With the evolution of web pages being given more elaborate APIs, their opportunity to fingerprint visitors has increased. As the web pages normally use the programming language JavaScript to perform the fingerprinting, the ability to track the propagation of information to help protect a user is crucial. In this talk, I will present an approach to use *Information-Flow Control (IFC)* to detect and prevent browser fingerprinting.

To help detect fingerprinting, we leverage JSFlow, an IFC aware interpreter for JavaScript, to handle web pages. JSFlow is then added via a shim to the Brave browser, to allow analysis of all executed JavaScript code when visiting a web page.

We analyze known fingerprinting APIs to figure out which DOM API functions they call when fingerprinting. Based on the called DOM API functions, the goal for our setup is to label the computations with the name of the DOM API function, and flag web pages to be fingerprinting if an accumulated label consists of too many specific functions.

This modified browser should then be used to crawl the Alexa top 10,000 to investigate how prevalent browser fingerprinting is. This is currently work in progress.