Actions Speak Louder than Words: Entity-sensitive Privacy Policy and Data Flow Analysis with PoliCheck

Benjamin Andow,¹* Samin Yaseer Mahmud,² Justin Whitaker,² William Enck,² Bradley Reaves,² Kapil Singh,³ and Serge Egelman⁴

¹Google
²North Carolina State University
³IBM T.J. Watson Research Center
⁴U.C. Berkeley / ICSI / AppCensus Inc.

* This work was completed while the lead author was at NCSU and IBM Research
Disclaimer

This work was completed while I was at North Carolina State University and IBM Research. None of this work or statements made within this presentation reflects the views of Google and should not be construed as statements from Google or from my current role as a Google employee.
Flow-to-Policy Consistency

Are these data flows disclosed within the application’s privacy policy?
Privacy Policy Excerpt:
“When you launch any of our applications, we collect information regarding your device type, operating system and version, carrier provider, IP address, Media Access Control (MAC) address, International Equipment Mobile ID (IMEI), whether you are using a point package, the game version, the device’s geo-location, language settings, and unique device ID.”
Privacy Policy Excerpt:
“When you launch any of our applications, we collect information regarding your device type, operating system and version, carrier provider, IP address, Media Access Control (MAC) address, International Equipment Mobile ID (IMEI), whether you are using a point package, the game version, the device’s geo-location, language settings, and unique device ID.”
Prior Flow-to-Policy Consistency Models

• Prior works [1, 2, 3] would incorrectly mark this data flow as consistent with its privacy policy due to entity-insensitive analysis.

Prior Flow-to-Policy Consistency Models

- Prior works [1, 2, 3] would incorrectly mark this data flow as consistent with its privacy policy due to entity-insensitive analysis.

Entity-insensitive analysis may result in:
- Reasoning over up to 55.8% of data flows with irrelevant policy statements
- Misclassifying up to 37.1% of data flows as consistent

---

Goals

1. Formally specify a flow-to-policy consistency model that is entity-sensitive, sentiment-sensitive, and contradiction-sensitive
   – Entity-sensitivity: Who is receiving the data
   – Sentiment-sensitivity [4]: Precise modeling of both positive and negative sentiment
   – Contradiction-sensitivity [4]: Holistic analysis of the policy
2. Large-scale empirical study to characterize the space of privacy policy disclosures and violations

PoliCheck

- Entity, sentiment, and contradiction-sensitive data flow to privacy policy consistency model
- Built on top of PolicyLint and AppCensus
  - Heuristics to resolve domains to entities and classify first party flows
  - Extends PolicyLint’s ontologies to ensure coverage over entities/data types

AppCensus: [https://search.appcensus.io/](https://search.appcensus.io/)

See our paper for complete design details.
Privacy Policy Disclosures

- **Goal:** Characterize how data flows are disclosed (or not disclosed) by privacy policies

- **Intuition:**
  - Consistent does not always mean transparent:
    - Explicitly discloses the data flow in exact terms (**Clear**)
    - Discloses the data flow using broad terms (**Vague**)
  - Several ways for an inconsistency to occur:
    - Does not disclose the data flow (**Omitted**)
    - States that the data flow does not occur (**Incorrect**)
    - States that the data flow both does and does not occur (**Ambiguous**)

See paper for our formalization of each disclosure type
Consistent: Clear Disclosure

- A *clear disclosure* discusses that the flow occurs in exact terms

Privacy Policy Excerpt: *Unity collects* the following information through our Games: unique device ID and AD ID.

A clear disclosure of a data flow $f$ occurs iff:
- A policy statement discusses that the flow exists in exact terms for the data type and entity
- No policy statement discusses that the flow does not occur

- $\exists p \in P_f \text{ s.t. } p.c = \text{collect} \land f.d \equiv_{\delta} p.d \land f.e \equiv_{\varepsilon} p.e$ and $\nexists p' \in P_f \text{ s.t. } p'.c = \text{not}_\text{collect}$
Consistent: Vague Disclosure

- A **vague disclosure** discusses that the data flow occurs using broad terms for the entity, data type, or both.

A vague disclosure of a data flow $f$ occurs iff:

- No clear disclosure exists
- A policy statement discusses that the flow occurs in broad terms for the data type or entity
- No policy statement discusses that the flow does not occur

$\forall p \in P_f \text{ s.t. } p.c = \text{collect} \land f.d \equiv_\delta p.d \land f.e \equiv_\varepsilon p.e$ and $\exists p' \in P_f \text{ s.t. } p'.c = \text{collect} \land f.d \sqsubseteq_\delta p'.d \land f.e \sqsubseteq_\varepsilon p'.e$ and $\nexists p'' \in P_f \text{ s.t. } p''.c = \text{not}_\text{collect}$

---

**Privacy Policy Excerpt:**

A device identifier and in-game or user session activity may be **shared** with the advertiser.
Privacy Policy Disclosures

- **Goal:** Characterize how data flows are disclosed (or not disclosed) by privacy policies

- **Intuition:**
  - Consistent does not always mean transparent:
    - Explicitly discloses the data flow in exact terms (*Clear*)
    - Discloses the data flow using broad terms (*Vague*)
  - **Several ways for an inconsistency to occur:**
    - Does not disclose the data flow (*Omitted*)
    - States that the data flow does not occur (*Incorrect*)
    - States that the data flow both does and does not occur (*Ambiguous*)

See paper for our formalization of each disclosure type
Inconsistent: Omitted Disclosure

• An omitted disclosure does not discuss the data flow in the policy

Privacy Policy Excerpt:
When you access our Services, we automatically record and upload information from your device including, but not limited to attributes such as the operating system, hardware version, device settings, battery and signal strength, device identifiers...”

