



WASHINGTON STATE
UNIVERSITY



WASHINGTON STATE
UNIVERSITY

Web Browser Fingerprinting: Revealing Stealth Tracking Techniques and Defenses

Xu Lin

Washington State University

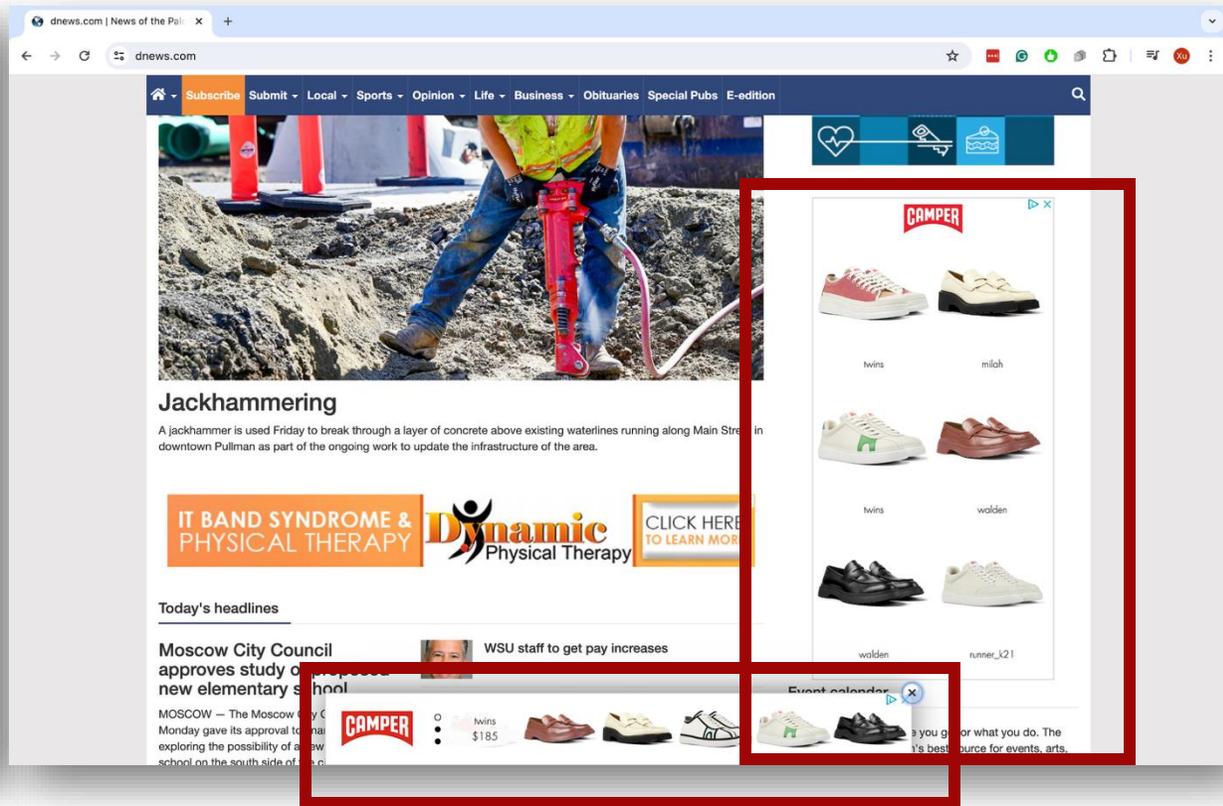
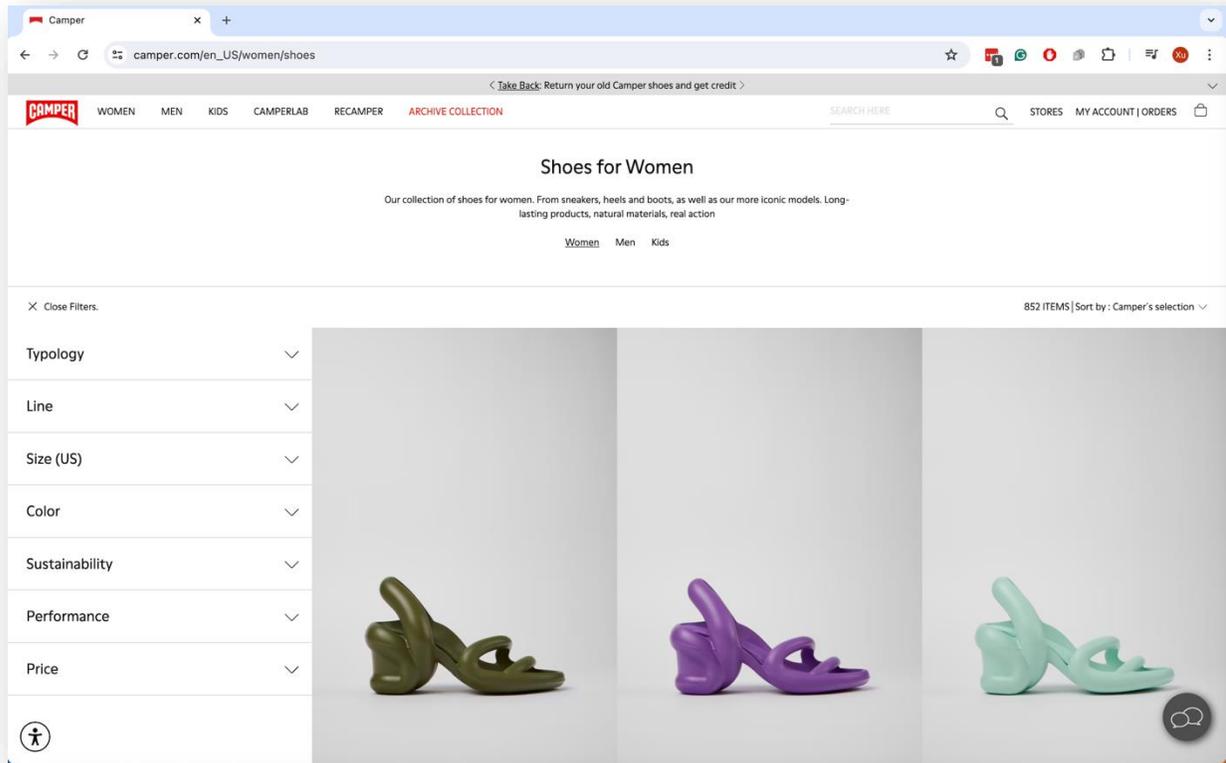
CySER May 21, 2025

What is Web Tracking?





Why Does Tracking Exist?

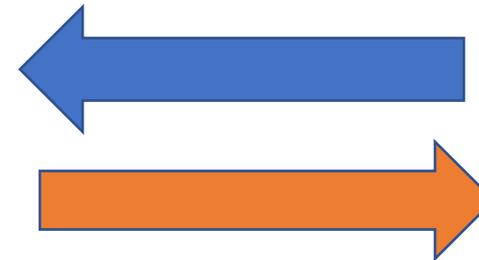
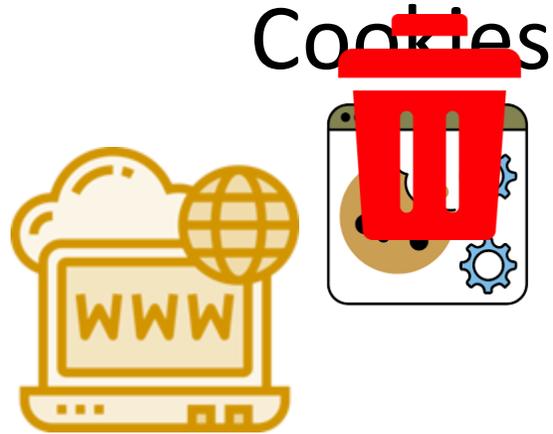




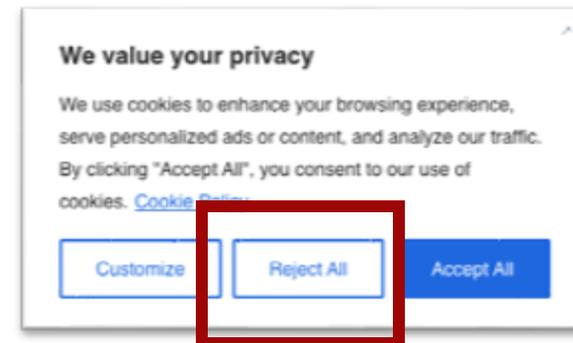
Classic Tracking



Cookies



- Website stores an id on the client
- The client returns the id to the server
- The id is what allows re-identification
- “Stateful”



Do websites still know who you are?





