Fortify Standalone Report Generator

Developer Workbook

scan_results

Table of Contents

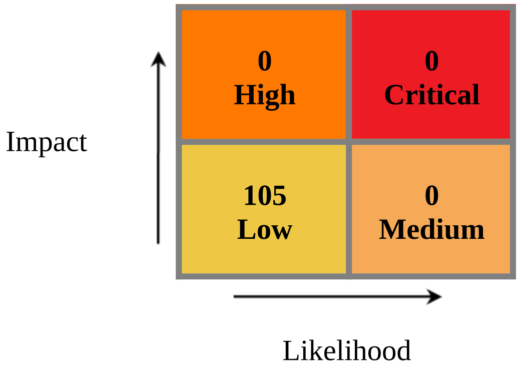
Executive Summary

This workbook is intended to provide all necessary details and information for a developer to understand and remediate the different issues discovered during the scan_results project audit. The information contained in this workbook is targeted at project managers and developers.

This section provides an overview of the issues uncovered during analysis.

Project Name:	scan_results
Project Version:	
SCA:	Results Present
WebInspect:	Results Not Present
WebInspect Agent:	Results Not Present
Other:	Results Not Present

Issues by Priority



0 **High**	**0** **Critical**
105 **Low**	**0** **Medium**

Top Ten Critical Categories

This project does not contain any critical issues

Project Description

This section provides an overview of the Fortify scan engines used for this project, as well as the project meta-information.

SCA

Date of Last Analysis:	Aug 5, 2025, 1:17 PM	**Engine Version:**	21.1.2.0002
Host Name:	df80d5ba2227	**Certification:**	VALID
Number of Files:	58	**Lines of Code:**	7,581

Rulepack Name	Rulepack Version
Fortify Secure Coding Rules, Community, Cloud	2025.3.0.0007
Fortify Secure Coding Rules, Community, Universal	2025.3.0.0007
Fortify Secure Coding Rules, Core, Android	2025.3.0.0007
Fortify Secure Coding Rules, Core, Annotations	2025.3.0.0007
Fortify Secure Coding Rules, Core, Cloud	2025.3.0.0007
Fortify Secure Coding Rules, Core, Java	2025.3.0.0007
Fortify Secure Coding Rules, Core, Universal	2025.3.0.0007
Fortify Secure Coding Rules, Extended, Configuration	2025.3.0.0007
Fortify Secure Coding Rules, Extended, Content	2025.3.0.0007
Fortify Secure Coding Rules, Extended, Java	2025.3.0.0007
Fortify Secure Coding Rules, Extended, JSP	2025.3.0.0007

Issue Breakdown by Fortify Categories

The following table depicts a summary of all issues grouped vertically by Fortify Category. For each category, the total number of issues is shown by Fortify Priority Order, including information about the number of audited issues.

Category	Fortify Priority (audited/total)				Total Issues
	Critical	High	Medium	Low	
Code Correctness: Constructor Invokes Overridable Function	0	0	0	0 / 1	0 / 1
J2EE Bad Practices: Threads	0	0	0	0 / 14	0 / 14
Poor Error Handling: Empty Catch Block	0	0	0	0 / 4	0 / 4
Poor Error Handling: Overly Broad Catch	0	0	0	0 / 42	0 / 42
Poor Style: Value Never Read	0	0	0	0 / 37	0 / 37
Portability Flaw: Locale Dependent Comparison	0	0	0	0 / 3	0 / 3
Race Condition: Format Flaw	0	0	0	0 / 1	0 / 1
Unchecked Return Value	0	0	0	0 / 2	0 / 2
Weak Cryptographic Hash	0	0	0	0 / 1	0 / 1

Results Outline

Code Correctness: Constructor Invokes Overridable Function (1 issue)

Abstract

A constructor of the class calls a function that can be overridden.

Explanation

When a constructor calls an overridable function, it may allow an attacker to access the `this` reference prior to the object being fully initialized, which can in turn lead to a vulnerability. **Example 1:** The following calls a method that can be overridden.

```
...
class User {
  private String username;
  private boolean valid;
  public User(String username, String password){
    this.username = username;
    this.valid = validateUser(username, password);
  }
  public boolean validateUser(String username, String password){
    //validate user is real and can authenticate
    ...
  }
  public final boolean isValid(){
    return valid;
  }
}
```

Since the function `validateUser` and the class are not `final`, it means that they can be overridden, and then initializing a variable to the subclass that overrides this function would allow bypassing of the `validateUser` functionality. For example:

```
...
class Attacker extends User{
  public Attacker(String username, String password){
    super(username, password);
  }
  public boolean validateUser(String username, String password){
    return true;
  }
}
...
class MainClass{
  public static void main(String[] args){
    User hacker = new Attacker("Evil", "Hacker");
    if (hacker.isValid()){
      System.out.println("Attack successful!");
    }else{
      System.out.println("Attack failed");
    }
  }
}
```

The code in `Example 1` prints "Attack successful!", since the `Attacker` class overrides the `validateUser()` function that is called from the constructor of the superclass `User`, and Java will first look in the subclass for functions called from the constructor.

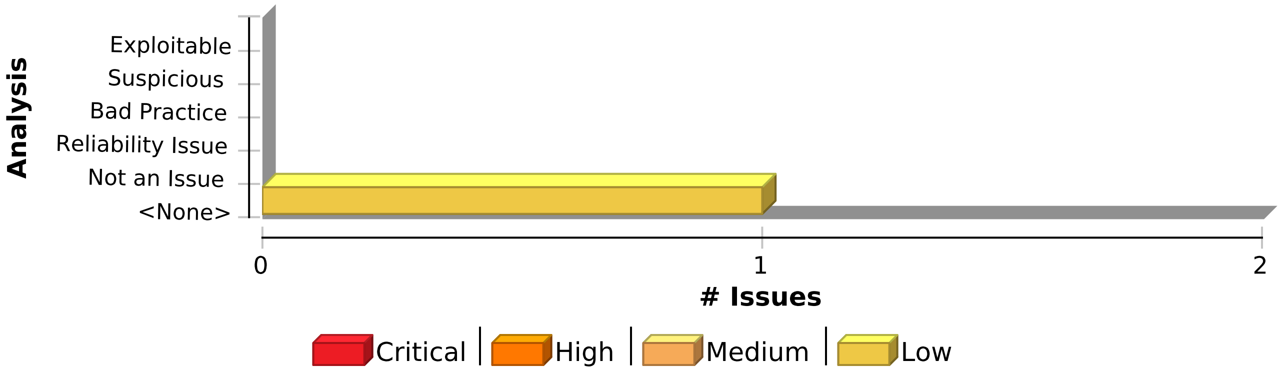
Recommendation

Constructors should not call functions that can be overridden, either by specifying them as `final`, or specifying the class as `final`. Alternatively if this code is only ever needed in the constructor, the `private` access specifier can be used, or the logic could be placed directly into the constructor of the superclass. **Example 2:** The following makes the class `final` to prevent the function from being overridden elsewhere.

```
  ...
  final class User {
    private String username;
    private boolean valid;
    public User(String username, String password){
      this.username = username;
      this.valid = validateUser(username, password);
    }
    private boolean validateUser(String username, String password){
      //validate user is real and can authenticate
      ...
    }
    public final boolean isValid(){
      return valid;
    }
  }
```

This example specifies the class as `final`, so that it cannot be subclassed, and changes the `validateUser()` function to `private`, since it is not needed elsewhere in this application. This is programming defensively, since at a later date it may be decided that the `User` class needs to be subclassed, which would result in this vulnerability reappearing if the `validateUser()` function was not set to `private`.

Issue Summary



Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Code Correctness: Constructor Invokes Overridable Function	1	0	0	1
Total	**1**	**0**	**0**	**1**

Code Correctness: Constructor Invokes Overridable Function	**Low**
Package: com.skyfi.atak.plugin	
AOIVisualizationManager.java, line 41 (Code Correctness: Constructor Invokes Overridable Function)	**Low**

Issue Details

Code Correctness: Constructor Invokes Overridable Function — Low

Package: com.skyfi.atak.plugin

AOIVisualizationManager.java, line 41 (Code Correctness: Constructor Invokes Overridable Function) — Low

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: loadAndDisplayAllAOIs
Enclosing Method: AOIVisualizationManager()
File: AOIVisualizationManager.java:41
Taint Flags:

```
38  this.aoiMapItems = new HashMap<>();
39
40  initializeMapGroup();
41  loadAndDisplayAllAOIs();
42  }
43
44  private void initializeMapGroup() {
```

J2EE Bad Practices: Threads (14 issues)

Abstract

Thread management in a web application is forbidden in some circumstances and is always highly error prone.

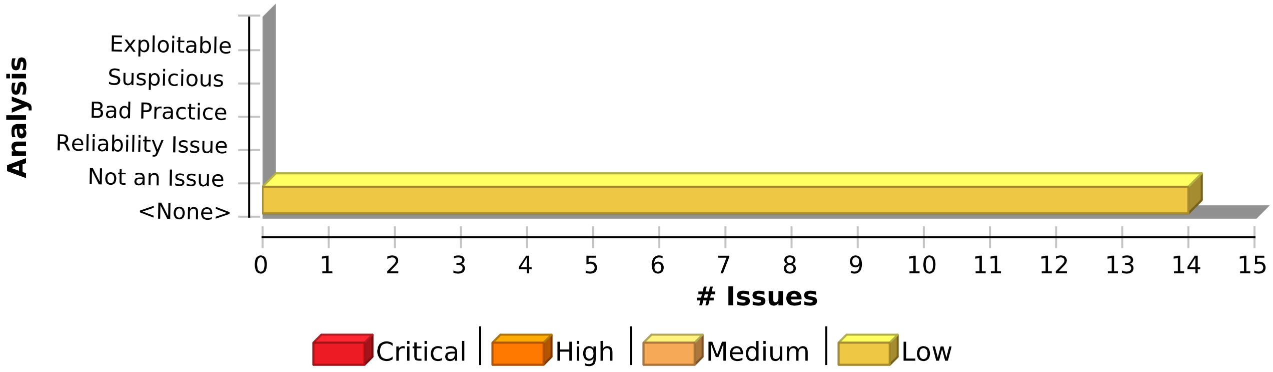
Explanation

Thread management in a web application is forbidden by the J2EE standard in some circumstances and is always highly error prone. Managing threads is difficult and is likely to interfere in unpredictable ways with the behavior of the application container. Even without interfering with the container, thread management usually leads to bugs that are hard to detect and diagnose like deadlock, race conditions, and other synchronization errors.

Recommendation

Avoid managing threads directly from within the web application. Instead use standards such as message driven beans and the EJB timer service that are provided by the application container.

