Spying through your Voice Assistants: Realistic Voice Command Fingerprinting

USENIX Security ‘23

Dilawer Ahmed, Aafaq Sabir, Anupam Das
What is voice command fingerprinting?

• Identify the **activity** being performed on the voice assistant

• Privacy attack which can result in sensitive information leakage

• Using **passively** sniffed **encrypted** network traffic
How a typical voice assistant works

Alexa, turn the light on

Okay
Voice Command Fingerprinting

Alexa, turn the light on
Okay

turn the light on
Non-local adversary challenges

• No effortless way to tell when voice assistant was used
• Due to NAT, traffic from devices isn’t easily distinguishable
• We split our attack into Invocation Detection and Activity Detection
• We introduce Traffic Flow Filtering to filter noisy traffic
Invocation Detection

• To detect ‘activation’ (invocation) of voice assistants
• Desired properties:
  – Continuous real-time detection
  – Low or no false positives
  – Lightweight
Activity Detection

• To detect the actual ‘activity’ performed on device after ‘invocation’
• Desired Properties:
  – Performance (Accuracy)
  – Captures varied commands
  – Resistant to noise
Invocation Detection: Windows

Network traffic (bytes) vs. time (s)

- Domain 1
- Domain 2

- invocation
- window_size (4s)
Spikes due to Invocation

- **Alexa**
  - avs-alexa-4-na.amazon.com (443, TCP)
  - unagi-na.amazon.com (443, TCP)
- **Siri**
  - dejavu.apple.com (443, TCP)
- **Google Assistant**
  - www.google.com (443, UDP)
Invocation Detection: All 3 Platforms

<table>
<thead>
<tr>
<th>Voice Assistant</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alexa</td>
<td>99.81</td>
<td>99.63</td>
<td>100.0</td>
</tr>
<tr>
<td>Google Assistant</td>
<td>99.70</td>
<td>99.71</td>
<td>99.71</td>
</tr>
<tr>
<td>Siri</td>
<td>99.50</td>
<td>99.71</td>
<td>99.32</td>
</tr>
</tbody>
</table>

- Compared Multiple lightweight ML models
- Random Forest was selected based on overall performance
Activity Detection: Window

- **Window time (s):**
- **Network traffic (bytes):**

**Domain 1**
- **Domain 2**

**invocation**

**window_size (1 minute)**
Flow Filtering Method

- Predefined fixed flows are always included
- ‘m’ second window for inclusion of flows
Activity Detection: Across platforms

<table>
<thead>
<tr>
<th>Type</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th># Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alexa</td>
<td>87.70</td>
<td>87.46</td>
<td>88.20</td>
<td>50</td>
</tr>
<tr>
<td>Google Assistant</td>
<td>92.67</td>
<td>92.66</td>
<td>92.96</td>
<td>50</td>
</tr>
<tr>
<td>Siri</td>
<td>92.80</td>
<td>92.91</td>
<td>93.18</td>
<td>50</td>
</tr>
</tbody>
</table>

- Slightly better results for Google Assistant and Siri
- Overall good performance for all platforms
Activity Detection: Command types

<table>
<thead>
<tr>
<th>Type</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th># Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
<td>80.30</td>
<td>80.58</td>
<td>81.44</td>
<td>100</td>
</tr>
<tr>
<td>Skills</td>
<td>82.76</td>
<td>85.33</td>
<td>82.42</td>
<td>100</td>
</tr>
<tr>
<td>Stream</td>
<td>99.39</td>
<td>99.30</td>
<td>99.34</td>
<td>15</td>
</tr>
</tbody>
</table>

- Very high accuracy for streaming commands
- Performance suffers slightly for simple commands and skills
  - Similar commands
  - Same vendor/functionality skills
Design: **End-to-End Classification**

- Network Traffic
- Invocation Detection (Traffic 4s)
- Flow Filtering
- Activity Detection
- Open-World Detection (Optional)
- Output Label

Next command

Traffic (4s) → Invocation detected → Flow Filtering → Activity Detection → Output Label

- No Invocation (Go back)
- Unknown (Optional) → Known

Traffic (1 minute)
End-to-End: Real-world test

- Conducted an IRB-approved study across 5 days with 15 participants using Alexa
- Used pre-trained Invocation Detection and Activity Detection models
- Realistic background noise added by participants using laptop, TV and smart phone.
End-to-End: Evaluation

• All invocations were correctly detected without any false positives (100% accuracy)
• 91% precision and 92% recall in distinguishing novel unknown voice commands
• 77% End-to-End accuracy in detecting voice commands
Limitations

• Unable to fingerprint two same-platform voice assistants if active within ‘1-minute’ of each other
• Only focus on ~100 command set at one time
• Domains for Invocation Detection are region-specific and need manual work for selection in other regions (e.g., unagi-na.amazon.com)
Contributions

- Focused on top 3 most popular platforms
- Introduced Flow Filtering and Invocation Detection
- Improved state-of-the-art in voice command fingerprinting
- Used multiple types of commands e.g., skills, streaming
- Designed an End-to-End fingerprinting method
- Code and data is open-sourced

Dilawer Ahmed (dahmed2@ncsu.edu)
https://github.com/dilawer11/va-fingerprinting