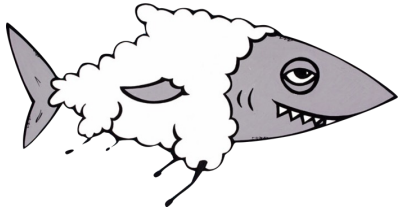
W3C TPAC 2022



# Phish in Sheep's Clothing: Exploring the Authentication Pitfalls of Browser Fingerprinting

**Xu Lin**, Panagiotis Ilia, Saumya Solanki, Jason Polakis
University of Illinois Chicago, USA

xlin48@uic.edu

usenix
THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

UIC

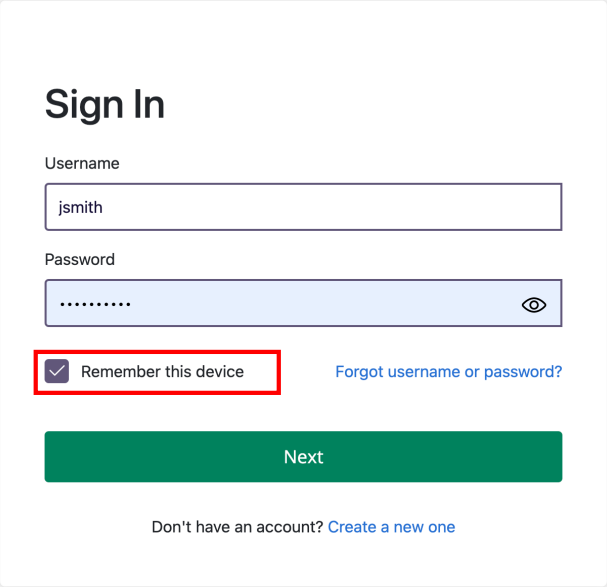
September, 2022

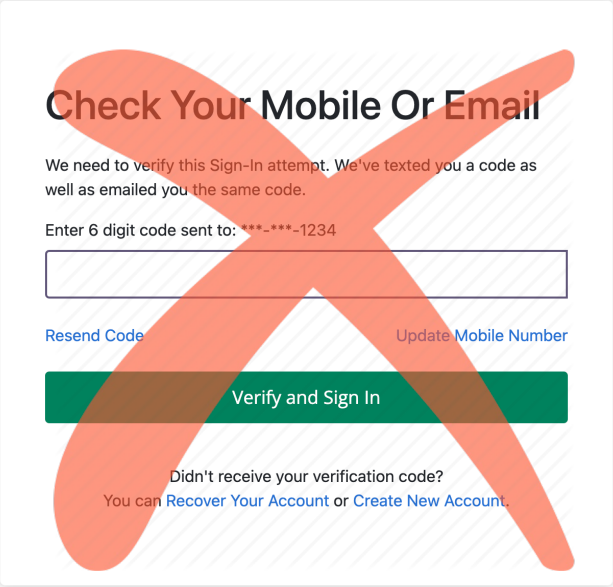
# Introduction

- Web integral to many facets of everyday life
  - User accounts contain sensitive and valuable data
- Account hijacking remains a major problem
- Phishing is a prevalent hijacking vector [1,2]
- Two-factor authentication (2FA) is a *critical* defense
  - Device-based challenges block >94% of phishing-based hijacking attempts, 100% of automated hijacking attempts [3]


[1] Bursztein et al. "Handcrafted fraud and extortion: Manual account hijacking in the wild." *IMC '14*.
[2] Thomas et al. "Data breaches, phishing, or malware? understanding the risks of stolen credentials." *CCS '17*
[3] Doerfler et al., "Evaluating login challenges as a defense against account takeover. " *WWW '19*

# Risk-based authentication and two-factor authentication (2FA)



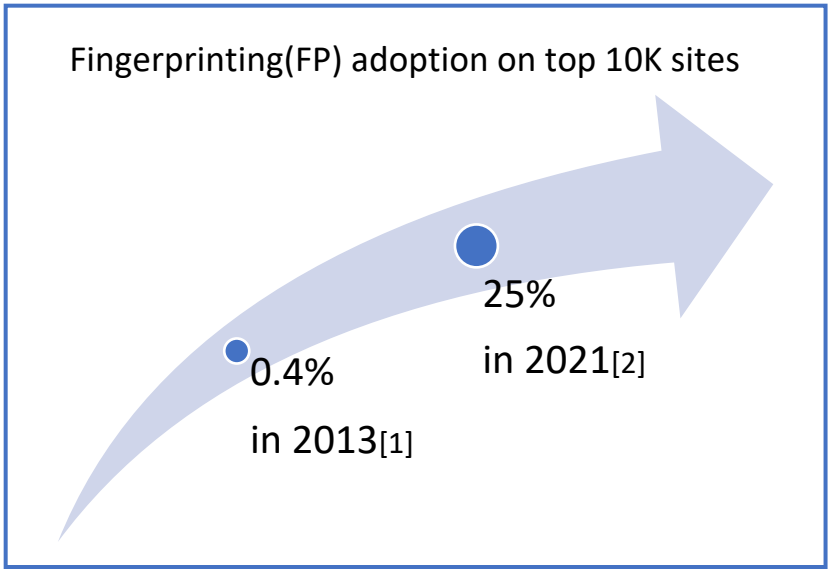




- 2FA creates friction for users
- Certain websites only trigger 2FA for *suspicious* login attempts

