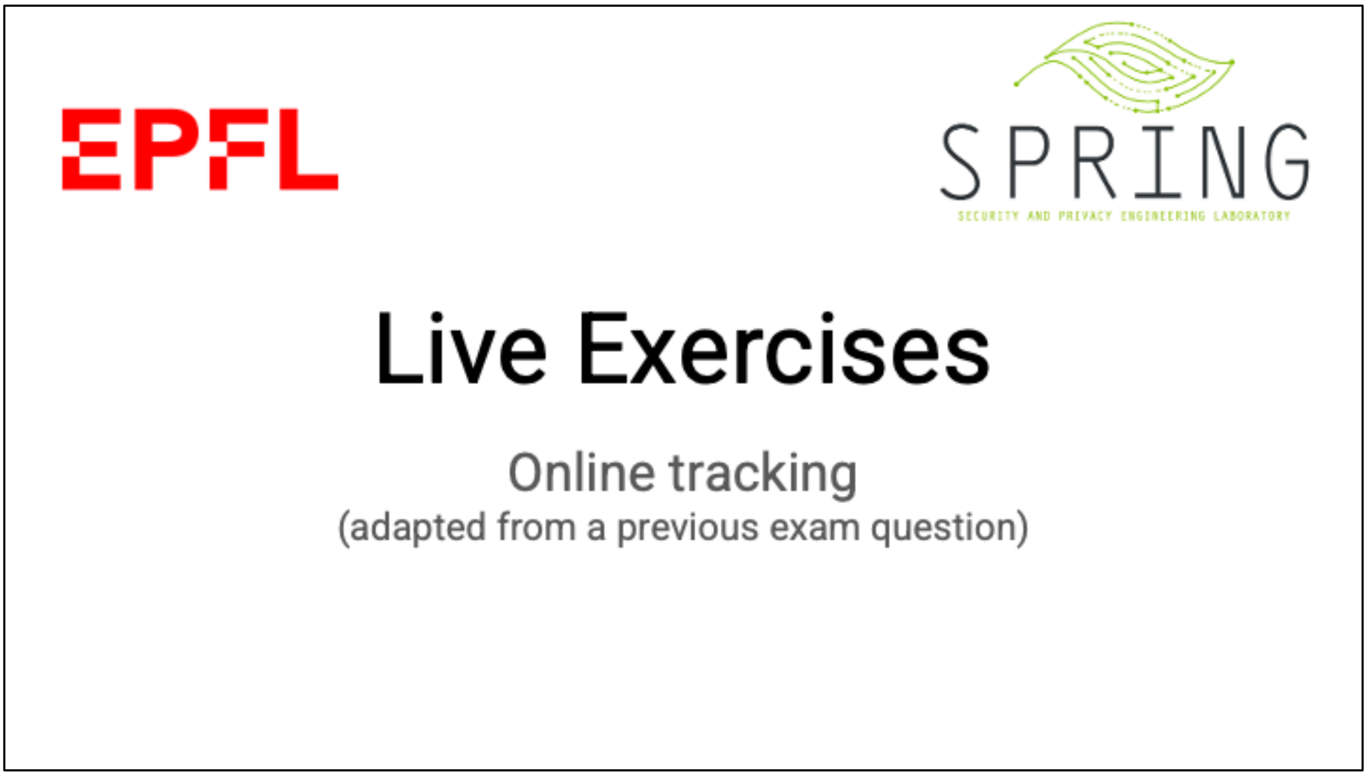


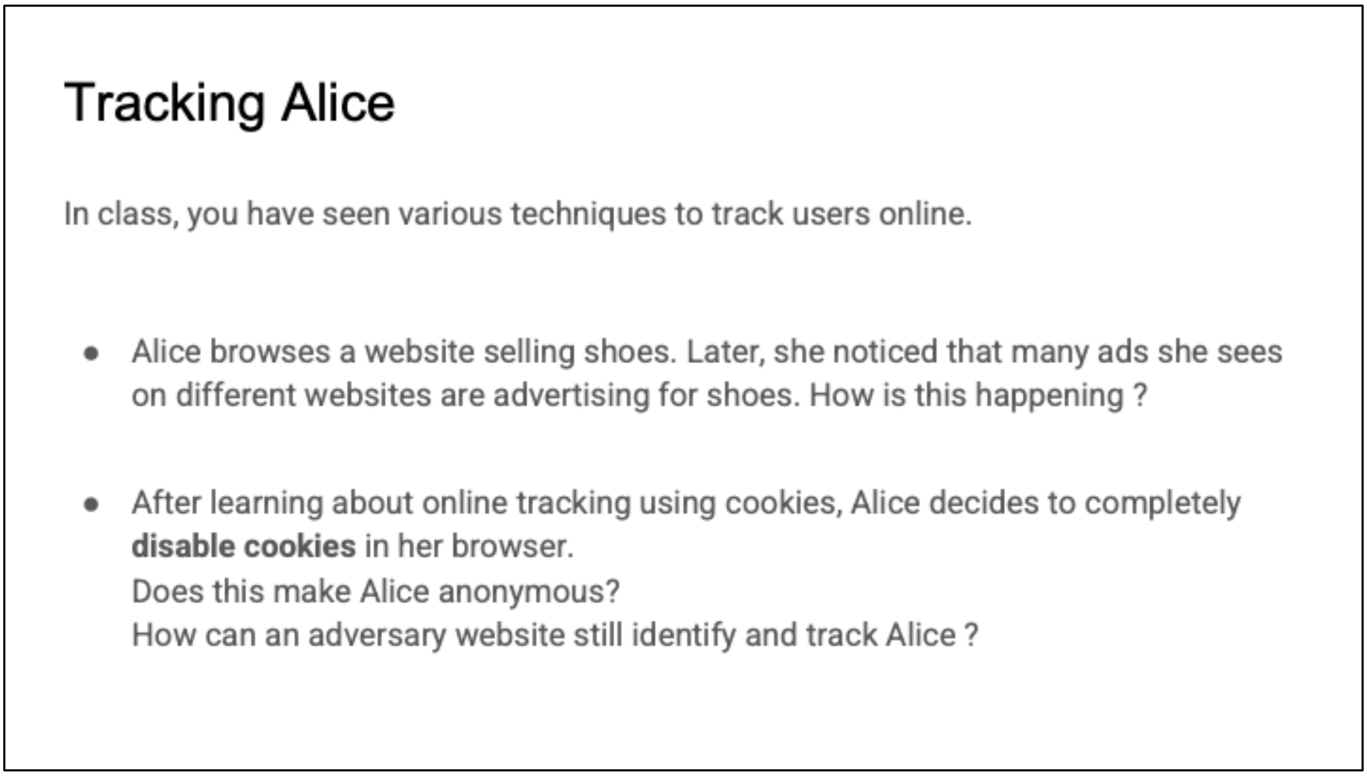
EPFL

SPRING
SECURITY AND PRIVACY ENGINEERING LABORATORY

Live Exercises

Online tracking

(adapted from a previous exam question)



Tracking Alice

In class, you have seen various techniques to track users online.

- Alice browses a website selling shoes. Later, she noticed that many ads she sees on different websites are advertising for shoes. How is this happening ?

- After learning about online tracking using cookies, Alice decides to completely **disable cookies** in her browser.
  Does this make Alice anonymous?
  How can an adversary website still identify and track Alice ?

1)
Advertising companies are storing cookies on visitors machines. When Alice browses a website selling shoes, a third-party cookie is created on her computer.
This cookie can tell any information about Alice behaviour, for example which website Alice browsed and what articles she bought.
When Alice browses a different website, the third-party cookie is read in order to show relevant ads that she has a higher probability to buy.

