Imperial College
London

Network and Web Security

Web user tracking

Dr Sergio Maffeis
Department of Computing
Course web page: https://331.websec.fun

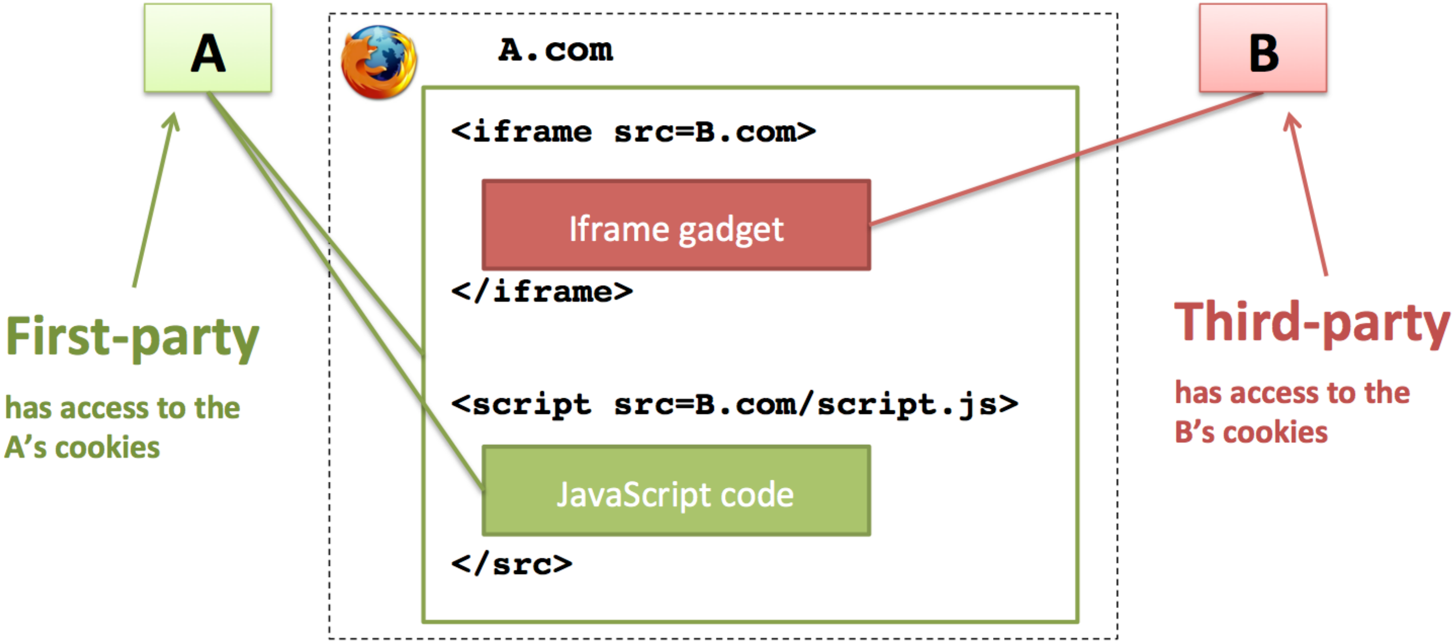
Web tracking

- Examples of tracking
  - User-specific website settings maintain a consistent browsing experience across a sequence of related HTTP requests and responses
  - Secure session recognize requests coming from the same user, that has already been authenticated, and provide privileged access
  - Browsing history, personal preferences and demographic data are harvested by marketers to profile users and provide relevant advertising
  - User presence online on different devices is correlated by law enforcement in order to identify individuals
- Tracking is a complex and pervasive issue
  - 1st party trackers
    - iframes and scripts from same origin as website visited by the user
  - 3rd party trackers
    - Cross-domain iframes and their resources, included by visited websites
  - Can happen across devices
    - Social network IDs in network traffic
    - Access to identifying web pages repeated across devices
    - Learn browsing patterns of user on the desktop, identify similar patterns on mobile
  - There are *legitimate* and *illegitimate* usages
    - Do not necessarily coincide with *desirable* and *undesirable* usages

