WASHINGTON STATE
UNIVERSITY

Web Browser Fingerprinting: Revealing Stealth Tracking Techniques and Defenses

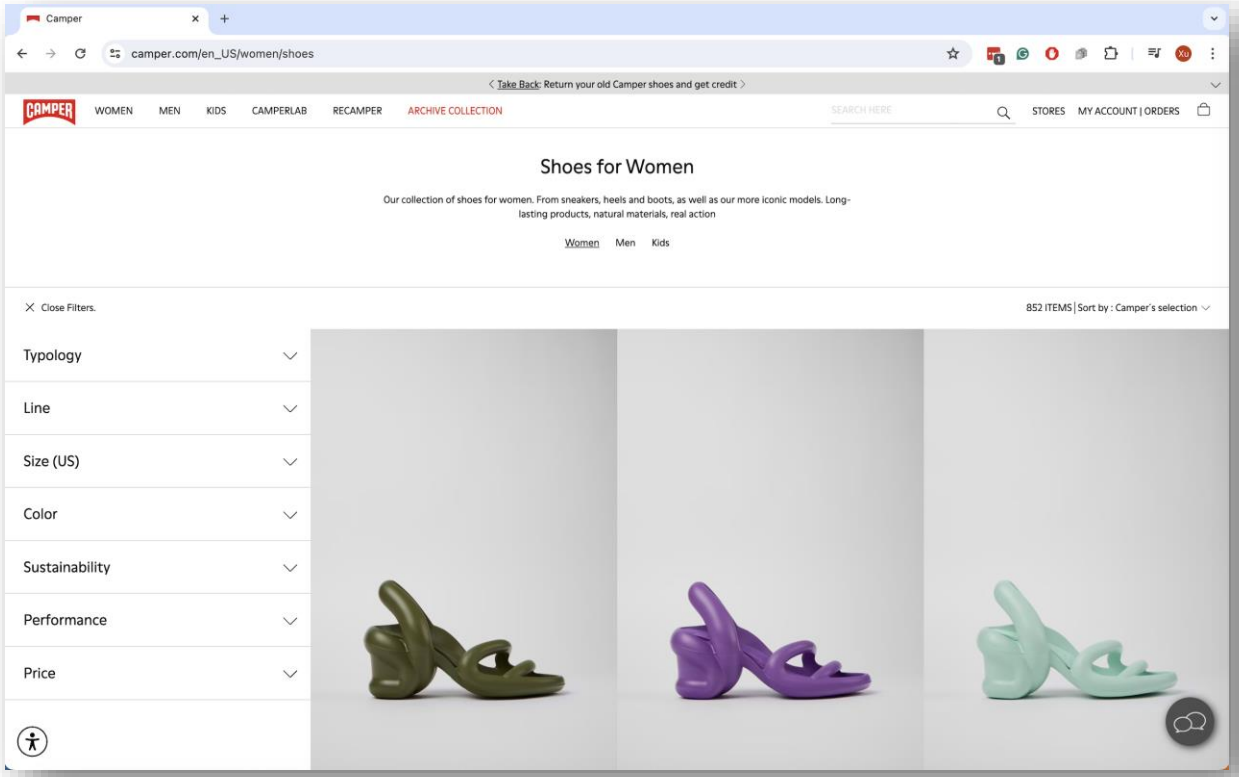
Xu Lin

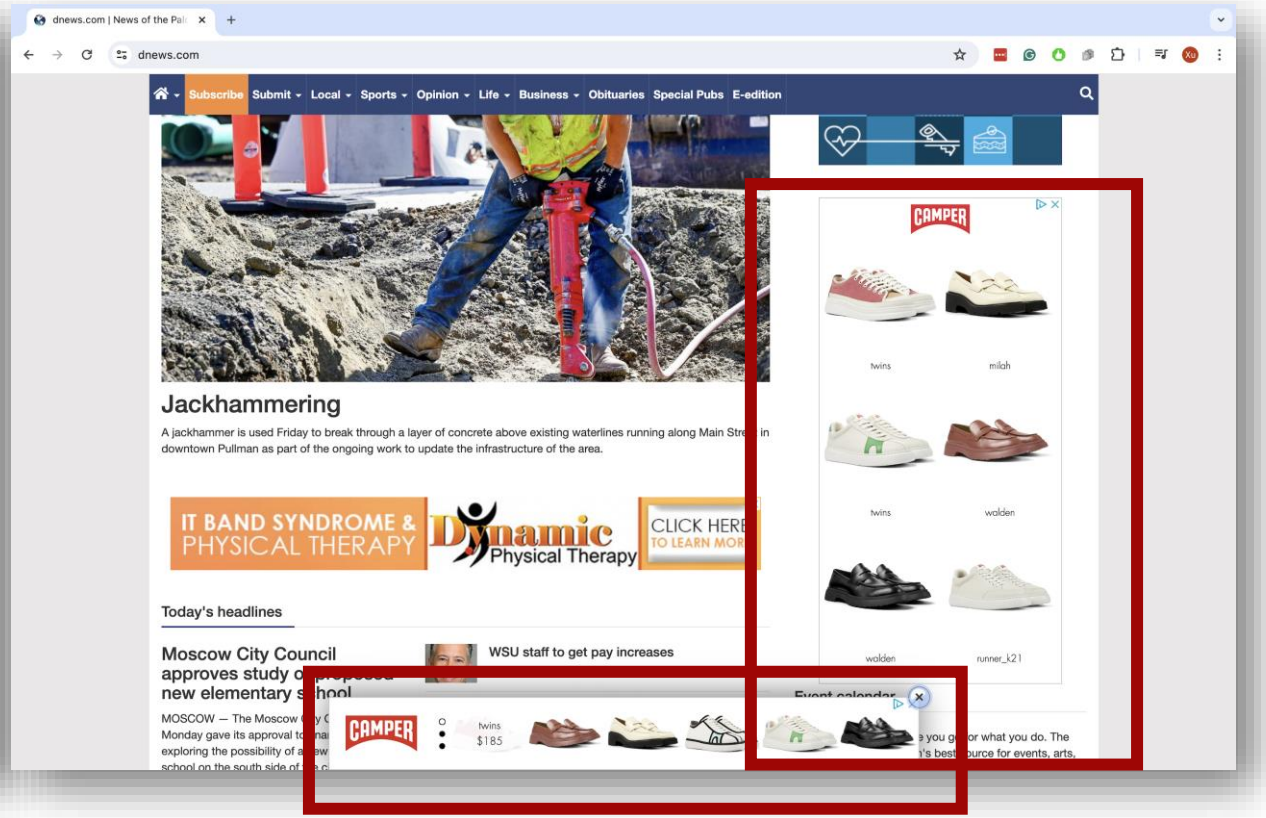
Washington State University

CySER May 22, 2024


What is Web Tracking?

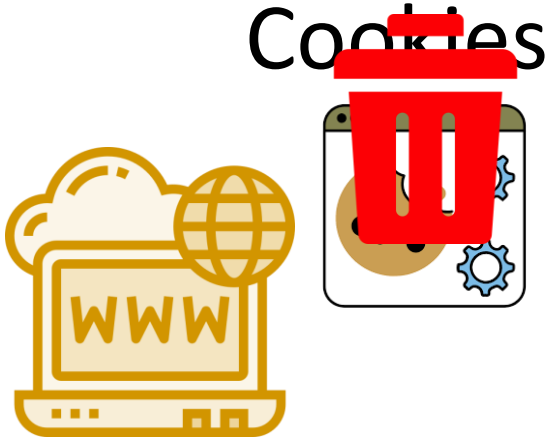
Why Does Tracking Exist?