# Browser fingerprint

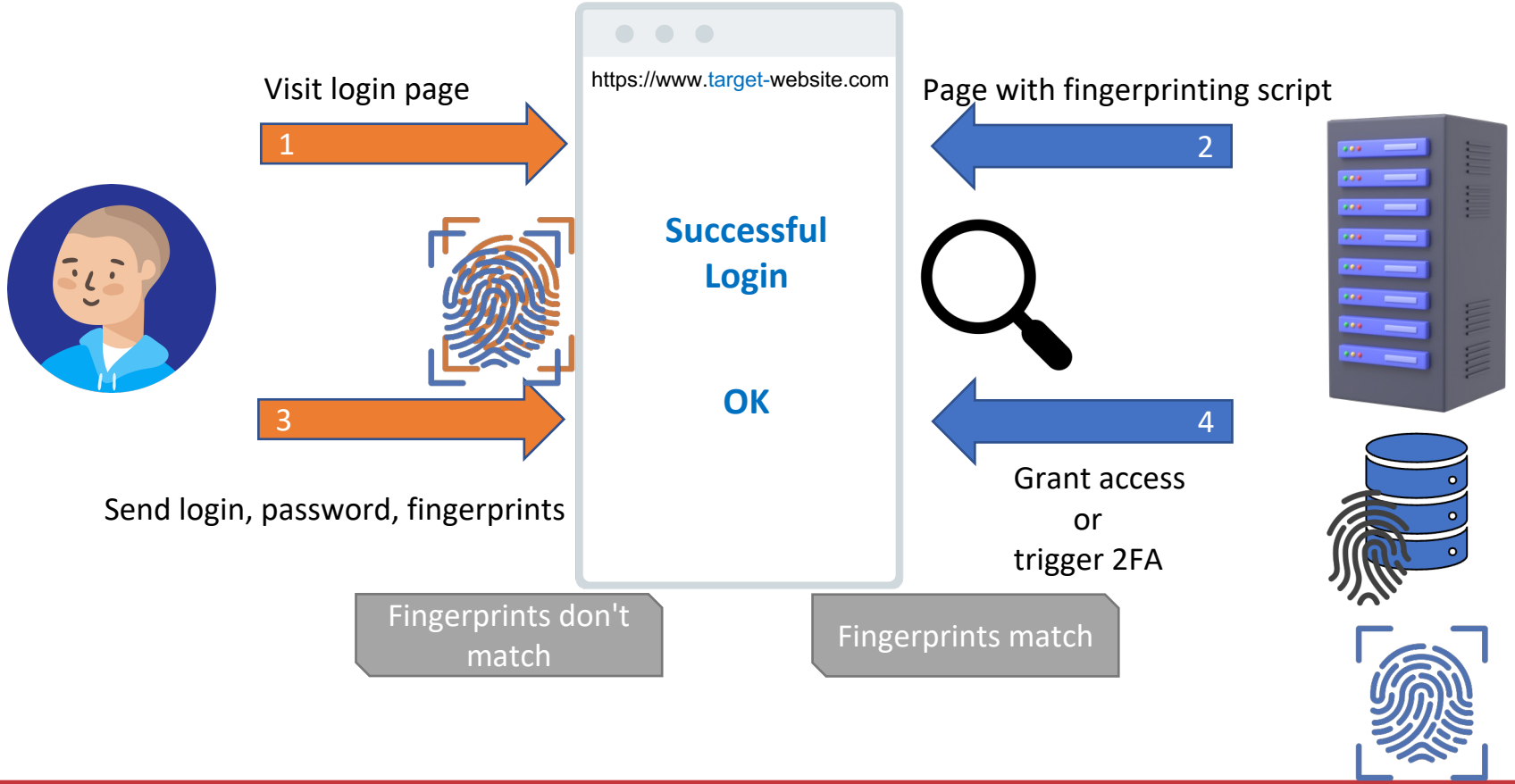
Browser fingerprint

- information collected about a device for the purpose of identification
- can be trivially collected by ***any*** website through a series of **JavaScript APIs**

