



# XS Leaks

**DD2525 Language-based Security**

Marco Campione, Tom Sorger

# XS Leaks Background

*“Cross-site leaks are a class of vulnerabilities derived from side-channels built into the web platform.”*

## History:

- In the mid-2010s, security researchers began systematically identifying and categorizing various XS-leaks.

## Root Cause:

- Web principle of the same-origin policy (SOP)
- XS-Leaks exploit minute side channels, so indirect information about resource states or user interactions that browsers inadvertently reveal.
  - timing discrepancies
  - error messages
  - status codes
  - even rendering behaviors

# XS Leaks Background

## Cross-Site Oracles:

The information leveraged in an XS-Leak is typically binary and called "oracles"

*"Is the word 'secret' present in the user's search results on another web application? "*



*" Does the query containing **?query=secret** return a **HTTP 200** status code? "*



*" Does loading a resource from **?query=secret** in the application trigger the 'onload' event? "*

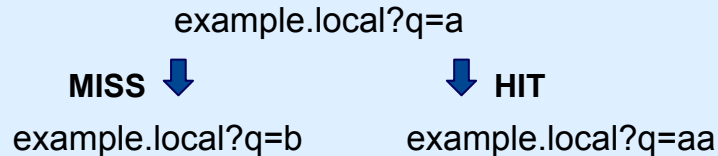
An attacker could repeat these queries with various keywords, allowing him to infer sensitive details about the user's data

# XS Search

XS-Search exploits the differences in response times or other observable behavior when a website processes search queries.

## Technique:

- This attack relies on timing
  - Measure request that return results (hit)
  - Measure request with no results (miss)
- Timing attack on the search endpoint by brute-force



## Real World Example:

- Mass XS-Search using Cache Attack on **Google** products in 2019

# Frame Counting

Window references allow cross-origin pages to get access to some of the attributes of other pages.

## Technique:

- Window references allow limited access to attributes of cross-origin pages adhering to the same-origin policy.
- “window.length”
  - Indicates the number of iframes within a window
  - Can disclose valuable information about the page's structure

## Real World Example:

- **Facebook** Vulnerability Exposed Private Information

# ID Attribute

Cross-origin websites can detect the presence of specific IDs on a page by using focus events and URL fragments.

## Technique:

- Loading a URL with a fragment in an iframe
  - E.g., `https://example.com/foo#secret`
- Triggers a focus event if an element with the corresponding ID exists
- Possible disclosure of sensitive user information

## Possible Scenario:

- A bank uses short numeric OTP as the id for a text box displaying the OTP
- Vulnerable to brute-force attacks: attacker can test all possible OTP values
- Can allow attackers to steal codes and compromise user accounts

# Error Event



If the server responds with an error status, the browser triggers an error event for the page to handle.

- Factors influencing this behavior include:
  - the loaded resources
  - HTML tags
  - presence of specific headers (e.g., nosniff, Content-Type)

## Real World Example:

- A bug allowed abusing a **Twitter API** endpoint. An attacker could exploit this behavior to deanonymize a user. [2019]

# Cache Probing

Cache probing is a technique used to detect whether a resource has been cached by a user's browser

## Technique:

1. **Resource Caching:** when a user visits a website, certain subresources get cached.
2. **Attacker-Controlled Page:** The user visits a page controlled by the attacker, which requests a resource typically cached by the target website
3. **Timing Analysis:** The attacker measures the response time. A quicker response indicates the resource was served from the cache, suggesting the user has visited the target site before

## Real World Example:

- Mass XS-Search using Cache Attack on **Google** product in 2019



# Mitigation Techniques

## Opt-in Mechanisms

- E.g., SameSite Cookies, Cross-Origin-Resource-Policy (CORP)

## Application Design

- Carefully designing the application in a way that prevents XS-Leaks
- E.g., Cache Protections, Subresource Protections

## Secure Browser Defaults

- Browser vendors are actively working on changing default behaviors to help mitigate XS-Leaks
- E.g., CORB, Partitioned Caches



# The Real World [Open-Source]

## Created Methodology:

- Identifying Goals and Interests
- Repository Hosting Services
- **Actively maintained and high starred projects**

## Did we find a vulnerability?



Thanks for listening!