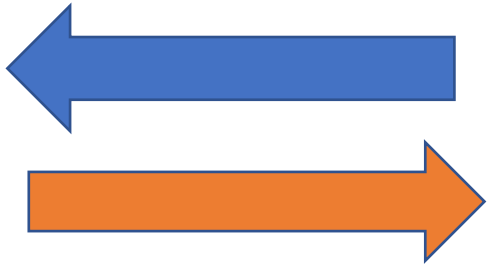




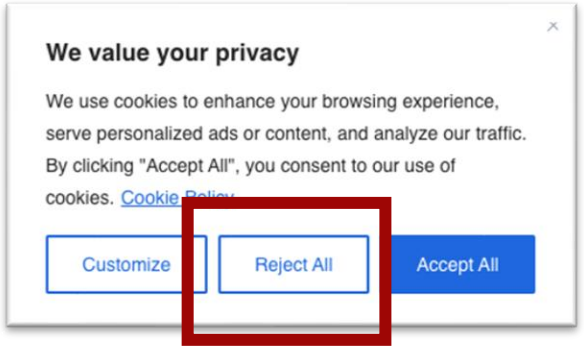
Classic Tracking

Cookies





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

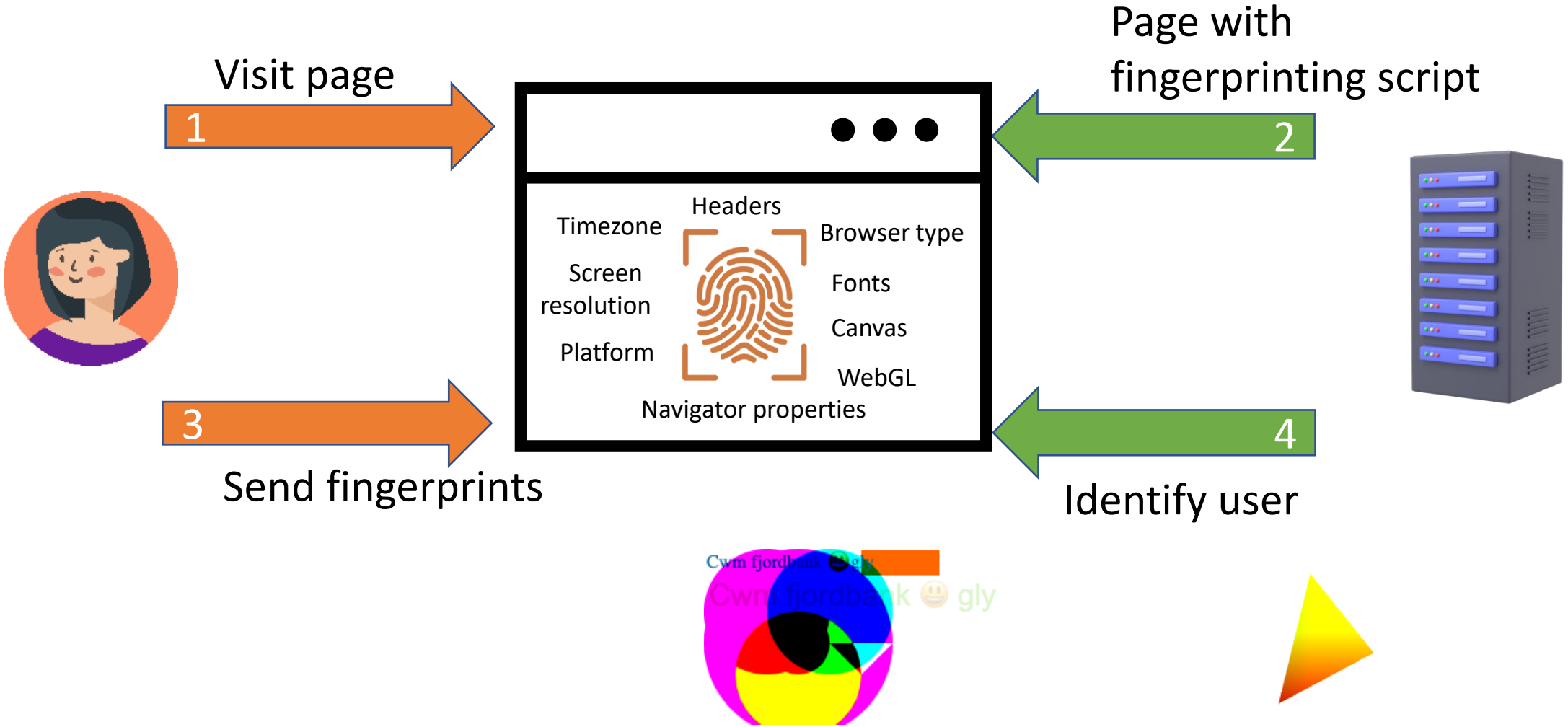


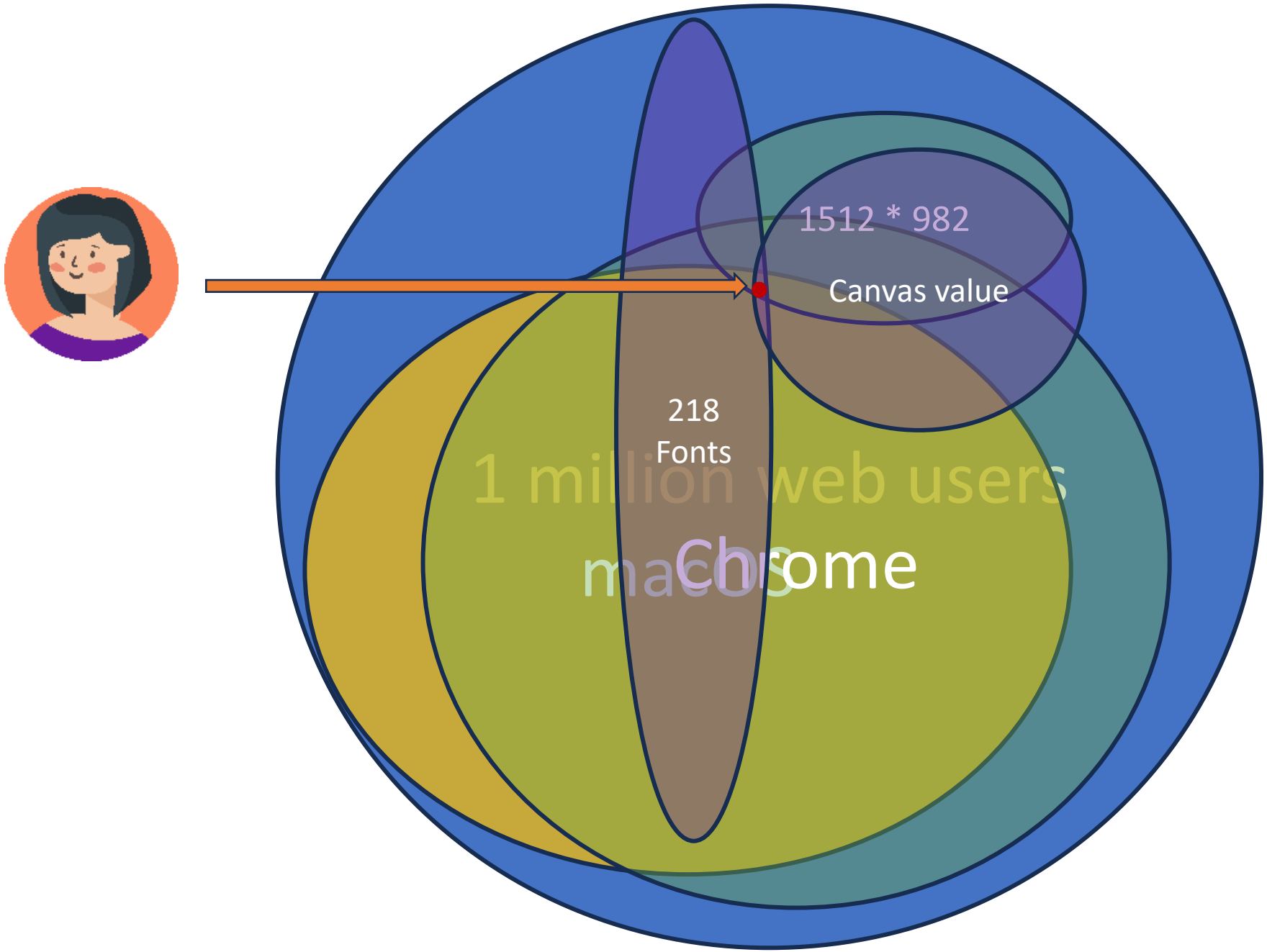

Do websites still know who you are?

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**.


Visit page
1
Page with fingerprinting script
2
Headers
Timezone
Browser type
Screen resolution
Fonts
Canvas
Platform
WebGL
Navigator properties
3
Send fingerprints
4
Identify user


1512 * 982
Canvas value
218
Fonts
1 million web users
macOS
Chrome

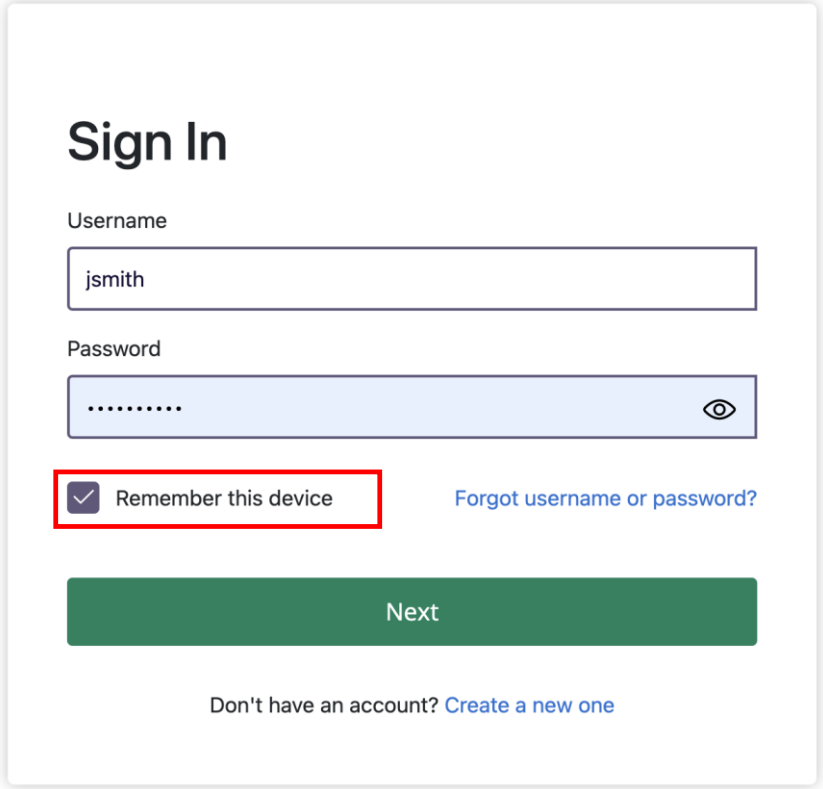
Browser fingerprinting is a double-edged sword