Issue Summary



Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
J2EE Bad Practices: Threads	14	0	0	14
Total	**14**	**0**	**0**	**14**

J2EE Bad Practices: Threads **Low**

Package: com.skyfi.atak.plugin

ImageCacheManager.java, line 360 (J2EE Bad Practices: Threads) **Low**

Issue Details

Kingdom: Time and State
Scan Engine: SCA (Semantic)

Sink Details

Sink: run()
Enclosing Method: cacheRegionImages()
File: ImageCacheManager.java:360
Taint Flags:

J2EE Bad Practices: Threads — Low

Package: com.skyfi.atak.plugin

ImageCacheManager.java, line 360 (J2EE Bad Practices: Threads) — Low

```
357 return;
358 }
359
360 executorService.execute(() -> {
361 // Filter images based on current AOR settings
362 List<String> filteredUrls = new ArrayList<>();
363 for (int i = 0; i < imageUrls.size(); i++) {
```

ArchivesBrowserRecyclerViewAdapter.java, line 227 (J2EE Bad Practices: Threads) — Low

Issue Details

Kingdom: Time and State
Scan Engine: SCA (Semantic)

Sink Details

Sink: run()
Enclosing Method: handleCacheButtonClick()
File: ArchivesBrowserRecyclerViewAdapter.java:227
Taint Flags:

```
224 holder.cacheProgressText.setText(context.getString(R.string.caching_progress));
225
226 // Start caching in background thread
227 Thread cacheThread = new Thread(() -> {
228 try {
229 URL url = new URL(imageUrl);
230 URLConnection connection = url.openConnection();
```

ImageCacheManager.java, line 97 (J2EE Bad Practices: Threads) — Low

Issue Details

Kingdom: Time and State
Scan Engine: SCA (Semantic)

Sink Details

Sink: run()
Enclosing Method: cacheImageData()
File: ImageCacheManager.java:97
Taint Flags:

```
94 return;
95 }
96
97 executorService.execute(() -> {
98 try {
99 Bitmap bitmap = BitmapFactory.decodeByteArray(data, 0, data.length);
```

J2EE Bad Practices: Threads — Low

Package: com.skyfi.atak.plugin

ImageCacheManager.java, line 97 (J2EE Bad Practices: Threads) — Low

```
100 if (bitmap != null) {
```

ImageCacheManager.java, line 156 (J2EE Bad Practices: Threads) — Low

Issue Details

Kingdom: Time and State
Scan Engine: SCA (Semantic)

Sink Details

Sink: run()
Enclosing Method: clearCache()
File: ImageCacheManager.java:156
Taint Flags:

```
153 public void clearCache() {
154 memoryCache.evictAll();
155
156 executorService.execute(() -> {
157 File[] files = diskCacheDir.listFiles();
158 if (files != null) {
159 for (File file : files) {
```

PreviewThumbnailAdapter.java, line 115 (J2EE Bad Practices: Threads) — Low

Issue Details

Kingdom: Time and State
Scan Engine: SCA (Semantic)

Sink Details

Sink: run()
Enclosing Method: loadThumbnailImage()
File: PreviewThumbnailAdapter.java:115
Taint Flags:

```
112
113 private void loadThumbnailImage(ImageView imageView, String imageUrl) {
114 Handler handler = new Handler(Looper.getMainLooper());
115 Thread thread = new Thread(() -> {
116 try {
117 URL thumbnailUrl = new URL(imageUrl);
118 URLConnection connection = thumbnailUrl.openConnection();
```

ImageCacheManager.java, line 244 (J2EE Bad Practices: Threads) — Low

Issue Details

Kingdom: Time and State

J2EE Bad Practices: Threads — Low

Package: com.skyfi.atak.plugin

ImageCacheManager.java, line 244 (J2EE Bad Practices: Threads) — Low

Scan Engine: SCA (Semantic)

Sink Details

Sink: run()
Enclosing Method: cleanupOldCache()
File: ImageCacheManager.java:244
Taint Flags:

```
241 }
242
243 private void cleanupOldCache() {
244 executorService.execute(() -> {
245 long totalSize = getCacheSize();
246 if (totalSize > DISK_CACHE_SIZE) {
247 // Delete oldest files until under limit
```

ImageCacheManager.java, line 278 (J2EE Bad Practices: Threads) — Low

Issue Details

Kingdom: Time and State
Scan Engine: SCA (Semantic)

Sink Details

Sink: run()
Enclosing Method: cacheHighResImages()
File: ImageCacheManager.java:278
Taint Flags:

```
275 return;
276 }
277
278 executorService.execute(() -> {
279 int total = imageUrls.size();
280 int completed = 0;
281 int failed = 0;
```

ArchivesBrowserRecyclerViewAdapter.java, line 266 (J2EE Bad Practices: Threads) — Low

Issue Details

Kingdom: Time and State
Scan Engine: SCA (Semantic)

Sink Details

Sink: start()
Enclosing Method: handleCacheButtonClick()
File: ArchivesBrowserRecyclerViewAdapter.java:266

J2EE Bad Practices: Threads — Low

Package: com.skyfi.atak.plugin

ArchivesBrowserRecyclerViewAdapter.java, line 266 (J2EE Bad Practices: Threads) — Low

Taint Flags:

```
263 }
264 });
265
266 cacheThread.start();
267 }
268
269 }
```

ArchivesBrowserRecyclerViewAdapter.java, line 86 (J2EE Bad Practices: Threads) — Low

Issue Details

Kingdom: Time and State
Scan Engine: SCA (Semantic)

Sink Details

Sink: start()
Enclosing Method: onBindViewHolder()
File: ArchivesBrowserRecyclerViewAdapter.java:86
Taint Flags:

```
83 }
84 });
85
86 thread.start();
87
88
89 holder.productType.setText(archive.getProductType());
```

PreviewThumbnailAdapter.java, line 133 (J2EE Bad Practices: Threads) — Low

Issue Details

Kingdom: Time and State
Scan Engine: SCA (Semantic)

Sink Details

Sink: start()
Enclosing Method: loadThumbnailImage()
File: PreviewThumbnailAdapter.java:133
Taint Flags:

```
130 handler.post(() -> imageView.setImageResource(R.drawable.placeholder));
131 }
132 });
133 thread.start();
134 }
```

J2EE Bad Practices: Threads — Low

Package: com.skyfi.atak.plugin

PreviewThumbnailAdapter.java, line 133 (J2EE Bad Practices: Threads) — Low

```
135
136 public static class ViewHolder extends RecyclerView.ViewHolder {
```

ImageCacheManager.java, line 83 (J2EE Bad Practices: Threads) — Low

Issue Details

Kingdom: Time and State
Scan Engine: SCA (Semantic)

Sink Details

Sink: run()
Enclosing Method: cacheImage()
File: ImageCacheManager.java:83
Taint Flags:

```
80 memoryCache.put(key, bitmap);
81
82 // Save to disk cache asynchronously
83 executorService.execute(() -> {
84 boolean success = saveToDisk(key, bitmap);
85 if (callback != null) {
86 callback.onCached(success);
```

ArchivesBrowserRecyclerViewAdapter.java, line 68 (J2EE Bad Practices: Threads) — Low

Issue Details

Kingdom: Time and State
Scan Engine: SCA (Semantic)

Sink Details

Sink: run()
Enclosing Method: onBindViewHolder()
File: ArchivesBrowserRecyclerViewAdapter.java:68
Taint Flags:

```
65 }*/
66
67 Handler handler = new Handler(Looper.getMainLooper());
68 Thread thread = new Thread(() -> {
69 try {
70 Map.Entry<String, String> entry = archive.getThumbnailUrls().entrySet().iterator().next();
71 String key = entry.getKey();
```

ImageCacheManager.java, line 398 (J2EE Bad Practices: Threads) — Low

Issue Details

J2EE Bad Practices: Threads — Low

Package: com.skyfi.atak.plugin

ImageCacheManager.java, line 398 (J2EE Bad Practices: Threads) — Low

Kingdom: Time and State
Scan Engine: SCA (Semantic)

Sink Details

Sink: run()
Enclosing Method: getCacheStats()
File: ImageCacheManager.java:398
Taint Flags:

```
395 * Get cache statistics for region vs world
396 */
397 public void getCacheStats(CacheStatsCallback callback) {
398 executorService.execute(() -> {
399 File[] files = diskCacheDir.listFiles();
400 if (files == null) {
401 if (callback != null) {
```

ImageCacheManager.java, line 437 (J2EE Bad Practices: Threads) — Low

Issue Details

Kingdom: Time and State
Scan Engine: SCA (Semantic)

Sink Details

Sink: run()
Enclosing Method: clearNonRegionCache()
File: ImageCacheManager.java:437
Taint Flags:

```
434 return;
435 }
436
437 executorService.execute(() -> {
438 // In a full implementation, you would need to track which cached files
439 // correspond to which geographic locations to selectively clear them
440 // For now, this is a placeholder that could be enhanced with metadata storage
```

Poor Error Handling: Empty Catch Block (4 issues)

Abstract

Ignoring an exception can cause the program to overlook unexpected states and conditions.

Explanation

Just about every serious attack on a software system begins with the violation of a programmer's assumptions. After the attack, the programmer's assumptions seem flimsy and poorly founded, but before an attack many programmers would defend their assumptions well past the end of their lunch break. Two dubious assumptions that are easy to spot in code are "this method call can never fail" and "it doesn't matter if this call fails". When a programmer ignores an exception, they implicitly state that they are operating under one of these assumptions. **Example 1:** The following code excerpt ignores a rarely-thrown exception from `doExchange()`.

```
try {
  doExchange();
}
catch (RareException e) {
  // this can never happen
}
```

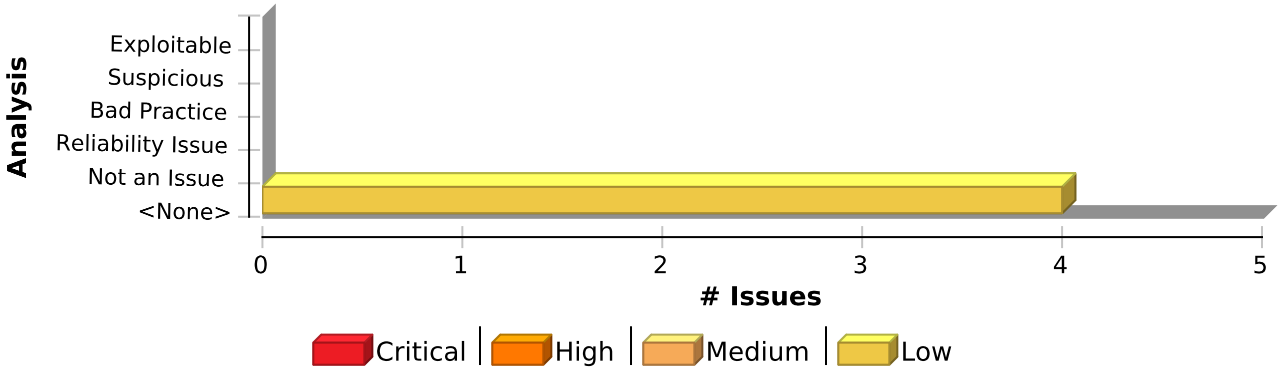
If a `RareException` were to ever be thrown, the program would continue to execute as though nothing unusual had occurred. The program records no evidence indicating the special situation, potentially frustrating any later attempt to explain the program's behavior.

