Escaping the Confines of Time: Continuous Browser Extension Fingerprinting Through Ephemeral Modifications

Konstantinos Solomos, Panagiotis Ilia, Nick Nikiforakis, Jason Polakis

ksolom6@uic.edu

acm SIGSAC

Stony Brook University

UIC

Browser Extensions

3 Million users

10 Million users

1 Million users

10 Million users

1 Million users

10 Million users

1 Million users

Fingerprinting Browser Extensions

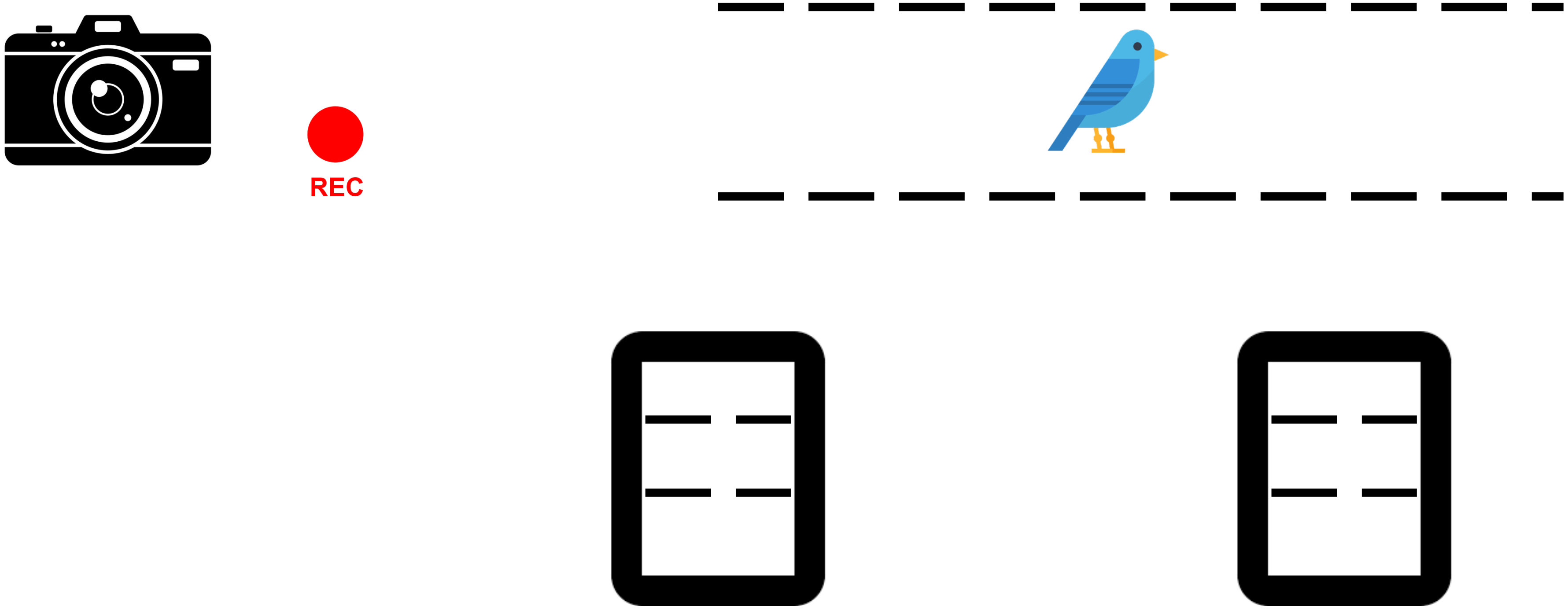
- Privacy invasive websites detect extensions
    - Track and target the device and the user
    - No permissions
    - Reveal personal-sensitive information
- Extension-fingerprinting is becoming mainstream
    - FingerprintJS framework
    - Device authentication & identification
    - Bot prevention