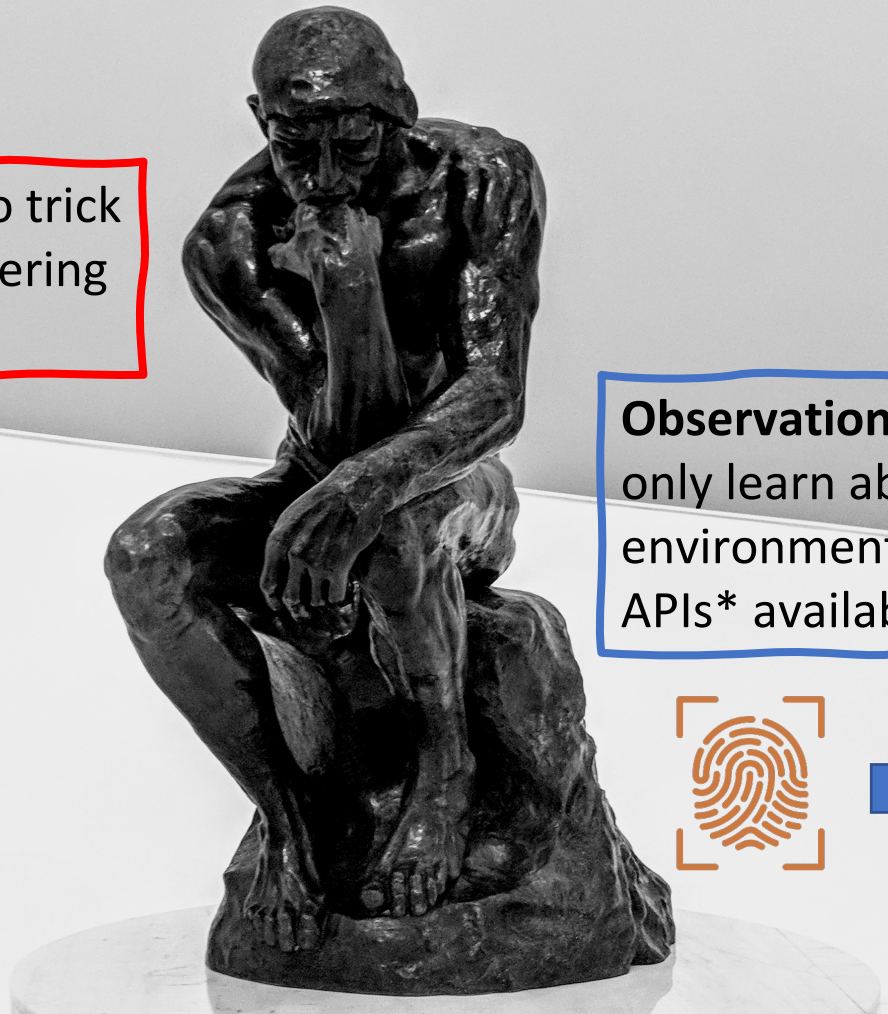



[1]N. Nikiforakis et al. " Cookieless monster: Exploring the ecosystem of web-based device fingerprinting, " S&P '13.
[2]U. Iqbal et al. " Fingerprinting the fingerprinters: Learning to detect browser fingerprinting behaviors, " S&P '21.

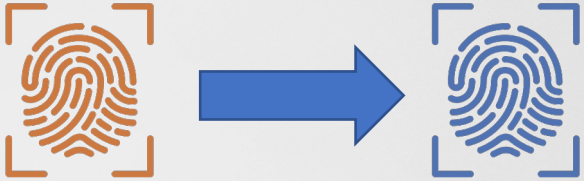
# Advanced risk-based authentication that uses browser fingerprinting



What can attackers do to trick websites into **not** considering a login suspicious?



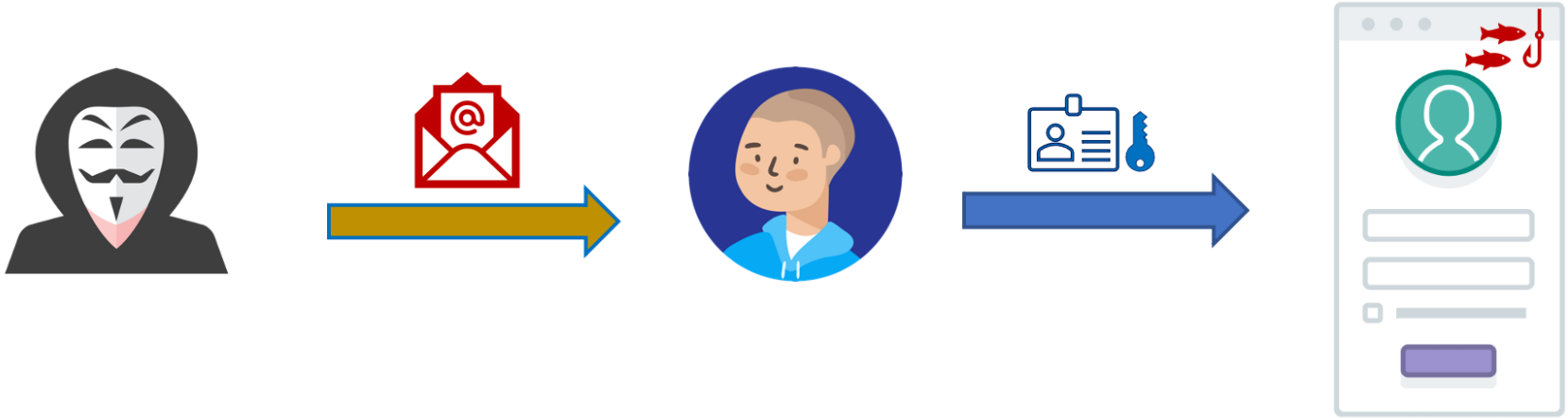
**Observation**: websites can only learn about the user's environment through browser APIs* available to any website.



* And HTTP headers, which can be easily spoofed.

# Threat Model

The attacker tricks the user into visiting a malicious website and entering their credentials.