Recommendation

At a minimum, log the fact that the exception was thrown so that it will be possible to come back later and make sense of the resulting program behavior. Better yet, abort the current operation. If the exception is being ignored because the caller cannot properly handle it but the context makes it inconvenient or impossible for the caller to declare that it throws the exception itself, consider throwing a `RuntimeException` or an `Error`, both of which are unchecked exceptions. As of JDK 1.4, `RuntimeException` has a constructor that makes it easy to wrap another exception. **Example 2:** The code in `Example 1` could be rewritten in the following way:

```
try {
  doExchange();
}
catch (RareException e) {
  throw new RuntimeException("This can never happen", e);
}
```

Issue Summary



Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Poor Error Handling: Empty Catch Block	4	0	0	4
Total	**4**	**0**	**0**	**4**

Poor Error Handling: Empty Catch Block — Low

Package: com.skyfi.atak.plugin

ImageCacheManager.java, line 207 (Poor Error Handling: Empty Catch Block) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: saveToDisk()
File: ImageCacheManager.java:207
Taint Flags:

```
204 if (fos != null) {
205 try {
206 fos.close();
207 } catch (IOException e) {
208 // Ignore
209 }
210 }
```

CoordinateInputDialog.java, line 188 (Poor Error Handling: Empty Catch Block) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: getCurrentLocation()
File: CoordinateInputDialog.java:188
Taint Flags:

```
185 return new GeoPoint(selfPoint.getLatitude(), selfPoint.getLongitude());
186 }
187 }
188 } catch (Exception e) {
189 // Fall back to default location if ATAK location is not available
190 }
191
```

ImageCacheManager.java, line 228 (Poor Error Handling: Empty Catch Block) — Low

Issue Details

Poor Error Handling: Empty Catch Block | Low

Package: com.skyfi.atak.plugin

ImageCacheManager.java, line 228 (Poor Error Handling: Empty Catch Block) | Low

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: saveRawToDisk()
File: ImageCacheManager.java:228
Taint Flags:

```
225  if (fos != null) {
226  try {
227  fos.close();
228  } catch (IOException e) {
229  // Ignore
230  }
231  }
```

SkyFiPlugin.java, line 96 (Poor Error Handling: Empty Catch Block) | Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: SkyFiPlugin()
File: SkyFiPlugin.java:96
Taint Flags:

```
93
94  try {
95  Looper.prepare();
96  } catch (Exception e) {}
97
98  // initialize the toolbar button for the plugin
99
```

Poor Error Handling: Overly Broad Catch (42 issues)

Abstract

The catch block handles a broad swath of exceptions, potentially trapping dissimilar issues or problems that should not be dealt with at this point in the program.

Explanation

Multiple catch blocks can get repetitive, but "condensing" catch blocks by catching a high-level class such as `Exception` can obscure exceptions that deserve special treatment or that should not be caught at this point in the program. Catching an overly broad exception essentially defeats the purpose of Java's typed exceptions, and can become particularly dangerous if the program grows and begins to throw new types of exceptions. The new exception types will not receive any attention. **Example 1:** The following code excerpt handles three types of exceptions in an identical fashion.

```
  try {
    doExchange();
  }
  catch (IOException e) {
    logger.error("doExchange failed", e);
  }
  catch (InvocationTargetException e) {
    logger.error("doExchange failed", e);
  }
  catch (SQLException e) {
    logger.error("doExchange failed", e);
  }
```

At first blush, it may seem preferable to deal with these exceptions in a single catch block, as follows:

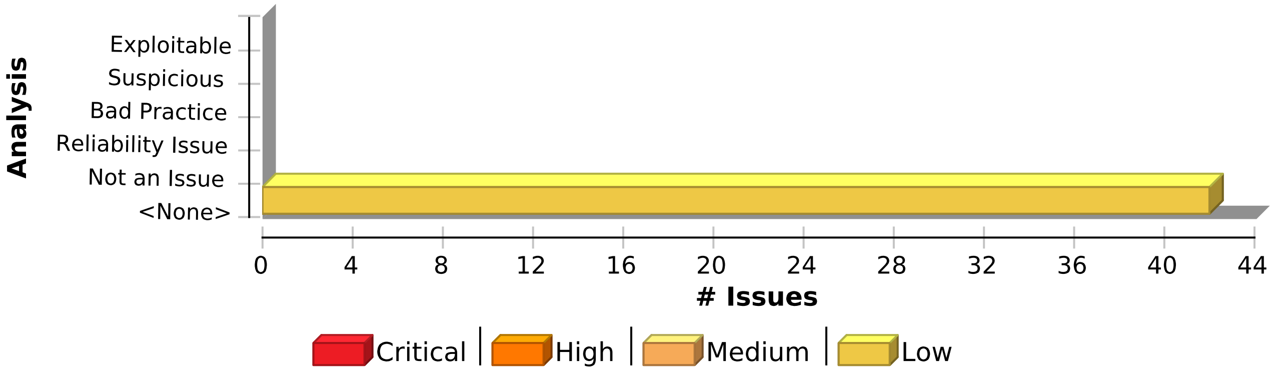
```
  try {
    doExchange();
  }
  catch (Exception e) {
    logger.error("doExchange failed", e);
  }
```

However, if `doExchange()` is modified to throw a new type of exception that should be handled in some different kind of way, the broad catch block will prevent the compiler from pointing out the situation. Further, the new catch block will now also handle exceptions derived from `RuntimeException` such as `ClassCastException`, and `NullPointerException`, which is not the programmer's intent.

Recommendation

Do not catch broad exception classes such as `Exception`, `Throwable`, `Error`, or `RuntimeException` except at the very top level of the program or thread.

Issue Summary



Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Poor Error Handling: Overly Broad Catch	42	0	0	42
Total	**42**	**0**	**0**	**42**

Poor Error Handling: Overly Broad Catch — Low

Package: com.skyfi.atak.plugin

ImageryPreviewManager.java, line 441 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: extractCenterFromArchive()
File: ImageryPreviewManager.java:441
Taint Flags:

```
438
439  return new GeoPoint(lat, lon);
440
441  } catch (Exception e) {
442  Log.w(LOGTAG, "Could not parse archive footprint: " + archive.getFootprint(), e);
443
444  // Fallback: try to extract center point from footprint string using simple parsing
```

OrdersRecyclerViewAdapter.java, line 93 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: onBindViewHolder()
File: OrdersRecyclerViewAdapter.java:93
Taint Flags:

Poor Error Handling: Overly Broad Catch — Low

Package: com.skyfi.atak.plugin

OrdersRecyclerViewAdapter.java, line 93 (Poor Error Handling: Overly Broad Catch) — Low

```
90 } else {
91 holder.opacityControlSection.setVisibility(View.GONE);
92 }
93 } catch (Exception e) {
94 Log.d(LOGTAG, "Failed", e);
95 }
96 }
```

TaskingOrderFragment.java, line 731 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: placeOrder()
File: TaskingOrderFragment.java:731
Taint Flags:

```
728 showError(context.getString(R.string.order_error), context.getString(R.string.window_end_error));
729 return;
730 }
731 } catch (Exception e) {
732 Log.d(LOGTAG, "Invalid window start: " + taskingOrder.getWindowStart(), e);
733 showError(context.getString(R.string.order_error), context.getString(R.string.window_start_error));
734 return;
```

ArchivesBrowser.java, line 219 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: parseResponse()
File: ArchivesBrowser.java:219
Taint Flags:

```
216 synchronized (recyclerViewAdapter) {
217 recyclerViewAdapter.notifyDataSetChanged();
218 }
219 } catch (Exception e) {
220 Log.e(LOGTAG, "Failed to search archives", e);
221 showError("Failed to search archives", e.getMessage());
```

Poor Error Handling: Overly Broad Catch — Low

Package: com.skyfi.atak.plugin

ArchivesBrowser.java, line 219 (Poor Error Handling: Overly Broad Catch) — Low

```
222 }
```

ArchiveSearch.java, line 231 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: onTextChanged()
File: ArchiveSearch.java:231
Taint Flags:

```
228 request.setMinOverlapRatio(value);
229 else
230 Toast.makeText(context, R.string.min_overlap_ratio_error, Toast.LENGTH_LONG).show();
231 } catch (Exception e) {
232 Toast.makeText(context, R.string.min_overlap_ratio_error, Toast.LENGTH_LONG).show();
233 }
234 }
```

TaskingOrderFragment.java, line 254 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: onTextChanged()
File: TaskingOrderFragment.java:254
Taint Flags:

```
251 float value = Float.parseFloat(charSequence.toString());
252 taskingOrder.setMaxOffNadirAngle(value);
253 }
254 } catch (Exception e) {
255 showError(context.getString(R.string.max_off_nadir_error), e.getMessage());
256 }
257 }
```

TaskingOrderFragment.java, line 1183 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors

Poor Error Handling: Overly Broad Catch — Low

Package: com.skyfi.atak.plugin

TaskingOrderFragment.java, line 1183 (Poor Error Handling: Overly Broad Catch) — Low

Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: convertPointsToWKT()
File: TaskingOrderFragment.java:1183
Taint Flags:

```
1180 factory.createPolygon(coordinates.toArray(new Coordinate[coordinates.size()]));
1181 org.locationtech.jts.io.WKTWriter wktWriter = new org.locationtech.jts.io.WKTWriter();
1182 return wktWriter.write(polygon);
1183 } catch (Exception e) {
1184 Log.e(LOGTAG, "Failed to convert points to WKT", e);
1185 return null;
1186 }
```

AORFilterManager.java, line 214 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: extractCenterFromFootprint()
File: AORFilterManager.java:214
Taint Flags:

```
211 double lon = Double.parseDouble(parts[1]);
212 return new GeoPoint(lat, lon);
213 }
214 } catch (Exception e) {
215 Log.w(TAG, "Could not parse coordinates from footprint: " + footprint, e);
216 }
217
```

ImageryPreviewManager.java, line 453 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: extractCenterFromArchive()
File: ImageryPreviewManager.java:453

Poor Error Handling: Overly Broad Catch — Low

Package: com.skyfi.atak.plugin

ImageryPreviewManager.java, line 453 (Poor Error Handling: Overly Broad Catch) — Low

Taint Flags:

```
450 double lon = Double.parseDouble(parts[1]);
451 return new GeoPoint(lat, lon);
452 }
453 } catch (Exception e2) {
454 Log.w(LOGTAG, "Could not extract center point from footprint", e2);
455 }
456 }
```

TaskingOrderFragment.java, line 862 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: updateTotalPrice()
File: TaskingOrderFragment.java:862
Taint Flags:

```
859 }
860 break;
861 }
862 catch (Exception e) {
863 Log.e(LOGTAG, "Failed to calculate price", e);
864 }
865 }
```

ImageryPreviewManager.java, line 416 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: createAOIAroundPoint()
File: ImageryPreviewManager.java:416
Taint Flags:

```
413 WKTWriter wktWriter = new WKTWriter();
414
415 return wktWriter.write(polygon);
416 } catch (Exception e) {
417 Log.e(LOGTAG, "Failed to create AOI around point", e);
```

Poor Error Handling: Overly Broad Catch

Low

Package: com.skyfi.atak.plugin

ImageryPreviewManager.java, line 416 (Poor Error Handling: Overly Broad Catch)

Low

```
418 return null;
419 }
```

AOIVisualizationManager.java, line 109 (Poor Error Handling: Overly Broad Catch)

Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: displayAOI()
File: AOIVisualizationManager.java:109
Taint Flags:

```
106
107 Log.d(TAG, "Displayed AOI: " + aoi.name);
108
109 } catch (Exception e) {
110 Log.e(TAG, "Failed to display AOI: " + aoi.name, e);
111 }
112 }
```

SkyFiPlugin.java, line 496 (Poor Error Handling: Overly Broad Catch)

Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: squareWkt()
File: SkyFiPlugin.java:496
Taint Flags:

```
493 Polygon polygon = factory.createPolygon(coordinates.toArray(new Coordinate[coordinates.size()]));
494 WKTWriter wktWriter = new WKTWriter();
495 return wktWriter.write(polygon);
496 } catch (Exception e) {
497 Log.e(LOGTAG, "Failed to make square WKT", e);
498 }
499
```

AORFilterManager.java, line 183 (Poor Error Handling: Overly Broad Catch)

Low

Issue Details

Poor Error Handling: Overly Broad Catch

Low

Package: com.skyfi.atak.plugin

AORFilterManager.java, line 183 (Poor Error Handling: Overly Broad Catch)