Fingerprinting Browser Extensions

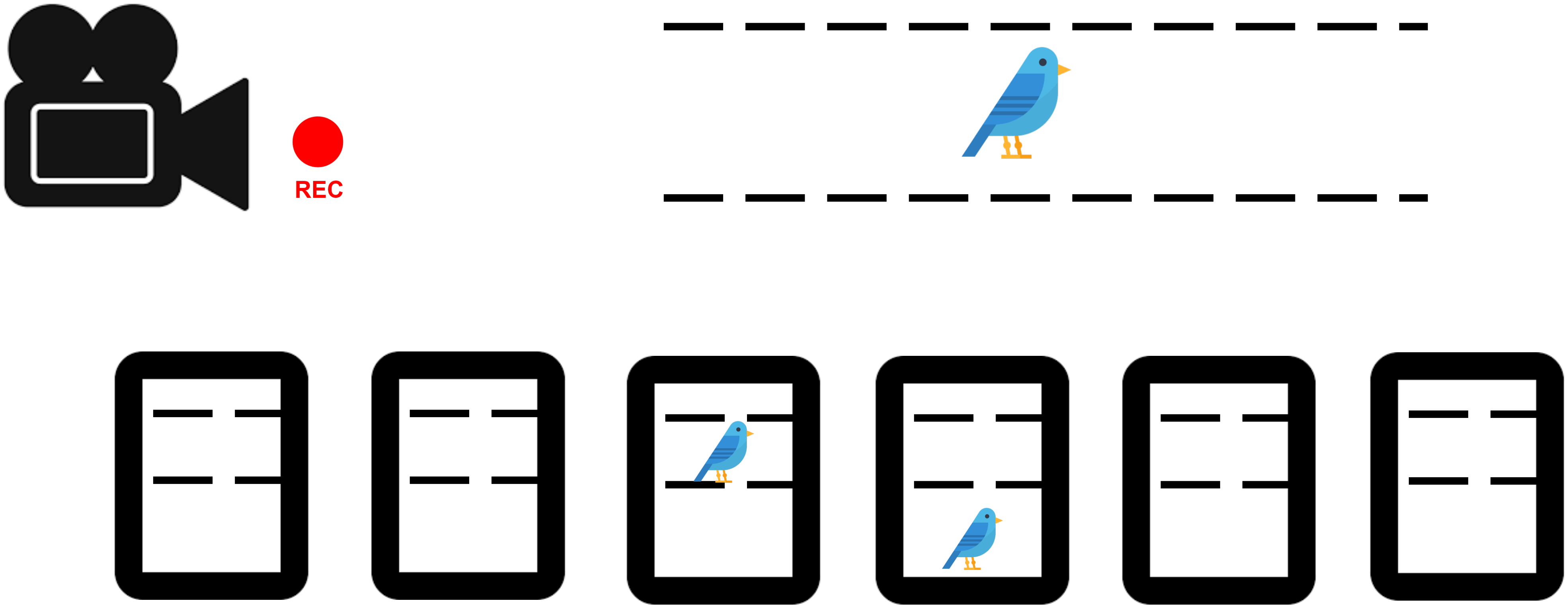
- Side channel inference techniques
  - Web Accessible Resources (Sjosten et al. CODASPY ‘17)
  - Style Modifications (Laperdrix et al. USENIX Security ‘21)
  - Behavioral fingerprints (Starov & Nikiforakis IEEE S&P ‘17, Karami et al. NDSS ’20)
  - User Interactions (Solomos et al. USENIX Security ‘22)

- Limitations
  - Analyze only a single snapshot
  - Ignore the extension’s execution life cycle

