CS6290 Privacy-enhancing Technologies
Tutorial 7: Unmasking Your Digital Fingerprint

This note guides you through the hands-on exercises related to browser fingerprinting. We'll explore how fingerprinting works, test its effectiveness, and discuss mitigation strategies.

Prerequisites:

- A web browser (Chrome, Firefox, etc.)
- Stable internet connection.

1. Understanding Browser Fingerprinting

Classic Tracking vs. Fingerprinting: Traditional online tracking relies on identifiers stored on your device (like cookies). Fingerprinting, however, identifies you based on the **unique characteristics** of your browser and system configuration.
Semi-Identifiers: Fingerprinting combines many "semi-identifiers" pieces of information that aren't necessarily unique on their own, but become unique in combination. Examples include:

- Browser size
- Installed fonts (especially non-system fonts)
- Audio/Video hardware details
- Installed plugins
- Color depth
- User Agent string
- Canvas/WebGL rendering characteristics
- Hardware identifiers (CPU, GPU, memory)
- Screen height/width
- Time Zone
- Language Preferences
- Do Not Track (DNT) Status

Visualizing Uniqueness: Each attribute contributes to the uniqueness of the fingerprint.
Font Fingerprinting: Websites can detect installed fonts using JavaScript and CSS. The presence of non-standard ("local") fonts significantly increases uniqueness.
Canvas/WebGL Fingerprinting: These techniques use your graphics hardware to generate a unique identifier. Even subtle differences in how your GPU renders a hidden image can distinguish you.
Hardware Identifiers: Various Web APIs (HTML, Web Audio, WebGL, Device Memory, WebRTC) can leak information about your hardware, further contributing to your fingerprint.
Height/Width: Even something as simple as your browser window's dimensions can be used as part of a fingerprint.

2. Exercise 1: Exploring FingerprintJS2

Goal: To understand the **breadth** of fingerprinting techniques by examining a real-world fingerprinting library.
Task:

1. Visit the GitHub repository for FingerprintJS2: https://github.com/LukasDrgon/fingerprintjs2/blob/master/fingerprint2.js
2. Read through the JavaScript code (you don't need to understand every line, but try to get a general sense).
3. **List as many different fingerprinting techniques as you can find.** Look for different `navigator` properties being accessed, different APIs being used, etc.
4. **Try to understand *how* each technique is carried out.** What information is being collected, and how is it being used?
5. **Based on your understanding, predict which techniques you think are *most* identifying.** Which ones contribute the most to uniqueness?

Key Concepts: This exercise reinforces the idea that fingerprinting isn't just about one or two pieces of information. It's about a **wide range** of attributes.

3. Exercise 2: User Agent Spoofing and its Limitations

Goal: To understand the difference between changing the `User-Agent` HTTP header and modifying JavaScript attributes, and to see the limitations of simple User Agent spoofing.
Task:

1. Follow the instructions in this guide to **temporarily** change your User Agent in Chrome: https://www.browserstack.com/guide/change-user-agent-in-chrome

- **Important:** This method only changes the `User-Agent` HTTP header. It does not modify the values reported by JavaScript running within a webpage.

2. After changing your User Agent, visit AmIUnique: https://amiunique.org/fingerprint
3. Click "View my browser fingerprint" and examine the results.
4. **Answer this question:** Did changing the User Agent in Chrome's Developer Tools significantly change your fingerprint uniqueness? Did AmIUnique still correctly identify your operating system?

Key Concepts:

- **HTTP Header vs. JavaScript:** The `User-Agent` header is sent to the server *before* the page loads. Fingerprinting scripts use JavaScript running within the page to access much more information.
- **Limitations of Spoofing:** Simple User Agent spoofing is easily detected by sophisticated fingerprinting techniques.

4. Exercise 3: Fingerprinting Countermeasures

Goal:: To think critically about ways to mitigate fingerprinting and the challenges involved.
Background::

- **Remove the functionality:** Disable features that provide fingerprinting data (e.g. disable certain JS end point; remove the HTTP header; removing runtime capability)
- **Make functionality consistent:** All the browsers return the same values.
- **Restrict access:** Like permission prompts, user gesture.
- **Noise:** The technique used in differential privacy.
- **"Privacy Budget":** Allow some level of fingerprinting, up to a certain "budget," then take action.

Task:

1. **Choose two fingerprinting vectors (techniques) from FingerprintJS2 (Exercise 1) that you think are particularly effective at identifying users.**
2. **For each vector, propose one or more countermeasures that a browser could implement to reduce its effectiveness.** Think about the approaches listed above.
3. **Now, choose two fingerprinting vectors that you think would be very difficult to defend against.**

4. **Explain why countermeasures against these vectors would be hard to implement.** Consider technical challenges, usability impacts, and the potential for websites to circumvent the countermeasures.

5. **"Attacker/Defender" Game:** Imagine you're an attacker trying to fingerprint users. How would you respond to the defenses you proposed in step 2? Then, imagine you're a defender. How would you modify your defenses in response to the attacker's counter-moves?

Key Concepts:

- **No Silver Bullet:** There's no single, perfect solution to fingerprinting.
- **Trade-offs:** Many countermeasures involve trade-offs between privacy and usability or website functionality.
- **Arms Race:** Fingerprinting and anti-fingerprinting techniques are in a constant "arms race," with each side trying to outsmart the other.

5. Hands-On Activities - Visualization

We'll use two websites to analyze your browser fingerprint:

- **AmIUnique:** https://amiunique.org
- **Device Info:** https://www.deviceinfo.me (Optional, but recommended for further exploration)

Instructions:

1. **Open your web browser (Chrome, Firefox, Edge, etc.).**
2. **Go to AmIUnique (https://amiunique.org).**
3. **Click the button that says "View my browser fingerprint" (or similar wording).**
4. **Wait for the analysis to complete (this may take a few seconds).**
5. **Examine the Results:**
    - **Uniqueness Score:** This is the key metric. A higher score means your fingerprint is more unique and easier to track.
    - **Contributing Attributes:** Scroll down and look at the list of attributes that make up your fingerprint.
    - **Canvas Fingerprinting:** Look for a section related to Canvas fingerprinting.
6. (Optional): Visit Device Info (https://www.deviceinfo.me) and explore its fingerprinting analysis.

Questions to Consider:

- What was your uniqueness score on AmIUnique? Was it higher or lower than you expected?
- Which attributes seemed to contribute the most to your uniqueness?
- Were you surprised by any of the information that was collected?

6. Further Exploration

- Explore the extensions to help mitigate the fingerprinting.
- Try other tools introduced in the class to see your fingerprints.
- **Electronic Frontier Foundation (EFF):** https://www.eff.org - A great resource for information on digital privacy and security.
- **Mozilla:** https://www.mozilla.org - Learn more about Firefox and its privacy features.
- **PrivacyTests.org:** https://privacytests.org/ - Open-source tests of web browser privacy. This is an excellent resource to see how different browsers perform on various privacy tests, including fingerprinting resistance.

7. Mitigation Strategies

- **Privacy-Focused Browsers:** Brave; Firefox; Tor Browser.
- **Browser Extensions:** Privacy Badger (EFF); uBlock Origin; CanvasBlocker/Canvas Defender (Firefox); NoScript (Firefox - Advanced).
- **General Privacy Practices:** Clear your cookies regularly; Use strong, unique passwords; Be cautious about what information you share online; Use a VPN (Virtual Private Network) to mask your IP address.

Keep building and exploring the world of privacy-enhancing technologies!