What Is Browser Fingerprinting

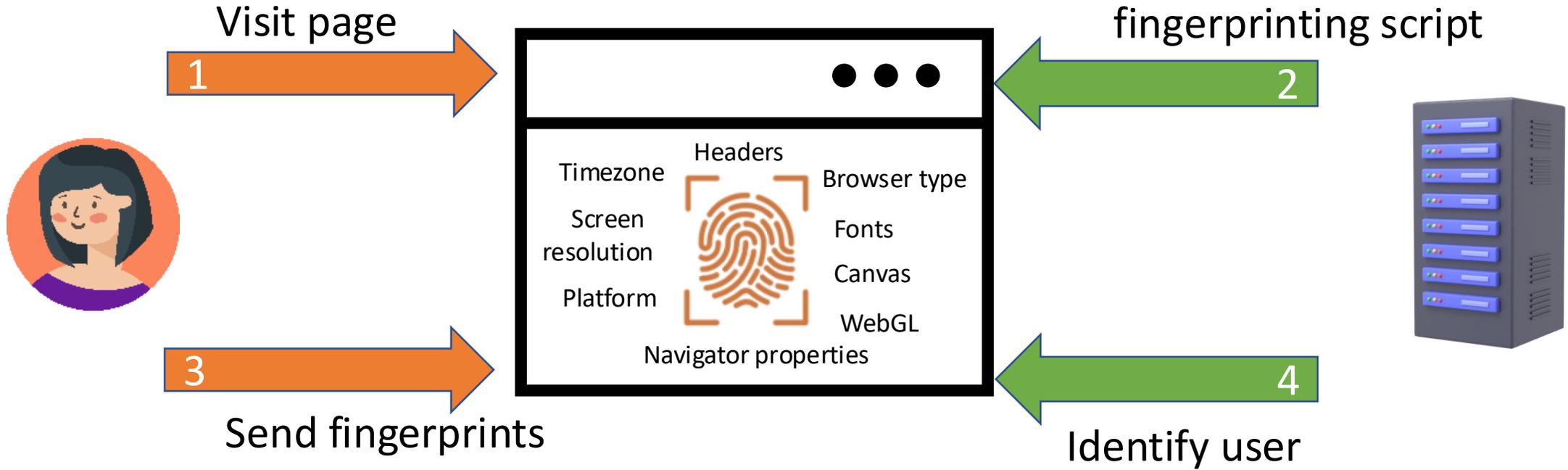


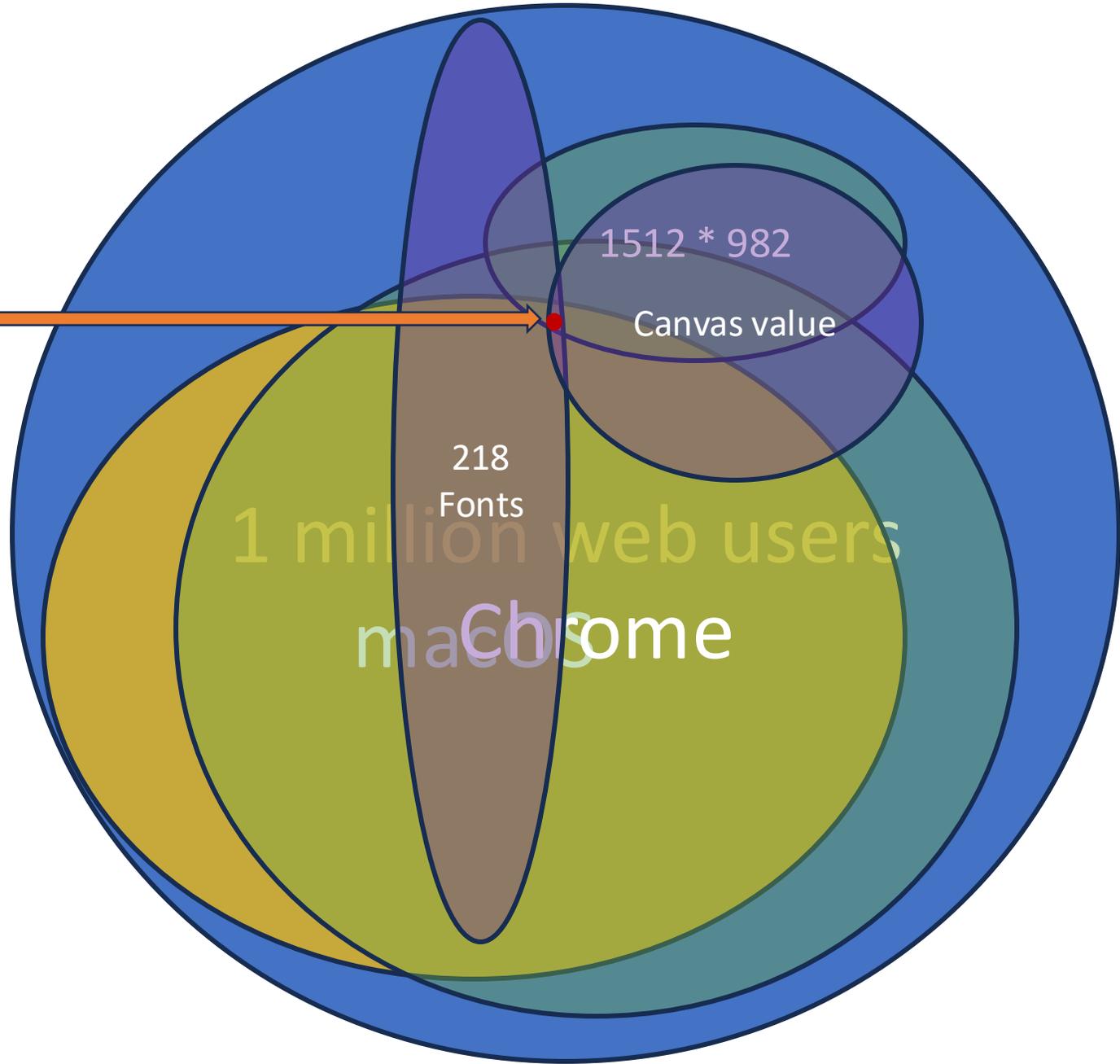
*“A device fingerprint, machine fingerprint, or browser fingerprint is information collected about a remote computing device for the purpose of identification. Fingerprints can be used to fully or partially **identify individual users or devices** even when cookies are turned off.”*

- Browser fingerprints can be trivially collected by **any** website the user visits through a series of **JavaScript APIs**.



Page with fingerprinting script







Browser fingerprinting is a double-edged sword

Usage scenarios

- **Security Enhancement**

- **Bot Detection:** Differentiate between bots and legitimate web users
- **Authentication:** Differentiate between a legitimate owner of an account and someone impersonating that owner

- **Privacy Concerns**

- **Track users** against their will (stateless tracking)

Risk-Based Authentication and Two-Factor Authentication (2FA)



WASHINGTON STATE
UNIVERSITY

Sign In

Username

Password

Remember this device [Forgot username or password?](#)

[Next](#)

Don't have an account? [Create a new one](#)

Check Your Mobile Or Email

We need to verify this Sign-In attempt. We've texted you a code as well as emailed you the same code.

Enter 6 digit code sent to: ***-***-1234

[Resend Code](#) [Update Mobile Number](#)

[Verify and Sign In](#)