Snapshot vs Continuous Recording



Snapshot vs Continuous Recording



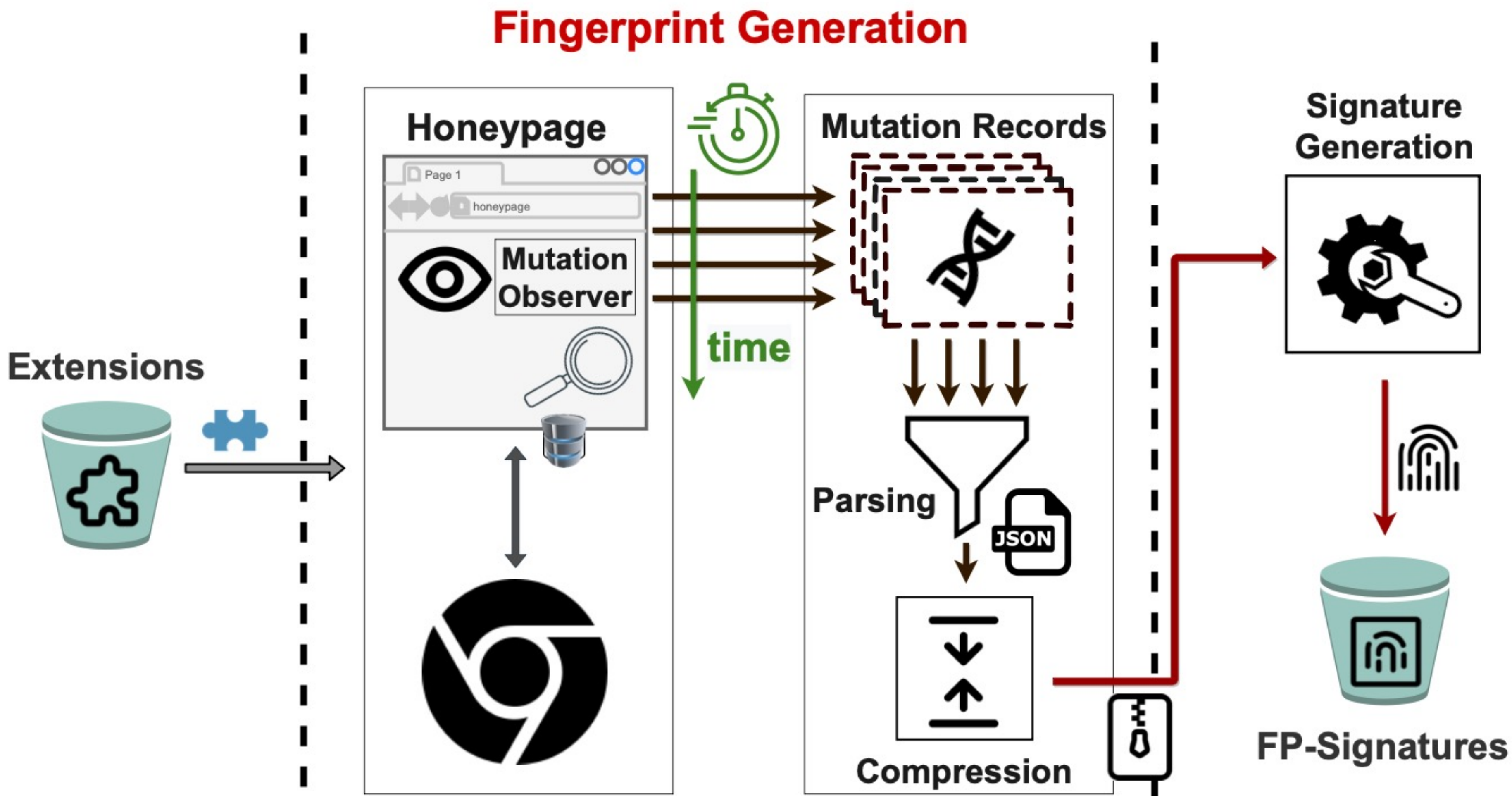
Our Work

- Propose continuous extension fingerprinting to overcome the time-based limitations of prior works

- Develop a system (*Chronos*) to collect **all** the changes that the extensions introduce

- Explore multiple aspects of continuous fingerprinting and compare with the state-of-the-art techniques