# Overview of our attack workflow

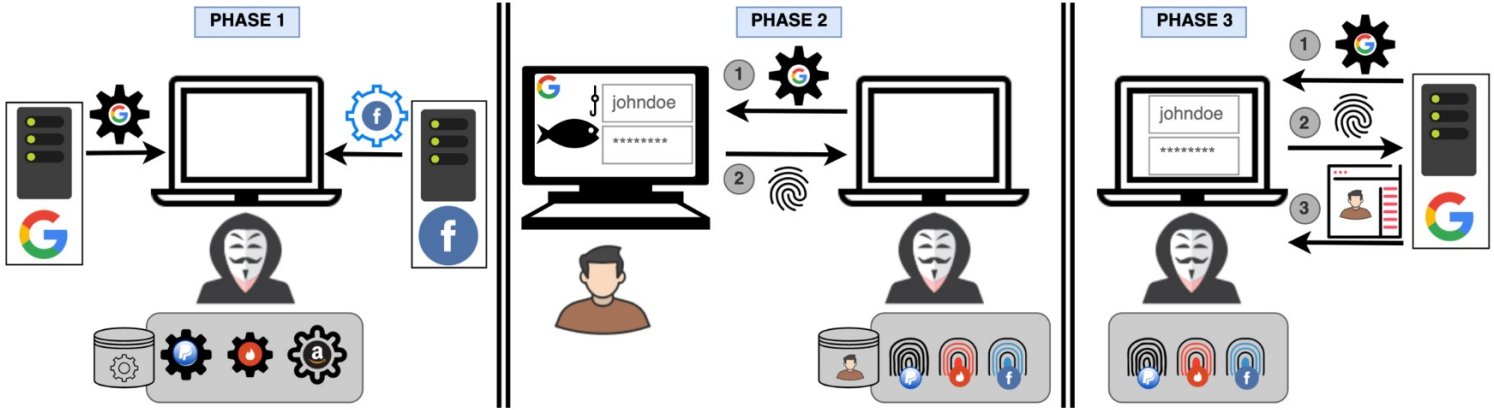
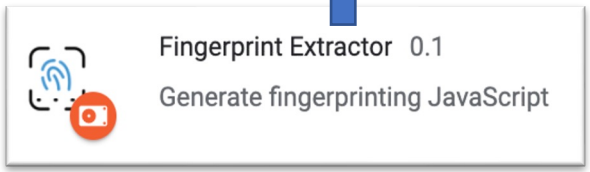
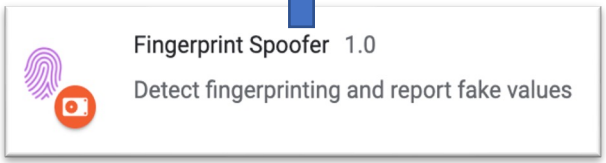



Figure 1: Overview of our attack workflow that misuses browser fingerprints for bypassing ancillary security checks.