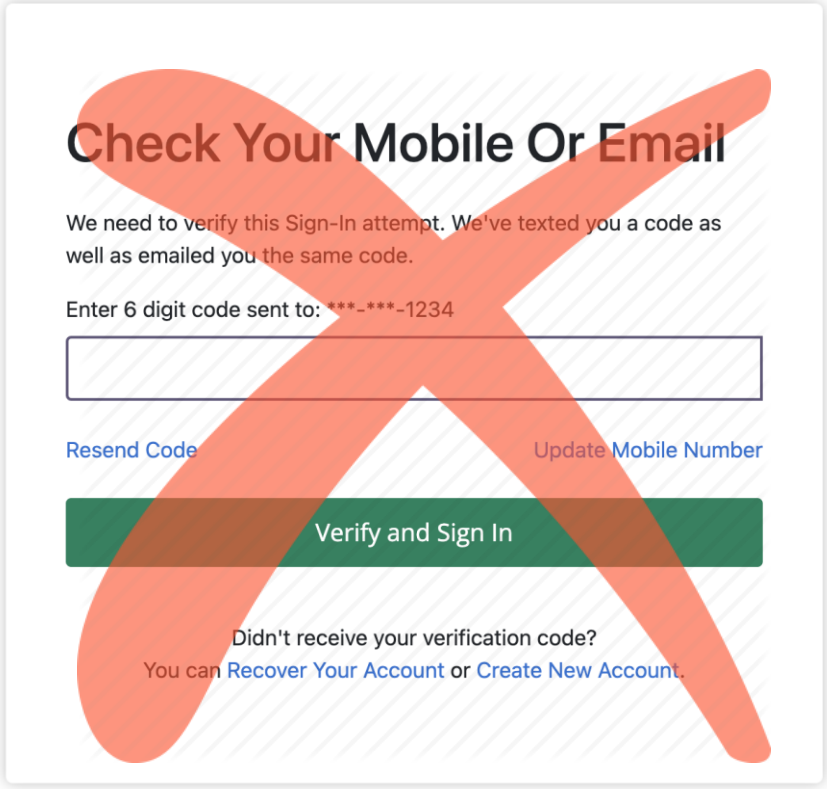
WASHINGTON STATE UNIVERSITY

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)

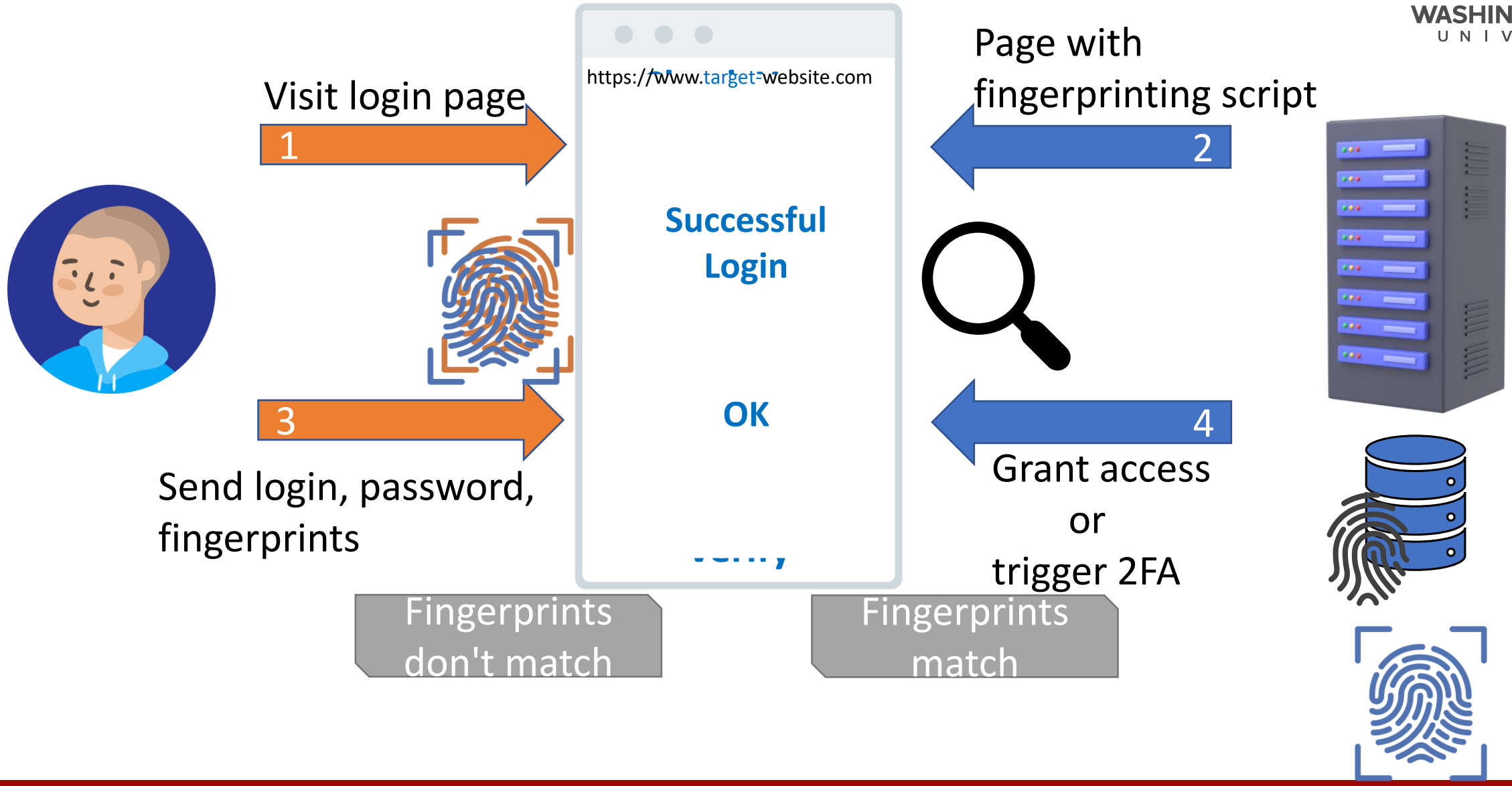
Risk-Based Authentication and Two-Factor Authentication (2FA)