2)
Even by disabling cookies, a website can still fingerprint Alice.
When a connection from a client to a website is made, the client sends information about its environment. This information is mostly unique and can be traced.
Some example of data fields that are often different among users :

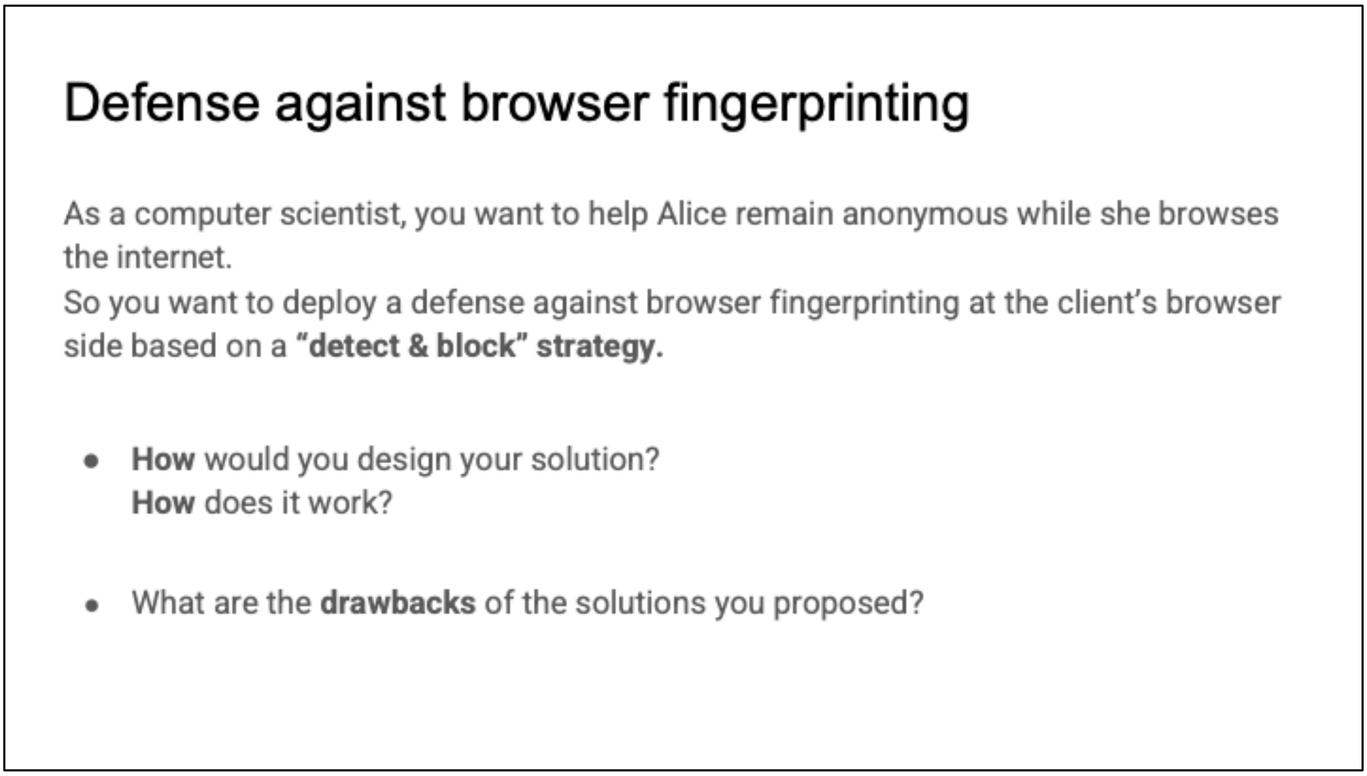
- User agent (information about the browser itself, version, operating system, …)
- Language (which language the browser requested)
- Accept (type of file accepted by the browser as a response from the website)