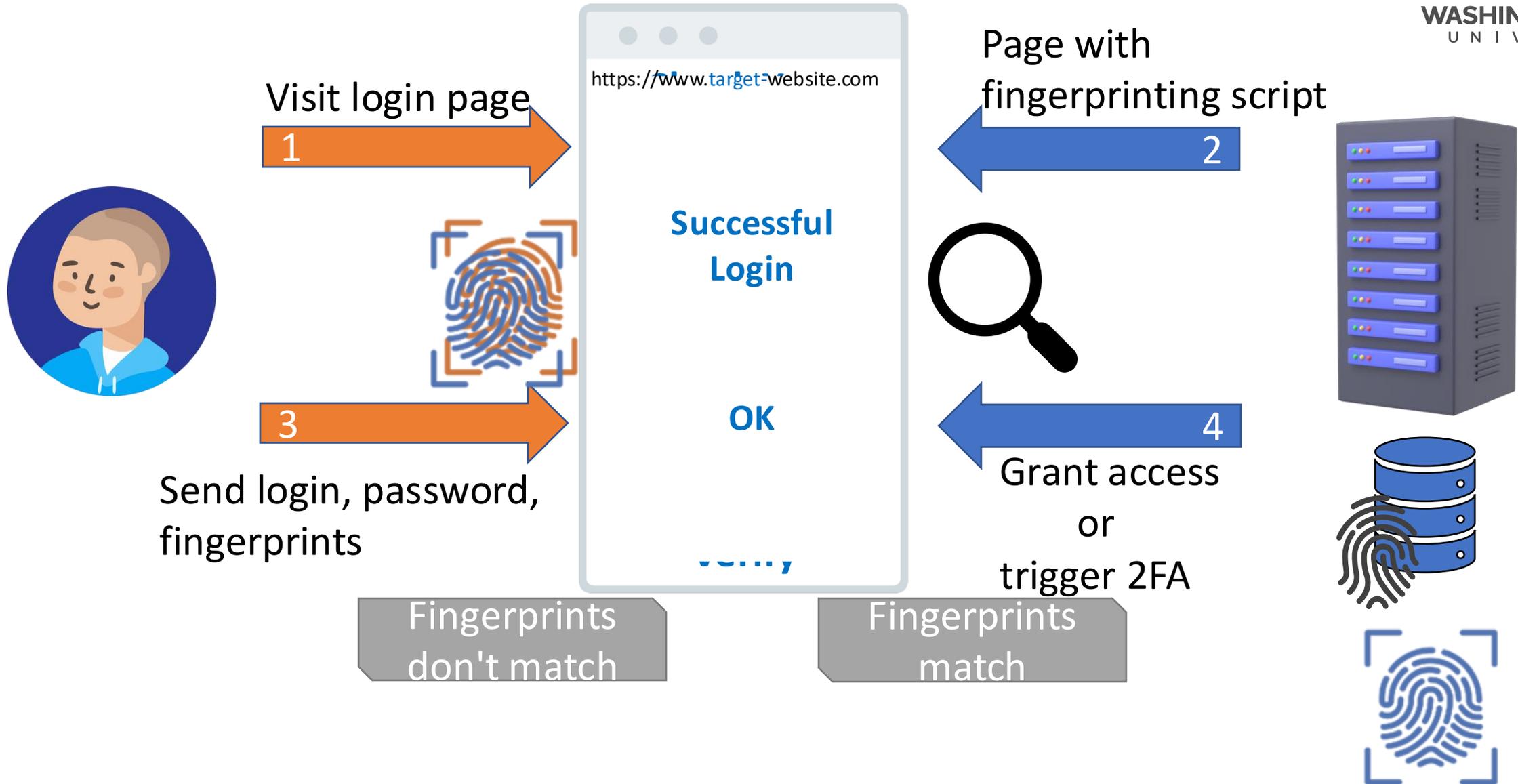
Didn't receive your verification code?
You can [Recover Your Account](#) or [Create New Account](#).

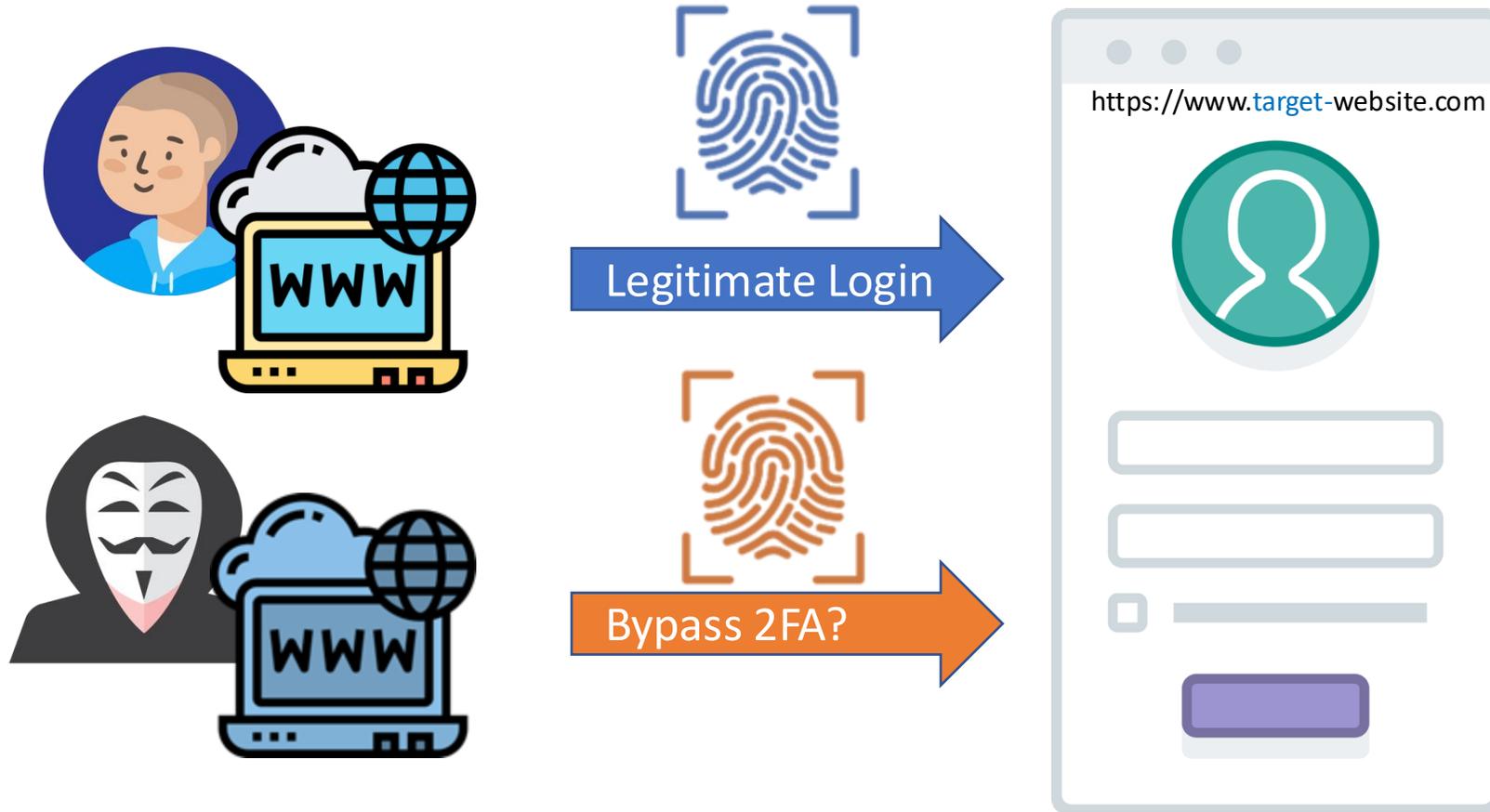
- 2FA creates friction for users
- Certain websites only trigger 2FA for *suspicious* login attempts

Advanced Risk-Based Authentication That Uses Browser Fingerprinting



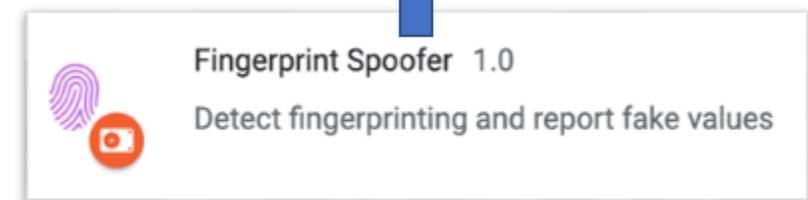
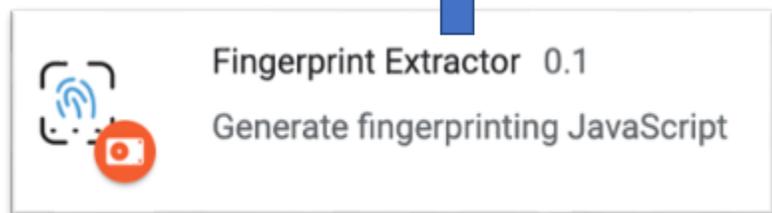
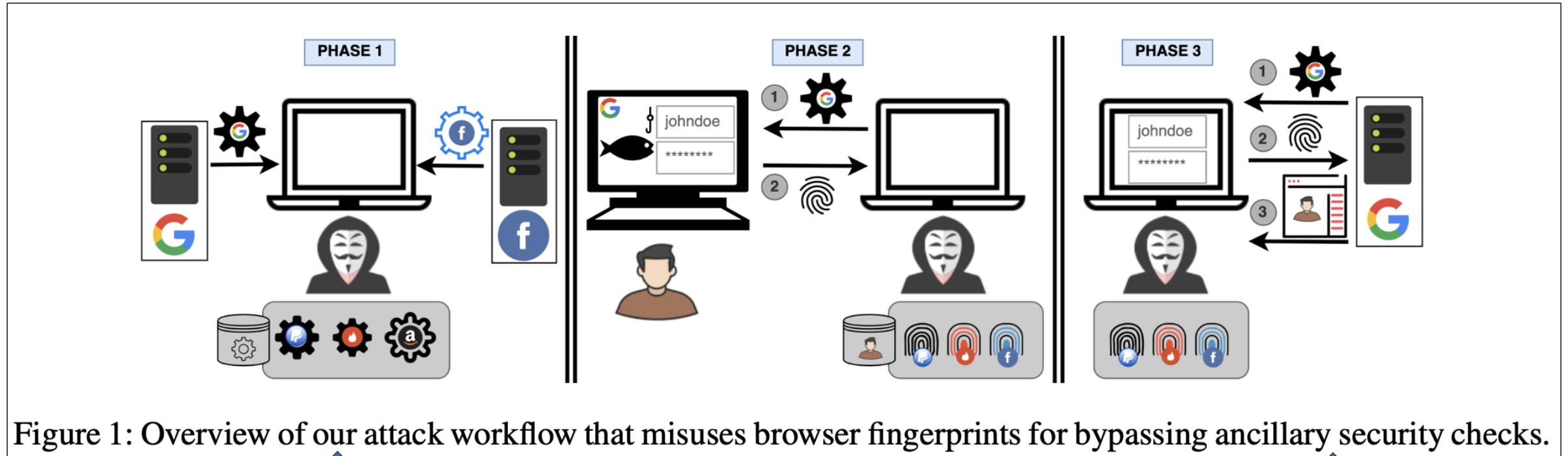
WASHINGTON STATE UNIVERSITY