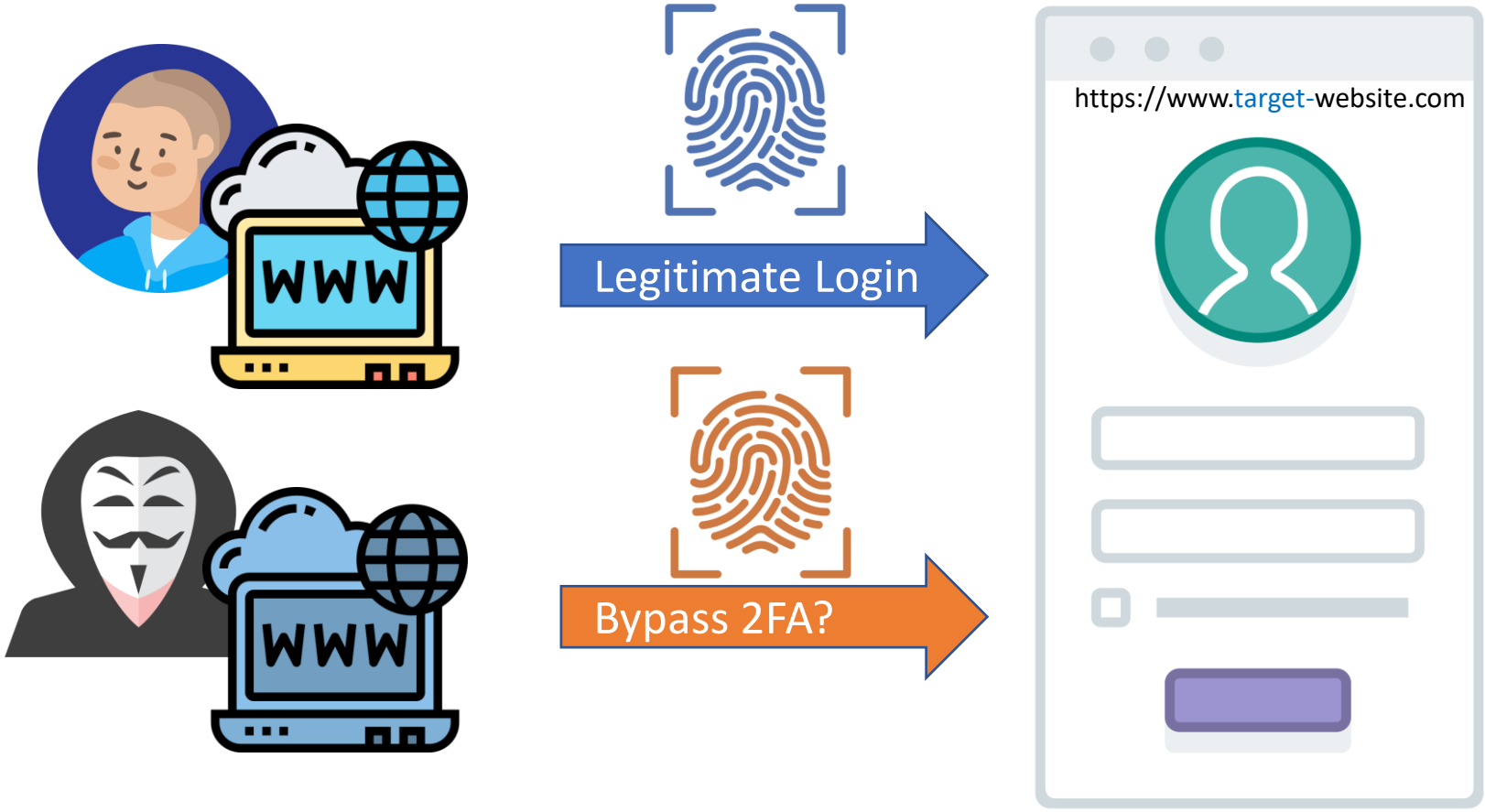




- 2FA creates friction for users
- Certain websites only trigger 2FA for *suspicious* login attempts

Advanced Risk-Based Authentication That Uses Browser Fingerprinting




WWW
WWW
Legitimate Login
Bypass 2FA?
https://www.target-website.com

Overview of our attack workflow

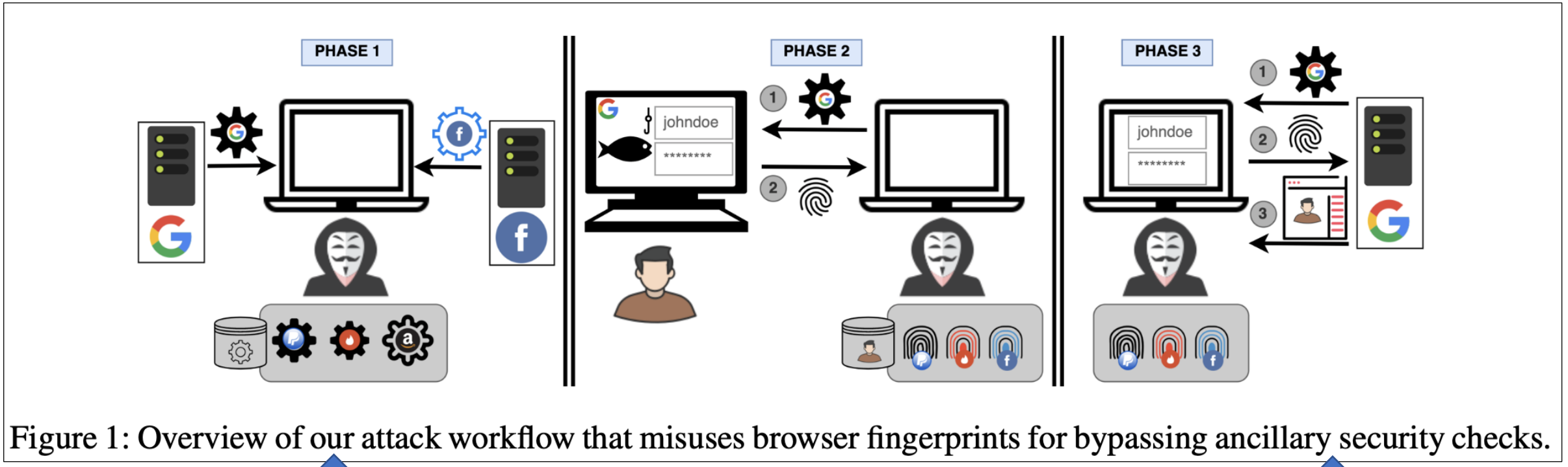
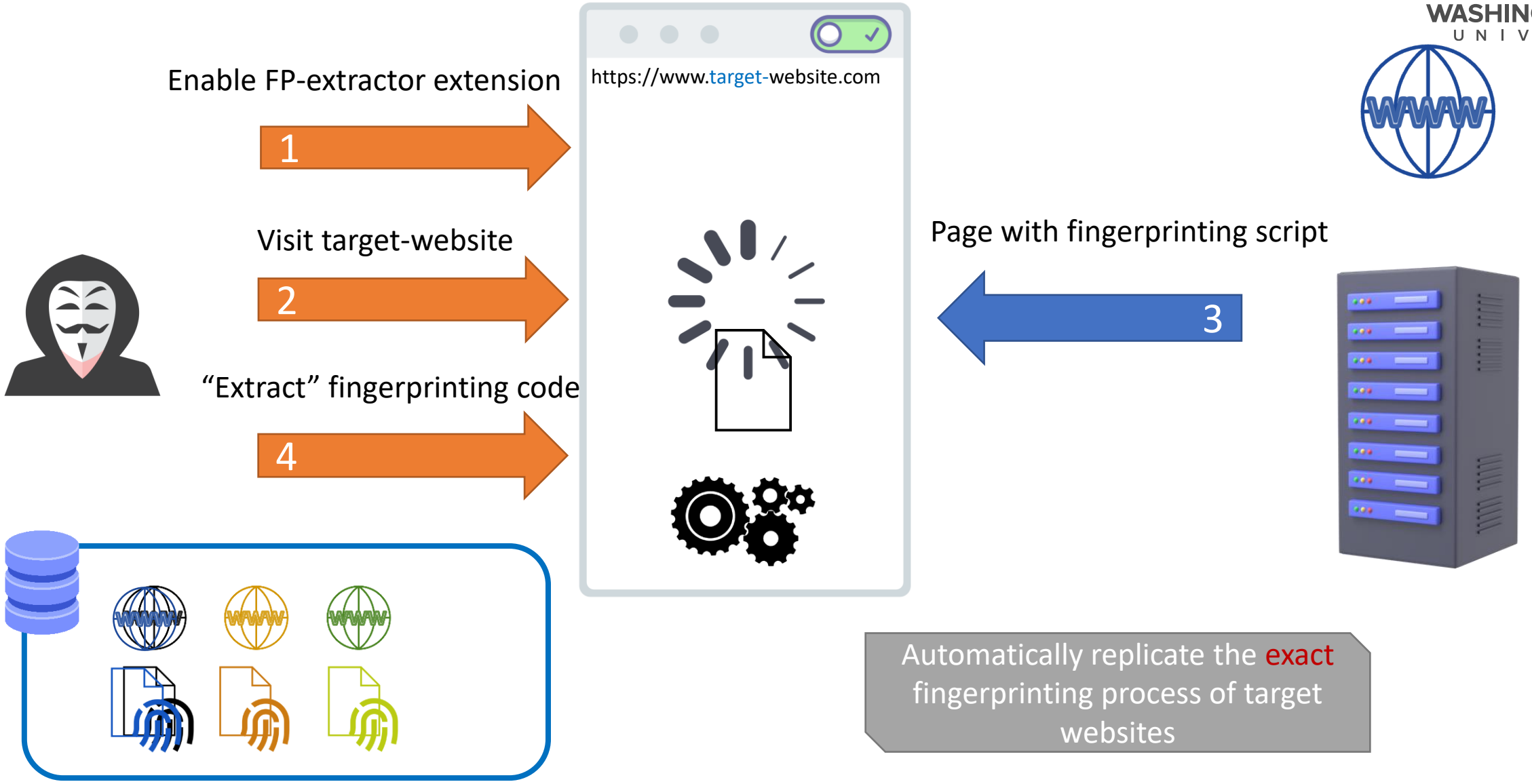