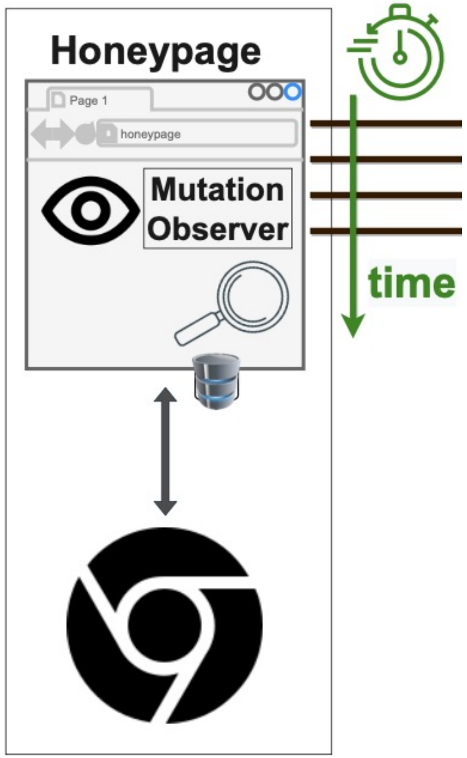
Chronos: Continuous Fingerprinting



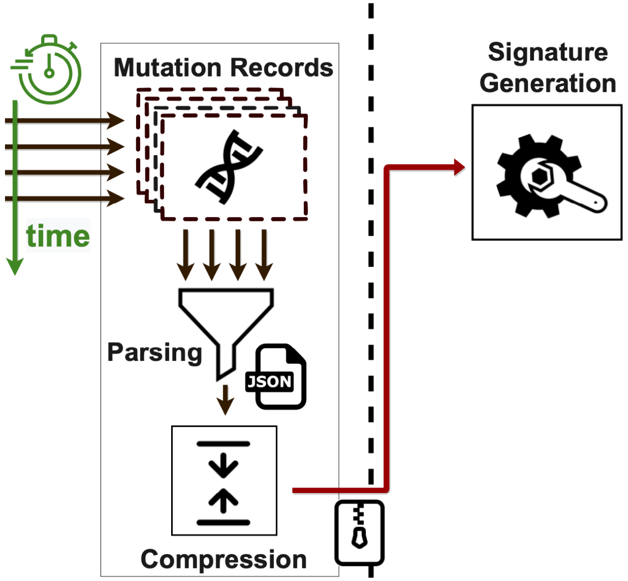
Detecting DOM-Based Modifications

- Mutation Observer Interface
  - Monitors DOM continuously for alterations
  - Asynchronous trigger when modification is detected
  - Mutation record types
    - ChildList : added & removed elements
    - Attributes : alteration of existing element's attributes

- Honey Page for extension exercising
  - Adopted by Carnus [Karami et al. NDSS '20]
  - Record modification information through Mutation Observer



Fingerprint Generation & Collection



- Extract information from Mutation Records
  - Mutation target
    - `head, body, element`
  - OuterHTML
    - `<h1 id=“foo">bar</h1>`
- Replace the dynamic and unstable parts of the record
  - `{cdn.com/content.js?rand=1234}` → `{cdn.com/content.js?rand=ID}`
- Each fingerprint contains a set of unique and shared mutation records

Experimental Evaluation

- 2 Datasets [2018-2021]
  - 35K extensions
  - Fingerprinted : **11,219** (31%)

- Overview
  - Increased coverage by **67%** over the state of the art (Carnus)
  - 40% of extensions perform **ephemeral modifications** only visible to our system

Signature Characteristics

- Signature stability
  - 99.5% same number of mutation records across runs
  - 94% with at least one **unique** mutation record
- 80% signatures < 20 records
  - Deterministic modifications
  - Size < 1.5 KB

➢ **Efficient fingerprint generation with low network and storage demands**

Multi-Extension Fingerprinting

- Distinguish between multiple installed extensions of the same browser
  - Evaluate the fingerprint matching algorithm
  - Randomly install a set of extensions (N=2..10)
  - Repeat 100 times

- Accuracy & Performance
  - Detected 98% of installed extensions
  - No misclassifications (False Positives)
  - Execution time 1.5 second

Countermeasure Effects

- CloakX [Trickel et al. USENIX Security ‘19]
  - Randomizes the values of ID, class and WAR paths
  - Injects random tags and attributes into the page
  - No major effect on our signatures
    - 92% signatures with unique mutation records

- Simulacrum [Karami et al. USENIX Security ‘22]
  - Intercept JS APIs and separates DOM
  - Impacts our system’s efficiency and efficacy

Countermeasure Effects

- CloakX [Trickel et al. USENIX Security ‘19]
  - Randomizes the values of ID, class and WAR paths

Our work highlights the importance of browsers adopting extension-fingerprinting defenses