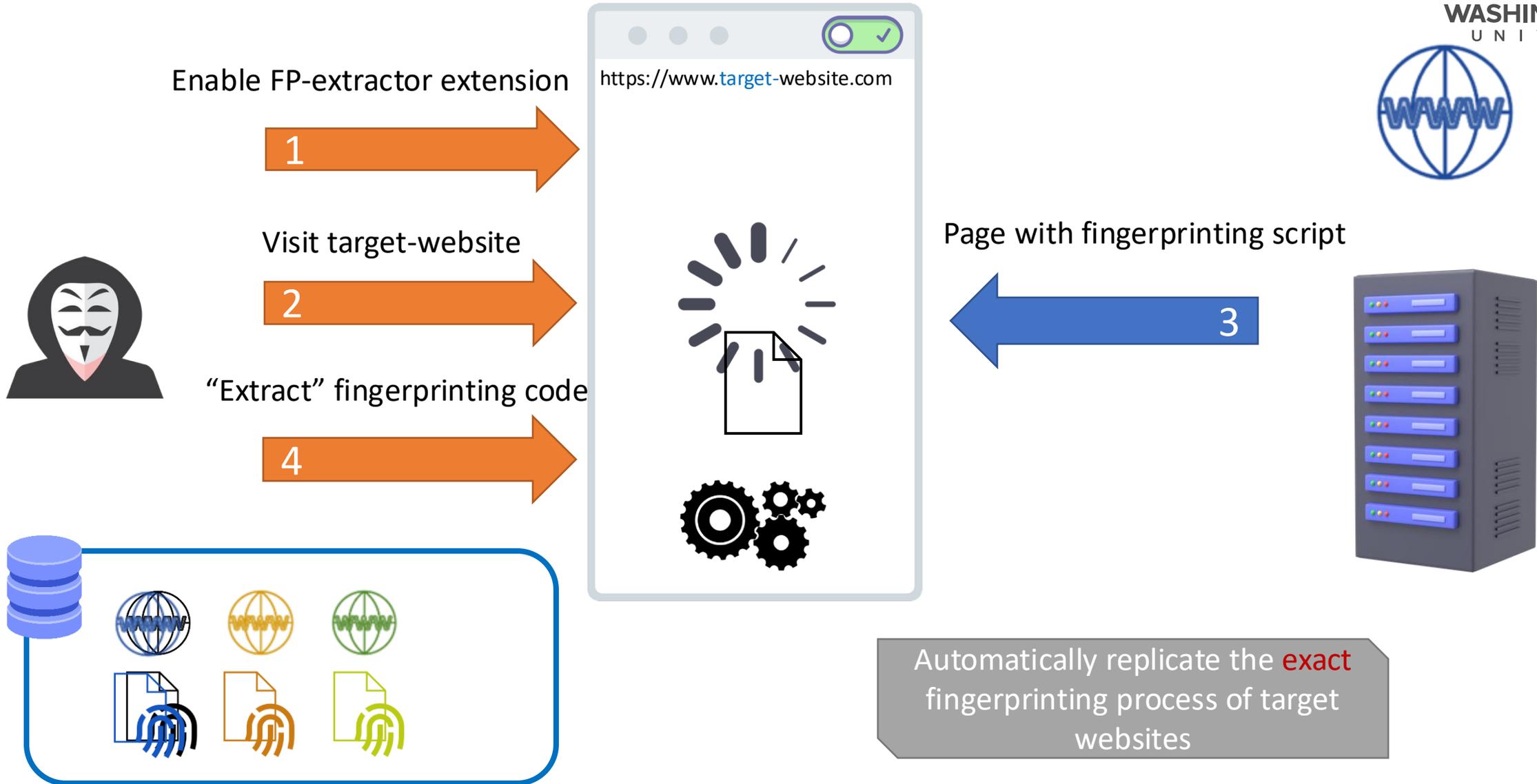
Lin, Xu, Panagiotis Ilia, Saumya Solanki, and Jason Polakis. "Phish in sheep's clothing: Exploring the authentication pitfalls of browser fingerprinting." In 31st USENIX Security Symposium (USENIX Security 22), pp. 1651-1668. 2022.