Fingerprint Extractor 0.1

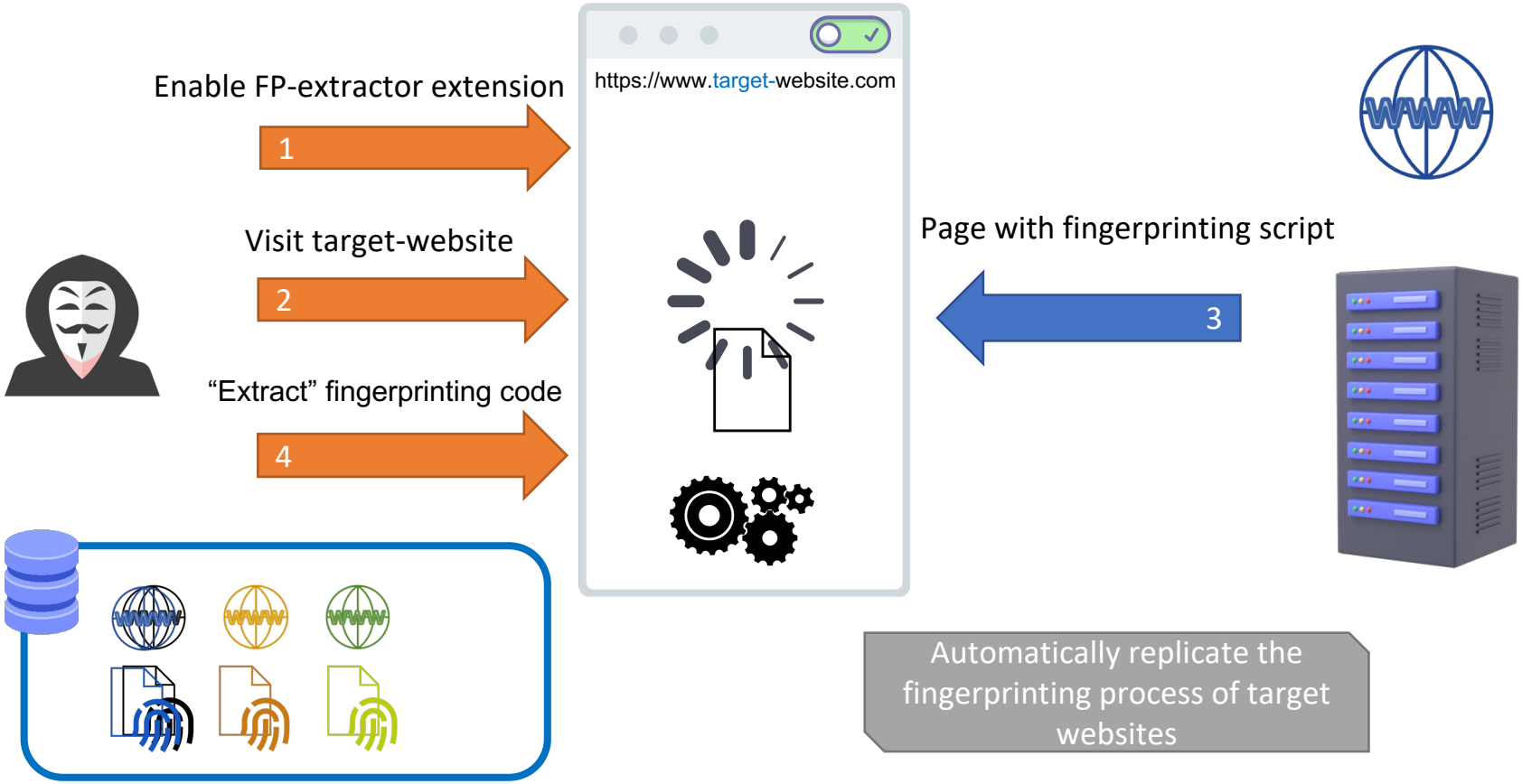
Generate fingerprinting JavaScript



Fingerprint Spoofer 1.0

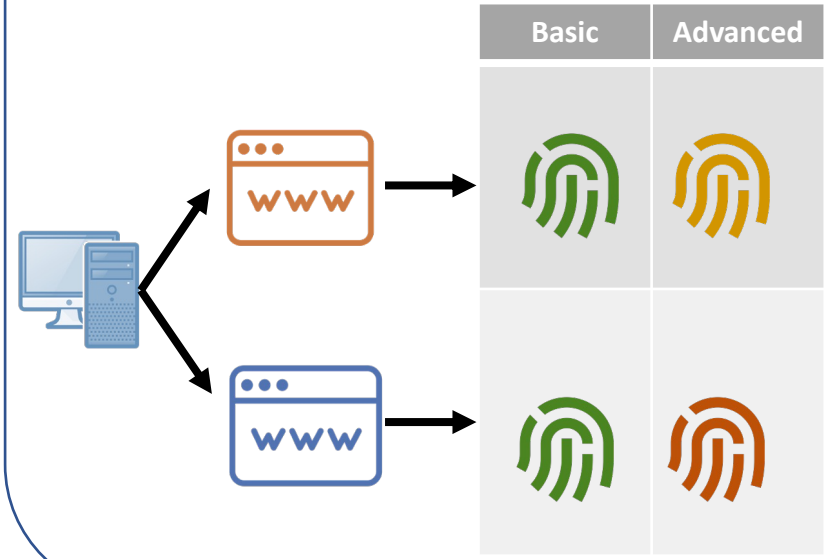
Detect fingerprinting and report fake values

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



# Replicate target website's *exact* fingerprint-generation

**Same** device has **different** fingerprints across websites




- **Basic** fingerprints are identical across websites
- **Advanced** fingerprints vary depending on the fingerprint generation
  - Canvas FP: render different images