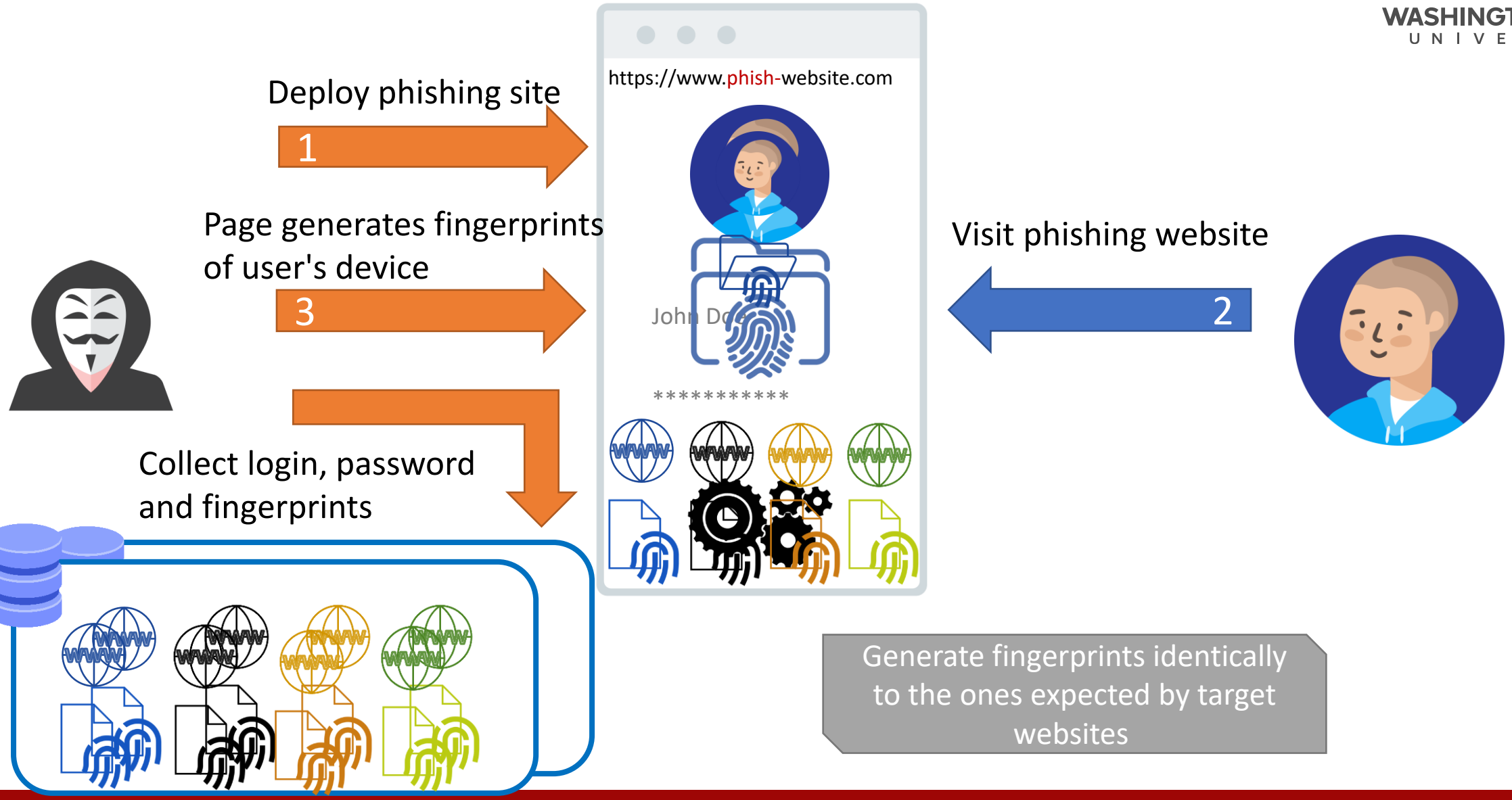
Figure 1: Overview of our attack workflow that misuses browser fingerprints for bypassing ancillary security checks.

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

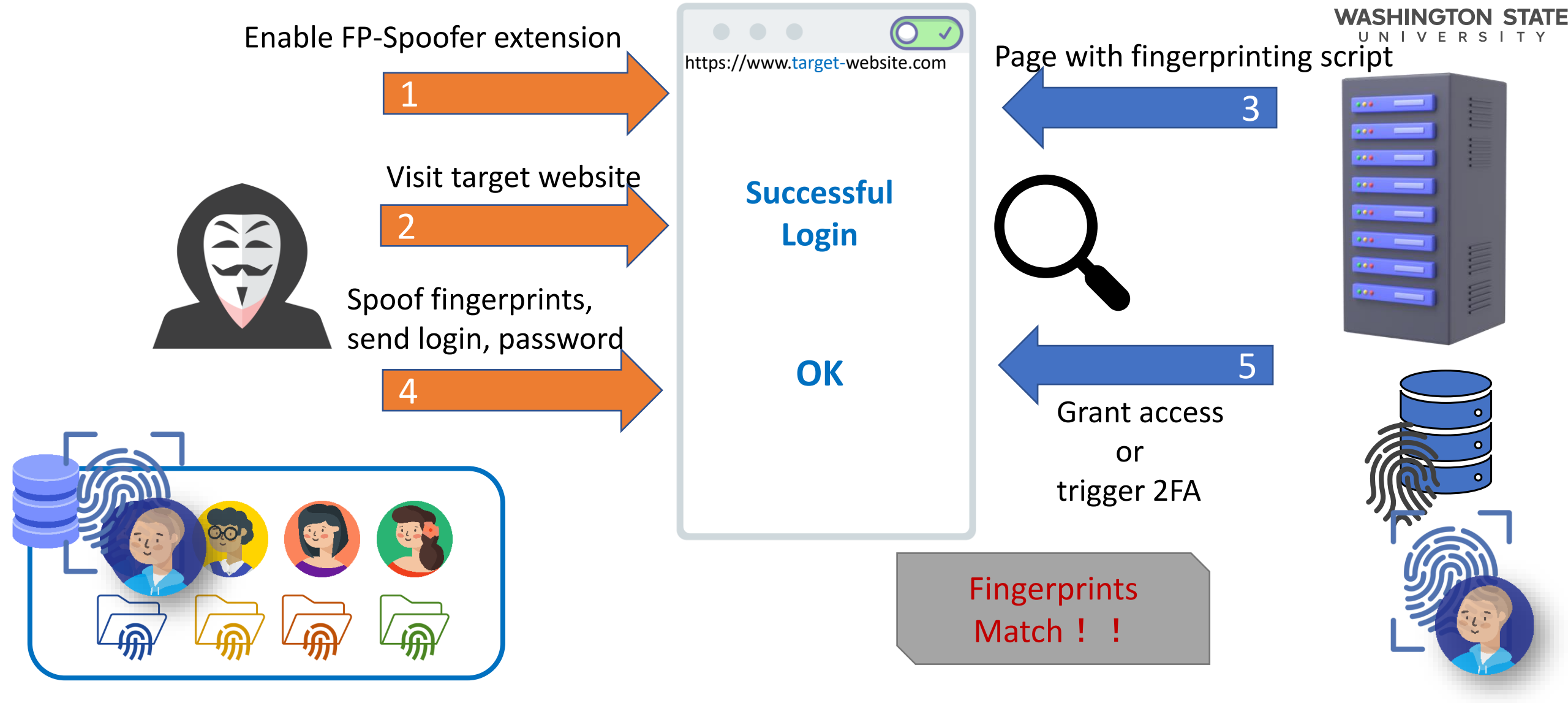


Automatically replicate the exact fingerprinting process of target websites

Phase2: attacker obtains user's credentials and fingerprints



Phase3: attacker spoofs fingerprints and bypasses 2FA mechanism




What about the real world ?

Risk-based authentication mechanisms in popular web services

WASHINGTON STATE UNIVERSITY

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 (○).

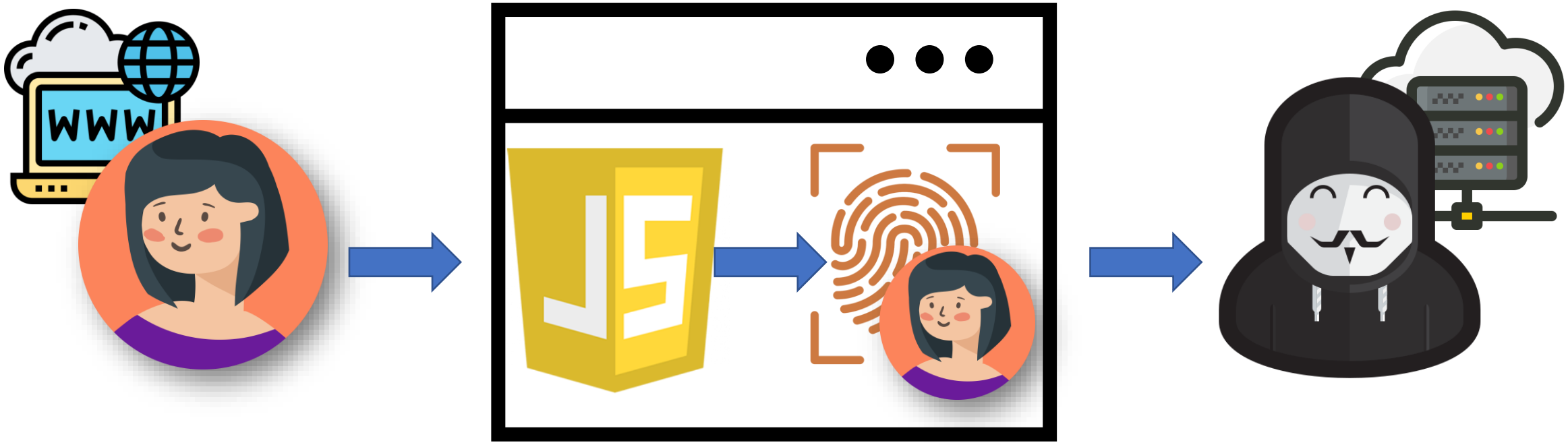
Browser fingerprinting is a double-edged sword