Browser support for tracking

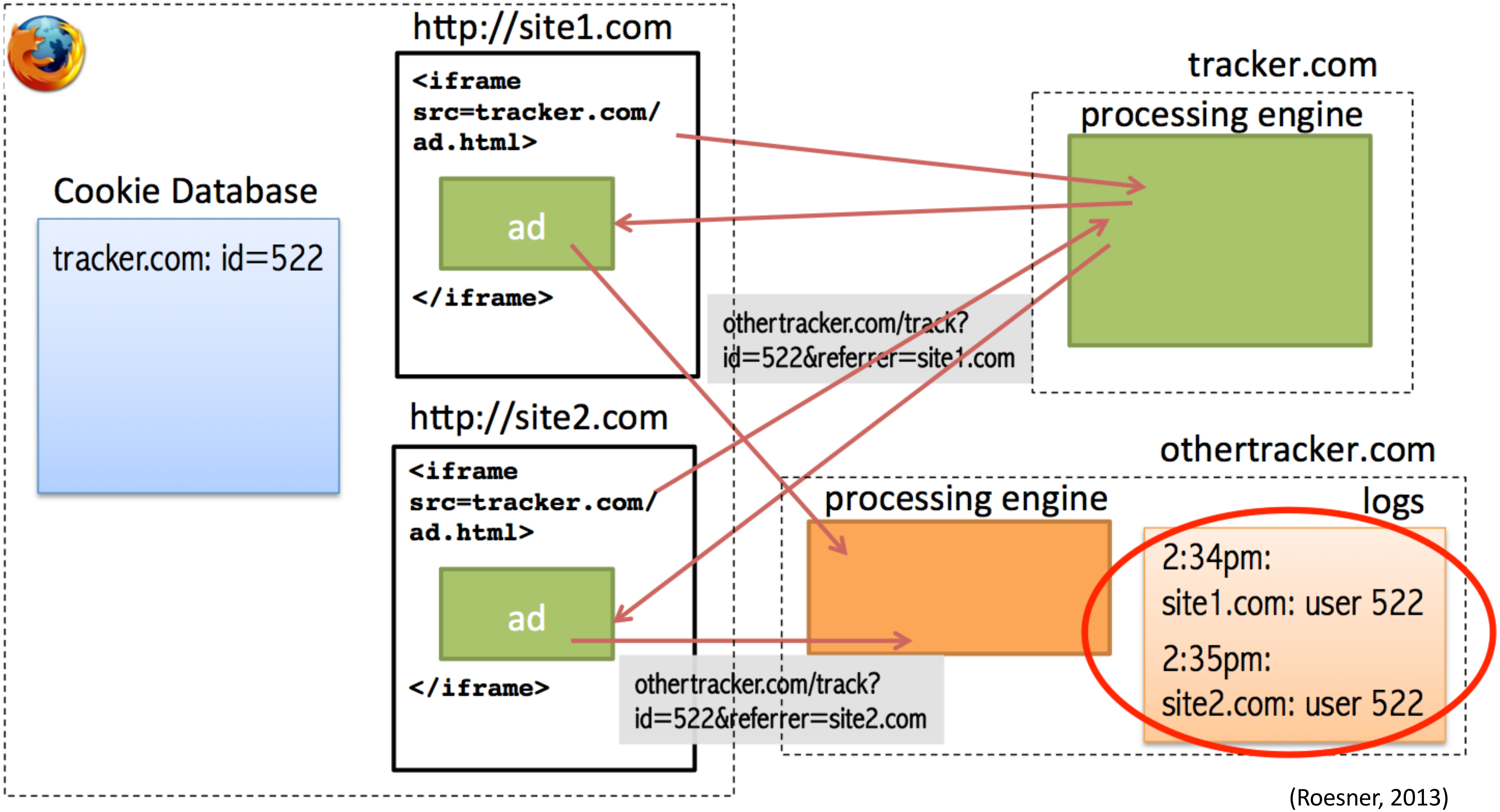
Imperial College London

- **Trackers need to store information in the browser about the user**
- Cookies: again 1st/3rd party distinction



Example: cross-site tracking

Imperial College London



(Roesner, 2013)

Browser support for tracking

- **Trackers need to store information in the browser about the user**
- Supercookies
  - HTML5 storage: Local/sessionStorage, Web SQL, IndexedDB …
  - Plugin storage: Flash Local Shared Objects
- Cached resources
  - Server tells browser to cache a script `track.js?v=1.0`
    - Response header `Cache-Control: private, Max-Age=31536000`
  - Script saves state to a variable `var store= [interesting information];`
  - Other page loads including same script get same value in variable `store`
  - Change resource name to force cache miss and reset state: `track.js?v=1.1`
- Cache headers
  - ETag header is intended for cache validation
  - Response header sets ETAG value `ETag: "A23C42BF890DFE"`
  - Next request headers reflect the Etag value `If-none-match: "A23C42BF890DFE"`
    - A 304 response means use cached version, else new resource is sent back
  - Can be used like a simplified cookie
    - Put tracking id in Etag header: it is sent every time to the server
  - Other request/response header pairs similar to ETag
    - `If-Modified-Since/Last-Modified`