Added to that, a website can run some scripts on the client side producing different output based on the client machine:

- Canvas fingerprinting (producing images with pictures and fonts from the client machine)
- Font fingerprinting (detect variations in fonts used by the browser)
- Hardware capabilities (number of cores, graphics manufacturer, screen size)

- Software capabilities (libraries versions, plugins enabled/disabled/, etc)

For a list of often used metrics :
https://amiunique.org/
https://coveryourtracks.eff.org/



Defense against browser fingerprinting

As a computer scientist, you want to help Alice remain anonymous while she browses the internet.
So you want to deploy a defense against browser fingerprinting at the client’s browser side based on a **“detect & block” strategy.**

- **How** would you design your solution?
  How does it work?

- What are the **drawbacks** of the solutions you proposed?

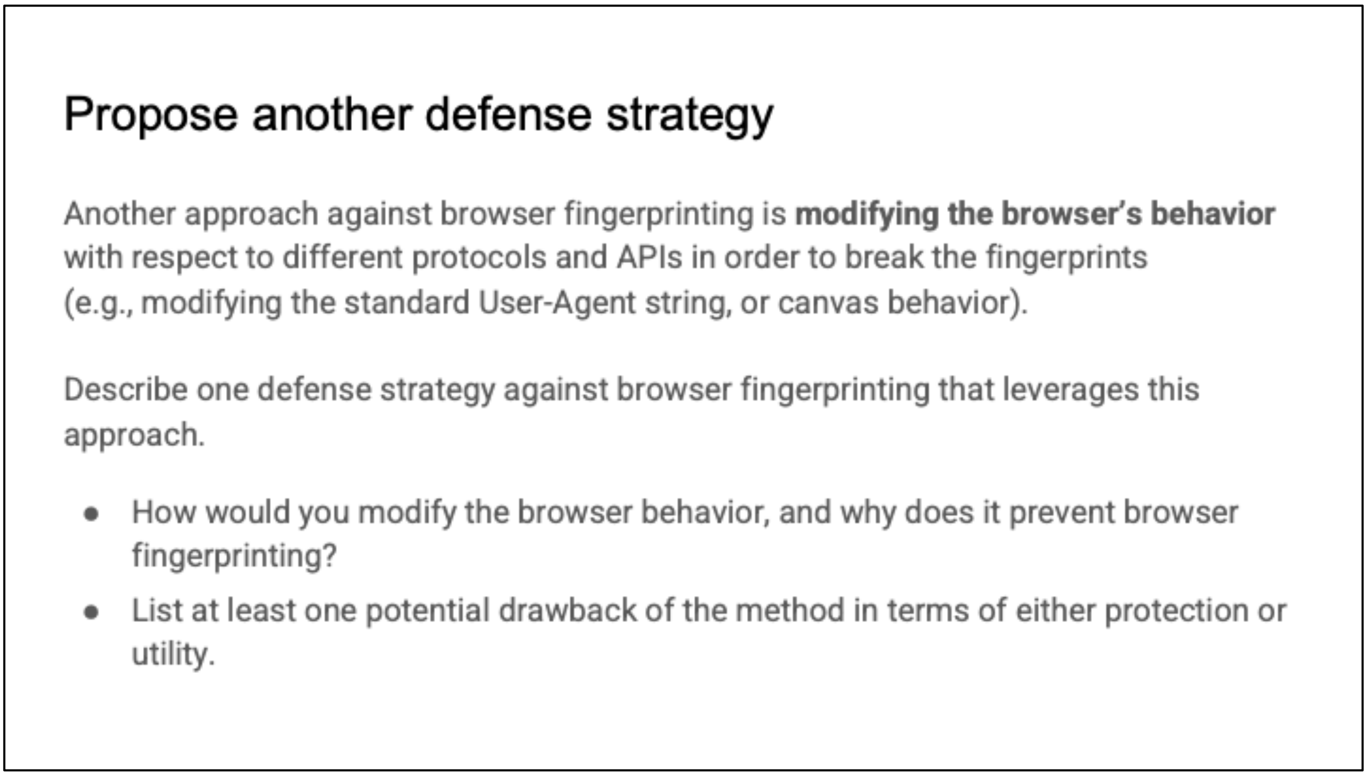
First, one may want to generalize some information in the request header. For example confuse the language fr-CH (swiss french) with fr-FR (france french)
This works, but is not really “detect & block” strategy. It is discussed in the next slide