Overview of our attack workflow



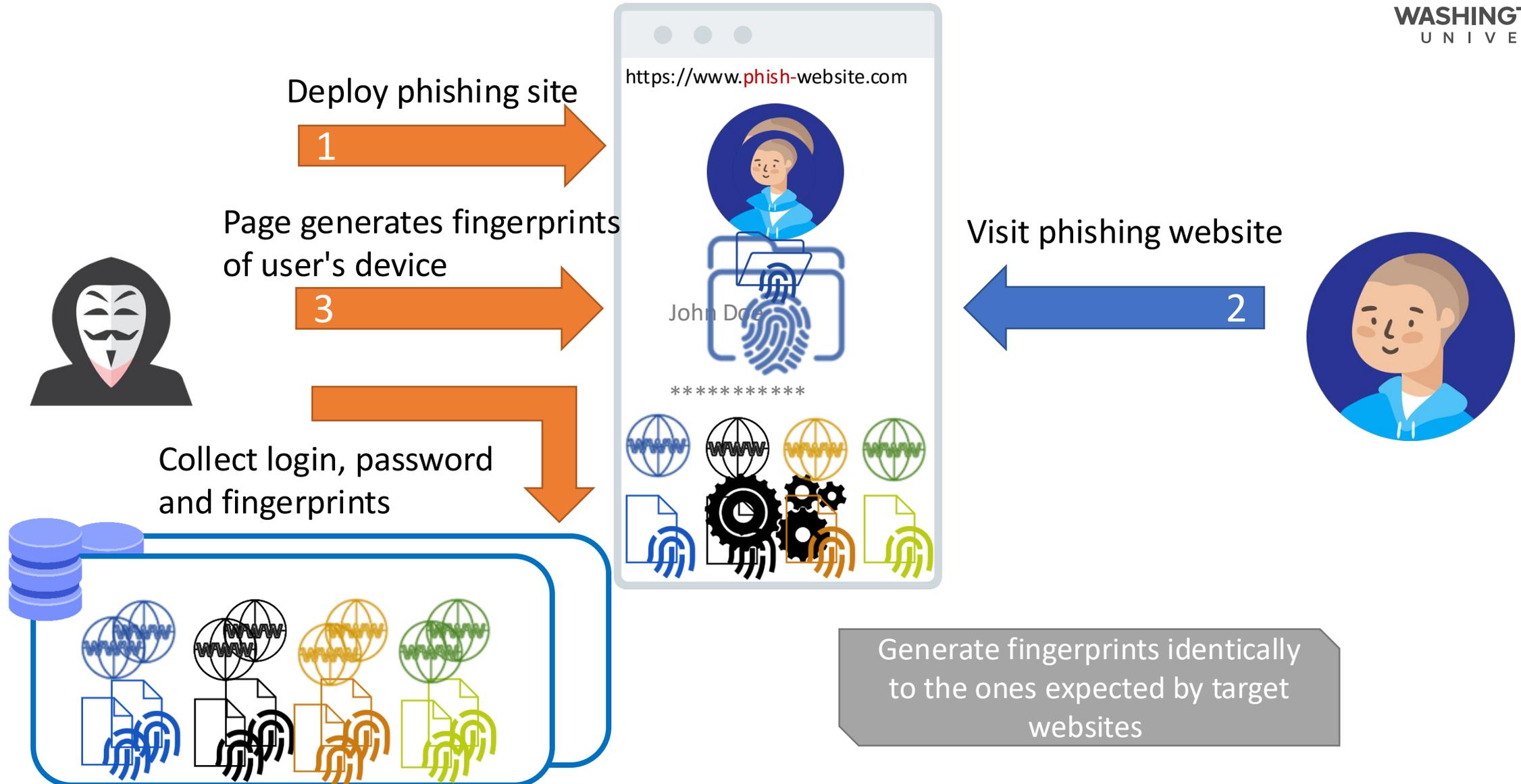


Phase1: attacker visits target websites and "extracts" their fingerprinting code





Phase2: attacker obtains user's credentials and fingerprints



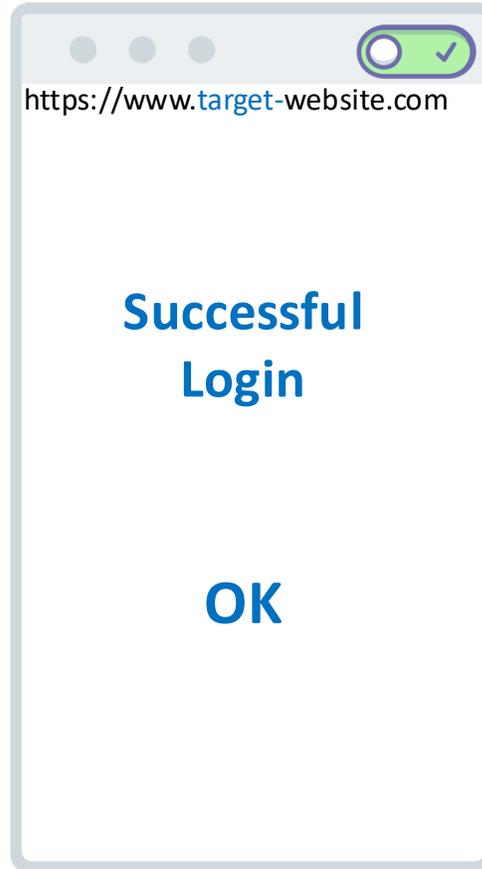


Phase3: attacker spoofs fingerprints and bypasses 2FA mechanism

Enable FP-Spoof extension



Spoof fingerprints, send login, password



Page with fingerprinting script



Grant access or trigger 2FA



Fingerprints Match !!



What about the real world ?





Risk-based authentication mechanisms in popular web services

Website	Fingerprinting Technique				IP Address Restrictions		Vulnerable
	BasicFP	Canvas/WebGL	Fonts	Audio	IP Check	Bypass	
Bank-A	✓	✗	✗	✗	✗	-	✓
Bank-B	✗	✗	✗	✗	✓	✗	✗
CreditCard	✓	✗	✗	✗	✓	→	✓
Trading-A	✓	✗	✗	✗	✗	-	✓
Trading-B	✗	✗	✗	✗	✓	→	✓
Tax-A	✓	✓	✗	✗	✓	✗	✗
Tax-B	✓	✓	✓	✗	✗	-	✓
Tax-C	✓	✓	✓	✓	✗	-	✓
Tax-D	✓	✓	✓	✓	✓	✗	✗
eCommerce-A	✓	✓	✗	✗	✗	-	✓
eCommerce-B	✓	✗	✗	✗	✓	✗	✗
RideSharing	✓	✓	✓	✗	✓	→	✓
Food&Beverage-A	✓	✗	✗	✗	✓	○	✓
Food&Beverage-B	✓	✗	✗	✗	✓	✗	✗

- We **completely bypass** 2FA in 9/14 websites that use FPs for authentication!
- Attack only prevented by IP address checks.
- We inject X-Forwarded-For header (used by proxies) with the user's IP to bypass IP-checks (→).
- Certain sites only require an IP from the same city (○).



So what can be done to prevent this?

- Web services:
 - Always trigger 2FA challenges (most secure, least user-friendly)
 - Chain sessions using one new and one old Canvas element [Laperdrix et al., DIMVA '19] (susceptible to other attacks)
 - Use strict IP address checks and require the presence of specific cookies
 - Follow layered multi-modal strategy to enhance security
- Users (common best-practice guidelines):
 - Always enable 2FA when possible
 - Use stronger second factors (e.g., authenticator apps, U2F keys)
 - Use password managers, never reuse passwords across services
 - Anti-fingerprinting browsers/extensions can help in certain cases



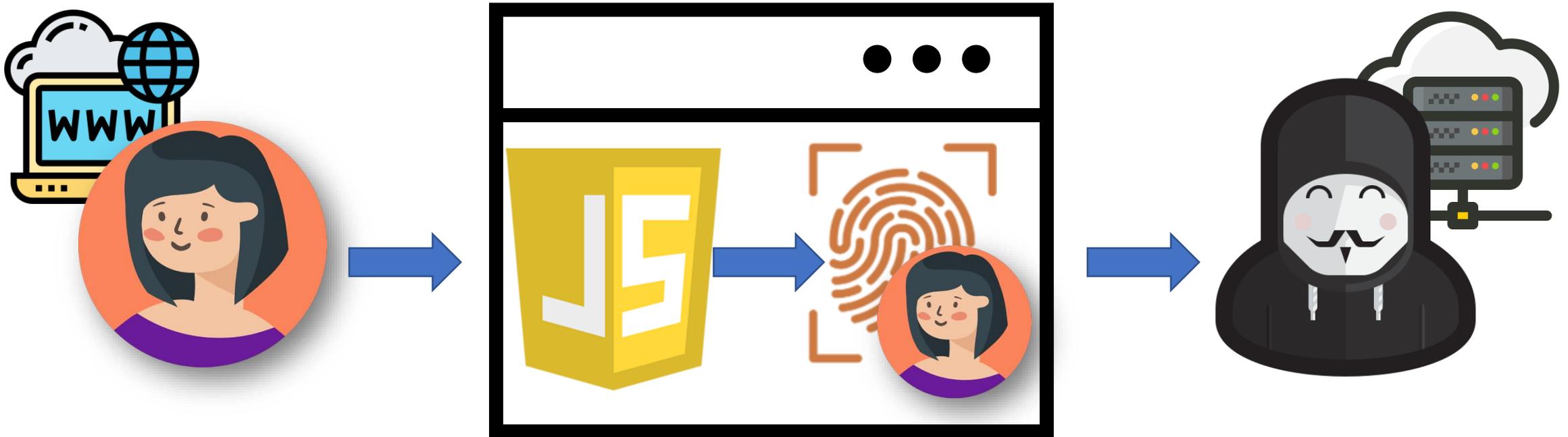
Browser fingerprinting is a double-edged sword

Usage scenarios

- **Security Enhancement**
 - **Bot Detection:** Differentiate between bots and legitimate web users
 - **Authentication:** Differentiate between a legitimate owner of an account and someone impersonating that owner
- **Privacy Concerns**
 - **Track users** against their will (stateless tracking)



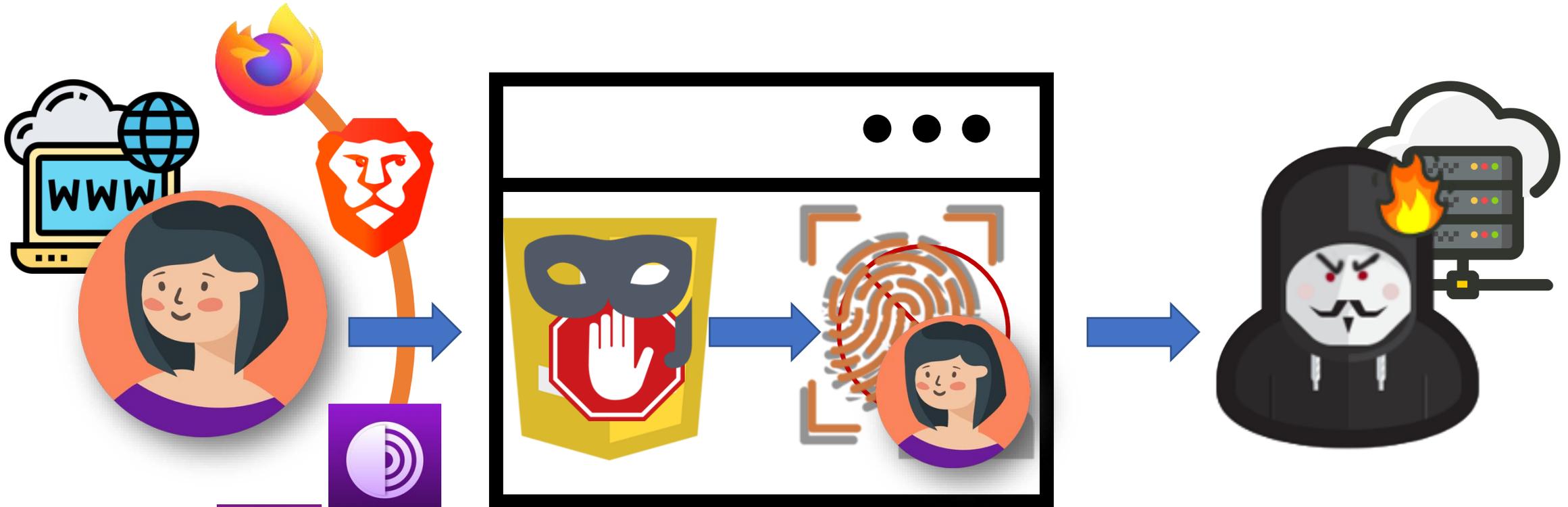
Online tracking



Browser fingerprinting heavily relies on JavaScript.