WASHINGTON STATE UNIVERSITY

Usage scenarios

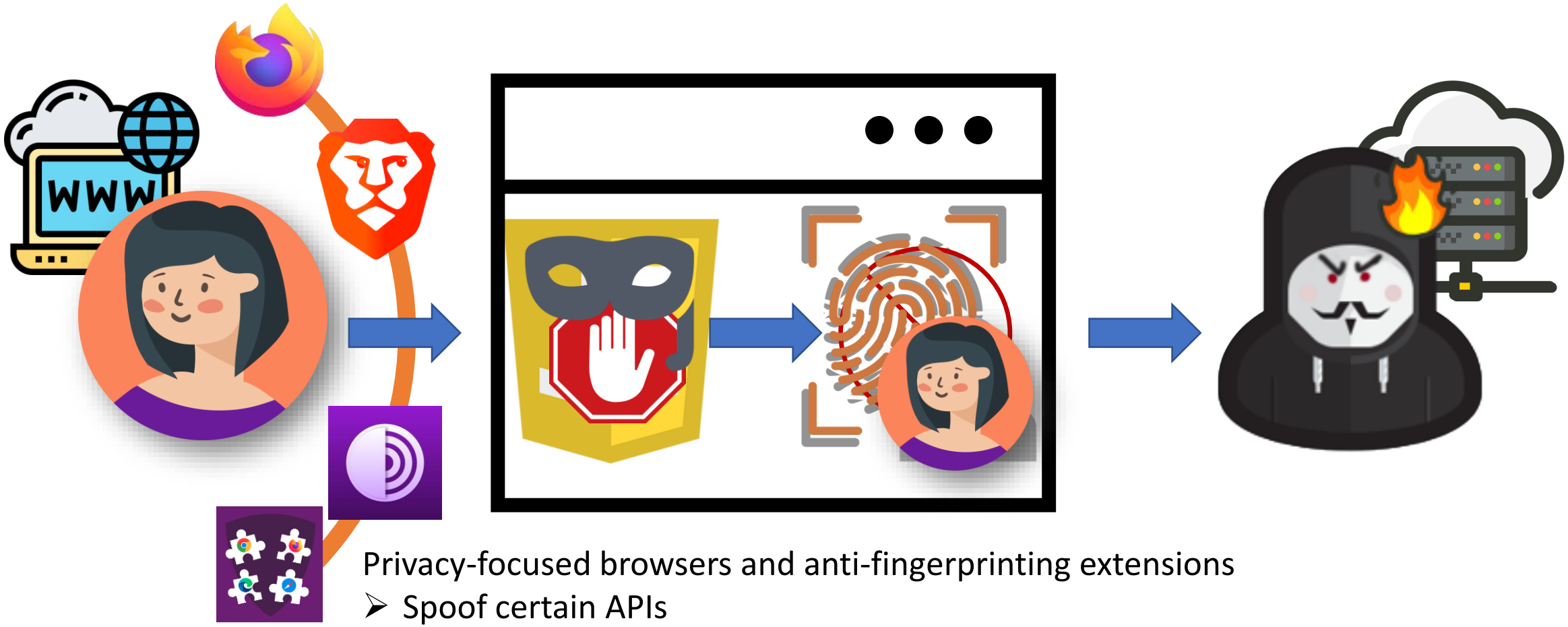
- **Security Enhancement**
  - **Bot Detection**: Differentiate between bots and legitimate web users
  - **Authentication**: Differentiate between a legitimate owner of an account and someone impersonating that owner
- **Privacy Concerns**
  - **Track users** against their will (stateless tracking)

Online tracking



Browser fingerprinting heavily relies on JavaScript.

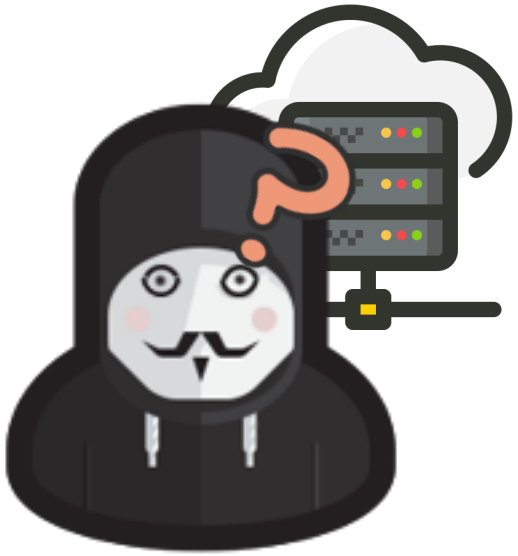
Fingerprinting countermeasures

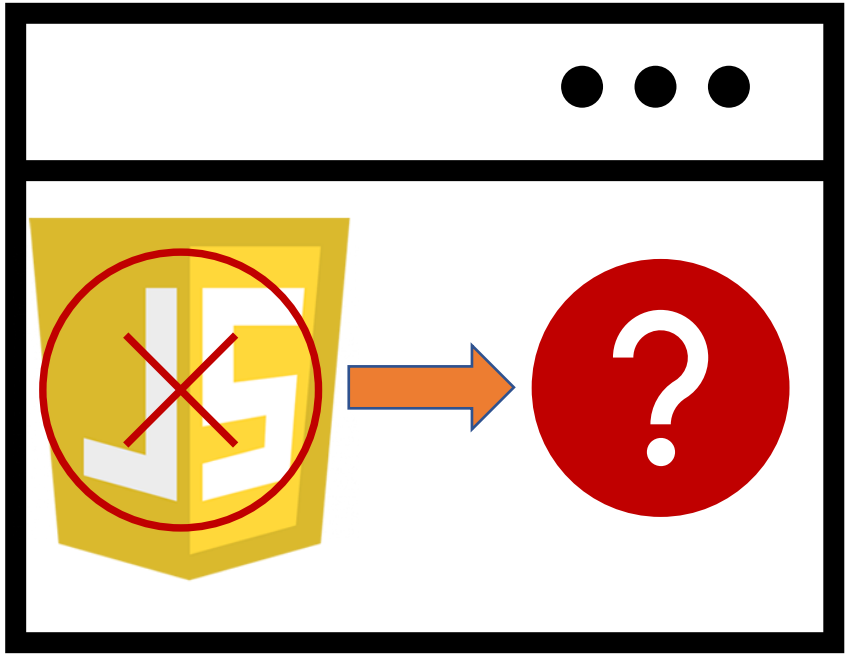


Privacy-focused browsers and anti-fingerprinting extensions

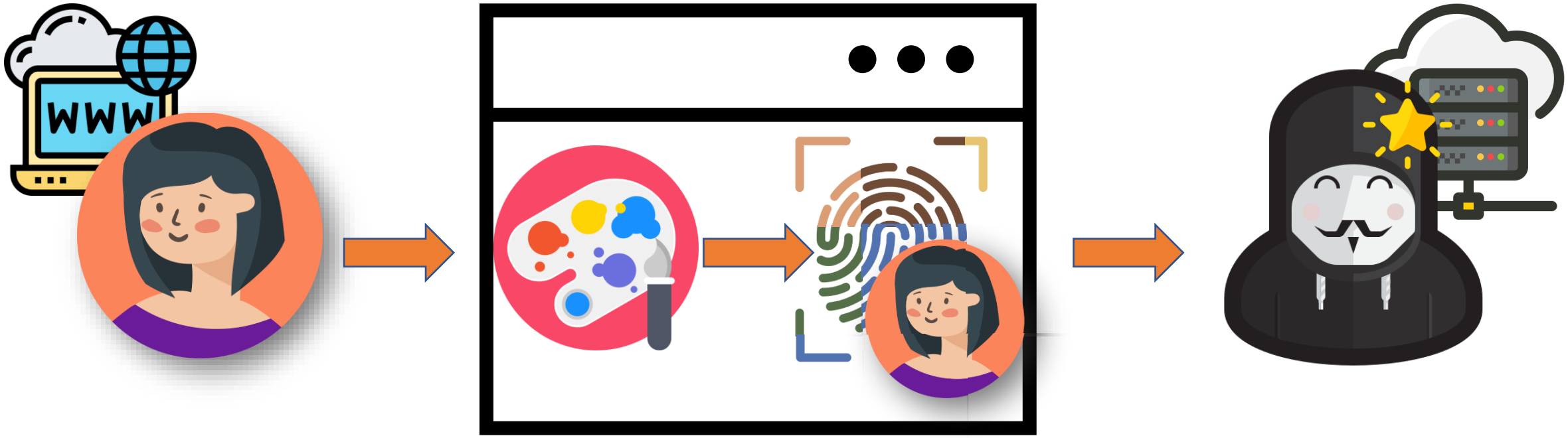
- Spoof certain APIs
- Disable JavaScript (entirely or partially)

Is fingerprinting possible without JavaScript?





