Week 3 Static Analysis Report

1. Introduction & Methodology

This report presents a static analysis of ten published Alexa skills to compare their declared permissions against what their core functionality actually requires. By identifying cases of permission creep or overbroad data requests, we flag potential privacy and security concerns.

Methodology Steps: 1. Collected each skill's manifest.json (declared `alexa::…` permissions). 2. Defined expected permissions based on documented functionality (voice-only interaction, account linking, etc.). 3. Compared declared vs. expected; flagged inconsistencies.

2. Summary Table

Skill	Declared Permissions	Expected Permissions	Flags
Amazon Shopping	`shopping:list`	`shopping:list`	None
Philips Hue	`smartHome:devices:readWrite`	`smartHome:devices:readWrite`	None
Uber	`device:address:full`	`device:address:full`	None
Fitbit	`health:fitness:read`	Account Linking Only	Overbroad PII request
Capital One	`payments:read`	Account Linking Only	Overbroad financial data
Jeopardy!	None	None	None
Order Pizza	`device:address:full`	Manual Address Entry	Unnecessary location access
Ask My Buddy	`notifications:push`	`notifications:push`	None
Sleep Sounds	None	None	None
Tinder	`profile:read`	Account Linking Only	Overbroad profile data

3. Individual Skill Analyses

Amazon Shopping

- **Manifest**: Requests `shopping:list` to read user's shopping list.

- **Functionality**: Users add/view items; list access is needed and scoped appropriately.
- **Assessment**: No permission creep.

Philips Hue

- **Manifest**: `smartHome:devices:readWrite` to control lights.
- **Functionality**: Turn devices on/off and adjust settings.
- **Assessment**: Permissions align with functionality.

Uber

- **Manifest**: `device:address:full` for pickup suggestions.
- **Functionality**: Provide ride estimates based on device address.
- **Assessment**: Legitimate use of full address permission.

Fitbit

- **Manifest**: `health:fitness:read` to fetch activity data.
- **Functionality**: Summarize steps, heart rate.
- **Expected**: Only account linking; health data should be protected via OAuth scopes outside Alexa.
- **Flag**: Excessive PII request within skill manifest.

Capital One

- **Manifest**: `payments:read` to retrieve transaction history.
- **Functionality**: Report balances and recent transactions.
- **Expected**: OAuth-based account linking to bank's APIs; manifest-level read is overbroad.
- **Flag**: Overbroad financial data declared.

Jeopardy!

- **Manifest**: No permissions.
- **Functionality**: Trivia Q&A.
- **Assessment**: No data requests needed; manifest correctly minimal.

Order Pizza

- **Manifest**: `device:address:full` for delivery location.
- **Functionality**: Take orders; address could be entered by voice.
- **Expected**: Manual entry; full address permission not strictly required.
- **Flag**: Unnecessary location permission.

Ask My Buddy

- **Manifest**: `notifications:push` for alerts to emergency contacts.
- **Functionality**: Send notifications to buddy list.
- **Assessment**: Permission appropriate.

Sleep Sounds

- **Manifest**: No permissions.
- **Functionality**: Play white noise sounds.
- **Assessment**: No data required; manifest minimal.

Tinder

- **Manifest**: `profile:read` to fetch basic user info.
- **Functionality**: Provide matches, bio snippets.
- **Expected**: OAuth-based account linkage; manifest scope is overbroad.
- **Flag**: Excessive profile data requested.

---

4. Overall Observations & Recommendations

- **Common Good Practices:** Skills with purely informational or media playback needs (e.g., Jeopardy!, Sleep Sounds) correctly declare minimal or no permissions.
- **Permission Creep:** Consumer-focused skills (e.g., Fitbit, Capital One, Tinder) conflate Alexa manifest scopes with OAuth scopes, leading to overbroad PII/financial data requests.
- **UX vs. Privacy Tradeoff:** Skills like Order Pizza and Uber request address access; only Uber’s use is defensible given real-time pickup; pizza ordering could rely on voice-entered address to reduce risk.
- **Recommendations:**
    1. **Least Privilege:** Limit manifest scopes to only what Alexa absolutely needs; defer sensitive data retrieval to back-end OAuth flows.
    2. **User Transparency:** Clearly disclose in privacy policies why each permission is requested.
    3. **Review & Audit:** Regularly audit skill manifests against functionality changes to catch emergent permission creep.