Fingerprinting countermeasures

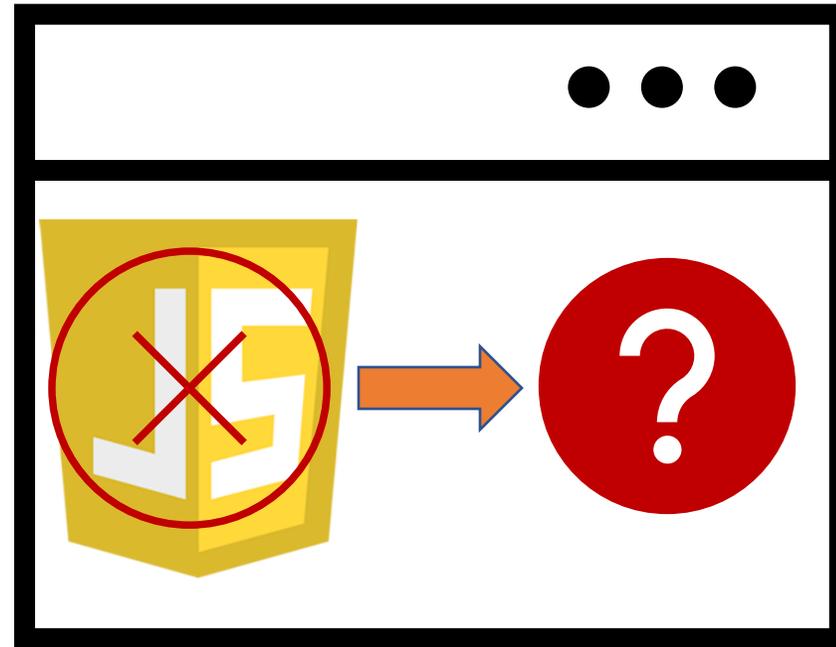


Privacy-focused browsers and anti-fingerprinting extensions

- Spoof certain APIs
- Disable JavaScript (entirely or partially)



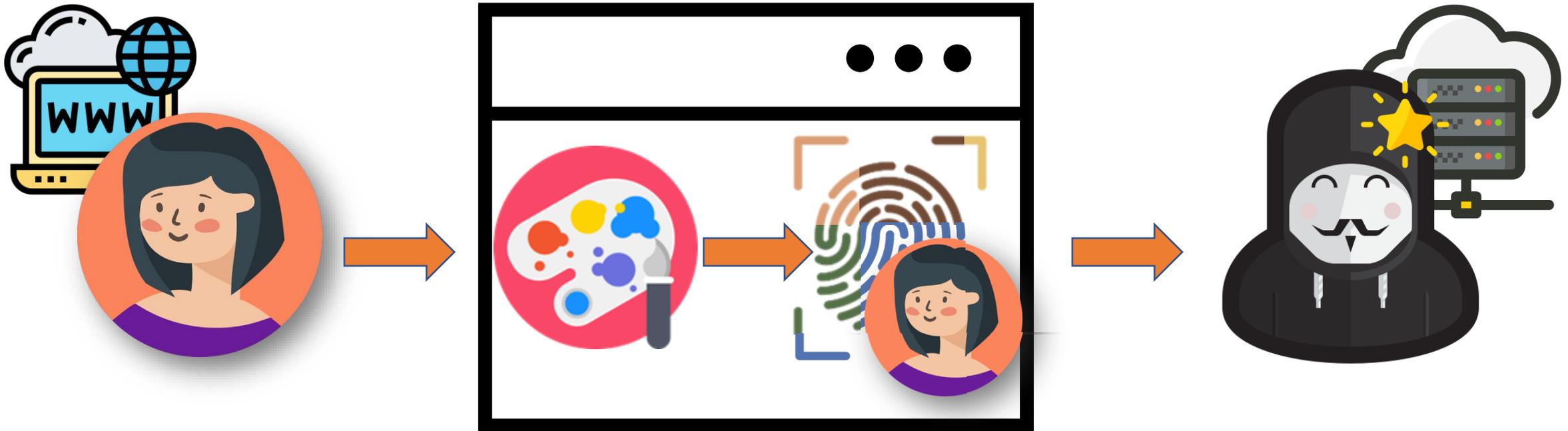
Is fingerprinting possible **without** JavaScript?





Our approach: Implicit stylistic browser fingerprinting

Lin, Xu, Frederico Araujo, Teryl Taylor, Jiyong Jang, and Jason Polakis. "Fashion faux pas: Implicit stylistic fingerprints for bypassing browsers' anti-fingerprinting defenses." In 2023 IEEE Symposium on Security and Privacy (SP), pp. 987-1004. IEEE, 2023.



Implicit stylistic browser fingerprinting

- Does not use any JavaScript
- Provides highly discriminating fingerprints



What can we use to detect the stylistic differences?

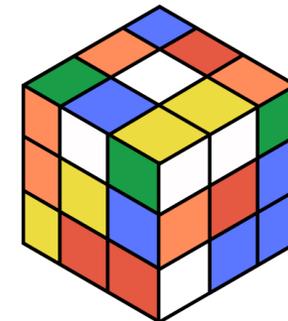
 Chrome	 Firefox	 Safari
40px/15px	183px/16px	34px/13px

12-hour Time	24-hour Time
146px/32px	99px/32px

English OS	Chinese OS
425px/35px	425px/41px



Dimensions!





Fingerprinting attributes

Certain HTML elements have different sizes depending on certain environmental factors.

