

CS-523 Advanced topics on Privacy Enhancing Technologies

Tracking Live exercises

Karel Kubicek

Waterwolf 2.0

The Waterwolf Browser company collects usage statistics from its users to better understand which websites its users visit most frequently and how this behaviour changes over time. In the Waterwolf database, each user is identified by a unique identifier and the usage_table contains the following information about each user: the origin country of their IP address, their total browsing time in minutes, and a list of binary values that indicates, for a pre-defined set of 1000 websites, whether the user has ever visited this website. A snapshot of this database is shown below

user_id	country	usage_time (in minutes)	google.com	amazon.ch	...	protonmail.com
uid198	CH	121	0	0	...	1
uid847	CH	76	1	1	...	0
...
uid272	FR	876	1	0	...	1

The Waterwolf database gets updated with the most up-to-date statistics on a daily basis. This means that when a new user has started using the Waterwolf Browser, a new entry is created; and that when an existing user visits a website they had not previously visited, the corresponding entry is flipped from 0 to 1.

Waterwolf 2.0

Part 1: Waterwolf decides to reduce cookie usage for its users and instead launches a service for “cookie-replacement”: instead of giving real cookies to the web server, the browser sends `user_id` and allows the web server to query relevant information (i.e. country, ...) from the database.

- 1) Can this solution work as real cookies?
- 2) Does it have better or worse privacy/security properties?

user_id	country	usage_time (in minutes)	google.com	amazon.ch	...	protonmail.com
uid198	CH	121	0	0	...	1
uid847	CH	76	1	1	...	0
...
uid272	FR	876	1	0	...	1

1) It will work. Honest web servers will ask the browser for the `user_id`, query the Waterwolf database and get the info from the corresponding record, and use that to provide personalized content, resume session, etc.
Depending on the kind of cookie you want to replace, Waterwolf might need to add columns to the database.

2) Not that much of an improvement over the cookie approach. Now the browser is collecting your data and sending to a database which can be accessed by potentially malicious servers. Waterwolf needs to build an access control mechanism and allow some web servers to access some information.

The user has no control about who accesses this database, nor any way to clear it, compared to clearing its cache.

Only point where it is a bit better (arguably) : if computer is hacked, the only local info that can be extracted from the computer is the `user_id`, which is useless without database access. But following this logic, if Waterwolf is hacked, the whole database is compromised.

Waterwolf 2.0

Part 2: After being fined for sharing private user data with third-parties, Waterwolf decides to record only public information about its users, along with the list of websites used.

- 1) Which information is public for any browser user?
- 2) What are the privacy implications?

- 1) OS / browser version / timezone / language and some others. See the fingerprinting slide or the amunique website.
- 2) This data is still valuable, waterwolf can still sell their database because users are still uniquely identified through fingerprinting, and some information can be used by third parties to adapt to their audience (preferred language is one example).

Waterwolf 2.0

Part 3: For debugging purposes, Waterwolf software engineer Joe modified the source code. The browser, instead of sending its version, sends the time in milliseconds required for execution of the last Javascript snippet. He did it right before Christmas, went on holidays, and although the vulnerability was discovered later on, this browser version was already downloaded and installed by thousands of new users.

1) Why is it a vulnerability?

2) How can the company fix this issue?

See Fantastic Timers and Where to Find Them if you are interested in this topic.

1) While it is hard to fingerprint with only the browser version, this time leaks information about the user's hardware and can be used to differentiate users. Furthermore, a malicious website that wants to increase the precision of this fingerprinting can add a custom JS snippet to their website that helps them further identify users. Indeed, some operations are CPU dependent.

2) Some companies can force users to download each newer version (hi Apple). In the case where you cannot do that it gets tricky. You can notify honest web servers to include a pop up when such users are visiting the website among other options to be discussed.