Browser support for tracking

- **Trackers need ways to send user information back to the server**
- HTTP request and responses
  - Explicit communication
    - User clicks on a link
    - Loading of page resources (iframes, images, etc.)
    - AJAX and JavaScript-triggered page loading or navigation events
  - Implicit communication
    - W3C Beacons
      - *“asynchronous and non-blocking delivery of data that minimizes resource contention with other time-critical operations”*
      - `navigator.sendBeacon('/collector', data);`
    - In Chrome, opening a new tab sends a `new_tab` request to Google
    - Search bar may send in the background one request per character you type
    - *Pre-rendering*
      - Browser loads resources linked on current page in case you later click
- Other Plugin communications
  - Flash, Java, Active X controls can use sockets

Zombie cookies

- Deleted your cookies?
- Tracking data saved in other headers, cache or supercookies can be used to resuscitate them!
  - *Cookie respawning,* aka ***Zombie cookies***
  - In fact, who needs cookies if you have JavaScript + localStorage?
- Cleared also cache and local/sessionStorage?
  - Respawning via Flash cookies (LSOs)
  - Thanks to Flash, zombie cookies can migrate across browsers!
    - (LSOs can be shared by various Flash plugin instances)
- Key role of **fingerprinting** in tracking
  - Respawn tracking data associated to a known fingerprint even if browser and Flash data was reset

KISSmetrics and Hulu got sued for that trick in 2011

Tracking countermeasures

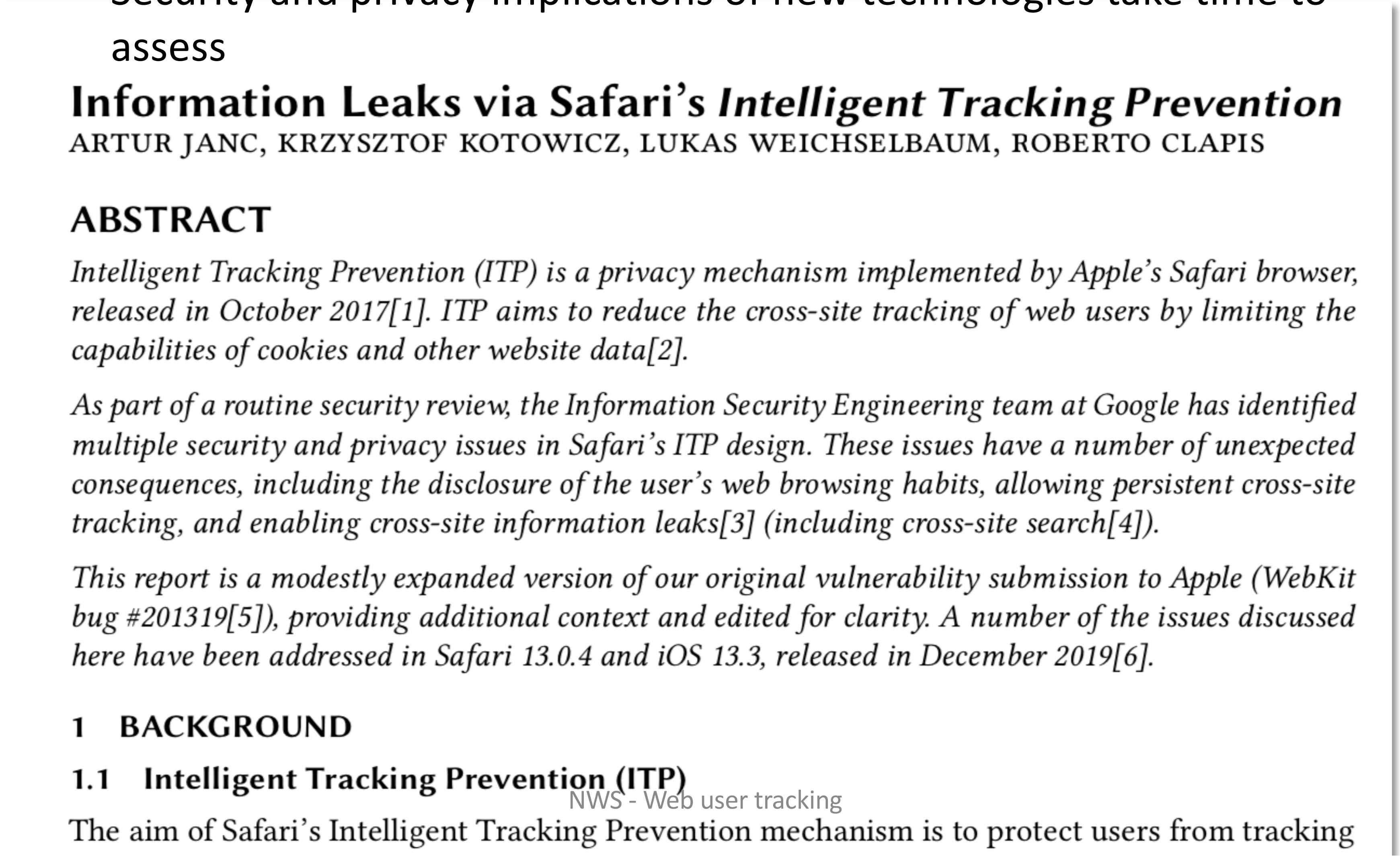
- HTTP level defenses
  - Do Not Track header
    - W3C Tracking Protection Working Group’s brainchild
    - Request header: `DNT : 1`
    - Mostly interpreted as *do not target the users based on collected data*
    - Data is still collected by the tracker
  - Referrer-Policy header
    - Prevent cross-domain Referer leaks
    - See Module 12 - HTTP
- Privacy solutions
  - Private browsing/Incognito mode
    - Prevents caching, history, cookies, preferences
    - A bit of a drastic solution
    - A lot can still be achieved using JavaScript, side-channels, etc.
    - *“Going incognito doesn’t hide your browsing from your employer, your internet service provider, or the websites you visit.”*
  - Anti-tracking extensions: Ghostery, AdBlock+, Privacy Badger, ShareMeNot…
  - Privacy focussed browsers: TOR, Vivaldi, …