Low

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: isArchiveInAOR()
File: AORFilterManager.java:183
Taint Flags:

```
180 // Check if the archive geometry intersects with the AOR
181 return aorPolygon.intersects(archiveGeometry);
182
183 } catch (Exception e) {
184 Log.w(TAG, "Could not parse archive footprint: " + archive.getFootprint(), e);
185
186 // Fallback: try to extract center point from footprint string
```

SkyFiPlugin.java, line 433 (Poor Error Handling: Overly Broad Catch)

Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: onShapeComplete()
File: SkyFiPlugin.java:433
Taint Flags:

```
430 .setNegativeButton("Later", null)
431 .show();
432
433 } catch (Exception e) {
434 Log.e(LOGTAG, "Failed to save AOI", e);
435 Toast.makeText(pluginContext, "Failed to save AOI: " + e.getMessage(),
436 Toast.LENGTH_SHORT).show();
```

Orders.java, line 345 (Poor Error Handling: Overly Broad Catch)

Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: onShowAdvancedOpacityDialog()

Poor Error Handling: Overly Broad Catch — Low

Package: com.skyfi.atak.plugin

Orders.java, line 345 (Poor Error Handling: Overly Broad Catch) — Low

File: Orders.java:345
Taint Flags:

```
342 ordersRecyclerViewAdapter.notifyDataSetChanged();
343 }
344 );
345 } catch (Exception e) {
346 Log.e(LOGTAG, "Failed to show advanced opacity dialog", e);
347 }
348 }
```

Orders.java, line 324 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: onOpacityChanged()
File: Orders.java:324
Taint Flags:

```
321 AtakBroadcast.getInstance().sendBroadcast(opacityIntent);
322
323 Log.d(LOGTAG, "Set opacity for layer " + layerName + " to " + opacity + "%");
324 } catch (Exception e) {
325 Log.e(LOGTAG, "Failed to set layer opacity", e);
326 }
327 }
```

TaskingOrderFragment.java, line 233 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: onTextChanged()
File: TaskingOrderFragment.java:233
Taint Flags:

```
230 float value = Float.parseFloat(charSequence.toString());
231 taskingOrder.setMaxCloudCoveragePercent(value);
232 }
233 } catch (Exception e) {
```

Poor Error Handling: Overly Broad Catch — Low

Package: com.skyfi.atak.plugin

TaskingOrderFragment.java, line 233 (Poor Error Handling: Overly Broad Catch) — Low

```
234 showError(context.getString(R.string.error), context.getString(R.string.max_cloud_coverage_error));
235 }
236 }
```

ArchiveSearch.java, line 209 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: onTextChanged()
File: ArchiveSearch.java:209
Taint Flags:

```
206 request.setMaxOffNadirAngle(value);
207 else
208 Toast.makeText(context, R.string.max_off_nadir_error, Toast.LENGTH_LONG).show();
209 } catch (Exception e) {
210 Toast.makeText(context, R.string.max_off_nadir_error, Toast.LENGTH_LONG).show();
211 }
212 }
```

ArchivesBrowser.java, line 338 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: onResponse()
File: ArchivesBrowser.java:338
Taint Flags:

```
335 .setPositiveButton(context.getString(R.string.ok), null)
336 .create()
337 .show();
338 } catch (Exception e) {
339 Log.e(LOGTAG, "Failed to fail", e);
340 }
341 }
```

SkyFiPlugin.java, line 790 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Poor Error Handling: Overly Broad Catch — Low

Package: com.skyfi.atak.plugin

SkyFiPlugin.java, line 790 (Poor Error Handling: Overly Broad Catch) — Low

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: calculateAreaFromWKT()
File: SkyFiPlugin.java:790
Taint Flags:

```
787 org.locationtech.jts.io.WKTReader reader = new org.locationtech.jts.io.WKTReader();
788 org.locationtech.jts.geom.Geometry geometry = reader.read(wkt);
789 return TaskingOrderFragment.calculatePolygonArea(geometry.getCoordinates());
790 } catch (Exception e) {
791 Log.e(LOGTAG, "Failed to calculate area from WKT", e);
792 return 1.0; // Default fallback
793 }
```

Orders.java, line 149 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: onResponse()
File: Orders.java:149
Taint Flags:

```
146 synchronized (ordersRecyclerViewAdapter) {
147 ordersRecyclerViewAdapter.notifyDataSetChanged();
148 }
149 } catch (Exception e) {
150 Log.e(LOGTAG, "Failed to get orders", e);
151 showAlert(context.getString(R.string.failed_to_get_orders), e.toString());
152 }
```

AORFilterManager.java, line 161 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: filterArchivesByAOR()

Poor Error Handling: Overly Broad Catch — Low

Package: com.skyfi.atak.plugin

AORFilterManager.java, line 161 (Poor Error Handling: Overly Broad Catch) — Low

File: AORFilterManager.java:161
Taint Flags:

```
158 filteredArchives.add(archive);
159 }
160 }
161 } catch (Exception e) {
162 Log.e(TAG, "Error filtering archives by AOR", e);
163 return archives; // Return original list if filtering fails
164 }
```

SkyFiPlugin.java, line 340 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: onItemClick()
File: SkyFiPlugin.java:340
Taint Flags:

```
337 alertDialog.setView(linearLayout);
338
339 alertDialog.create().show();
340 } catch (Exception e) {
341 Log.e(LOGTAG, "Failed", e);
342 }
343 break;
```

TaskingOrderFragment.java, line 1194 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: calculateAreaFromWKT()
File: TaskingOrderFragment.java:1194
Taint Flags:

```
1191 org.locationtech.jts.io.WKTReader reader = new org.locationtech.jts.io.WKTReader();
1192 org.locationtech.jts.geom.Geometry geometry = reader.read(wkt);
1193 return calculatePolygonArea(geometry.getCoordinates());
1194 } catch (Exception e) {
```

Poor Error Handling: Overly Broad Catch — Low

Package: com.skyfi.atak.plugin

TaskingOrderFragment.java, line 1194 (Poor Error Handling: Overly Broad Catch) — Low

```
1195 Log.e(LOGTAG, "Failed to calculate area from WKT", e);
1196 return 1.0; // Default fallback
1197 }
```

TaskingOrderFragment.java, line 926 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: updateFeasibility()
File: TaskingOrderFragment.java:926
Taint Flags:

```
923 // Update UI
924 updateFeasibilityDisplay(feasibility);
925
926 } catch (Exception e) {
927 Log.e(LOGTAG, "Error updating feasibility", e);
928 resetFeasibilityDisplay();
929 }
```

SkyFiPlugin.java, line 96 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: SkyFiPlugin()
File: SkyFiPlugin.java:96
Taint Flags:

```
93
94 try {
95 Looper.prepare();
96 } catch (Exception e) {}
97
98 // initialize the toolbar button for the plugin
99
```

SkyFiDrawingToolsHandler.java, line 194 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Poor Error Handling: Overly Broad Catch — Low

Package: com.skyfi.atak.plugin

SkyFiDrawingToolsHandler.java, line 194 (Poor Error Handling: Overly Broad Catch) — Low

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: saveShapeAsAOI()
File: SkyFiDrawingToolsHandler.java:194
Taint Flags:

```
191 aoiVisualizationManager.displayAOI(aoi);
192 }
193
194 } catch (Exception e) {
195 Log.e(TAG, "Failed to save AOI", e);
196 Toast.makeText(context, "Failed to save AOI: " + e.getMessage(), Toast.LENGTH_SHORT).show();
197 }
```

SatelliteFeasibilityCalculator.java, line 61 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: calculateFeasibility()
File: SatelliteFeasibilityCalculator.java:61
Taint Flags:

```
58 return new FeasibilityInfo(expectedPasses, level, sensorType, explanation,
59 latitude, longitude, dateRange);
60
61 } catch (Exception e) {
62 return createDefaultFeasibility();
63 }
64 }
```

ArchivesBrowserRecyclerViewAdapter.java, line 99 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock

Poor Error Handling: Overly Broad Catch — Low

Package: com.skyfi.atak.plugin

ArchivesBrowserRecyclerViewAdapter.java, line 99 (Poor Error Handling: Overly Broad Catch) — Low

Enclosing Method: onBindViewHolder()
File: ArchivesBrowserRecyclerViewAdapter.java:99
Taint Flags:

```
96  // Update button states based on preferences and cache status
97  updateButtonStates(holder, archive.getArchiveId());
98  updateCacheButtonState(holder, archive);
99  } catch (Exception e) {
100  Log.d(LOGTAG, "Failed", e);
101  }
102  }
```

SkyFiPlugin.java, line 779 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: convertPointsToWKT()
File: SkyFiPlugin.java:779
Taint Flags:

```
776  Polygon polygon = factory.createPolygon(coordinates.toArray(new Coordinate[coordinates.size()]));
777  WKTWriter wktWriter = new WKTWriter();
778  return wktWriter.write(polygon);
779  } catch (Exception e) {
780  Log.e(LOGTAG, "Failed to convert points to WKT", e);
781  return null;
782  }
```

Profile.java, line 73 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: onResponse()
File: Profile.java:73
Taint Flags:

```
70  .show();
71  try {
```

Poor Error Handling: Overly Broad Catch — Low

Package: com.skyfi.atak.plugin

Profile.java, line 73 (Poor Error Handling: Overly Broad Catch) — Low

```
72  Log.e(LOGTAG, response.errorBody().string() + " " + response.code());
73  } catch (Exception e) {
74  Log.e(LOGTAG, "no error message " + response.code());
75  }
76
```

PreviewThumbnailAdapter.java, line 79 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: onBindViewHolder()
File: PreviewThumbnailAdapter.java:79
Taint Flags:

```
76  } else {
77  holder.dateText.setText("N/A");
78  }
79  } catch (Exception e) {
80  holder.dateText.setText("N/A");
81  }
82
```

TaskingOrderFragment.java, line 788 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: onResponse()
File: TaskingOrderFragment.java:788
Taint Flags:

```
785  Log.d(LOGTAG, call.request().body().toString());
786  Log.e(LOGTAG, message);
787  showError(context.getString(R.string.get_pricing_failed), message);
788  } catch (Exception e) {
789  Log.e(LOGTAG, "Failed to fail", e);
790  }
791  }
```

Poor Error Handling: Overly Broad Catch — Low

Package: com.skyfi.atak.plugin

Orders.java, line 307 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: onItemClick()
File: Orders.java:307
Taint Flags:

```
304 GeoPoint[] points = geoPoints.toArray(new GeoPoint[geoPoints.size()]);
305
306 ATAKUtilities.scaleToFit(MapView.getMapView(), points, 1000, 1000);
307 } catch (Exception e) {
308 Log.e(LOGTAG, "Failed to make map source", e);
309 }
310 }
```

ArchiveSearch.java, line 187 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: onTextChanged()
File: ArchiveSearch.java:187
Taint Flags:

```
184 else
185 maxCloudCoverage.setError(context.getString(R.string.max_cloud_coverage_error));
186 //Toast.makeText(context, R.string.max_cloud_coverage_error, Toast.LENGTH_LONG).show();
187 } catch (Exception e) {
188 Toast.makeText(context, R.string.max_cloud_coverage_error, Toast.LENGTH_LONG).show();
189 }
190 }
```

TaskingOrderFragment.java, line 755 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Poor Error Handling: Overly Broad Catch — Low

Package: com.skyfi.atak.plugin

TaskingOrderFragment.java, line 755 (Poor Error Handling: Overly Broad Catch) — Low

