gallery app. In my time using this app, the only time the app required the internet was to donate and to share both of which can be handled without the Internet. Although Wallpaper features may require internet access, this feature is not available yet.	High
android.permission.READ_EXTERNAL_STORAGE	dangerous	read SD card contents	Allows an application to read from SD Card.	This permission is a feature of the app as it needs external storage access to read the media in the storage.	Medium
android.permission.WRITE_EXTERNAL_STORAGE	dangerous	read/modify/delete SD card contents	Allows an application to write to the SD card.	This permission is a feature of the app as it needs external storage access to write/modify/delete the media in the storage.	Medium
com.android.launcher.permission.INSTALL_SHORTCUT	normal	Install shortcuts	Allows an application to install a shortcut in Launcher.	I'll categorize this permission as a feature as it is harmless and used to create a shortcut on the home screen.	Low
com.android.vending.BILLING	dangerous	Unknown permission from android reference	Unknown permission from android reference	I couldn't find the use this permission in the app as the only time it may need this permission is for donations, which are being handled through paypal link and bitcoin.	High

Application Screenshots:



Verification after Reverse Engineering

ISSUE	SEVERITY	DESCRIPTION	OBSERVATION	VERDICT (THREAT LEVEL)
<p>Activity is not Protected. An intent-filter exists.</p> <ol style="list-style-type: none"> 1. org.horaapps.leafpic.activities.AboutActivity 2. org.horaapps.leafpic.activities.DonateActivity 3. org.horaapps.leafpic.activities.ExcludedAlbumsActivity 4. org.horaapps.leafpic.activities.MainActivity 5. org.horaapps.leafpic.activities.PlayerActivity 6. org.horaapps.leafpic.activities.SecurityActivity 7. org.horaapps.leafpic.activities.SettingsActivity 8. org.horaapps.leafpic.a 	High	<p>An Activity is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device. The presence of intent-filter indicates that the Activity is explicitly exported.</p>	<p>Checked and verified all activities all intent listed are features of the app. Screenshots are provided below.</p> <p>But calling them features and vulnerabilities in some cases can vary user to user.</p>	Medium

activities.SingleMediaActivity				
Application Data can be Backed up [android:allowBackup=true]	medium	This flag allows anyone to backup your application data via adb. It allows users who have enabled USB debugging to copy application data off of the device.	This can be harmful for certain users, like myself, who normally have USB Debugging activated. Data can be backed up to a pc using adb.	high

Screenshots of code for verification:

SettingsActivity.java

```

findViewById(R.id.ll_security).setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        if (!SettingsActivity.this.securityObj.isActiveSecurity()) {
            SettingsActivity.this.startActivity(new Intent(SettingsActivity.this.getApplicationContext(), SecurityActivity.class));
        } else {
            SettingsActivity.this.askPasswordDialog();
        }
    }
}

```

```

findViewById(R.id.ll_excluded_album).setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        SettingsActivity.this.startActivity(new Intent(SettingsActivity.this, ExcludedAlbumsActivity.class));
    }
}

```

```

public void onClick(View v) {
    if (SettingsActivity.this.securityObj.checkPassword(editTextPassword.getText().toString())) {
        passwordDialog.dismiss();
        SettingsActivity.this.startActivity(new Intent(SettingsActivity.this.getApplicationContext(), SecurityActivity.class));
        return;
    }
}

```

MainActivity.java

```

public void onClick(View v) {
    MainActivity.this.startActivity(new Intent("android.media.action.STILL_IMAGE_CAMERA"));
}

```

```

if (MainActivity.this.pickMode) {
    MainActivity.this.setResult(-1, new Intent().setData(MainActivity.this.getAlbum().getMedia(index).getUri()));
    MainActivity.this.finish();
} else if (MainActivity.this.editMode) {
    MainActivity.this.mediaAdapter.notifyItemChanged(MainActivity.this.getAlbum().toggleSelectPhoto(index));
    MainActivity.this.invalidateOptionsMenu();
} else {
    MainActivity.this.getAlbum().setCurrentPhotoIndex(index);
    Intent intent = new Intent(MainActivity.this, SingleMediaActivity.class);
    intent.setAction("android.intent.action.pagerAlbumMedia");
    MainActivity.this.startActivity(intent);
}

```

```

        public void onClick(DialogInterface dialogInterface, int i) {
            if (Build.VERSION.SDK_INT >= 21) {
                MainActivity.this.startActivityForResult(new Intent("android.intent.action.OPEN_DOCUMENT_TREE"),
MainActivity.this.REQUEST_CODE_SD_CARD_PERMISSIONS);
            }
        }

```

```

case R.id.sharePhotos:
    Intent intent = new Intent();
    intent.setAction("android.intent.action.SEND_MULTIPLE");
    intent.putExtra("android.intent.extra.SUBJECT", getString(R.string.sent_to_action));
    ArrayList<Uri> files = new ArrayList<>();
    Iterator<Media> it = getAlbum().selectedMedias.iterator();
    while (it.hasNext()) {
        files.add(it.next().getUri());
    }
    intent.putParcelableArrayListExtra("android.intent.extra.STREAM", files);
    intent.setType(StringUtils.getGenericMIME(getAlbum().selectedMedias.get(0).getMimeType()));
    finishEditMode();
    startActivity(Intent.createChooser(intent, getResources().getText(R.string.send_to)));
    return true;

```

SingleMediaActivity.java

```

    public void onClick(View v) {
        if (SingleMediaActivity.this.SP.getBoolean("set_internal_player", false)) {
            SingleMediaActivity.this.startActivity(new Intent(SingleMediaActivity.this,
PlayerActivity.class).setData(SingleMediaActivity.this.getAlbum().getCurrentMedia().getUri()));
            return;
        }
        Intent intentOpenWith = new Intent("android.intent.action.VIEW");
        intentOpenWith.setDataAndType(SingleMediaActivity.this.getAlbum().getMedia().get(SingleMediaActivity.this.mViewPager.getCurrentItem()).getUri(),
SingleMediaActivity.this.getAlbum().getMedia().get(SingleMediaActivity.this.mViewPager.getCurrentItem()).getMimeType());
        SingleMediaActivity.this.startActivity(intentOpenWith);
    }
}