Our approach: Implicit stylistic browser fingerprinting



Implicit stylistic browser fingerprinting

- Does not use any JavaScript
- Provides highly discriminating fingerprints

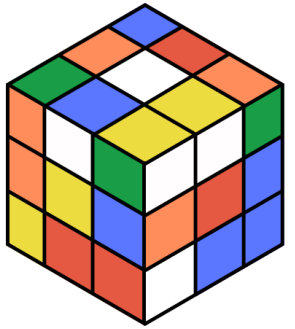
What can we use to detect the stylistic differences?

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

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

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

Dimensions!



Fingerprinting attributes

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

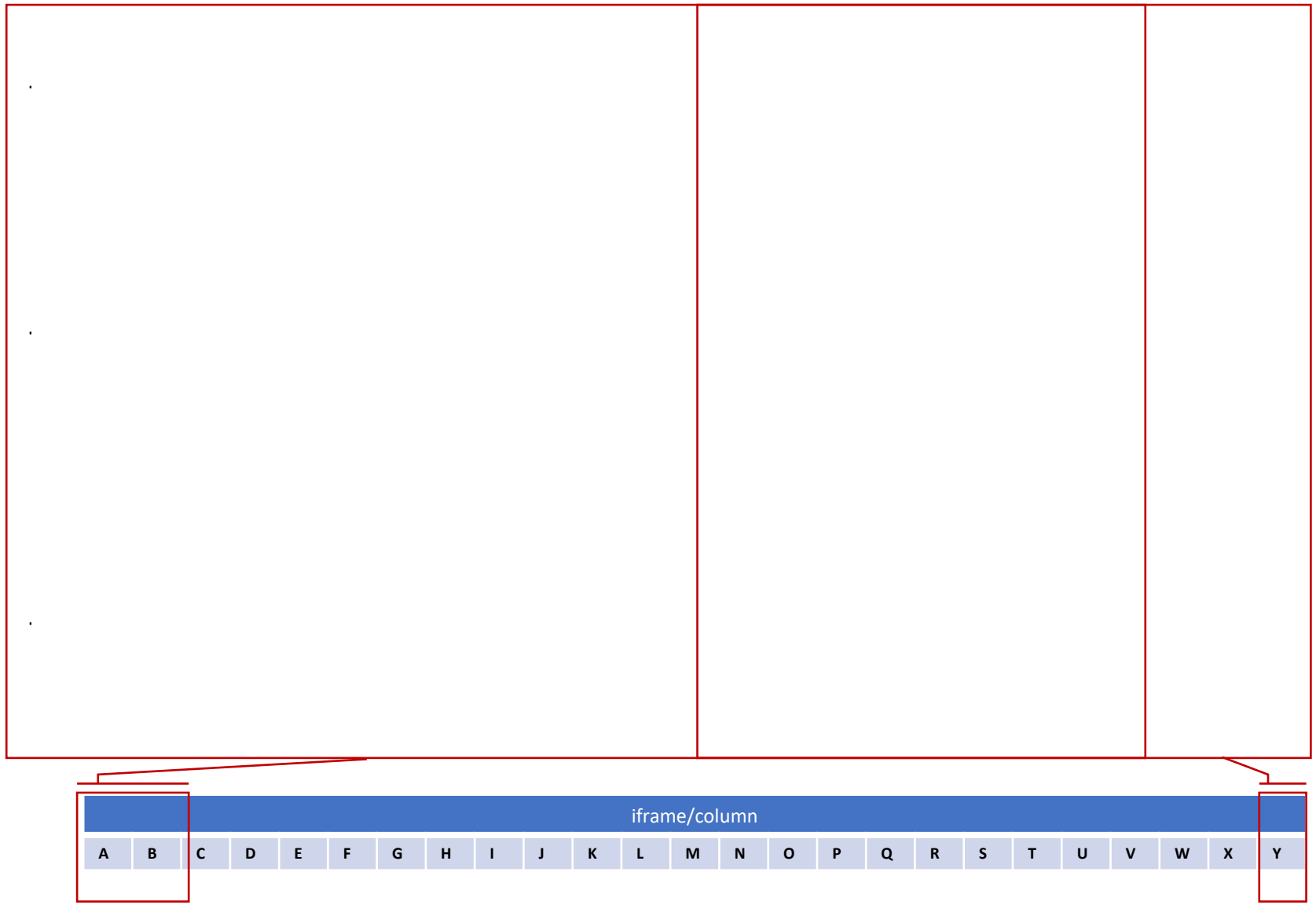
339
Fingerprinting Elements

dimensions

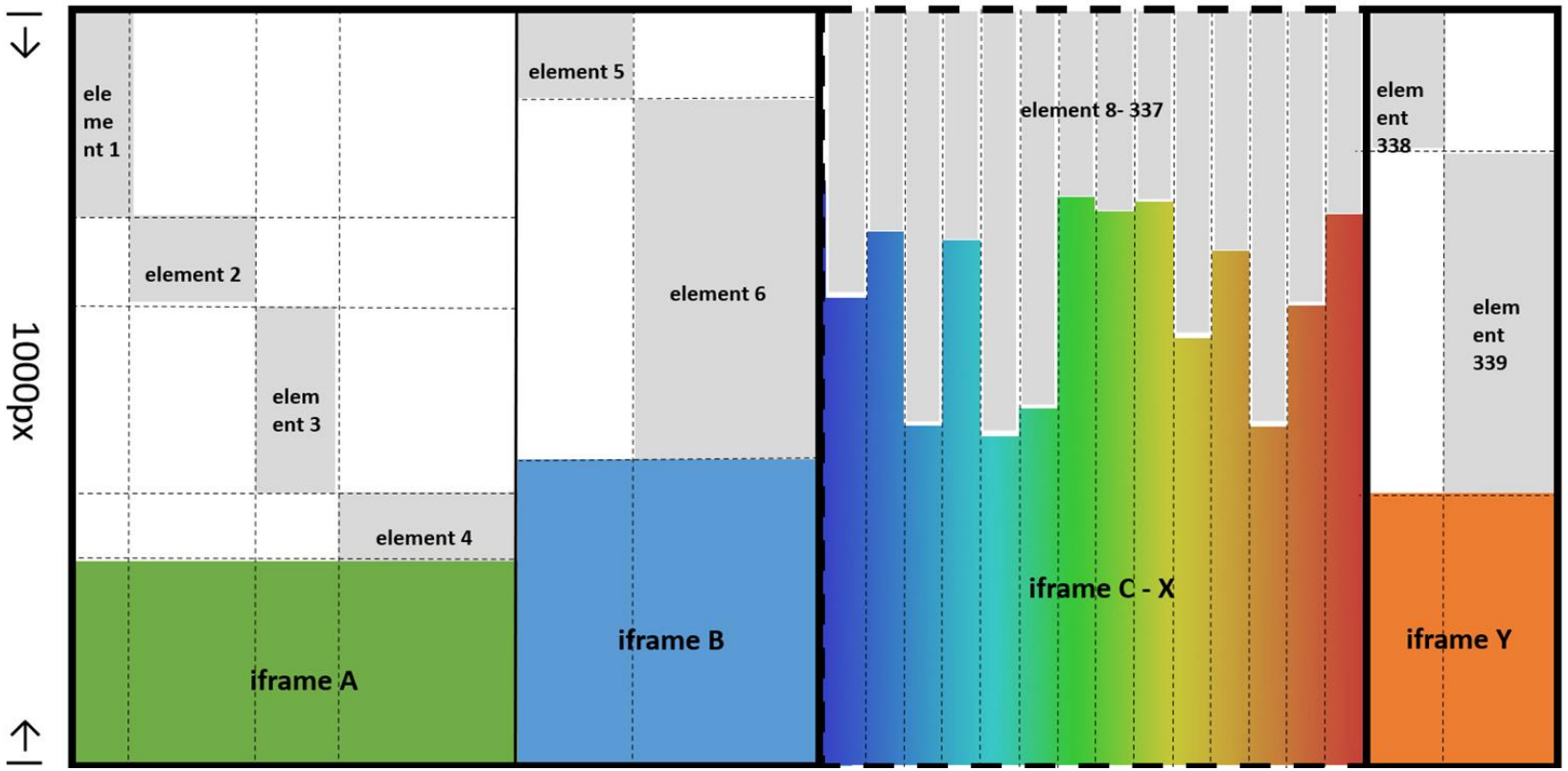
Category	Fingerprint attributes	AIU	FPJS
Environment	browser	●	●
	browser major version	●	●
	operating system	●	●
	platform	◐	◐
	operating system language		
	scrollbar settings		
	JS disabled		
Fonts	font preferences		●
	supported fonts	●	●
	supported shadow fonts		
Ad blocker	presence of ad blocker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