Sink: CatchBlock
Enclosing Method: onResponse()
File: TaskingOrderFragment.java:755
Taint Flags:

```
752 Log.d(LOGTAG, call.request().body().toString());
753 Log.e(LOGTAG, message);
754 showError(context.getString(R.string.order_error), message);
755 } catch (Exception e) {
756 Log.e(LOGTAG, "Failed to fail", e);
757 }
758 }
```

CoordinateInputDialog.java, line 188 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: getCurrentLocation()
File: CoordinateInputDialog.java:188
Taint Flags:

```
185 return new GeoPoint(selfPoint.getLatitude(), selfPoint.getLongitude());
186 }
187 }
188 } catch (Exception e) {
189 // Fall back to default location if ATAK location is not available
190 }
191
```

TaskingOrderFragment.java, line 941 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: extractCenterFromAOI()
File: TaskingOrderFragment.java:941
Taint Flags:

```
938 Geometry geometry = reader.read(aoi);
939 Coordinate centroid = geometry.getCentroid().getCoordinate();
```

Poor Error Handling: Overly Broad Catch — Low

Package: com.skyfi.atak.plugin

TaskingOrderFragment.java, line 941 (Poor Error Handling: Overly Broad Catch) — Low

```
940 return new double[]{centroid.y, centroid.x}; // lat, lon
941 } catch (Exception e) {
942 Log.e(LOGTAG, "Error extracting center from AOI", e);
943 return null;
944 }
```

ArchivesBrowser.java, line 183 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: parseResponse()
File: ArchivesBrowser.java:183
Taint Flags:

```
180 }
181 }
182 }
183 catch (Exception e) {
184 Log.e(LOGTAG, "Failed to parse page hash", e);
185 }
186
```

ArchivesBrowser.java, line 233 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: parseResponse()
File: ArchivesBrowser.java:233
Taint Flags:

```
230 Log.d(LOGTAG, call.request().body().toString());
231 Log.e(LOGTAG, message);
232 showError("Error searching archives", message);
233 } catch (Exception e) {
234 Log.e(LOGTAG, "Failed to fail", e);
235 }
236 }
```

Poor Error Handling: Overly Broad Catch — Low

Package: com.skyfi.atak.plugin

AORFilterManager.java, line 193 (Poor Error Handling: Overly Broad Catch) — Low

Issue Details

Kingdom: Errors
Scan Engine: SCA (Structural)

Sink Details

Sink: CatchBlock
Enclosing Method: isArchiveInAOR()
File: AORFilterManager.java:193
Taint Flags:

```
190 Point point = geometryFactory.createPoint(new Coordinate(centerPoint.getLongitude(), centerPoint.getLatitude()));
191 return aorPolygon.contains(point);
192 }
193 } catch (Exception e2) {
194 Log.w(TAG, "Could not extract center point from footprint", e2);
195 }
196 }
```

Poor Style: Value Never Read (37 issues)

Abstract

The variable's value is assigned but never used, making it a dead store.

Explanation

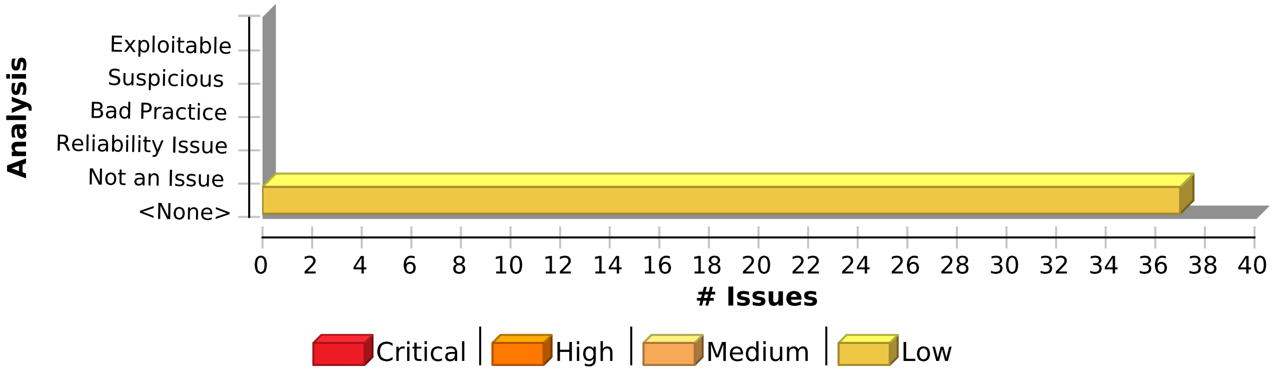
This variable's value is not used. After the assignment, the variable is either assigned another value or goes out of scope. **Example 1:** The following code excerpt assigns to the variable `r` and then overwrites the value without using it.

```
  r = getName();
  r = getNewBuffer(buf);
```

Recommendation

Remove unnecessary assignments in order to make the code easier to understand and maintain.

Issue Summary



Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Poor Style: Value Never Read	37	0	0	37
Total	**37**	**0**	**0**	**37**

Poor Style: Value Never Read — Low

Package: <none>

ArchivesBrowserRecyclerViewAdapter.java, line 71 (Poor Style: Value Never Read) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: key
Enclosing Method: run()
File: ArchivesBrowserRecyclerViewAdapter.java:71

Poor Style: Value Never Read — Low

Package: <none>

ArchivesBrowserRecyclerViewAdapter.java, line 71 (Poor Style: Value Never Read) — Low

Taint Flags:

```
68 Thread thread = new Thread(() -> {
69 try {
70 Map.Entry<String, String> entry = archive.getThumbnailUrls().entrySet().iterator().next();
71 String key = entry.getKey();
72 String value = entry.getValue();
73 URL thumbnailUrl = new URL(value);
74 URLConnection connection = thumbnailUrl.openConnection();
```

ImageCacheManager.java, line 365 (Poor Style: Value Never Read) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: lat
Enclosing Method: run()
File: ImageCacheManager.java:365
Taint Flags:

```
362 List<String> filteredUrls = new ArrayList<>();
363 for (int i = 0; i < imageUrls.size(); i++) {
364 String url = imageUrls.get(i);
365 double lat = latitudes.get(i);
366 double lon = longitudes.get(i);
367
368 // Check if this image should be cached based on AOR filter settings
```

ImageCacheManager.java, line 366 (Poor Style: Value Never Read) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: lon
Enclosing Method: run()
File: ImageCacheManager.java:366
Taint Flags:

```
363 for (int i = 0; i < imageUrls.size(); i++) {
364 String url = imageUrls.get(i);
365 double lat = latitudes.get(i);
366 double lon = longitudes.get(i);
367
```

Poor Style: Value Never Read

Low

Package: <none>

ImageCacheManager.java, line 366 (Poor Style: Value Never Read)

Low

```
368 // Check if this image should be cached based on AOR filter settings
369 // For now, cache all images since we don't have specific point-in-region checking
```

ArchivesBrowserRecyclerViewAdapter.java, line 75 (Poor Style: Value Never Read)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: image
Enclosing Method: run()
File: ArchivesBrowserRecyclerViewAdapter.java:75
Taint Flags:

```
72 String value = entry.getValue();
73 URL thumbnailUrl = new URL(value);
74 URLConnection connection = thumbnailUrl.openConnection();
75 Bitmap image = BitmapFactory.decodeStream(connection.getInputStream());
76 handler.post(() -> {
77 holder.thumbnail.setImageBitmap(image);
78 holder.thumbnail.setAlpha(1f);
```

PreviewThumbnailAdapter.java, line 119 (Poor Style: Value Never Read)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: image
Enclosing Method: run()
File: PreviewThumbnailAdapter.java:119
Taint Flags:

```
116 try {
117 URL thumbnailUrl = new URL(imageUrl);
118 URLConnection connection = thumbnailUrl.openConnection();
119 Bitmap image = BitmapFactory.decodeStream(connection.getInputStream());
120 handler.post(() -> {
121 if (image != null) {
122 imageView.setImageBitmap(image);
```

Poor Style: Value Never Read — Low

Package: com.skyfi.atak.plugin

SkyFiMapOverlay.java, line 230 (Poor Style: Value Never Read) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: phase
Enclosing Method: createAnimatedPathEffect()
File: SkyFiMapOverlay.java:230
Taint Flags:

```
227 }
228
229 private android.graphics.PathEffect createAnimatedPathEffect() {
230 float phase = (System.currentTimeMillis() % 1000) / 50f;
231 return new android.graphics.DashPathEffect(new float[]{20, 10}, phase);
232 }
233
```

ArchivesBrowser.java, line 297 (Poor Style: Value Never Read) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: archive
Enclosing Method: onItemClick()
File: ArchivesBrowser.java:297
Taint Flags:

```
294
295 @Override
296 public void onItemClick(View view, int position) {
297 Archive archive = archives.get(position);
298
299 new AlertDialog.Builder(MapView.getMapView().getContext())
300 .setTitle(context.getString(R.string.place_order))
```

ImageryPreviewManager.java, line 401 (Poor Style: Value Never Read) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Poor Style: Value Never Read — Low

Package: com.skyfi.atak.plugin

ImageryPreviewManager.java, line 401 (Poor Style: Value Never Read) — Low

Sink: VariableAccess: west
Enclosing Method: createAOIAroundPoint()
File: ImageryPreviewManager.java:401
Taint Flags:

```
398 double north = centerPoint.getLatitude() + latOffset;
399 double south = centerPoint.getLatitude() - latOffset;
400 double east = centerPoint.getLongitude() + lonOffset;
401 double west = centerPoint.getLongitude() - lonOffset;
402
403 // Create polygon coordinates
404 ArrayList<Coordinate> coordinates = new ArrayList<>();
```

SkyFiPlugin.java, line 827 (Poor Style: Value Never Read) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: apiStatus
Enclosing Method: updateDashboardMetrics()
File: SkyFiPlugin.java:827
Taint Flags:

```
824 TextView satelliteCount = mainView.findViewById(R.id.satellite_count);
825 TextView coveragePercent = mainView.findViewById(R.id.coverage_percent);
826 TextView activeOrders = mainView.findViewById(R.id.active_orders);
827 TextView apiStatus = mainView.findViewById(R.id.api_status);
828
829 // Check API connection status
830 apiClient.ping().enqueue(new Callback<com.skyfi.atak.plugin.skyfiapi.Pong>() {
```

SkyFiPlugin.java, line 451 (Poor Style: Value Never Read) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: lon
Enclosing Method: squareWkt()
File: SkyFiPlugin.java:451
Taint Flags:

```
448 private String squareWkt(double diameter) {
449 try {
```

Poor Style: Value Never Read

Low

Package: com.skyfi.atak.plugin

SkyFiPlugin.java, line 451 (Poor Style: Value Never Read)

Low

```
450 double lat = MapView.getMapView().getSelfMarker().getPoint().getLatitude();
451 double lon = MapView.getMapView().getSelfMarker().getPoint().getLongitude();
452
453 // Convert to meters
454 diameter = diameter * 1000;
```

SkyFiPlugin.java, line 417 (Poor Style: Value Never Read)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: aoiId
Enclosing Method: onShapeComplete()
File: SkyFiPlugin.java:417
Taint Flags:

```
414 AOIManager aoiManager = new AOIManager(pluginContext);
415 String aoiName = "AOI_" + System.currentTimeMillis();
416 AOIManager.AOI aoi = aoiManager.createAOI(aoiName, points, areaSqKm, "default");
417 String aoiId = aoi.id;
418
419 // Show success and offer to create order
420 new AlertDialog.Builder(MapView.getMapView().getContext())
```