```

```

case R.id.action_share:
    Intent share = new Intent("android.intent.action.SEND");
    share.setType(getAlbum().getCurrentMedia().getMimeType());
    share.putExtra("android.intent.extra.STREAM", getAlbum().getCurrentMedia().getUri());
    startActivity(Intent.createChooser(share, getString(R.string.send_to)));
    return true;
case R.id.action_settings:
    startActivity(new Intent(getApplicationContext(), SettingsActivity.class));
    break;

```

```

case R.id.action_use_as:
    Intent intent = new Intent("android.intent.action.ATTACH_DATA");
    intent.setDataAndType(getAlbum().getCurrentMedia().getUri(), getAlbum().getCurrentMedia().getMimeType());
    startActivity(Intent.createChooser(intent, getString(R.string.use_as)));
    return true;

```

```

case R.id.action_edit_with:
    Intent editIntent = new Intent("android.intent.action.EDIT");
    editIntent.setDataAndType(getAlbum().getCurrentMedia().getUri(), getAlbum().getCurrentMedia().getMimeType());
    editIntent.setFlags(1);
    startActivity(Intent.createChooser(editIntent, "Edit with"));
    break;
case R.id.action_open_with:
    Intent intentOpenWith = new Intent("android.intent.action.VIEW");
    intentOpenWith.setDataAndType(getAlbum().getCurrentMedia().getUri(), getAlbum().getCurrentMedia().getMimeType());
    startActivity(Intent.createChooser(intentOpenWith, getString(R.string.open_with)));
    break;

```

PlayerActivity.java

```

case R.id.action_share:
    Intent share = new Intent("android.intent.action.SEND");
    share.setType(new Media(ContentHelper.getMediaPath(getApplicationContext(), getIntent().getData())).getMimeType());
    share.putExtra("android.intent.extra.STREAM", getIntent().getData());
    startActivity(Intent.createChooser(share, getString(R.string.send_to)));
    return true;

```

AboutActivity.java

```

case R.id.action_share:
    Intent share = new Intent("android.intent.action.SEND");
    share.setType(new Media(ContentHelper.getMediaPath(getApplicationContext(), getIntent().getData())).getMimeType());
    share.putExtra("android.intent.extra.STREAM", getIntent().getData());
    startActivity(Intent.createChooser(share, getString(R.string.send_to)));
    return true;
}

'''
findViewById(R.id.about_author_gilbert_mail_item).setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        Intent intent = new Intent();
        intent.setAction("android.intent.action.SENDTO");
        intent.setData(Uri.parse("mailto: jibo95@gmail.com"));
        try {
            AboutActivity.this.startActivity(intent);
        } catch (Exception e) {
            Toast.makeText(AboutActivity.this, AboutActivity.this.getString(R.string.send_mail_error), 0).show();
        }
    }
}

```

Issues found in Static code Analysis

ISSUE	SEVERITY	STANDARDS	FILES
App can read/write to External Storage. Any App can read data written to External Storage.	high	CVSS V2: 5.5 (medium) CWE: CWE-276 - Incorrect Default Permissions OWASP Top 10: M2: Insecure Data Storage OWASP MASVS: MSTG-STORAGE-2	org/horaapps/leafpic/SelectAlbumBottomSheet.java org/horaapps/leafpic/data/CustomAlbumsHelper.java org/horaapps/leafpic/util/Affix.java org/horaapps/leafpic/util/ContentHelper.java
App uses SQLite Database and execute raw SQL query. Untrusted user input in raw SQL queries can cause SQL Injection. Also sensitive information should be encrypted and written to the database.	high	CVSS V2: 5.9 (medium) CWE: CWE-89 - Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') OWASP Top 10: M7: Client Code Quality	org/horaapps/leafpic/data/CustomAlbumsHelper.java

Amaze File Manager

2. App Name: Amaze File Manager

- Details: Light and smooth file manager following the Material Design guidelines.

Features:

- Basic features like cut, copy, delete, compress, extract etc. easily accessible
- Work on multiple tabs at same time
- Multiple themes with cool icons
- Navigation drawer for quick navigation
- App Manager to open, backup, or directly uninstall any app
- Quickly access history, access bookmarks or search for any file
- Root explorer for advanced users
- Permissions:

Version 3.1.2 RC1 (54) - Added on 2017-04-04

This version requires Android 4.0 or newer.

It is built and signed by F-Droid, and guaranteed to correspond to [this source tarball](#).



[NoSourceSince](#)

▼ Permissions

- view Wi-Fi connections

Allows the app to view information about Wi-Fi networking, such as whether Wi-Fi is enabled and name of connected Wi-Fi devices.

- view network connections

Allows the app to view information about network connections such as which networks exist and are connected.

- modify or delete the contents of your shared storage

Allows the app to write the contents of your shared storage.

- install shortcuts

Allows an application to add Homescreen shortcuts without user intervention.

- have full network access

Allows the app to create network sockets and use custom network protocols. The browser and other applications provide means to send data to the internet, so this permission is not required to send data to the internet.

- read the contents of your shared storage

Allows the app to read the contents of your shared storage.