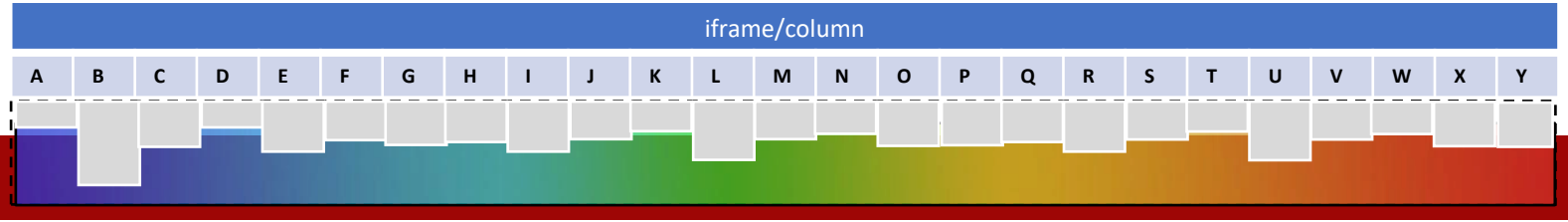
AIU: captured by AmIUnique FPJS: captured by FingerprintJS
◐ : partial feature support ●: full feature support

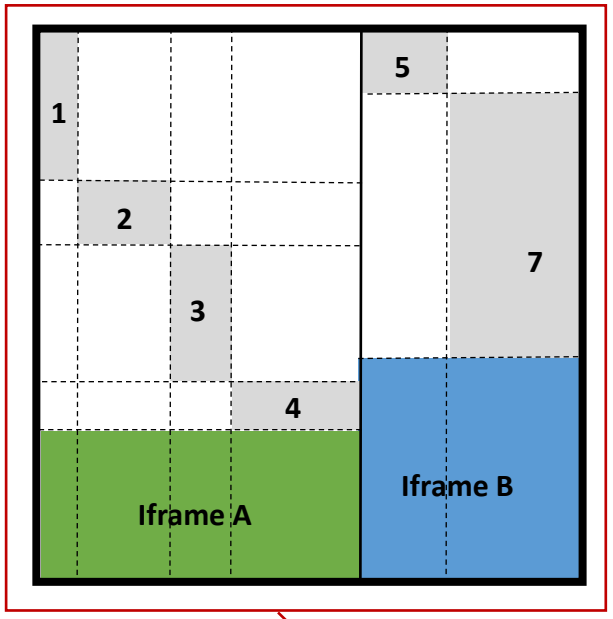
- The page only needs **25** iframes.
- All elements are placed in an 800px by 1000px iframe (main iframe) to ensure that their dimensions remain consistent across different browser window sizes.

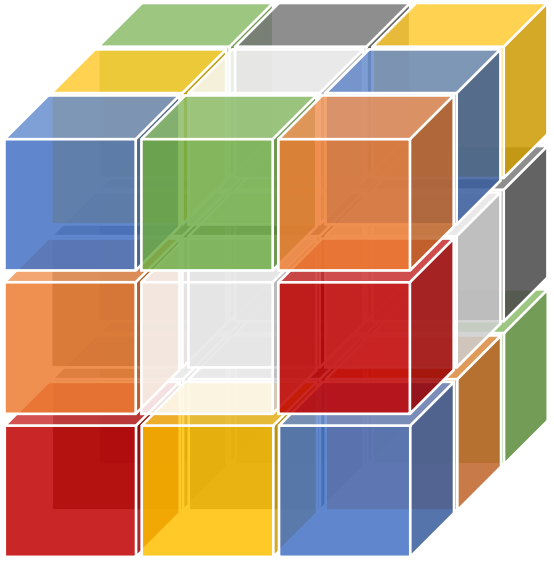


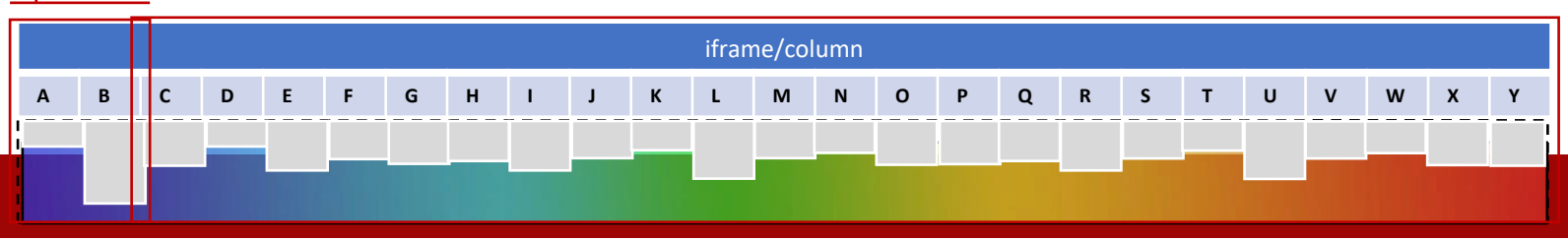
Main iframe


1000px
element 1
element 2
element 3
element 4
element 5
element 6
element 8- 337
element 338
element 339
iframe A
iframe B
iframe C - X
iframe Y


iframe/column
A B C D E F G H I J K L M N O P Q R S T U V W X Y


1
2
3
4
5
7
Iframe A
Iframe B




iframe/column
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y

Other Tracking Vectors: Using Extensions

WASHINGTON STATE UNIVERSITY

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

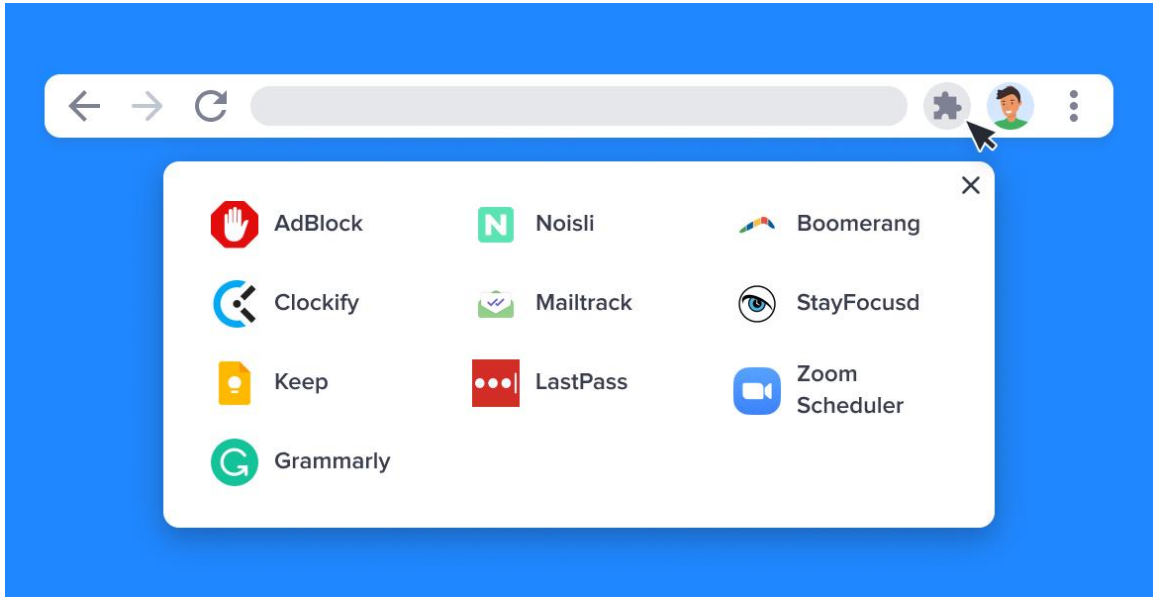
ARTIFACT EVALUATED usenix ASSOCIATION PASSED

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

WASHINGTON STATE UNIVERSITY

Tales of FAVICONS and Caches: Persistent Tracking in Modern Browsers

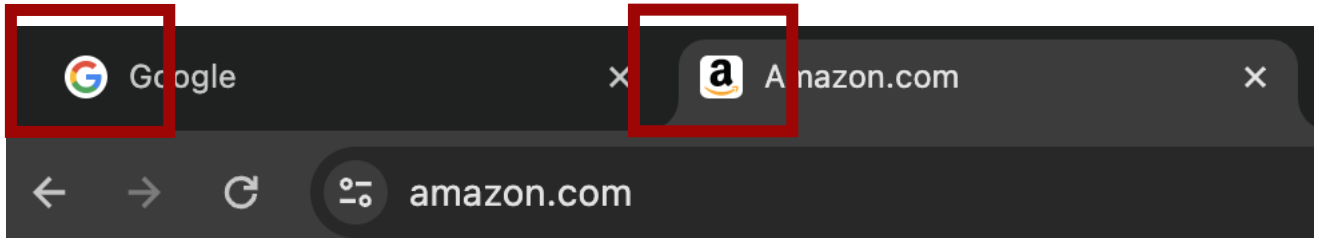
Konstantinos Solomos, John Kristoff, Chris Kanich, Jason Polakis
University of Illinois at Chicago
{ksolom6, jkrist3, ckanich, polakis}@uic.edu

***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 subdomain. 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**

I. Introduction

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."

2015

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

If you're interested in breaking and fixing on the Web, please contact me at: xu.lin@wsu.edu