CoordinateInputDialog.java, line 47 (Poor Style: Value Never Read)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: lonInput
Enclosing Method: show()
File: CoordinateInputDialog.java:47
Taint Flags:

```
44
45 RadioGroup formatGroup = dialogView.findViewById(R.id.coordinate_format_group);
46 EditText latInput = dialogView.findViewById(R.id.latitude_input);
47 EditText lonInput = dialogView.findViewById(R.id.longitude_input);
48 EditText mgrsInput = dialogView.findViewById(R.id.mgrs_input);
49 TextView currentLocationText = dialogView.findViewById(R.id.current_location_text);
50
```

Poor Style: Value Never Read

Low

Package: com.skyfi.atak.plugin

TaskingOrderFragment.java, line 939 (Poor Style: Value Never Read)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: centroid
Enclosing Method: extractCenterFromAOI()
File: TaskingOrderFragment.java:939
Taint Flags:

```
936 try {
937 WKTReader reader = new WKTReader();
938 Geometry geometry = reader.read(aoi);
939 Coordinate centroid = geometry.getCentroid().getCoordinate();
940 return new double[]{centroid.y, centroid.x}; // lat, lon
941 } catch (Exception e) {
942 Log.e(LOGTAG, "Error extracting center from AOI", e);
```

ImageryPreviewManager.java, line 399 (Poor Style: Value Never Read)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: south
Enclosing Method: createAOIAroundPoint()
File: ImageryPreviewManager.java:399
Taint Flags:

```
396 double lonOffset = radiusMeters / (111000.0 * Math.cos(Math.toRadians(centerPoint.getLatitude())));
397
398 double north = centerPoint.getLatitude() + latOffset;
399 double south = centerPoint.getLatitude() - latOffset;
400 double east = centerPoint.getLongitude() + lonOffset;
401 double west = centerPoint.getLongitude() - lonOffset;
402
```

AORFilterManager.java, line 211 (Poor Style: Value Never Read)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Poor Style: Value Never Read — Low

Package: com.skyfi.atak.plugin

AORFilterManager.java, line 211 (Poor Style: Value Never Read) — Low

Sink: VariableAccess: lon
Enclosing Method: extractCenterFromFootprint()
File: AORFilterManager.java:211
Taint Flags:

```
208  String[] parts = footprint.split("[,\\s]+");
209  if (parts.length >= 2) {
210  double lat = Double.parseDouble(parts[0]);
211  double lon = Double.parseDouble(parts[1]);
212  return new GeoPoint(lat, lon);
213  }
214  } catch (Exception e) {
```

CoordinateInputDialog.java, line 46 (Poor Style: Value Never Read) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: latInput
Enclosing Method: show()
File: CoordinateInputDialog.java:46
Taint Flags:

```
43  View dialogView = LayoutInflater.from(context).inflate(R.layout.coordinate_input_dialog, null);
44
45  RadioGroup formatGroup = dialogView.findViewById(R.id.coordinate_format_group);
46  EditText latInput = dialogView.findViewById(R.id.latitude_input);
47  EditText lonInput = dialogView.findViewById(R.id.longitude_input);
48  EditText mgrsInput = dialogView.findViewById(R.id.mgrs_input);
49  TextView currentLocationText = dialogView.findViewById(R.id.current_location_text);
```

ImageryPreviewManager.java, line 398 (Poor Style: Value Never Read) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: north
Enclosing Method: createAOIAroundPoint()
File: ImageryPreviewManager.java:398
Taint Flags:

```
395  double latOffset = radiusMeters / 111000.0; // Approximate meters per degree latitude
396  double lonOffset = radiusMeters / (111000.0 * Math.cos(Math.toRadians(centerPoint.getLatitude())));
```

Poor Style: Value Never Read — Low

Package: com.skyfi.atak.plugin

ImageryPreviewManager.java, line 398 (Poor Style: Value Never Read) — Low

```
397
398 double north = centerPoint.getLatitude() + latOffset;
399 double south = centerPoint.getLatitude() - latOffset;
400 double east = centerPoint.getLongitude() + lonOffset;
401 double west = centerPoint.getLongitude() - lonOffset;
```

SkyFiMapOverlay.java, line 122 (Poor Style: Value Never Read) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: xy
Enclosing Method: drawTaskedArea()
File: SkyFiMapOverlay.java:122
Taint Flags:

```
119 boolean first = true;
120
121 for (GeoPoint point : area.points) {
122 PointF xy = mapView.forward(point);
123 if (first) {
124 path.moveTo(xy.x, xy.y);
125 first = false;
```

SkyFiMapOverlay.java, line 146 (Poor Style: Value Never Read) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: xy
Enclosing Method: drawTaskedArea()
File: SkyFiMapOverlay.java:146
Taint Flags:

```
143 // Draw status label at centroid
144 GeoPoint centroid = calculateCentroid(area.points);
145 if (centroid != null) {
146 PointF xy = mapView.forward(centroid);
147
148 // Draw background for text
149 String label = String.format("%.1f km² - %s", area.areaKm2, area.status);
```

Poor Style: Value Never Read — Low

Package: com.skyfi.atak.plugin

SkyFiPlugin.java, line 826 (Poor Style: Value Never Read) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: activeOrders
Enclosing Method: updateDashboardMetrics()
File: SkyFiPlugin.java:826
Taint Flags:

```
823 private void updateDashboardMetrics() {
824 TextView satelliteCount = mainView.findViewById(R.id.satellite_count);
825 TextView coveragePercent = mainView.findViewById(R.id.coverage_percent);
826 TextView activeOrders = mainView.findViewById(R.id.active_orders);
827 TextView apiStatus = mainView.findViewById(R.id.api_status);
828
829 // Check API connection status
```

TaskingOrderFragment.java, line 882 (Poor Style: Value Never Read) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: p2
Enclosing Method: calculatePolygonArea()
File: TaskingOrderFragment.java:882
Taint Flags:

```
879 for (int i = 0; i < coordinates.length-1; i++) {
880 Coordinate p1, p2;
881 p1 = coordinates[i];
882 p2 = coordinates[i + 1];
883 area += Math.toRadians(p2.x - p1.x) * (2 + Math.sin(Math.toRadians(p1.y)) + Math.sin(Math.toRadians(p2.y)));
884 }
885
```

CoordinateInputDialog.java, line 49 (Poor Style: Value Never Read) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Poor Style: Value Never Read — Low

Package: com.skyfi.atak.plugin

CoordinateInputDialog.java, line 49 (Poor Style: Value Never Read) — Low

Sink: VariableAccess: currentLocationText
Enclosing Method: show()
File: CoordinateInputDialog.java:49
Taint Flags:

```
46 EditText latInput = dialogView.findViewById(R.id.latitude_input);
47 EditText lonInput = dialogView.findViewById(R.id.longitude_input);
48 EditText mgrsInput = dialogView.findViewById(R.id.mgrs_input);
49 TextView currentLocationText = dialogView.findViewById(R.id.current_location_text);
50
51 // Set up format switching
52 formatGroup.setOnCheckedChangeListener((group, checkedId) -> {
```

SkyFiPlugin.java, line 824 (Poor Style: Value Never Read) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: satelliteCount
Enclosing Method: updateDashboardMetrics()
File: SkyFiPlugin.java:824
Taint Flags:

```
821 }
822
823 private void updateDashboardMetrics() {
824 TextView satelliteCount = mainView.findViewById(R.id.satellite_count);
825 TextView coveragePercent = mainView.findViewById(R.id.coverage_percent);
826 TextView activeOrders = mainView.findViewById(R.id.active_orders);
827 TextView apiStatus = mainView.findViewById(R.id.api_status);
```

SkyFiPlugin.java, line 1125 (Poor Style: Value Never Read) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: deltaLon
Enclosing Method: createSquareAround()
File: SkyFiPlugin.java:1125
Taint Flags:

```
1122 double metersPerDegreeLon = 111412.84 * Math.cos(Math.toRadians(lat));
1123
```

Poor Style: Value Never Read

Low

Package: com.skyfi.atak.plugin

SkyFiPlugin.java, line 1125 (Poor Style: Value Never Read)

Low

```
1124  double deltaLat = radiusMeters / metersPerDegreeLat;
1125  double deltaLon = radiusMeters / metersPerDegreeLon;
1126
1127  // Create square corners
1128  points.add(new com.atakmap.coremap.maps.coords.GeoPoint(lat - deltaLat, lon - deltaLon)); // SW
```

SkyFiPlugin.java, line 1124 (Poor Style: Value Never Read)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: deltaLat
Enclosing Method: createSquareAround()
File: SkyFiPlugin.java:1124
Taint Flags:

```
1121  double metersPerDegreeLat = 111132.92 - 559.82 * Math.cos(2 * Math.toRadians(lat));
1122  double metersPerDegreeLon = 111412.84 * Math.cos(Math.toRadians(lat));
1123
1124  double deltaLat = radiusMeters / metersPerDegreeLat;
1125  double deltaLon = radiusMeters / metersPerDegreeLon;
1126
1127  // Create square corners
```

ImageryPreviewManager.java, line 400 (Poor Style: Value Never Read)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: east
Enclosing Method: createAOIAroundPoint()
File: ImageryPreviewManager.java:400
Taint Flags:

```
397
398  double north = centerPoint.getLatitude() + latOffset;
399  double south = centerPoint.getLatitude() - latOffset;
400  double east = centerPoint.getLongitude() + lonOffset;
401  double west = centerPoint.getLongitude() - lonOffset;
402
403  // Create polygon coordinates
```

Poor Style: Value Never Read — Low

Package: com.skyfi.atak.plugin

CoordinateInputDialog.java, line 48 (Poor Style: Value Never Read) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: mgrsInput
Enclosing Method: show()
File: CoordinateInputDialog.java:48
Taint Flags:

```
45 RadioGroup formatGroup = dialogView.findViewById(R.id.coordinate_format_group);
46 EditText latInput = dialogView.findViewById(R.id.latitude_input);
47 EditText lonInput = dialogView.findViewById(R.id.longitude_input);
48 EditText mgrsInput = dialogView.findViewById(R.id.mgrs_input);
49 TextView currentLocationText = dialogView.findViewById(R.id.current_location_text);
50
51 // Set up format switching
```

SkyFiMapOverlay.java, line 174 (Poor Style: Value Never Read) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: xy
Enclosing Method: drawCurrentPolygon()
File: SkyFiMapOverlay.java:174
Taint Flags:

```
171 boolean first = true;
172
173 for (GeoPoint point : currentPolygon) {
174 PointF xy = mapView.forward(point);
175 if (first) {
176 path.moveTo(xy.x, xy.y);
177 first = false;
```

SkyFiMapOverlay.java, line 205 (Poor Style: Value Never Read) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Poor Style: Value Never Read

Low

Package: com.skyfi.atak.plugin

SkyFiMapOverlay.java, line 205 (Poor Style: Value Never Read)

Low

Sink: VariableAccess: xy
Enclosing Method: drawCurrentPolygon()
File: SkyFiMapOverlay.java:205
Taint Flags:

```
202 double area = calculatePolygonArea(currentPolygon);
203 GeoPoint centroid = calculateCentroid(currentPolygon);
204 if (centroid != null) {
205 PointF xy = mapView.forward(centroid);
206 String areaText = String.format("%.2f km²", area);
207 float textWidth = areaPaint.measureText(areaText);
208 canvas.drawText(areaText, xy.x - textWidth/2, xy.y, areaPaint);
```