[Download APK](#) 5 MiB [PGP Signature](#) | [Build Log](#)

- GitHub Repository: <https://github.com/TeamAmaze/AmazeFileManager>
- F-droid Link (Apk download link):
<https://f-droid.org/packages/com.amaze.filemanager/>

Observations:

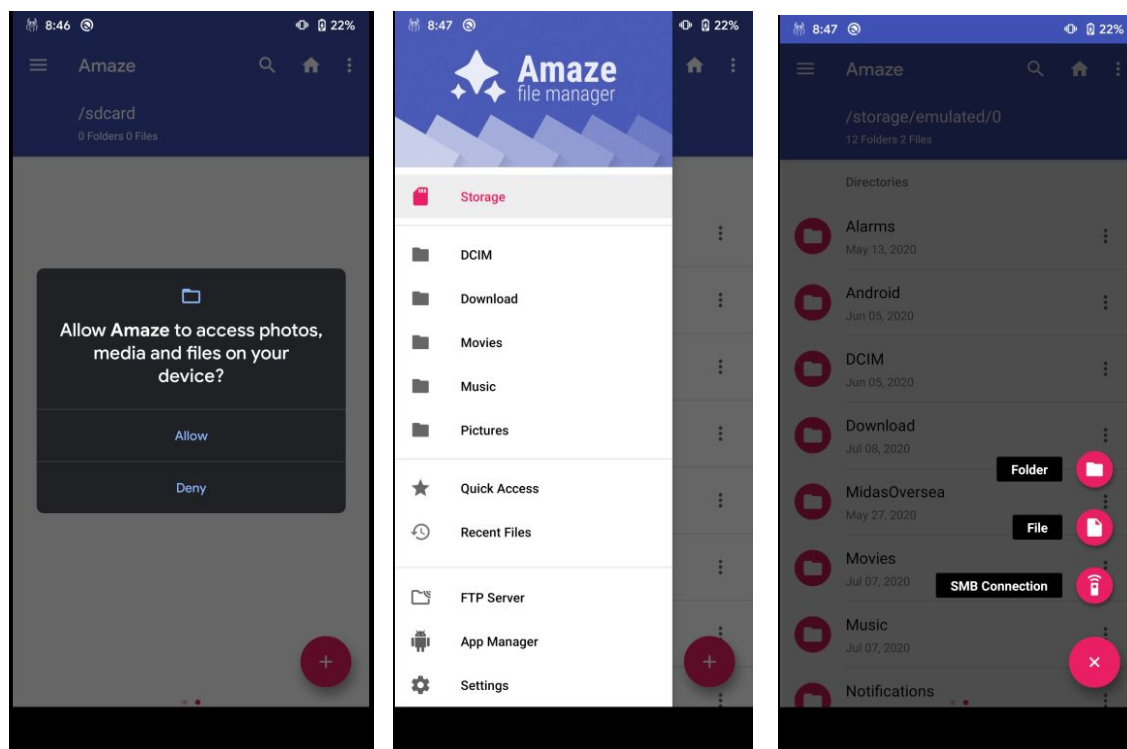
- Detailed Reports:
<https://github.com/hritikchaudhary/SecurityAnalysisOfAndroidApplications>
- Reverse Engineering and Verification of vulnerabilities:

PERMISSION REPORT

PERMISSION	STATUS	INFO	DESCRIPTION	OBSERVATION	VERDICT (THREAT LEVEL)
android.permission.ACCESS_NETWORK_STATE	normal	view network status	Allows an application to view the status of all networks.	It is a normal permission to view the network status of the phone.	medium
android.permission.ACCESS_SUPERUSER	dangerous	Unknown permission from android reference	Unknown permission from android reference	Although as mentioned in the required permissions that this feature is for advanced users, it's still a very dangerous permission, and accidental deletion of some important file can result in system failure and other catastrophic events including bricking of the device (not be able to turn on due to operating system failure).	dangerous
android.permission.ACCESS_WIFI_STATE	normal	view Wi-Fi status	Allows an application to view the information about the status of Wi-Fi.	It is a normal permission to view the Wi-Fi status of the phone.	Low
android.permission.INTERNET	dangerous	full Internet access	Allows an application to create network sockets.	In the case of this app it is normal to have internet access permission as it is a feature of the app.	medium
android.permission.WRITE_EXTERNAL_STORAGE	dangerous	read/modify/delete SD card contents	Allows an application to write to the SD card.	As this is a file manager it is required to have this permission for basic working of the app.	Medium

com.android.launcher.permission.INSTALL_SHORTCUT	normal		Allows an application to install a shortcut in Launcher.	I'll categorize this permission as a feature as it is harmless and used to create a shortcut on the home screen	low
--	--------	--	--	---	-----

Application Screenshots:



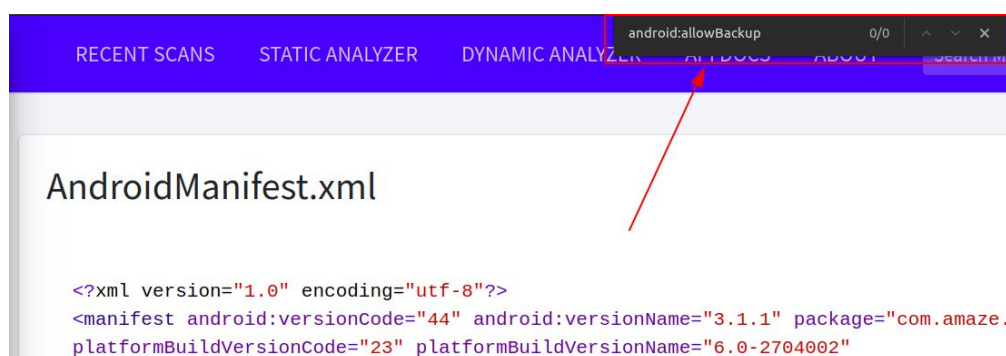
Verification after Reverse Engineering

ISSUE	SEVERITY	DESCRIPTION	OBSERVATION	VERDICT (THREAT LEVEL)
Activity is not Protected. An intent-filter exists. 1. com.amaze.filemanager.activities.DbViewer 2. com.amaze.filemanager.activities.Preferences 3. com.amaze.filemanager.activities.TextReader	high	An Activity is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device. The presence of intent-filter indicates that the Activity is explicitly exported.	Checked and verified all activities all intent listed are features of the app. Screenshots are provided below. But calling them features and vulnerabilities in some cases can vary user to user.	medium

Application Data can be Backed up [android:allowBackup] flag is missing.	medium	The flag [android:allowBackup] should be set to false. By default it is set to true and allows anyone to backup your application data via adb.	It allows users who have enabled USB debugging to copy application data off of the device. The flag [android:allowBackup] should be set to false.	high
1. Broadcast Receiver (com.amaze.filemanager.services.ftpservice.FTPReceiver) is not Protected. [android:exported=true] 2. Broadcast Receiver (com.amaze.filemanager.ui.notifications.FTPNotification) is not Protected. [android:exported=true]	high	A Broadcast Receiver is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device.	These are required for working of the app as these are the features.	medium
Service (com.amaze.filemanager.services.ftpservice.FTPService) is not Protected. [android:exported=true]	high	A Service is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device.	These are required for working of the app as these are the features.	medium