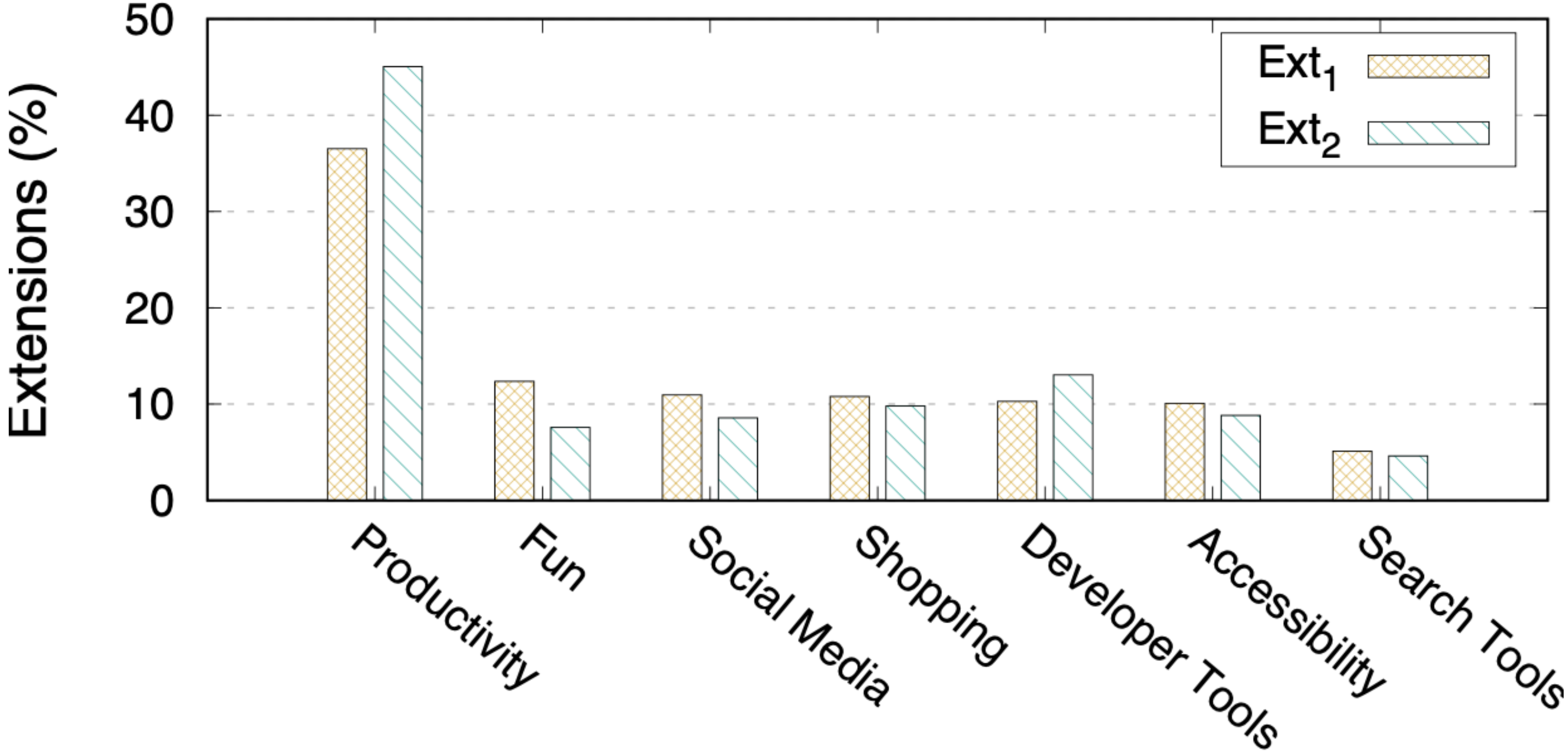
- Simulacrum [Karami et al. USENIX Security ‘22]
  - Intercept JS APIs and separates DOM
  - Impacts our system’s efficiency and efficacy

Conclusion

- Novel continuous fingerprinting strategy that significantly augmented extension fingerprinting frameworks
- Experimental evaluation revealed thousands of non detectable extensions
- Demonstrated that our fine-grained approach is highly accurate in realistic deployments
- Evaluated state-of-the-art countermeasures and highlighted the need for additional privacy protections

Thank you!
Feel free to reach out with any questions:
ksolom6@uic.edu

Extension Categorization


Extensions (%)
Ext$_1$
Ext$_2$
Productivity
Fun
Social Media
Shopping
Developer Tools
Accessibility
Search Tools