An omitted disclosure of a data flow $f$ occurs iff:

- No policy statements discuss the data flow
  - $P_f = \emptyset$
An incorrect disclosure of a data flow \( f \) occurs iff:

- All policy statements discuss that the flow does not occur
- Or a narrowing definitions exist, but not a contradictory statement

\[ \forall p \in P_f, p.c = \text{not\_collect} \text{ or } (P_f \cap P_N \neq \emptyset \land P_f \cap P_C = \emptyset) \]
An ambiguous disclosure occurs for a data flow $f$ iff:

- The policy contains contradictory policy statements
  - $P_f \cap P_c \neq \emptyset$

Privacy Policy Excerpt:

On our apps, these third party advertising companies will collect and use your data to provide you with targeted advertising…

…

We don’t give or sell your data to third parties for them to market to you.
Empirical Study

• **Dataset:** 13,796 Android applications with 45,603 data flows
  - Selected top 100 free Google Play apps across 35 categories (3,500 apps)
  - Randomly selected 42,129 apps from AppCensus
    - Exclude apps with no data flows reported by AppCensus (23,488 apps)
    - Exclude apps with no privacy policy, no reachable policy, and not written in English (6,039 apps)

• **Research Questions:**
  1. How are apps disclosing their client-side third-party data flows in their privacy policies?
  2. How does entity-sensitive analysis impact the soundness of flow-to-policy consistency?
Flow-to-Policy Consistency Results

- 43.5% of flows were classified as omitted, incorrect, or ambiguous disclosures
  - Some apps had over 15 omitted or incorrect disclosures!

<table>
<thead>
<tr>
<th></th>
<th>Clear</th>
<th>Vague</th>
<th>Omit</th>
<th>Incorrect</th>
<th>Ambig</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>First Party</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flows</td>
<td>216</td>
<td>2,211</td>
<td>208</td>
<td>18</td>
<td>358</td>
</tr>
<tr>
<td>Apps</td>
<td>206</td>
<td>1,600</td>
<td>154</td>
<td>11</td>
<td>224</td>
</tr>
<tr>
<td><strong>Third Party</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flows</td>
<td>7</td>
<td>23,367</td>
<td>14,201</td>
<td>1,912</td>
<td>3,105</td>
</tr>
<tr>
<td>Apps</td>
<td>7</td>
<td>6,833</td>
<td>5,076</td>
<td>708</td>
<td>892</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flows</td>
<td>223</td>
<td>25,578</td>
<td>14,409</td>
<td>1,930</td>
<td>3,463</td>
</tr>
<tr>
<td>Apps</td>
<td>213</td>
<td>7,626</td>
<td>5,155</td>
<td>719</td>
<td>1,101</td>
</tr>
</tbody>
</table>
Highlighted Finding

- **49.5% of applications are disclosing their third party sharing practices using vague terms**

- Flows to third parties involving Ad IDs and Android IDs were disclosed by:
  - (3rd party, collect, PII): 40.7% of flows
  - (3rd party, collect, info): 25.2% of flows
Highlighted Finding

- **719 applications make incorrect statements about their data practices.**
  - Ad IDs and Android IDs most common data type involved in incorrect disclosures
    - 15.7% of flows involve to Crashlytics
    - 13.7% of flows involve to Unity
    - 9.6% of flows involve Flurry
  - (third party, not_collect, PII) accounted for 63.4% of these disclosures.
8.0% of apps have ambiguous privacy policies

- Android IDs and Ad IDs are involved in 88.8% (3,074 / 3,494) of ambiguous disclosures.
- C₁ contradictions involving PII are most common where policy states that a specific entity both collects and does not collect PII.

**Highlighted Finding**

**Privacy Policy Excerpt:**

We DO NOT collect your unique identifier [sic].

... Anonymous identifiers, we use anonymous identifiers when you interact with services, such as advertising services and others.
Highlighted Finding

• **31.1% of data flows were classified as omitted disclosures***
  – Only 6.9% (208) of first party flows were omitted
  – Crashlytics and Unity receiving Android IDs and Ad IDs accounted for 27.8% (3,168) and 24.7% of third party omitted disclosures, respectively.
• Raises questions on developers’ understandings of third party SDKs and their responsibility for disclosing privacy practices

*Note: PoliCheck had lower precision for classifying omitted disclosures, but the paper discusses that this may be indicative of confusing policy language
Thank You!

*PoliCheck provides a flow-to-policy consistency model that is entity-sensitive, sentiment-sensitive, and contradiction-sensitive.*

Benjamin Andow
Google*

Code Available on Github: [https://github.com/benandow/PrivacyPolicyAnalysis](https://github.com/benandow/PrivacyPolicyAnalysis)

*This work was completed while the lead author was at NCSU and IBM Research*
**Evaluation**

- **Dataset:** 153 data flows across 151 applications
  - POLICHECK achieves an overall **90.8%** overall precision (139/153) for performing flow-to-policy consistency analysis.
    - 95.6% precision for identifying clear and vague disclosures
    - 92.1% precision for identifying incorrect disclosures
    - 72.0% precision for identifying omitted disclosures
  - Primarily due to incomplete policy statement extraction
  - Policy statements distributed across multiple sentences or sections
  - Took much more effort to validate FPs for omitted disclosures, as many were difficult to comprehend

<table>
<thead>
<tr>
<th></th>
<th>Clear</th>
<th>Vague</th>
<th>Incorrect</th>
<th>Omitted</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP</td>
<td>30</td>
<td>56</td>
<td>35</td>
<td>18</td>
</tr>
<tr>
<td>FP</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>Precision</td>
<td>100%</td>
<td>93.3%</td>
<td>92.1%</td>
<td>72.0%</td>
</tr>
</tbody>
</table>