Screenshots of code for verification:

AndroidManifest.xml



```

<receiver android:name="com.amaze.filemanager.services.ftpservice.FTPReceiver" android:exported="true">
    <intent-filter>
        <action android:name="com.amaze.filemanager.services.ftpservice.FTPReceiver.ACTION_START_FTPSERVER" />
        <action android:name="com.amaze.filemanager.services.ftpservice.FTPReceiver.ACTION_STOP_FTPSERVER" />
    </intent-filter>
</receiver>
<receiver android:name="com.amaze.filemanager.ui.notifications.FTPNotification" android:exported="true">
    <intent-filter>
        <action android:name="com.amaze.filemanager.services.ftpservice.FTPReceiver.FTPSERVER_STARTED" />
        <action android:name="com.amaze.filemanager.services.ftpservice.FTPReceiver.FTPSERVER_STOPPED" />
    </intent-filter>
</receiver>
</application>
</manifest>

```

TextReader.java

```

try {
    if (getIntent().getData() != null) {
        this.uri = getIntent().getData();
        this.mFile = new File(getIntent().getData().getPath());
    } else {
        this.mFile = new File(getIntent().getStringExtra("path"));
    }
} catch (Exception e) {
    this.mFile = null;
}
String fileName = null;

```

Preferences.java





```

        selectItem(0);
    } else {
        Intent in = new Intent(this, MainActivity.class);
        in.setAction("android.intent.action.MAIN");
        overridePendingTransition(17432576, 17432577);
        finish();
        overridePendingTransition(17432576, 17432577);
        startActivity(in);
    }
}

public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case 16908332:
            if (this.select != 1 || this.changed != 1) {
                if (this.select != 1) {
                    Intent in = new Intent(this, MainActivity.class);
                    in.setAction("android.intent.action.MAIN");
                    overridePendingTransition(17432576, 17432577);
                    finish();
                }
            }
        }
    }
}

```

Issues found in Static code Analysis

ISSUE 	SEVERITY 	STANDARDS 	FILES 
App can read/write to External Storage. Any App can read data written to External Storage.	high	CVSS V2: 5.5 (medium) CWE: CWE-276 - Incorrect Default Permissions OWASP Top 10: M2: Insecure Data Storage OWASP MASVS: MSTG-STORAGE-2	com/amaze/filemanager/activities/MainActivity.java com/amaze/filemanager/adapters/AppsAdapter.java com/amaze/filemanager/filesystem/FileUtil.java com/amaze/filemanager/filesystem/MediaStoreHack.java com/amaze/filemanager/services/ftpService/FTPService.java com/amaze/filemanager/utills/HistoryManager.java com/amaze/filemanager/utills/Logger.java com/stericson/RootTools/internal/RootToolsInternalMethods.java
App creates temp file. Sensitive information should never be written into a temp file.	high	CVSS V2: 5.5 (medium) CWE: CWE-276 - Incorrect Default Permissions OWASP Top 10: M2: Insecure Data Storage OWASP MASVS: MSTG-STORAGE-2	com/amaze/filemanager/utills/SmbStreamer/StreamServer.java
App uses SQLite Database and execute raw SQL query. Untrusted user input in raw SQL queries can cause SQL Injection. Also sensitive information should be encrypted and written to the database.	high	CVSS V2: 5.9 (medium) CWE: CWE-89 - Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') OWASP Top 10: M7: Client Code Quality	com/amaze/filemanager/activities/DbViewer.java com/amaze/filemanager/database/TabHandler.java com/amaze/filemanager/utills/HistoryManager.java
Files may contain hardcoded sensitive informations like usernames, passwords, keys etc.	high	CVSS V2: 7.4 (high) CWE: CWE-312 - Cleartext Storage of Sensitive Information OWASP Top 10: M9: Reverse Engineering OWASP MASVS: MSTG-STORAGE-14	com/amaze/filemanager/activities/Preferences.java com/amaze/filemanager/utills/Constants.java com/jcraft/jsch/KeyExchange.java jcifs/dcerpc/DcerpcHandle.java jcifs/ntlmssp/Type3Message.java jcifs/smb/SmbComSessionSetupAndX.java jcifs/smb/SmbComTreeConnectAndX.java org.slf4j/impl/SimpleLogger.java org/sufficientlysecure/donations/DonationsFragment.java

Simple Calendar Pro

3. App Name: Simple Calendar Pro

- **Details:** A simple calendar with events and a customizable widget.
Simple Calendar Pro is a fully customizable, offline calendar designed to do exactly what a calendar should do. No complicated features, unnecessary permissions and no ads! Whether you're organizing single or recurring events, birthdays, anniversaries, business meetings, appointments or anything else, Simple Calendar Pro makes it easy to stay organized.
- **Permissions:**

Version 6.9.6 (179) suggested Added on 2020-06-17

This version requires Android 5.0 or newer.

It is built and signed by F-Droid, and guaranteed to correspond to [this source tarball](#).

▼ Permissions

- run at startup

Allows the app to have itself started as soon as the system has finished booting. This can make it take longer to start the phone and allow the app to slow down the overall phone by always running.

- read your contacts

Allows the app to read data about your contacts stored on your phone, including the frequency with which you've called, emailed, or communicated in other ways with specific individuals. This permission allows apps to save your contact data, and malicious apps may share contact data without your knowledge.

- control vibration

Allows the app to control the vibrator.

- Read calendar events and details

This app can read all calendar events stored on your phone and share or save your calendar data.

- add or modify calendar events and send email to guests without owners' knowledge

This app can add, remove, or change calendar events on your phone. This app can send messages that may appear to come from calendar owners, or change events without notifying their owners.

- prevent phone from sleeping

Allows the app to prevent the phone from going to sleep.