Image_1 Image_2


  - Font FP: detect different fonts

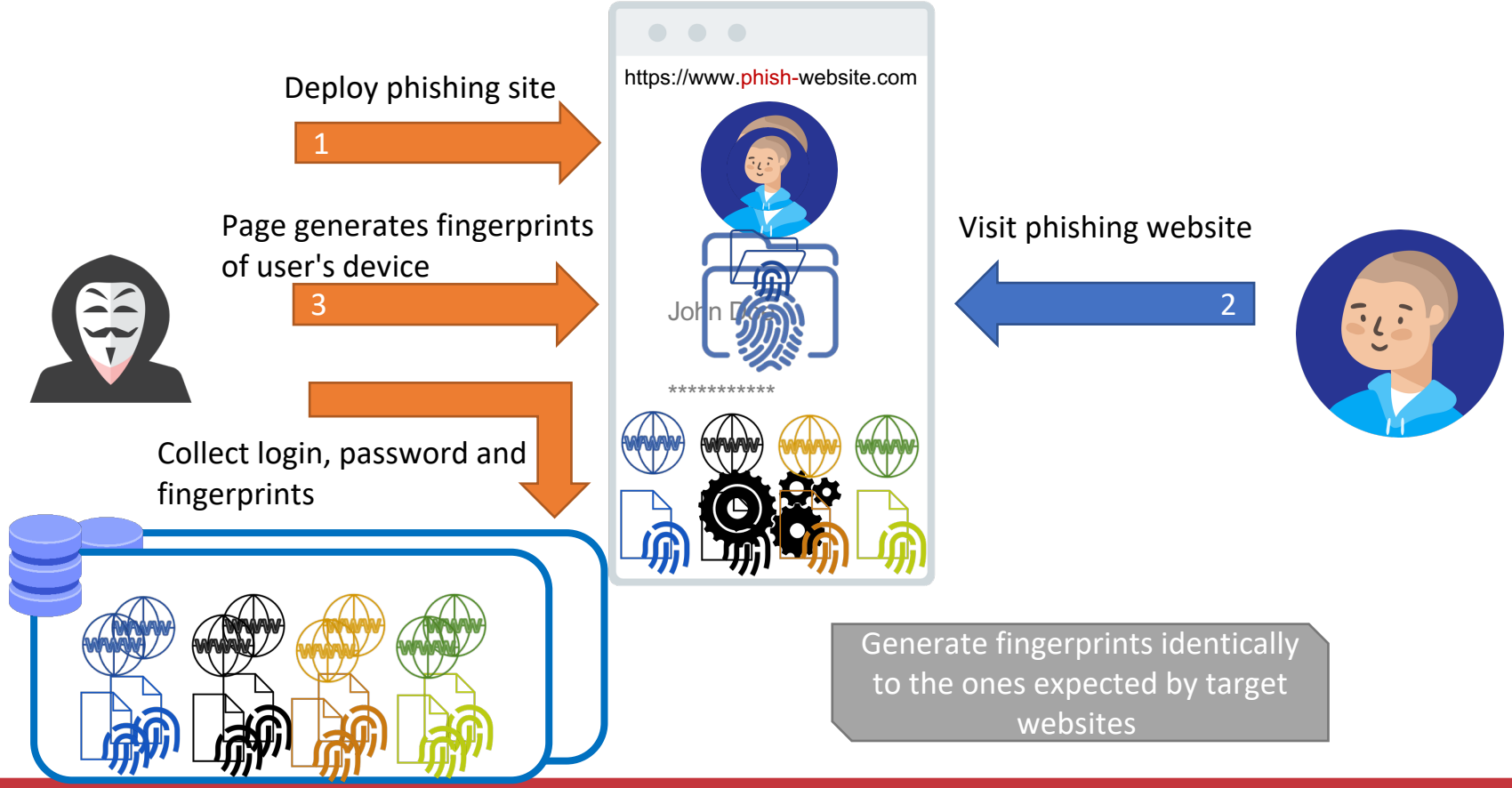
Font Families 1

Font a Font b
Font c Font d

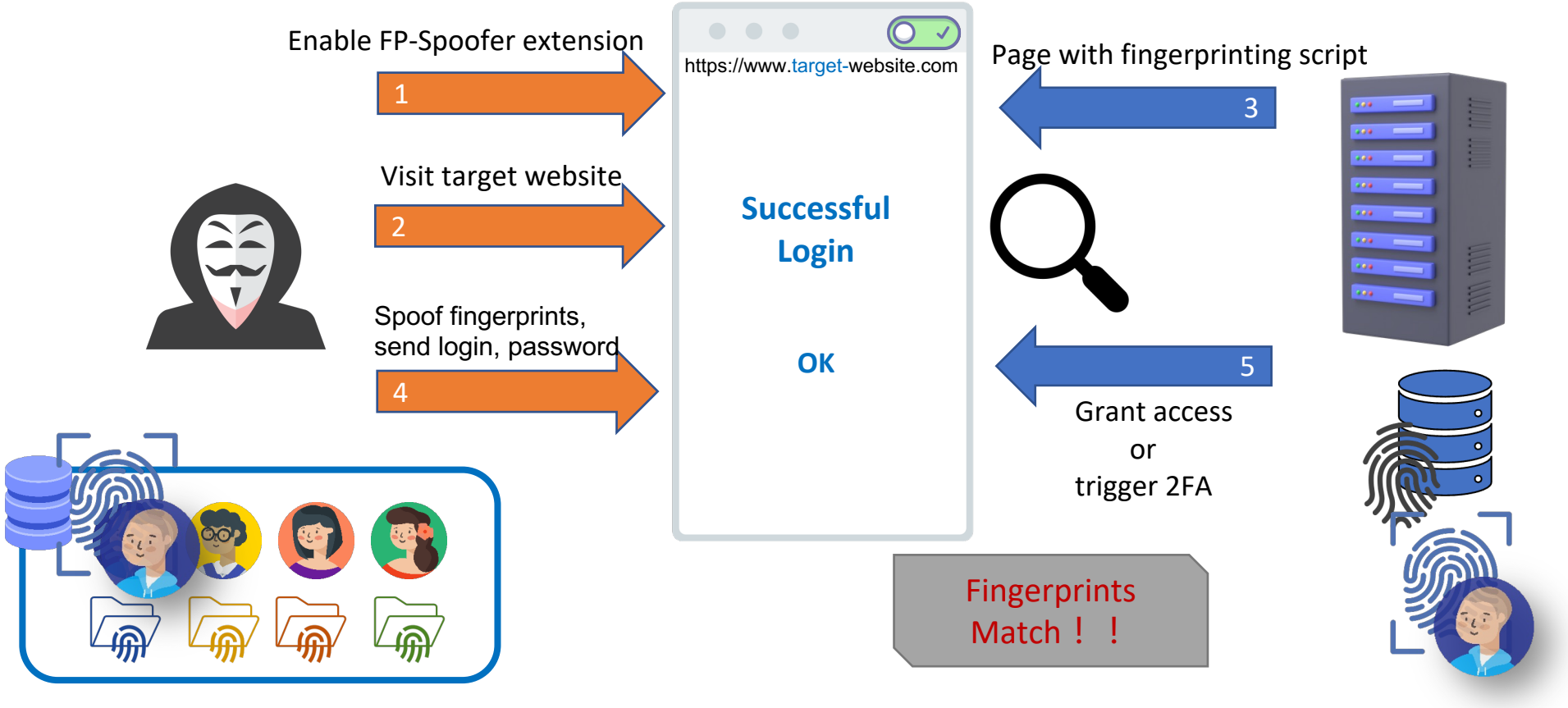
Font Families 2

Font e Font f
FONT G Font h

# Phase2: attacker obtains user's credentials and fingerprints



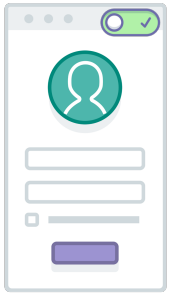
# Phase3: attacker spoofs fingerprints and bypasses 2FA mechanism



# How FP-Extractor Extension Works

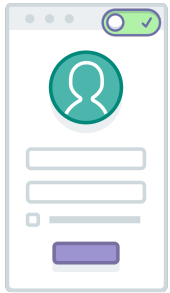
1. Inject code that hooks fingerprinting properties & methods.

```
Object.defineProperty(MediaDevices.prototype, 'enumerateDevices', {
    value: () => {
      fpTrace.push('enumerateDevices');
       return originalPromise;
  }
})
```



target website

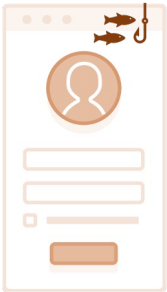
2. Code runs at "*document_start*".
3. keep track of accesses with their arguments.
   - Dynamic FP attributes (e.g., WebGL) can vary across websites.



