I still Know What You Visited Last Summer

Leaking browser history via user interaction and side channel attacks

Zachary Weinberg
Eric Y. Chen
Pavitha Ramesh Jayaraman
Collin Jackson

Presenter : Arakadakis K.
Schedule

- How do websites work?
- Threat model and defence
- Experiments and results
How do websites work?

- **HTML** structure
- **CSS** style
- **JavaScript** functionality
Where is the problem?

Javascript programs must NOT be permitted to exploit the user!
Same-origin policy

Partitions the web by its servers.

Javascript programs can only see data from the HTTP server that produced them.
BUT...

It was never used for the hyperlinks
BUT...

It was never used for the hyperlinks
Will my browser send my history back to the server?

No.
Where is the catch?

The server (website) can probe a bunch of links and test if they are in your history, based on the way they are rendered.

This is called *history sniffing*.
What is their motivation?

**Good purpose**
Check if users have visited known phishing sites.

**Track users**
Violate privacy, Ads

**Create more targeted phishing pages**
Types of history sniffing

- Direct
- Indirect
- Side-channel
Direct sniffing

1. Create a site that contains links you want to check if are visited
2. Let JavaScript check how these links were rendered
Indirect sniffing

First way

1. Make visited and unvisited links have different size
2. Some element of the site will move
3. Check their final position
Indirect sniffing

First way

1. Make visited and unvisited links have different size
2. Some element of the site will move
3. Check their final position

Second way

1. Make visited and unvisited links load different images
2. Check which images are loaded
Side-channel sniffing

A system leaks information through a mechanism that wasn’t intended to provide that information.

E.g. *Timing attacks*
Defence

In 2010 Baron of Mozilla developed a defence that blocks all known techniques for automated sniffing.

Direct attacks  Computed style APIs pretend that all links are unvisited.

Indirect/side-channel attacks  All links have the same size and take the same time to load.
What about interactive attacks?

Attacks that need users' interaction with the malicious site.

Unfortunately, Baron does not attempt to address these attacks.
Some examples of interactive attacks

Word Captcha

Please type all the words shown below, then press RETURN.

low hang  we life alone  line cost
Some examples of interactive attacks

Word Captcha

Please type all the words shown below, then press RETURN.

low hang  we life alone  line cost

And hide the unvisited links ;)}
Some examples of interactive attacks

Char Captcha

Please type the string of characters shown below, then press RETURN. You don’t have to match upper and lower case.

FA4A SABA A-65 A9-5
Some examples of interactive attacks

Char Captcha

Each character is actually 3 Characters (3 links) combined.

So, we have more probed links.
Some examples of interactive attacks

Chessboard puzzle

Just clink on the powns!
Some examples of interactive attacks

Pattern matching

Click the images from right that when assembled create the left one! Ez
Time for experiments

Tried to see how much information is extracted from browser history. (307 participants)
Results

And the winner is... **Word CAPTCHA**
Side-channel attacks experiments

Webcam attacks

1. Ask user permission to use her webcam
2. If a link is visited: set_background( color of choice ), let's say green
3. Use the webcam and see what the color of the room is.
4. Is it green? Then it is visited...
Side-channel attacks experiments

Webcam attacks - Results

Some users did not even give webcam permissions
Summary

- History sniffing aims to inferring our browser history.
- We can defence against **automated attacks**, thanks to Baron.
- We cannot defence against **interactive attacks**, unless we develop the ability to tell bogus CAPTCHAs and other applications that ask us to interact with.
Thank you!

Q&A