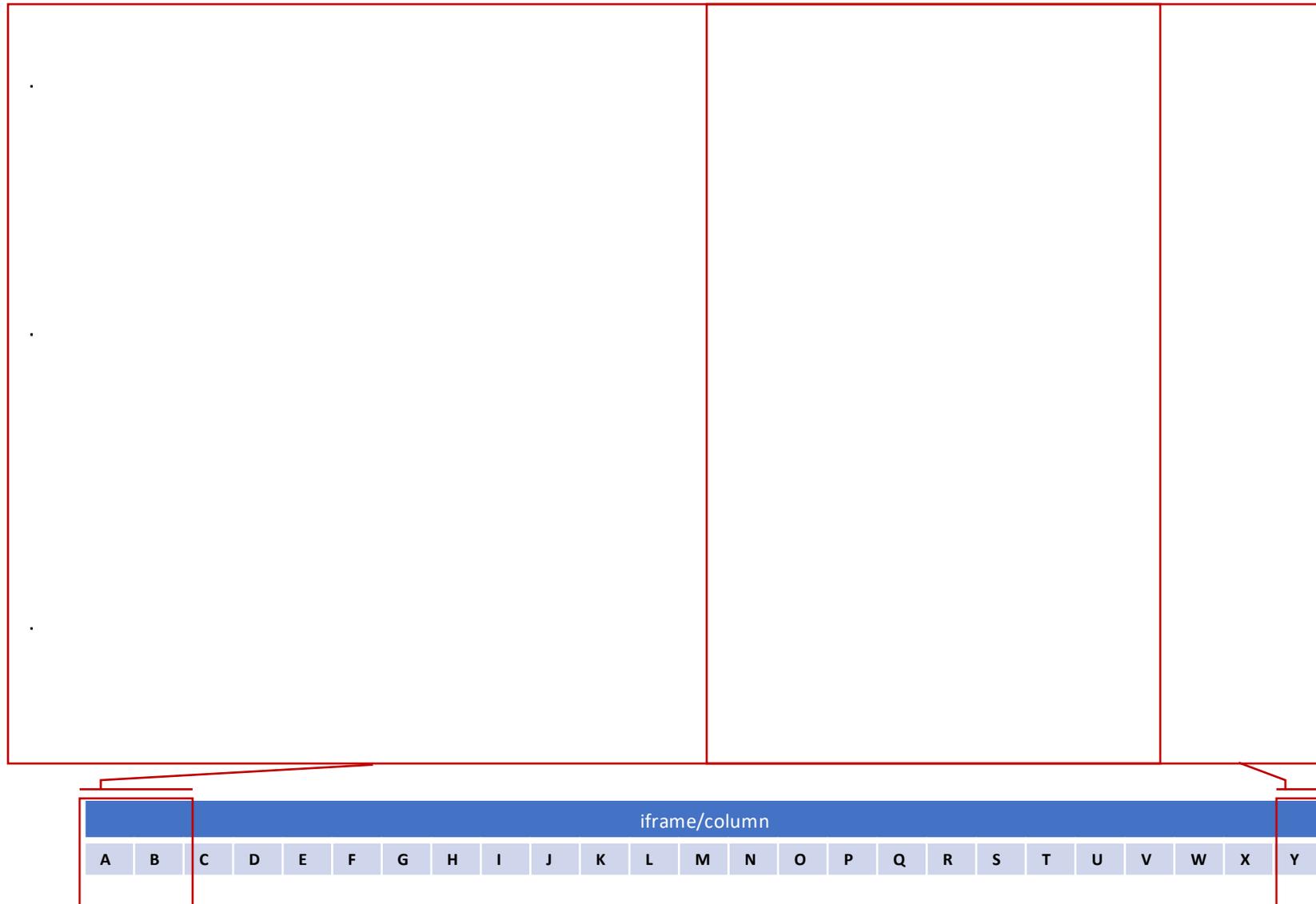
339
Fingerprinting Elements



Category	Fingerprint attributes	AIU	FPJS	ATE
Environment	browser	●	●	
	browser major version	●	●	
	operating system	●	●	
	platform	◐	◐	
	operating system language			
	scrollbar settings			
	JS disabled			
Fonts	font preferences		●	
	supported fonts	●	●	
	supported shadow fonts			
Ad blocker	presence of ad blocker	●		
	ad blocker identification			
Media properties	screen resolution	●	●	
	supported media features		◐	
	media features' values		◐	

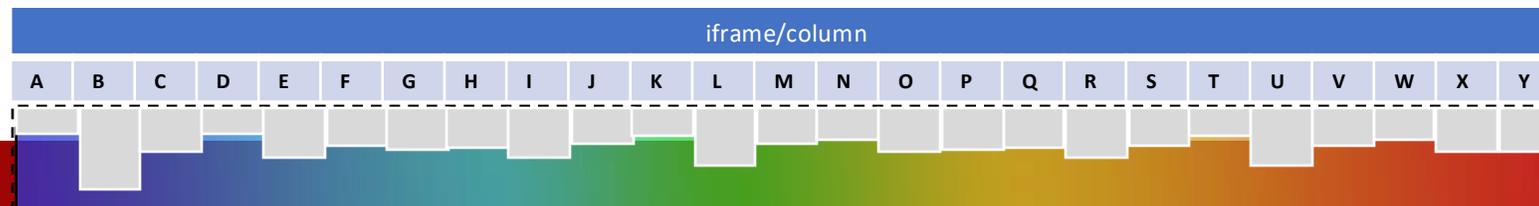
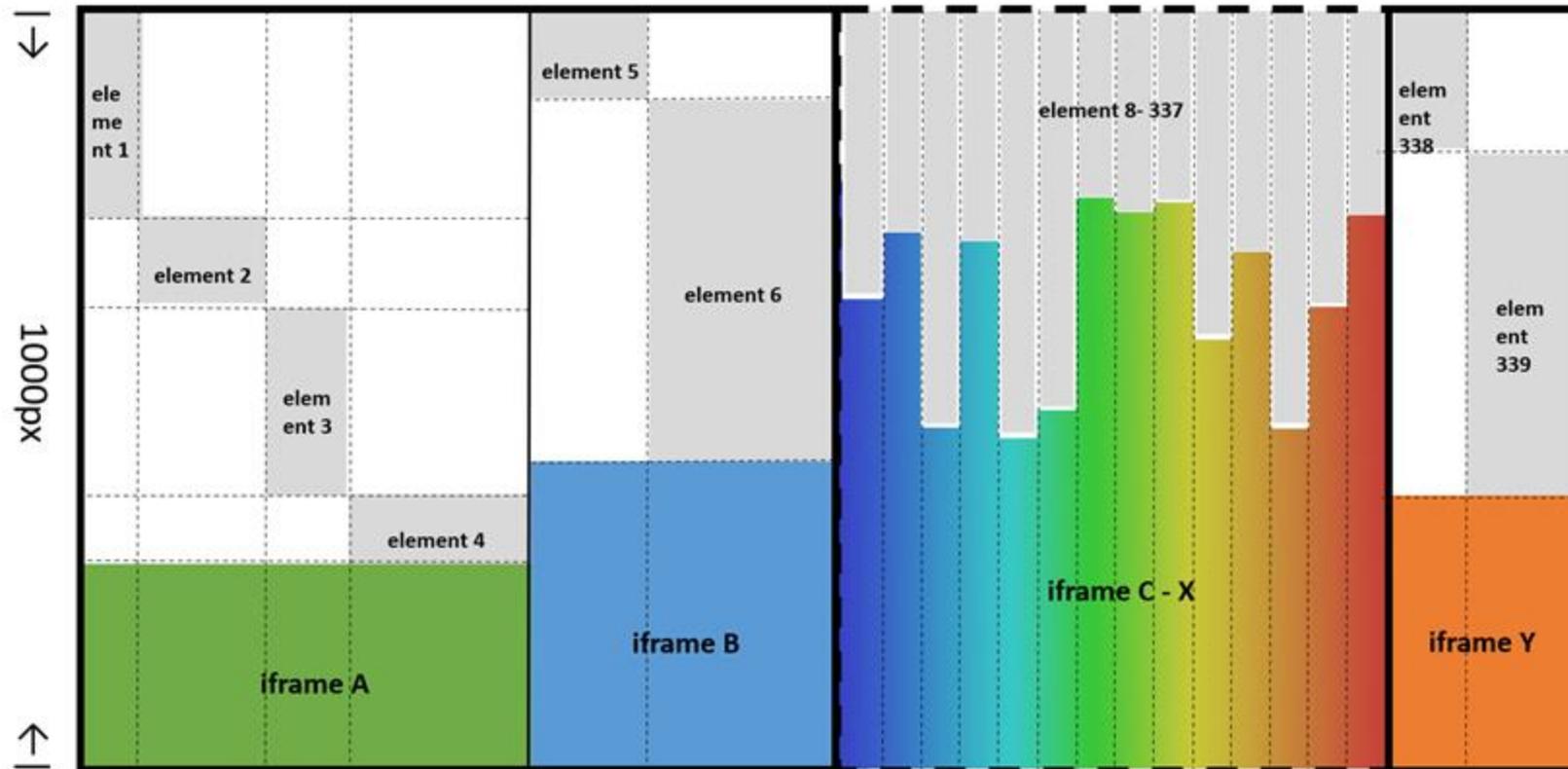
AIU: captured by AmlUnique FPJS: captured by FingerprintJS
◐ : partial feature support ● : full feature support

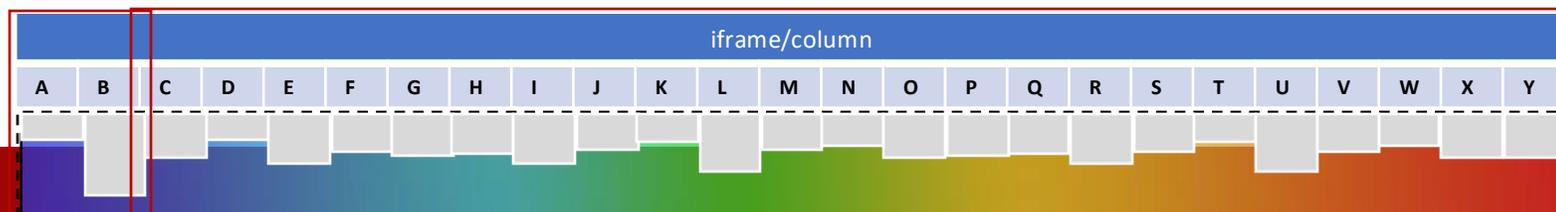
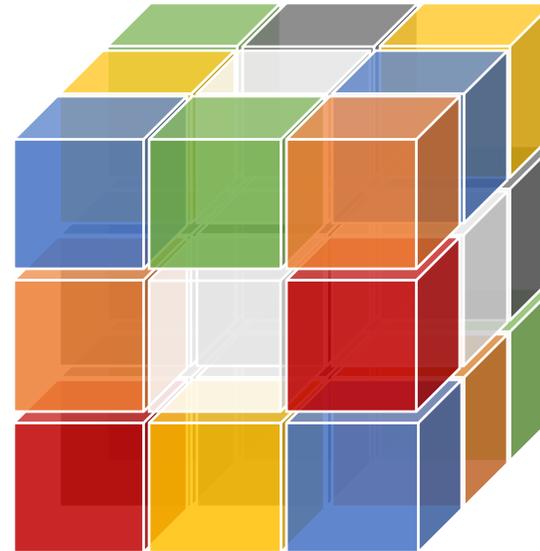
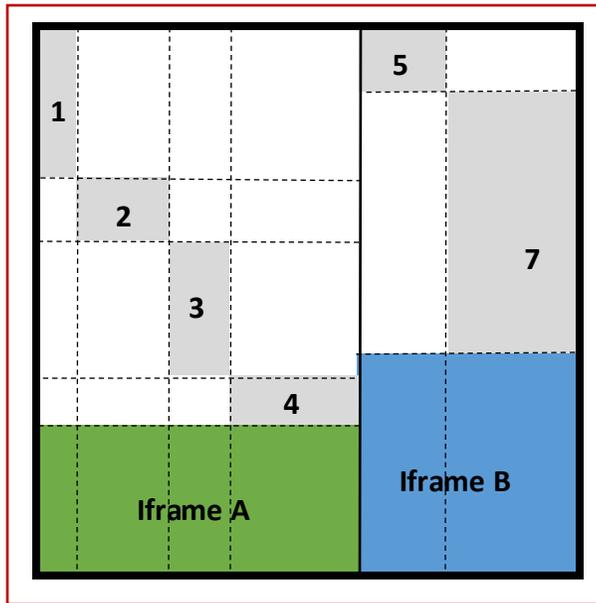
- The page only needs **25** iframes.
- All elements are placed in an 800px by 1000px iframe (main iframe) to ensure that their dimensions remain consistent across different browser window sizes.