SkyFiMapOverlay.java, line 219 (Poor Style: Value Never Read)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: xy
Enclosing Method: drawCurrentPolygon()
File: SkyFiMapOverlay.java:219
Taint Flags:

```
216 vertexPaint.setAntiAlias(true);
217
218 for (int i = 0; i < currentPolygon.length; i++) {
219 PointF xy = mapView.forward(currentPolygon[i]);
220 canvas.drawCircle(xy.x, xy.y, 8f, vertexPaint);
221
222 // Draw vertex number
```

AORFilterManager.java, line 210 (Poor Style: Value Never Read)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: lat
Enclosing Method: extractCenterFromFootprint()
File: AORFilterManager.java:210
Taint Flags:

```
207 // This is a basic implementation that might need refinement
208 String[] parts = footprint.split("[,\\s]+");
```

Poor Style: Value Never Read — Low

Package: com.skyfi.atak.plugin

AORFilterManager.java, line 210 (Poor Style: Value Never Read) — Low

```
209  if (parts.length >= 2) {
210  double lat = Double.parseDouble(parts[0]);
211  double lon = Double.parseDouble(parts[1]);
212  return new GeoPoint(lat, lon);
213  }
```

ImageryPreviewManager.java, line 437 (Poor Style: Value Never Read) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: lon
Enclosing Method: extractCenterFromArchive()
File: ImageryPreviewManager.java:437
Taint Flags:

```
434  // Get the centroid of the geometry
435  org.locationtech.jts.geom.Point centroid = geometry.getCentroid();
436  double lat = centroid.getY();
437  double lon = centroid.getX();
438
439  return new GeoPoint(lat, lon);
440
```

ImageryPreviewManager.java, line 436 (Poor Style: Value Never Read) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: lat
Enclosing Method: extractCenterFromArchive()
File: ImageryPreviewManager.java:436
Taint Flags:

```
433
434  // Get the centroid of the geometry
435  org.locationtech.jts.geom.Point centroid = geometry.getCentroid();
436  double lat = centroid.getY();
437  double lon = centroid.getX();
438
439  return new GeoPoint(lat, lon);
```

Poor Style: Value Never Read — Low

Package: com.skyfi.atak.plugin

SkyFiPlugin.java, line 825 (Poor Style: Value Never Read) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: coveragePercent
Enclosing Method: updateDashboardMetrics()
File: SkyFiPlugin.java:825
Taint Flags:

```
822
823 private void updateDashboardMetrics() {
824 TextView satelliteCount = mainView.findViewById(R.id.satellite_count);
825 TextView coveragePercent = mainView.findViewById(R.id.coverage_percent);
826 TextView activeOrders = mainView.findViewById(R.id.active_orders);
827 TextView apiStatus = mainView.findViewById(R.id.api_status);
828
```

TaskingOrderFragment.java, line 881 (Poor Style: Value Never Read) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: p1
Enclosing Method: calculatePolygonArea()
File: TaskingOrderFragment.java:881
Taint Flags:

```
878 if (coordinates.length > 2) {
879 for (int i = 0; i < coordinates.length-1; i++) {
880 Coordinate p1, p2;
881 p1 = coordinates[i];
882 p2 = coordinates[i + 1];
883 area += Math.toRadians(p2.x - p1.x) * (2 + Math.sin(Math.toRadians(p1.y)) + Math.sin(Math.toRadians(p2.y)));
884 }
```

ArchiveListAdapter.java, line 62 (Poor Style: Value Never Read) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Poor Style: Value Never Read — Low

Package: com.skyfi.atak.plugin

ArchiveListAdapter.java, line 62 (Poor Style: Value Never Read) — Low

Sink: VariableAccess: holder
Enclosing Method: getView()
File: ArchiveListAdapter.java:62
Taint Flags:

```
59  holder.linearLayout = convertView.findViewById(R.id.linear_layout);
60  convertView.setTag(holder);
61  } else {
62  holder = (ViewHolder) convertView.getTag();
63  }
64
65  // TODO: Bind data to views when properly implemented
```

SkyFiPlugin.java, line 1118 (Poor Style: Value Never Read) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: lon
Enclosing Method: createSquareAround()
File: SkyFiPlugin.java:1118
Taint Flags:

```
1115
1116  // Calculate corner points
1117  double lat = center.getLatitude();
1118  double lon = center.getLongitude();
1119
1120  // Approximate degrees per meter
1121  double metersPerDegreeLat = 111132.92 - 559.82 * Math.cos(2 * Math.toRadians(lat));
```

Portability Flaw: Locale Dependent Comparison (3 issues)

Abstract

Unexpected portability problems can be found when the locale is not specified.

Explanation

When comparing data that may be locale-dependent, an appropriate locale should be specified. **Example 1:** The following example tries to perform validation to determine if user input includes a

Recommendation

To prevent this from occurring, always make sure to either specify the default locale, or specify the locale with APIs that accept them such as `toUpperCase()`. **Example 2:** The following specifies the locale manually as an argument to `toUpperCase()`.

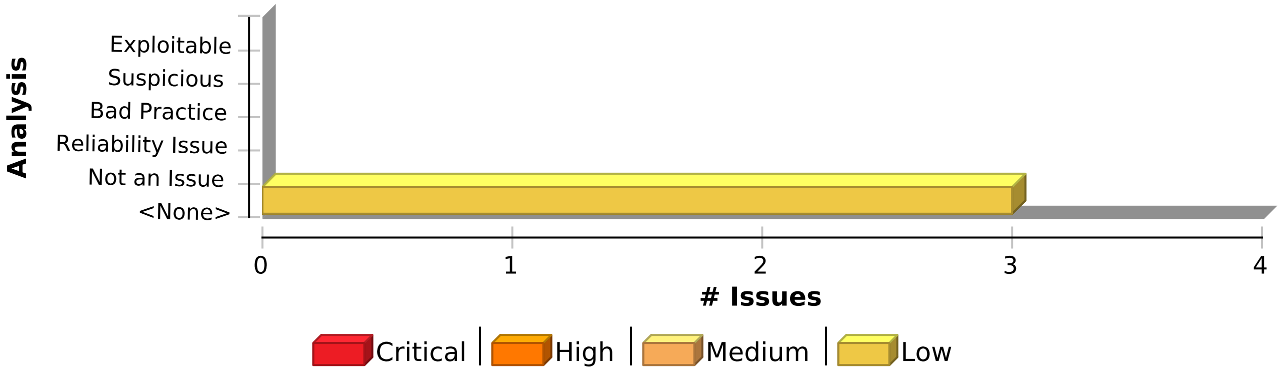
```
import java.util.Locale;
  ...
  public String tagProcessor(String tag){
    if (tag.toUpperCase(Locale.ENGLISH).equals("SCRIPT")){
      return null;
    }
    //does not contain SCRIPT tag, keep processing input
    ...
  }
  ...
```

Example 3: The following uses the function `java.lang.String.equalsIgnoreCase()` API to prevent this issue.

```
  ...
  public String tagProcessor(String tag){
    if (tag.equalsIgnoreCase("SCRIPT")){
      return null;
    }
    //does not contain SCRIPT tag, keep processing input
    ...
  }
  ...
```

This prevents the problem because `equalsIgnoreCase()` changes case similar to `Character.toLowerCase()` and `Character.toUpperCase()`. This involves creating temporary canonical forms of both strings using information from the `UnicodeData` file that is part of the Unicode Character Database maintained by the Unicode Consortium, and even though this may render them unreadable if they were to be read out, it makes comparison possible without being dependent upon locale.

Issue Summary



Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Portability Flaw: Locale Dependent Comparison	3	0	0	3
Total	**3**	**0**	**0**	**3**

Portability Flaw: Locale Dependent Comparison — Low

Package: com.skyfi.atak.plugin

SatelliteFeasibilityCalculator.java, line 74 (Portability Flaw: Locale Dependent Comparison) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Control Flow)

Sink Details

Sink: sensorType.toUpperCase().equals(...) : Comparison without checking locale
Enclosing Method: calculateSensorPasses()
File: SatelliteFeasibilityCalculator.java:74
Taint Flags:

```
71  double latitudeFactor = calculateLatitudeFactor(latitude);
72
73  switch (sensorType.toUpperCase()) {
74  case "ASAP":
75  // ASAP typically uses multiple constellation types for fastest response
76  // Higher pass frequency due to multiple satellite types
77  return (int) Math.round(daysDuration * 4.5 * latitudeFactor);
```

SatelliteFeasibilityCalculator.java, line 152 (Portability Flaw: Locale Dependent Comparison) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Control Flow)

Sink Details

Sink: sensorType.toUpperCase().equals(...) : Comparison without checking locale

Portability Flaw: Locale Dependent Comparison — Low

Package: com.skyfi.atak.plugin

SatelliteFeasibilityCalculator.java, line 152 (Portability Flaw: Locale Dependent Comparison) — Low

Enclosing Method: determineFeasibilityLevel()
File: SatelliteFeasibilityCalculator.java:152
Taint Flags:

```
149 double excellentThreshold, goodThreshold, fairThreshold;
150
151 switch (sensorType.toUpperCase()) {
152 case "ASAP":
153 excellentThreshold = 4.0;
154 goodThreshold = 2.5;
155 fairThreshold = 1.5;
```

SkyFiPolygonStyle.java, line 83 (Portability Flaw: Locale Dependent Comparison) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Control Flow)

Sink Details

Sink: status.toLowerCase().equals(...) : Comparison without checking locale
Enclosing Method: applyTaskedStyle()
File: SkyFiPolygonStyle.java:83
Taint Flags:

```
80 int fillColor;
81
82 switch (status.toLowerCase()) {
83 case "pending":
84 strokeColor = SKYFI_WARNING;
85 fillColor = adjustAlpha(SKYFI_WARNING, TASKED_FILL_ALPHA);
86 break;
```

Race Condition: Format Flaw (1 issue)

Abstract

The methods `parse()` and `format()` in `java.text.Format` contain a design flaw that can cause one user to see another user's data.

Explanation

The methods `parse()` and `format()` in `java.text.Format` contains a race condition that can cause one user to see another user's data. **Example 1:** The following code shows how this design flaw can manifest itself.

```
public class Common {

    private static SimpleDateFormat dateFormat;
    ...

    public String format(Date date) {
        return dateFormat.format(date);
    }
    ...

    final OtherClass dateFormatAccess=new OtherClass();
    ...

    public void function_running_in_thread1(){
        System.out.println("Time in thread 1 should be 12/31/69 4:00 PM,
found: "+ dateFormatAccess.format(new Date(0)));
    }

    public void function_running_in_thread2(){
        System.out.println("Time in thread 2 should be around 12/29/09 6:26
AM, found: "+ dateFormatAccess.format(new Date(System.currentTimeMillis())));
    }
}
```

While this code will behave correctly in a single-user environment, if two threads run it at the same time they could produce the following output: Time in thread 1 should be 12/31/69 4:00 PM, found: 12/31/69 4:00 PM Time in thread 2 should be around 12/29/09 6:26 AM, found: 12/31/69 4:00 PM In this case, the date from the first thread is shown in the output from the second thread due a race condition in the implementation of `format()`.