- modify or delete the contents of your shared storage

Allows the app to write the contents of your shared storage.
(9.0)

- read the contents of your shared storage

Allows the app to read the contents of your shared storage.
(9.0)

[Download APK](#) 7.0 MiB [PGP Signature](#) | [Build Log](#)

- GitHub Repository: <https://github.com/SimpleMobileTools/Simple-Calendar>
- F-droid Link (Apk download link):
<https://f-droid.org/packages/com.simplemobiletools.calendar.pro/>

Observations:

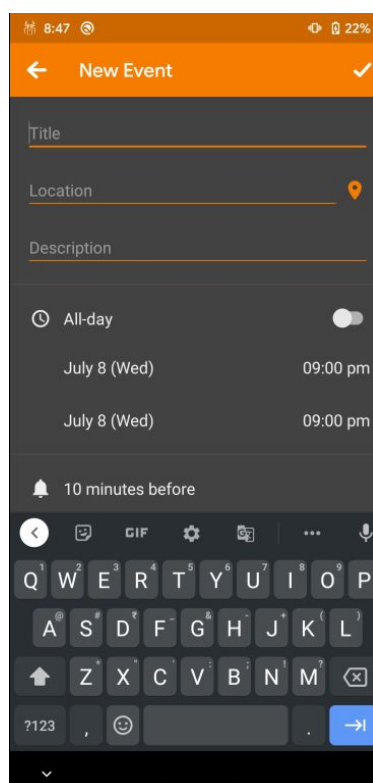
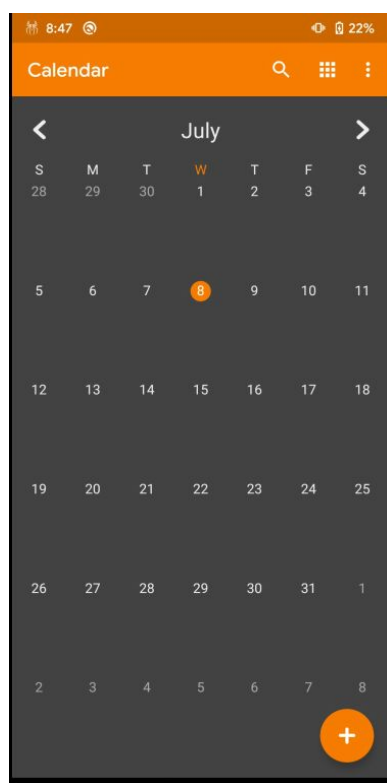
- Detailed Reports:
<https://github.com/hritikchaudhary/SecurityAnalysisOfAndroidApplications>
- Reverse Engineering and Verification of vulnerabilities:

PERMISSION REPORT

PERMISSION	STATUS	INFO	DESCRIPTION	OBSERVATION	VERDICT (THREAT LEVEL)
android.permission.READ_CALENDAR	dangerous	read calendar events	Allows an application to read all of the calendar events stored on your phone.	This is a basic feature for working of the app	low
android.permission.READ_CONTACTS	dangerous	read contact data	Allows an application to read all of the contact (address) data stored on your phone.	There are two uses of this permission in the app: 1. To add contact's birthdays 2. To add contact's anniversaries But nonetheless this app has access to all the contacts which is still harmful.	dangerous
android.permission.RECEIVE_BOOT_COMPLETED	normal	automatically start at boot	Allows an application to start itself as soon as the system has finished booting.	This can make it take longer to start the phone and allow the application to slow down the overall phone by always running.	high
android.permission.VIBRATE	normal	control vibrator	Allows the application to control the vibrator.	This is a feature as it takes this permission for notifications.	medium
android.permission.WAKE_LOCK	dangerous	prevent phone from sleeping	Allows an application to prevent the phone from going to sleep.	This is a feature that is usually not required in a calendar app. This can improve battery consumption	high

android.permission.WRITE_CALENDAR	dangerous	add or modify calendar events and send emails to guests	Allows an application to add or change the events on your calendar, which may send emails to guests.	Malicious applications can use this to erase or modify your calendar events or to send emails to guests. But since it is a calendar app, it is considered as a feature.	medium
android.permission.WRITE_EXTERNAL_STORAGE	dangerous	read/modify/delete SD card contents	Allows an application to write to the SD card.	It's is a feature only useful for importing and exporting calendar events. Giving it access to read/write external storage can be harmful	high

Application Screenshots:



Verification after Reverse Engineering

ISSUE	SEVERITY	DESCRIPTION	OBSERVATION	VERDICT (THREAT LEVEL)
Activity is not Protected. An intent-filter exists in found around 25 instance, most of them are of SplashActivity.Colour	high	An Activity is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device. The presence of intent-filter indicates that the Activity is explicitly exported.	Checked and verified all activities all intent listed are features of the app. Screenshots are provided below. But calling them features and vulnerabilities in some cases can vary user to user.	medium
Application Data can be Backed up [android:allowBackup=true]	medium	This flag allows anyone to backup your application data via adb. It allows users who have enabled USB debugging to copy application data off of the device.	It allows users who have enabled USB debugging to copy application data off of the device. The flag [android:allowBackup] should be set to false.	high
Broadcast Receiver is not Protected. An intent-filter exists. (5 instances)	high	A Broadcast Receiver is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device.	The presence of intent-filter indicates that the Broadcast Receiver is explicitly exported. These are required for working of the app as these are the features.	medium
Service (com.simplemobiletools.calendar.pro.jobs.CalDAVUpdateListener) is Protected by a permission, but the protection level of the permission should be checked. Permission: android.permission.BIND_JOB_SERVICE [android:exported=true]	high	A Service is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device. It is protected by a permission which is not defined in the analysed application. As a result, the protection level of the permission should be checked where it is defined.	If it is set to normal or dangerous, a malicious application can request and obtain the permission and interact with the component. If it is set to signature, only applications signed with the same certificate can obtain the permission.	medium

Code Screenshot for Verification:

AndroidManifest.xml

```
<service android:name="com.simplmobiletools.calendar.pro.jobs.CalDAVUpdateListener" android:permission="android.permission.BIND_JOB_SERVICE"
android:exported="true" />

<receiver android:name="com.simplmobiletools.commons.receivers.SharedThemeReceiver" android:exported="true">
```