Browser-based countermeasures

- User settings
  - Block 3rd party cookies
    - 3rd party trackers cannot set cookies
    - In some browsers, 3rd party cookies are still sent if they already exists
    - A 3rd party iframe can open a popup which is now a 1st-party cookie setter
  - Disable plugins
    - Finally Flash is no longer supported by major browsers
  - Disable JavaScript (stop browsing?)
- New trend: built-in tracking defenses
  - Firefox
    - Enhanced Tracking Protection: blocks by default 3rd party cookies
    - Since Jan 2020, stops also 3rd party requests for fingerprinting scripts
    - Based on Disconnect blacklist: https://disconnect.me/trackerprotection
  - Chrome's Privacy Sandbox
    - Use SameSite cookies, kill Flash, work in progress
  - Safari's Intelligent Tracking Prevention
    - Complex solution controlling cookies, local storage lifetime and scope, based on blacklisting and analysis of url parameters (e.g. website.example?clickID=0123456789)

Browser-based countermeasures

- Beware of complex solutions
  - Internet standards and browsers evolved a bit at a time, over the years
  - Security and privacy implications of new technologies take time to assess



Information Leaks via Safari's *Intelligent Tracking Prevention*

ARTUR JANC, KRZYSZTOF KOTOWICZ, LUKAS WEICHSELBAUM, ROBERTO CLAPIS

ABSTRACT

Intelligent Tracking Prevention (ITP) is a privacy mechanism implemented by Apple's Safari browser, released in October 2017[1]. ITP aims to reduce the cross-site tracking of web users by limiting the capabilities of cookies and other website data[2].

As part of a routine security review, the Information Security Engineering team at Google has identified multiple security and privacy issues in Safari's ITP design. These issues have a number of unexpected consequences, including the disclosure of the user's web browsing habits, allowing persistent cross-site tracking, and enabling cross-site information leaks[3] (including cross-site search[4]).

This report is a modestly expanded version of our original vulnerability submission to Apple (WebKit bug #201319[5]), providing additional context and edited for clarity. A number of the issues discussed here have been addressed in Safari 13.0.4 and iOS 13.3, released in December 2019[6].