1)
There are two way to prevent browser fingerprinting based on the detect & block strategy

1. Block scripts based on a known list of malicious scripts
2. Block scripts based on a dynamic analysis of the scripts executing on the client

2)

1. For solution 1 : The drawback is that only known fingerprinting methods will be blocked. Newer fingerprinting techniques / scripts will still work.
2. For solution 2 : The drawback is that there may be false positives, interfering with the legitimate functionality of the website by blocking a non-tracking script. The different will be especially difficult to make in complex websites or web apps that are heavily relying on scripts.



Propose another defense strategy

Another approach against browser fingerprinting is **modifying the browser's behavior** with respect to different protocols and APIs in order to break the fingerprints (e.g., modifying the standard User-Agent string, or canvas behavior).

Describe one defense strategy against browser fingerprinting that leverages this approach.

- How would you modify the browser behavior, and why does it prevent browser fingerprinting?
- List at least one potential drawback of the method in terms of either protection or utility.

Defence strategy:

The defense strategy aims to make browser fingerprints less unique from one-another by making them look more uniform.
Aim is to prevent sharing information that uniquely identifies the browser's fingerprints.

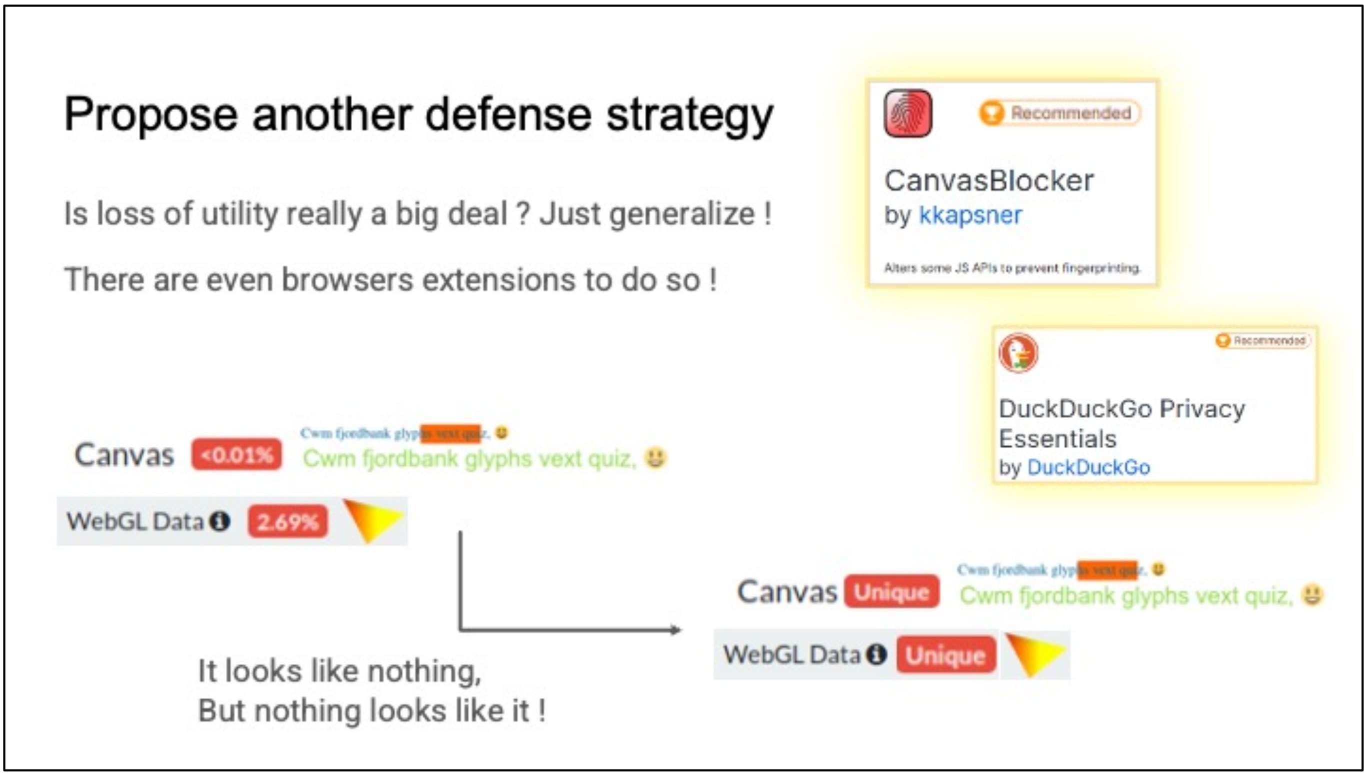
Can focus on any/all three techniques: (1) Generalizing (2) Modifying (3) Hiding.