MyWidgetDateProvider.java

```
private final void a(Context context, RemoteViews remoteViews) {
    Intent j = b.d.a.n.h.j(context);
    if (j == null) {
        j = new Intent(context, SplashActivity.class);
    }
    remoteViews.setOnClickPendingIntent(R.id.widget_date_holder, PendingIntent.getActivity(context, this.f2200a, j, 134217728));
}
```

Issues found in static code analysis:

ISSUE	SEVERITY	STANDARDS	FILES
App can read/write to External Storage. Any App can read data written to External Storage.	high	CVSS V2: 5.5 (medium) CWE: CWE-276 - Incorrect Default Permissions OWASP Top 10: M2: Insecure Data Storage OWASP MASVS: MSTG-STORAGE-2	b/d/a/m/i.java b/d/a/n/i.java
App creates temp file. Sensitive information should never be written into a temp file.	high	CVSS V2: 5.5 (medium) CWE: CWE-276 - Incorrect Default Permissions OWASP Top 10: M2: Insecure Data Storage OWASP MASVS: MSTG-STORAGE-2	a/m/c.java
App uses SQLite Database and execute raw SQL query. Untrusted user input in raw SQL queries can cause SQL Injection. Also sensitive information should be encrypted and written to the database.	high	CVSS V2: 5.9 (medium) CWE: CWE-89 - Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') OWASP Top 10: M7: Client Code Quality	a/o/a/g/a.java

Easy Sound Recorder

4. App Name: Easy Sound Recorder

- Details: Simple sound recorder implementing material design. Use Sound Recorder to record lectures, singing, business meetings, notes and more!

Features:

- Material Design
 - Easy to Use
 - No Time Limits
 - MP4 Format
 - AAC Encoder
 - Manage files easily
 - Open source
 - NO ADVERTISEMENTS
- Permissions:

Version 1.3.0 (130) suggested Added on 2017-05-25

This version requires Android 4.1 or newer.

It is built and signed by F-Droid, and guaranteed to correspond to [this source tarball](#).

▼ **Permissions**

- **record audio**
This app can record audio using the microphone at any time.
- **modify or delete the contents of your shared storage**
Allows the app to write the contents of your shared storage.
- **read the contents of your shared storage**
Allows the app to read the contents of your shared storage.

[Download APK](#) 1.2 MiB [PGP Signature](#) | [Build Log](#)

- GitHub Repository: <https://github.com/dkim0419/SoundRecorder>
- F-droid Link (Apk download link):
<https://f-droid.org/packages/com.danielkim.soundrecorder/>

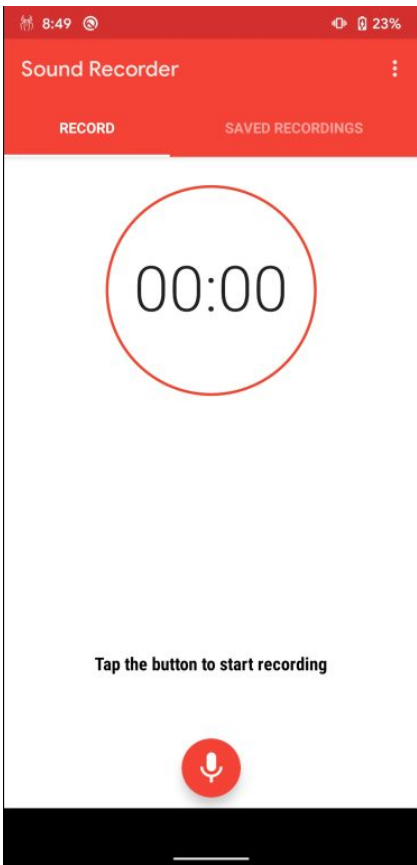
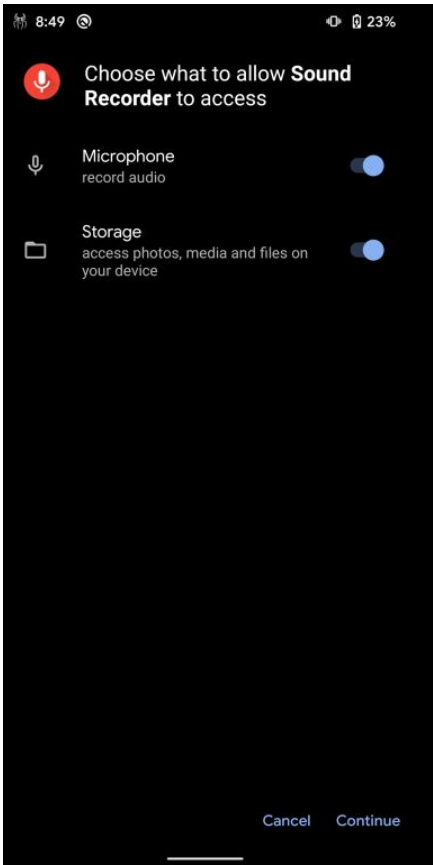
Observations:

- Detailed Reports:
<https://github.com/hritikchaudhary/SecurityAnalysisOfAndroidApplications>
- Reverse Engineering and Verification of vulnerabilities:

PERMISSION REPORT

PERMISSION	STATUS	INFO	DESCRIPTION	OBSERVATION	VERDICT (THREAT LEVEL)
android.permission.RECORD_AUDIO	dangerous	record audio	Allows application to access the audio record path.	This is a audio recording application, this permission is required for basic functioning of the app.	medium
android.permission.WRITE_EXTERNAL_STORAGE	dangerous	read/modify/delete SD card contents	Allows an application to write to the SD card.	This permission is required to save the audio recorded by the app.	medium

Application Screenshots:



Verification after Reverse Engineering

ISSUE	SEVERITY	DESCRIPTION	OBSERVATION	VERDICT (THREAT LEVEL)
Application Data can be Backed up [android:allowBackup=true]	medium	This flag allows anyone to backup your application data via adb. It allows users who have enabled USB debugging to copy application data off of the device.	It allows users who have enabled USB debugging to copy application data off of the device. The flag [android:allowBackup] should be set to false.	high