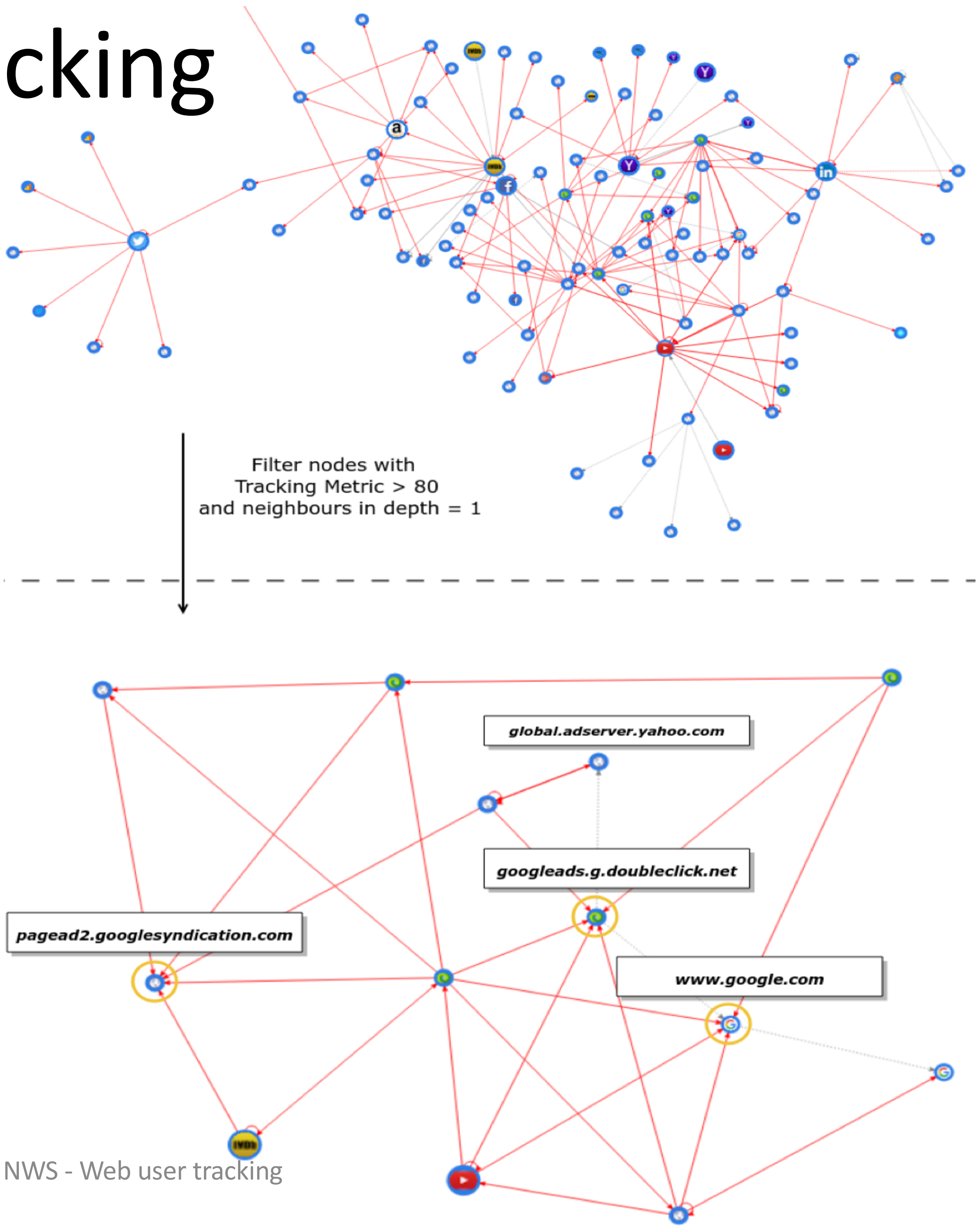
1 BACKGROUND

1.1 Intelligent Tracking Prevention (ITP)

The aim of Safari's Intelligent Tracking Prevention mechanism is to protect users from tracking

Research on tracking

- Anti-tracking extensions use blacklists to stop requests to tracking websites
  - How to automatically populate such blacklists?
  - Ongoing research on machine learning techniques to identify trackers
- Trackers leave a trail of information visible from the browser
  - Ongoing research on data analytic techniques to spot tracking patterns
  - Monitorito browser extension (distinguished student project)
    - Monitor and visualise network events in real time
    - Identify trackers using graph analytics


Filter nodes with
Tracking Metric > 80
and neighbours in depth = 1
global.adserver.yahoo.com
googleads.g.doubleclick.net
pagead2.googlesyndication.com
www.google.com