target website

4. Generate and export JavaScript Code.

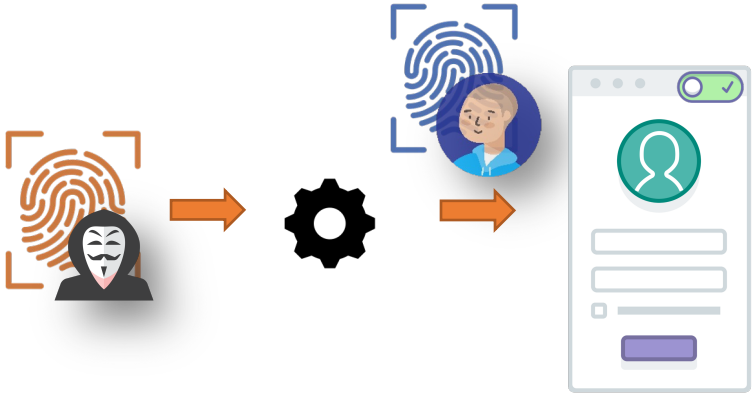
```
if (fpTrace.includes('enumerateDevices') {
   fpCode +=
`navigator.mediaDevices.enumerateDevices().then...`
})
```



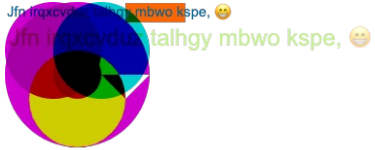
phishing website

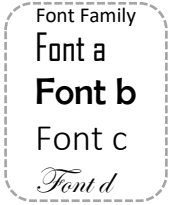
# How FP-Spoofer Extension Works

- Take victim's fingerprints as input
- Hook fingerprinting APIs
- Override/delete/add values to match the victim's values



- For advanced FPs
  - No need to manipulate intermediate values
  - Only spoof the **final** values, e.g., toDataURL for **Canvas**, offsetWidth and offsetHeight for **Fonts**




```
Object.defineProperty(HTMLSpanElement.prototype,
"offsetWidth", {
  get: function(){
    if (isSupportedFont) {
      return customWidth;
    } else {
      return fallbackFontWidth;
    }
  }
})
```

# Fingerprint Spoofing Demo

attacker spoofs their device's fingerprints to mimic those of the victim's device


What about the real world ?

# Experimental Evaluation

| | Top 10K | | Top 10K-20K | |
|---|---|---|---|---|
| **Website** | **Home** | **Login** | **Home** | **Login** |
| Navigator | 5,510 | 5,403 | 5,589 | 5,371 |
| Window | 5,261 | 5,104 | 5,272 | 4,968 |
| Screen | 5,209 | 4,682 | 5,231 | 4,473 |
| Timezone | 5,035 | 4,617 | 4,934 | 4,282 |
| Canvas | 1,224 | 1,254 | 1,077 | 879 |
| Canvas Fonts | 179 | 380 | 142 | 237 |
| WebRTC | 221 | 313 | 192 | 210 |
| AudioContext | 290 | 351 | 223 | 234 |

- Crawled Alexa top 20K
  - Logged FP APIs being used
- ➢ Top sites employ more advanced fingerprinting techniques on login pages vs home pages

- ➢ Select 300 popular sites that implement FP and support 2FA for manual analysis
  - ➢ 14 use fingerprints for remembering user's device
    - ➢ More prevalent among high-value financial services!
    - ➢ Risk-based authentication + FPs = emerging trend

# 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 | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ |
| AdBlocking | ✓ | ✗ | ✗ | ✗ | ✓ | ○ | ✓ |
| WebInfrastructure | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ |

no email alert

FP-checks for stolen cookies

- **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 (○).

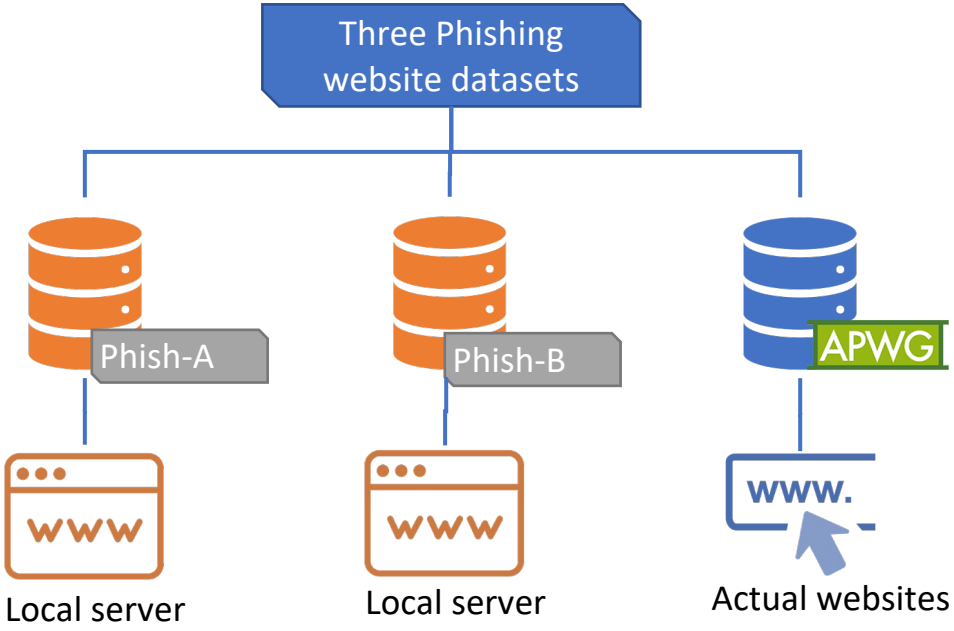
# Evade fingerprint spoofing detection

| Inconsistency checks | Spoof them all |
|---|---|
| e.g., userAgent and platform | |
| **"reflection"** | **Override "reflector"** |
| e.g., toString() | |

| | | |
|---|---|---|
| Native *toDataURL* | 'function toDataURL() { [native code] }'; | Object.defineProperty(Function.prototype, 'toString', {<br>value: () => {<br>return **'function toDataURL() { [native code] }'**;<br>} |
| Tampered *toDataURL* | 'function() {<br>return fakeImageData;<br>} ' | |

# What about phishing sites in the wild?

# Phishing website datasets




Zhang, Penghui, et al. "Crawlphish: Large-scale analysis of client-side cloaking techniques in phishing." *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2021.