Code Screenshot for Verification:

```
<application android:theme="@style/AppTheme" android:label="@string/app_name" android:icon="@drawable/ic_launcher" android:allowBackup="true">
    <activity android:label="@string/app_name" android:name="com.danielkim.soundrecorder.activities.MainActivity"
```

Issues found in Static code Analysis

ISSUE	SEVERITY	STANDARDS	FILES
App can read/write to External Storage. Any App can read data written to External Storage.	high	CVSS V2: 5.5 (medium) CWE: CWE-276 - Incorrect Default Permissions OWASP Top 10: M2: Insecure Data Storage OWASP MASVS: MSTG-STORAGE-2	com/danielkim/soundrecorder/RecordingService.java com/danielkim/soundrecorder/adapters/FileViewerAdapter.java com/danielkim/soundrecorder/fragments/FileViewerFragment.java com/danielkim/soundrecorder/fragments/RecordFragment.java
App uses SQLite Database and execute raw SQL query. Untrusted user input in raw SQL queries can cause SQL Injection. Also sensitive information should be encrypted and written to the database.	high	CVSS V2: 5.9 (medium) CWE: CWE-89 - Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') OWASP Top 10: M7: Client Code Quality	com/danielkim/soundrecorder/DBHelper.java

Timber (Music Player)

5. App Name: Timber

- Details: This app has been developed by naman14 and has been removed from F-Droid because of non-free dependencies. This is the pure FOSS fork of the app, where all these non-free dependencies have been removed.

Features:

- Material design
- 6 different now playing styles
- Homescreen widgets
- Browse device folders
- Dark theme and UI customizability
- Gestures for track switching
- LastFM scrobble
- Android Wear and Android Auto support
- Playing queue in notification (Xposed)
- Lyrics support
- Permissions:

Version 1.6.1 (21) suggested Added on 2019-07-19

This version requires Android 4.1 or newer.

It is built and signed by F-Droid, and guaranteed to correspond to [this source tarball](#).

▼ Permissions

- **modify or delete the contents of your shared storage**
Allows the app to write the contents of your shared storage.
- **read the contents of your shared storage**
Allows the app to read the contents of your shared storage.
- **view network connections**
Allows the app to view information about network connections such as which networks exist and are connected.
- **have full network access**
Allows the app to create network sockets and use custom network protocols. The browser and other applications provide means to send data to the internet, so this permission is not required to send data to the internet.
- **prevent phone from sleeping**
Allows the app to prevent the phone from going to sleep.
- **send sticky broadcast**
Allows the app to send sticky broadcasts, which remain after the broadcast ends. Excessive use may make the phone slow or unstable by causing it to use too much memory.
- **com.android.vending.BILLING**

[Download APK](#) 7.7 MiB [PGP Signature](#) | [Build Log](#)

- GitHub Repository: <https://github.com/naman14/Timber>
- F-droid Link (Apk download link):
<https://f-droid.org/en/packages/it.fossoft.timberfoss/>

Observations:

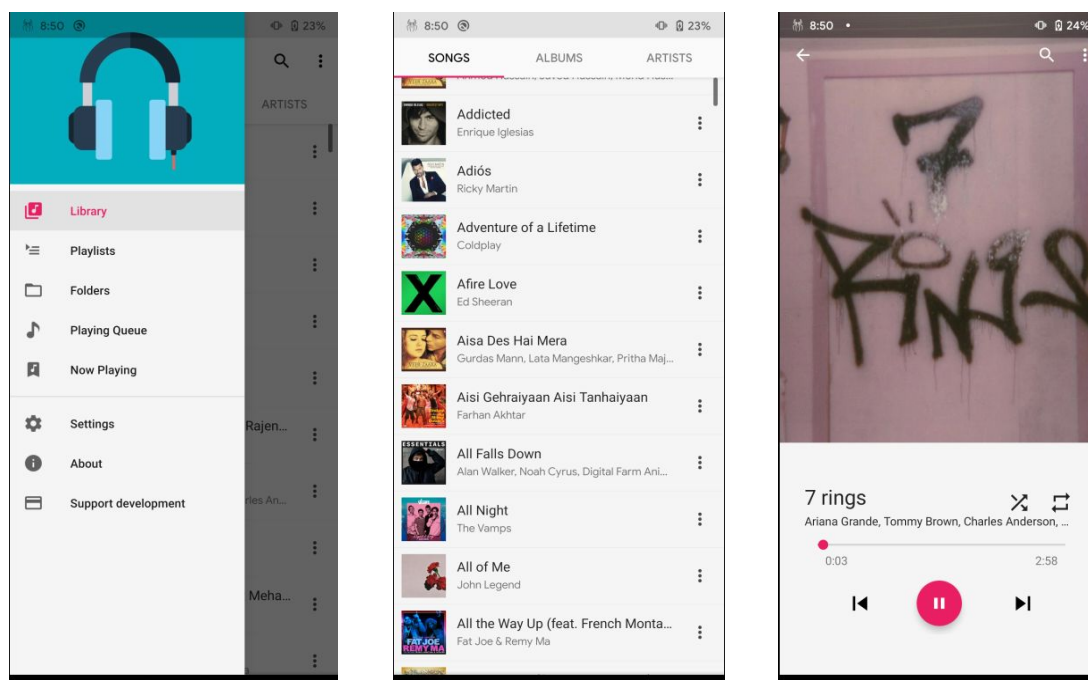
- Detailed Reports:
<https://github.com/hritikchaudhary/SecurityAnalysisOfAndroidApplications>
- Reverse Engineering and Verification of vulnerabilities:

PERMISSION REPORT:

PERMISSION	STATUS	INFO	DESCRIPTION	OBSERVATION	VERDICT (THREAT LEVEL)
android.permission.ACCESS_NETWORK_STATE	normal	view network status	Allows an application to view the status of all networks.	This permission is required for the app's lyrics and song's metadata downloading feature	low
android.permission.BROADCAST_STICKY	normal	send sticky broadcast	Allows an application to send sticky broadcasts, which remain after the broadcast ends.	Malicious applications can make the phone slow or unstable by causing it to use too much memory.	medium
android.permission.INTERNET	dangerous	full Internet access	Allows an application to create network sockets.	This permission is required for the app's lyrics and song's metadata downloading feature	medium
android.permission.READ_EXTERNAL_STORAGE	dangerous	read SD card contents	Allows an application to read from SD Card.	This permission is required to read songs from the storage	medium
android.permission.WAKE_LOCK	dangerous	Prevent phone from sleeping	Allows an application to prevent the phone from going to sleep.	This feature is required to keep app open and avoiding screen lock, but it can increase battery consumption	medium
android.permission.WRITE_EXTERNAL_STORAGE	dangerous	read/modify/delete SD card contents	Allows an application to write to the SD card.	This permission is required to modify the songs in the storage	medium

com.android.vending.BILLING	dangerous	Unknown permission from android reference	Unknown permission from android reference	In this app this is valid unlike LeafPic app above it do have In-app purchases. Though it do this using google play services but one should be careful anyway	high
-----------------------------	-----------	---	---	---	------

Application Screenshots:



Verification after Reverse Engineering

ISSUE	SEVERITY	DESCRIPTION	OBSERVATION	VERDICT (THREAT LEVEL)
<p>Broadcast Receiver is not Protected. An intent-filter exists.</p> <ol style="list-style-type: none"> com.naman14.timber.helpers.MediaButtonIntentReceiver com.naman14.timber.widgets.desktop.SmallWidget com.naman14.timber.widgets.desktop.StandardWidget 	high	A Broadcast Receiver is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device.	<p>The presence of intent-filter indicates that the Broadcast Receiver is explicitly exported. But after checking conclusion is that this is a feature and can be verified from the screenshot attached below</p>	medium

4. com.naman14.timber.wi dgets.desktop.WhiteWi dget				
Service (com.naman14.timber. WearBrowserService) is not Protected. [android:exported=true]	high	A Service is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device.	A Service is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device.	medium

Code Screenshots for verification:

AndroidManifest.xml

```
<service android:label="@string/app_name" android:name="com.naman14.timber.MusicService" android:process=":main" />
<service android:name="com.naman14.timber.WearBrowserService" android:exported="true">
```

MediaButtonIntentReceiver.java

```
case 1:
    if (!MediaButtonIntentReceiver.mLaunched) {
        Context context = (Context) message.obj;
        Intent intent = new Intent();
        intent.setClass(context, MainActivity.class);
        intent.setFlags(335544320);
        context.startActivity(intent);
        boolean unused = MediaButtonIntentReceiver.mLaunched = true;
        break;
    }
    break;
```

```
public static void startService(Context context, String str) {
    Intent intent = new Intent(context, MusicService.class);
    intent.setAction(MusicService.SERVICECMD);
    intent.putExtra(MusicService.CMDNAME, str);
    intent.putExtra(MusicService.FROM_MEDIA_BUTTON, true);
    startWakefulService(context, intent);
}
```

```
public void onReceive(Context context, Intent intent) {
    KeyEvent keyEvent;
    String action = intent.getAction();
    if ("android.media.AUDIO_BECOMING_NOISY".equals(action)) {
        if (PreferencesUtility.getInstance(context).pauseEnabledOnDetach()) {
            startService(context, MusicService.CMDPAUSE);
        }
    } else if ("android.intent.action.MEDIA_BUTTON".equals(action) && (keyEvent = (KeyEvent)
intent.getParcelableExtra("android.intent.extra.KEY_EVENT")) != null) {
        int keyCode = keyEvent.getKeyCode();
        int action2 = keyEvent.getAction();
    }
}
```

WhiteWidget.java





WhiteWidget.java

```
package com.naman14.timber.widgets.desktop;

import it.fossoft.timberfoss.R;

public class WhiteWidget extends StandardWidget {
    /* access modifiers changed from: package-private */
    public int getLayoutRes() {
        return R.layout.widget_white;
    }
}
```

Issue Found in Static code Analysis:

ISSUE 	SEVERITY 	STANDARDS 	FILES 
App can read/write to External Storage. Any App can read data written to External Storage.	high	CVSS V2: 5.5 (medium) CWE: CWE-276 - Incorrect Default Permissions OWASP Top 10: M2: Insecure Data Storage OWASP MASVS: MSTG-STORAGE-2	com/afollestad/materialdialogs/folderselector/FileChooserDialog.java com/afollestad/materialdialogs/folderselector/FolderChooserDialog.java com/naman14/timber/dialogs/StorageSelectDialog.java com/naman14/timber/utills/PreferencesUtility.java com/nostra13/universalimageloader/utills/StorageUtils.java
App creates temp file. Sensitive information should never be written into a temp file.	high	CVSS V2: 5.5 (medium) CWE: CWE-276 - Incorrect Default Permissions OWASP Top 10: M2: Insecure Data Storage OWASP MASVS: MSTG-STORAGE-2	fi/iki/elonon/NanoHTTDP.java
App uses SQLite Database and execute raw SQL query. Untrusted user input in raw SQL queries can cause SQL Injection. Also sensitive information should be encrypted and written to the database.	high	CVSS V2: 5.9 (medium) CWE: CWE-89 - Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') OWASP Top 10: M7: Client Code Quality	com/naman14/timber/provider/MusicPlaybackState.java com/naman14/timber/provider/RecentStore.java com/naman14/timber/provider/SearchHistory.java com/naman14/timber/provider/SongPlayCount.java
Files may contain hardcoded sensitive informations like usernames, passwords, keys etc.	high	CVSS V2: 7.4 (high) CWE: CWE-312 - Cleartext Storage of Sensitive Information OWASP Top 10: M9: Reverse Engineering OWASP MASVS: MSTG-STORAGE-14	com/afollestad/apptHEMEengine/ConfigBase.java com/anjlab/android/iab/v3/BillingCache.java com/anjlab/android/iab/v3/BillingProcessor.java com/naman14/timber/lastfmapi/LastFmClient.java com/naman14/timber/lastfmapi/models/LastfmUserSession.java