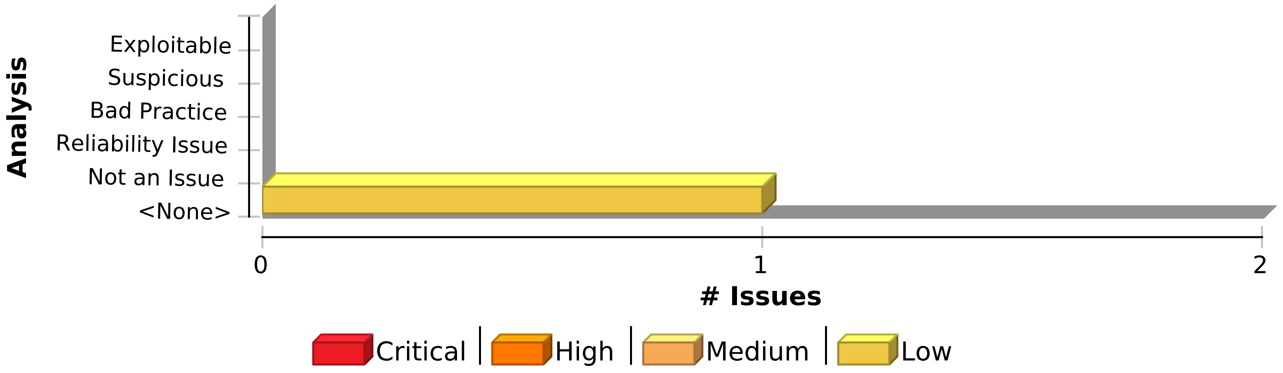
Recommendation

Use synchronization to protect against race conditions whenever `parse()` and `format()` in class `java.text.Format`. **Example 2:** The following code shows two ways that the code from `Example 1` can be corrected using synchronization constructs.

```
public class Common {

    private static SimpleDateFormat dateFormat;
    ...

    public synchronized String format1(Date date) {
        return dateFormat.format(date);
    }

    public String format2(Date date) {
        synchronized(dateFormat)
        {
            return dateFormat.format(date);
        }
    }
}
```

Alternatively, use `org.apache.commons.lang.time.FastDateFormat` class, which is a thread-safe version of `java.text.SimpleDateFormat`.

Issue Summary



Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Race Condition: Format Flaw	1	0	0	1
Total	**1**	**0**	**0**	**1**

Race Condition: Format Flaw — Low

Package: com.skyfi.atak.plugin

PreviewThumbnailAdapter.java, line 75 (Race Condition: Format Flaw) — Low

Issue Details

Kingdom: Time and State
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: format
Enclosing Method: onBindViewHolder()
File: PreviewThumbnailAdapter.java:75
Taint Flags:

```
72 // Set date text
73 try {
74 if (archive.getCaptureTimestamp() != null) {
75 holder.dateText.setText(dateFormat.format(archive.getCaptureTimestamp()));
76 } else {
77 holder.dateText.setText("N/A");
78 }
```

Unchecked Return Value (2 issues)

Abstract

Ignoring a method's return value can cause the program to overlook unexpected states and conditions.

Explanation

It is not uncommon for Java programmers to misunderstand `read()` and related methods that are part of many `java.io` classes. Most errors and unusual events in Java result in an exception being thrown. (This is one of the advantages that Java has over languages like C: Exceptions make it easier for programmers to think about what can go wrong.) But the stream and reader classes do not consider it unusual or exceptional if only a small amount of data becomes available. These classes simply add the small amount of data to the return buffer, and set the return value to the number of bytes or characters read. There is no guarantee that the amount of data returned is equal to the amount of data requested. This behavior makes it important for programmers to examine the return value from `read()` and other IO methods to ensure that they receive the amount of data they expect. **Example 1:** The following code loops through a set of users, reading a private data file for each user. The programmer assumes that the files are always exactly 1 kilobyte in size and therefore ignores the return value from `read()`. If an attacker can create a smaller file, the program will recycle the remainder of the data from the previous user and handle it as though it belongs to the attacker.

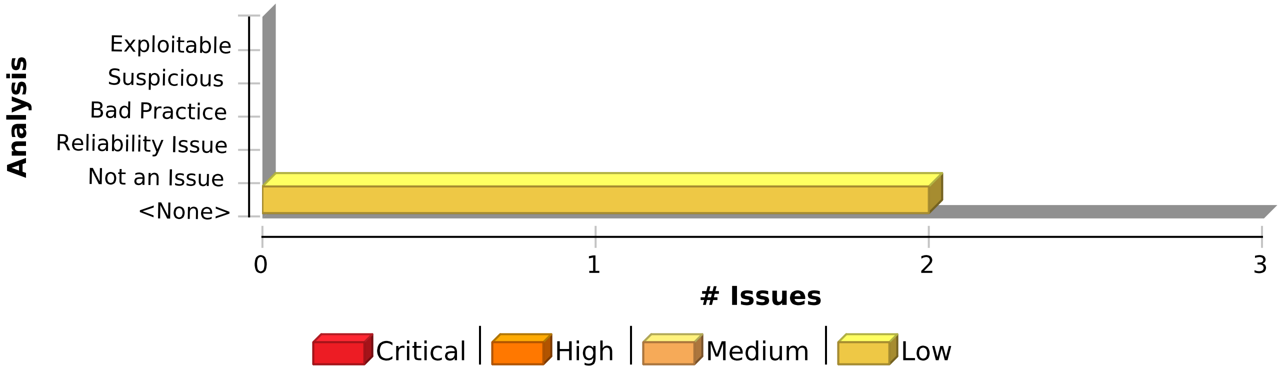
```
FileInputStream fis;
byte[] byteArray = new byte[1024];
for (Iterator i=users.iterator(); i.hasNext();) {
    String userName = (String) i.next();
    String pFileName = PFILE_ROOT + "/" + userName;
    FileInputStream fis = new FileInputStream(pFileName);
    fis.read(byteArray); // the file is always 1k bytes
    fis.close();
    processPFile(userName, byteArray);
}
```

Recommendation

```
  FileInputStream fis;
  byte[] byteArray = new byte[1024];
  for (Iterator i=users.iterator(); i.hasNext();) {
    String userName = (String) i.next();
    String pFileName = PFILE_ROOT + "/" + userName;
    fis = new FileInputStream(pFileName);
    int bRead = 0;
    while (bRead < 1024) {
        int rd = fis.read(byteArray, bRead, 1024 - bRead);
        if (rd == -1) {
          throw new IOException("file is unusually small");
        }
        bRead += rd;
    }
    // could add check to see if file is too large here
    fis.close();
    processPFile(userName, byteArray);
  }
```

Note: Because the fix for this problem is relatively complicated, you might be tempted to use a simpler approach, such as checking the size of the file before you begin reading. Such an approach would render the application vulnerable to a file system race condition, whereby an attacker could replace a well-formed file with a malicious file between the file size check and the call to read data from the file.

Issue Summary



Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Unchecked Return Value	2	0	0	2
Total	**2**	**0**	**0**	**2**

Unchecked Return Value — Low

Package: <none>

ImageCacheManager.java, line 160 (Unchecked Return Value) — Low

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Semantic)

Sink Details

Sink: delete()
Enclosing Method: run()
File: ImageCacheManager.java:160
Taint Flags:

```
157 File[] files = diskCacheDir.listFiles();
158 if (files != null) {
159 for (File file : files) {
160 file.delete();
161 }
162 }
163 });
```

Package: com.skyfi.atak.plugin

ImageCacheManager.java, line 57 (Unchecked Return Value) — Low

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Semantic)

Sink Details

Unchecked Return Value — Low

Package: com.skyfi.atak.plugin

ImageCacheManager.java, line 57 (Unchecked Return Value) — Low

Sink: mkdirs()
Enclosing Method: ImageCacheManager()
File: ImageCacheManager.java:57
Taint Flags:

```
54  // Initialize disk cache directory
55  diskCacheDir = new File(context.getCacheDir(), CACHE_DIR);
56  if (!diskCacheDir.exists()) {
57  diskCacheDir.mkdirs();
58  }
59
60  // Clean up old cache files
```

Weak Cryptographic Hash (1 issue)

Abstract

Weak cryptographic hashes cannot guarantee data integrity and should not be used in security-critical contexts.

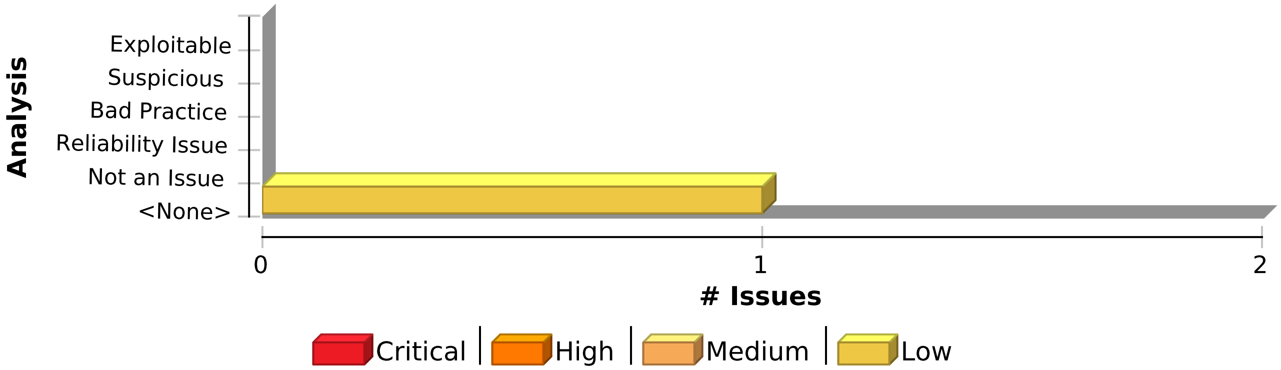
Explanation

MD2, MD4, MD5, RIPEMD-160, and SHA-1 are popular cryptographic hash algorithms often used to verify the integrity of messages and other data. However, as recent cryptanalysis research has revealed fundamental weaknesses in these algorithms, they should no longer be used within security-critical contexts. Effective techniques for breaking MD and RIPEMD hashes are widely available, so those algorithms should not be relied upon for security. In the case of SHA-1, current techniques still require a significant amount of computational power and are more difficult to implement. However, attackers have found the Achilles' heel for the algorithm, and techniques for breaking it will likely lead to the discovery of even faster attacks.

Recommendation

Discontinue the use of MD2, MD4, MD5, RIPEMD-160, and SHA-1 for data-verification in security-critical contexts. Currently, SHA-224, SHA-256, SHA-384, SHA-512, and SHA-3 are good alternatives. However, these variants of the Secure Hash Algorithm have not been scrutinized as closely as SHA-1, so be mindful of future research that might impact the security of these algorithms.

Issue Summary



Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Weak Cryptographic Hash	1	0	0	1
Total	**1**	**0**	**0**	**1**

Weak Cryptographic Hash **Low**

Package: com.skyfi.atak.plugin

ImageCacheManager.java, line 179 (Weak Cryptographic Hash) **Low**

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Semantic)

Sink Details

Weak Cryptographic Hash — Low

Package: com.skyfi.atak.plugin

ImageCacheManager.java, line 179 (Weak Cryptographic Hash) — Low

Sink: getInstance()
Enclosing Method: generateKey()
File: ImageCacheManager.java:179
Taint Flags:

```
176
177  private String generateKey(String url) {
178  try {
179  MessageDigest digest = MessageDigest.getInstance("MD5");
180  digest.update(url.getBytes());
181  byte[] bytes = digest.digest();
182  StringBuilder sb = new StringBuilder();
```

About Fortify Solutions

Fortify is the leader in end-to-end application security solutions with the flexibility of testing on-premise and on-demand to cover the entire software development lifecycle. Learn more at www.microfocus.com/solutions/application-security.