Use VisibleV8 to log native functions and property accesses

Jueckstock, Jordan, and Alexandros Kapravelos. "Visiblev8: In-browser monitoring of javascript in the wild." *Proceedings of the Internet Measurement Conference*. 2019

# Phishing and Fingerprinting

| Dataset | Time Period | Sites | JS | FP |
|---|---|---|---|---|
| Phish-A | 31/05/2018 – 19/06/2019 | 71,343 | 39,618 | 29,312 |
| Phish-B | 31/10/2018 – 05/05/2020 | 82,431 | 40,777 | 36,733 |
| APWG | 05/05/2020 – 12/04/2021 | 173,269 | 93,568 | 85,491 |

- Broad and representative view of the phishing ecosystem over a **3-year** period.
- The majority collect fingerprints, with **73.98%**, **90.08%** and **91.36%** across the 3 datasets respectively.
- An **increase** in the number of websites collecting browser fingerprints over time.

# Phishing sites that implement fingerprinting techniques

- Phishing sites aggressively collect FPs
- Upward trend in most categories
  - **72.00%**, **87.43%** and **91.34%** collect basic fingerprints
- Even advanced FPs being collected
  - between **9%** and **14%** collect advanced fingerprints

| Technique | Phishing Datasets | | |
|---|---|---|---|
| | **Phish-A** | **Phish-B** | **APWG** |
| Navigator | 27,578 | 34,650 | 84,239 |
| Window | 24,848 | 23,650 | 73,258 |
| Screen | 10,244 | 26,856 | 57,633 |
| Timezone | 22,636 | 28,549 | 59,251 |
| Canvas | 3,508 | 5,395 | 11,650 |
| Canvas Fonts | 56 | 91 | 399 |
| WebRTC | 536 | 165 | 1,938 |
| AudioContext | 275 | 363 | 1,795 |

# Phishing sites that obtain all necessary fingerprints for bypassing 2FA

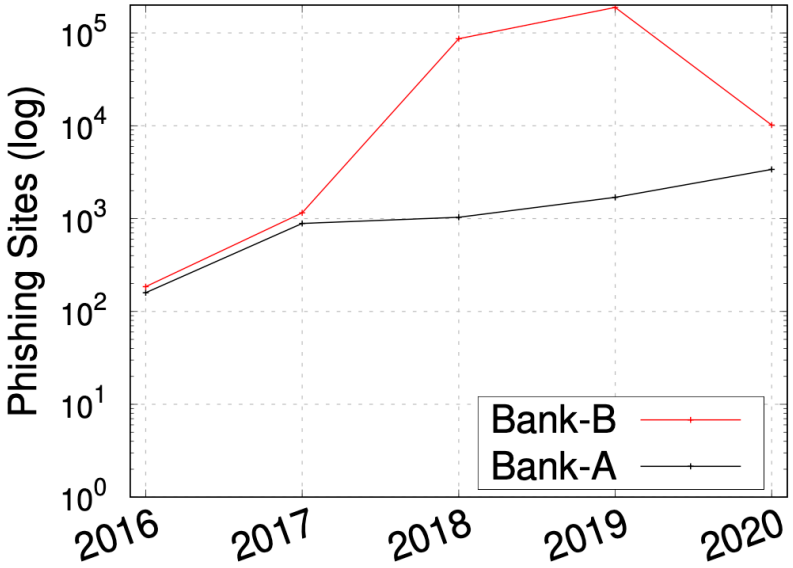
| Target | Phish-A | | Phish-B | | APWG | |
|---|---|---|---|---|---|---|
| | Sites | Bypass | Sites | Bypass | Sites | Bypass |
| Bank-A | 83 | 1 | 685 | 14 | 330 | 74 |
| Bank-B | 1549 | - | 2,683 | - | 327 | - |
| CreditCard | 89 | 61 | 0 | 0 | 12 | 0 |
| Trading-A | 0 | 0 | 0 | 0 | 6 | 6 |
| RideSharing | 7 | 0 | 363 | 1* | 1378 | 5* |
| WebInfrastructure | 0 | 0 | 1 | 1 | 220 | 219 |

APWG dataset

- more recent
- visited actual websites

* Indicates a mismatch in the arguments passed to fingerprinting functions.

# Are phishers adapting their targets?




- The sharp decline in phishing sites targeting Bank-B could be due to the IP address requirement.

# 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

# Summary

- First *fully automated* system for *replicating* and *replaying* fingerprints
- First *empirical analysis* of the use of browser fingerprinting for augmenting web authentication in the wild
- Practical attacks that *completely bypass 2FA* in high-value services
- A *large-scale study* on the use of browser fingerprinting techniques by phishing sites
- *Disclosure* of findings to affected vendors


# Questions?

Feel free to reach out
xlin48@uic.edu