Generalize certain attributes like OS distributions, media devices, graphic card details, etc.
Modify certain attributes like the user-agent (using a user-agent switcher for example), Canvas, etc.
Hide certain attributes like browser version, list of fonts, connection details, keyboard layout, etc.

Drawback:
First, generalizing / modifying / hiding information may result in a loss of utility. A website may no longer know which language the user speaks, what is the screen size (mobile, desktop ?), and no longer adapt the website to a specific browser.
Also, websites that find certain bugs that only occur for specific users may find it difficult to trace the cause of the bug in the absence of accurate information due to the modifications described above. Ex. Bug only occurring for users using chrome v99

will appear to be occurring for users with different chrome versions, making it seem like the version is irrelevant for debugging it.

In practice finding a good trade-off between privacy and usability is very hard and depends on the context. There is no simple solution to this problem.



Propose another defense strategy

Is loss of utility really a big deal ? Just generalize !

There are even browsers extensions to do so !

Recommended

CanvasBlocker
by kkapsner

Alters some JS APIs to prevent fingerprinting.

Recommended

DuckDuckGo Privacy Essentials
by DuckDuckGo

Canvas <0.01% Cwm fjordbank glyphs vext quiz, 😃

WebGL Data 2.69%

Canvas Unique Cwm fjordbank glyphs vext quiz, 😃

WebGL Data Unique

It looks like nothing,
But nothing looks like it !

In practice it is very hard to protect against browser fingerprinting kind of attacks because
(1) Generalizing : You want to replace a value by something more common, but how to define "more common" ? Are there common values for a set of fonts or a User-Agent ?
(2) Modifying : The value should be modified, but how to find a value that gives you a high anonymity set ? If you replace a value by another leaking less information but which is uncommon you are still identifiable
(3) Hiding : The absence of value is an information. There are not that many browsers that hides their screen resolution for example. This even makes the problem worse.

Most of those extensions are randomizing the information (adding slight random modifications on canvas for example), so the user appears to be a different one every time. This is a not-so-bad solution, but if not a lot of users are implementing this technology then a user can still be recognized based on the fact that they appear as different every time.

Another example: (outside the live-exercise) https://addons.mozilla.org/en-US/firefox/addon/user-agent-string-switcher/ - This extension claims the following: "*This extension allows you to spoof your browser "user-agent" string to a custom designation, making it impossible for websites to know specific details about your browsing arrangement.*"

But if you try to use it, there are a lot of options available to be picked as your “new” user-agent, ultimately doesn’t change anything for privacy.