Main iframe







Other Tracking Vectors: Using Extensions

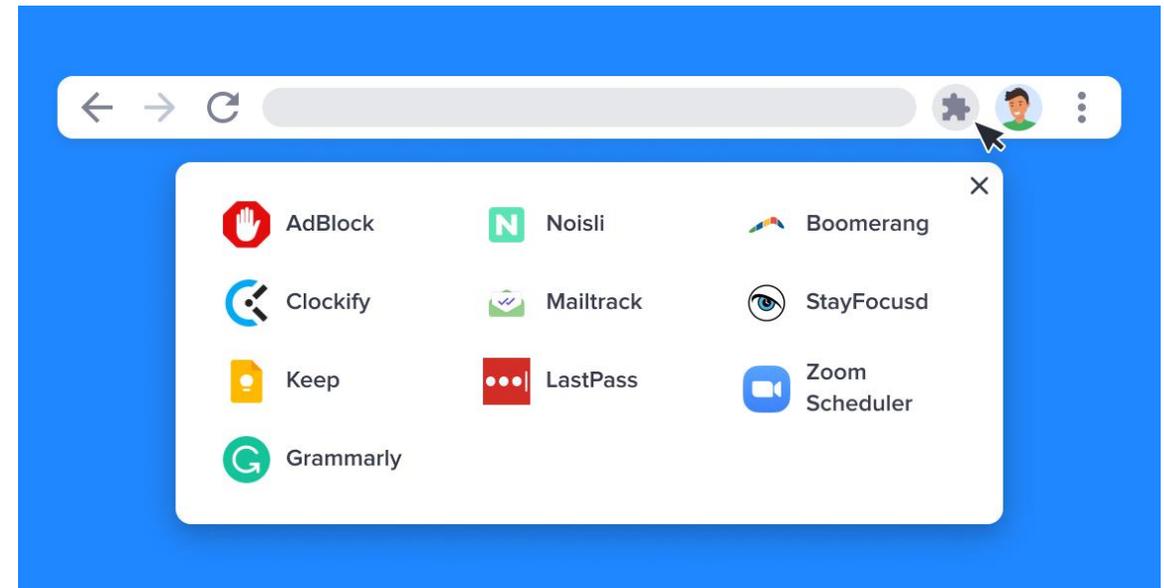
Carnus: Exploring the Privacy Threats of Browser Extension Fingerprinting

Soroush Karami, Panagiotis Ilia, Konstantinos Solomos, Jason Polakis
University of Illinois at Chicago, USA
{skaram5, pilia, ksolom6, polakis}@uic.edu

Abstract—With users becoming increasingly privacy-aware and browser vendors incorporating anti-tracking mechanisms, browser fingerprinting has garnered significant attention. Accordingly, prior work has proposed techniques for identifying browser extensions and using them as part of a device's fingerprint. While previous studies have demonstrated how extensions can be detected through their web accessible resources, there exists a significant gap regarding techniques that indirectly detect extensions through behavioral artifacts. In fact, no prior study

browsers still mediate a large portion of our online activities. As a result, the evolution of websites from static resources to functionality-rich applications has also necessitated the evolution of browsers into complex platforms with a rich set of APIs and features. To improve user experience, browsers allow users to further personalize them and extend their functionality by installing extensions.

Apart from the obvious benefits for users [26], [38], [48],



Fingerprinting in Style: Detecting Browser Extensions via Injected Style Sheets

Pierre Laperdrix
Univ. Lille, CNRS, Inria

Oleksii Starov
Palo Alto Networks

Quan Chen
North Carolina State University

Alexandros Kapravelos
North Carolina State University

Nick Nikiforakis
Stony Brook University



Abstract

Browser extensions enhance the web experience and have seen great adoption from users in the past decade. At the same time, past research has shown that online trackers can use

and a browser's private mode) stateless tracking techniques arose that enable third parties to track users across sessions, without relying on previously set cookies or other stateful identifiers. These stateless techniques essentially "fingerprint"



Other Tracking Vectors: Using Favicons

Tales of FAVICONS and Caches: Persistent Tracking in Modern Browsers

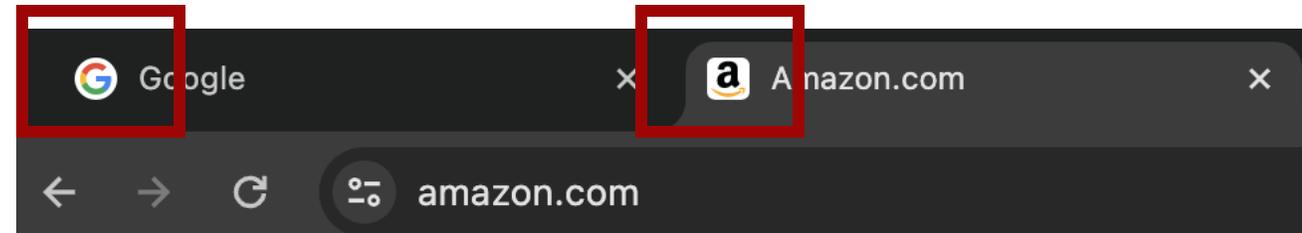
Konstantinos Solomos, John Kristoff, Chris Kanich, Jason Polakis
University of Illinois at Chicago
{ksolom6, jkrist3, ckanich, polakis}@uic.edu

I. INTRODUCTION

Abstract—The privacy threats of online tracking have garnered considerable attention in recent years from researchers and practitioners. This has resulted in users becoming more privacy-cautious and browsers gradually adopting countermeasures to mitigate certain forms of cookie-based and cookie-less tracking. Nonetheless, the complexity and feature-rich nature of modern browsers often lead to the deployment of seemingly innocuous functionality that can be readily abused by adversaries. In this paper we introduce a novel tracking mechanism that misuses a simple yet ubiquitous browser feature: *favicons*. In more detail, a website can track users across browsing sessions by storing a tracking identifier as a set of entries in the browser's dedicated favicon cache, where each entry corresponds to a specific sub-domain. In subsequent user visits the website can reconstruct the identifier by observing which favicons are requested by the browser while the user is automatically and rapidly redirected through a series of subdomains. More importantly, the caching of favicons in modern browsers exhibits several unique characteristics that render this tracking vector particularly powerful, as it is persistent (not affected by users clearing their browser data), non-destructive (reconstructing the identifier in subsequent visits does not alter the existing combination of cached entries), and even crosses the isolation of the *incognito* mode. We experimentally

Browsers lie at the heart of the web ecosystem, as they mediate and facilitate users' access to the Internet. As the Web continues to expand and evolve, online services strive to offer a richer and smoother user experience; this necessitates appropriate support from web browsers, which continuously adopt and deploy new standards, APIs and features [76]. These mechanisms may allow web sites to access a plethora of device and system information [55], [21] that can enable privacy-invasive practices, e.g., trackers leveraging browser features to exfiltrate users' Personally Identifiable Information (PII) [24]. Naturally, the increasing complexity and expanding set of features supported by browsers introduce new avenues for privacy-invasive or privacy-violating behavior, thus, exposing users to significant risks [53].

In more detail, while cookie-based tracking (e.g., through third-party cookies [57]) remains a major issue [29], [9], [69], tracking techniques that do not rely on HTTP cookies are on the rise [63], [16] and have attracted considerable attention from the research community (e.g., novel techniques





1993



"On the Internet, nobody knows you're a dog."

©The New Yorker Collection 1993, Peter Steiner
From cartoonbank.com. All rights reserved.

2015



"Remember when, on the Internet, nobody knew who you were?"

