Do Not Track

DOM Flag & Response Header

Jonathan Mayer Arvind Narayanan

questions@donottrack.us

http://donottrack.us

DOM Flag

`navigator.doNotTrack`

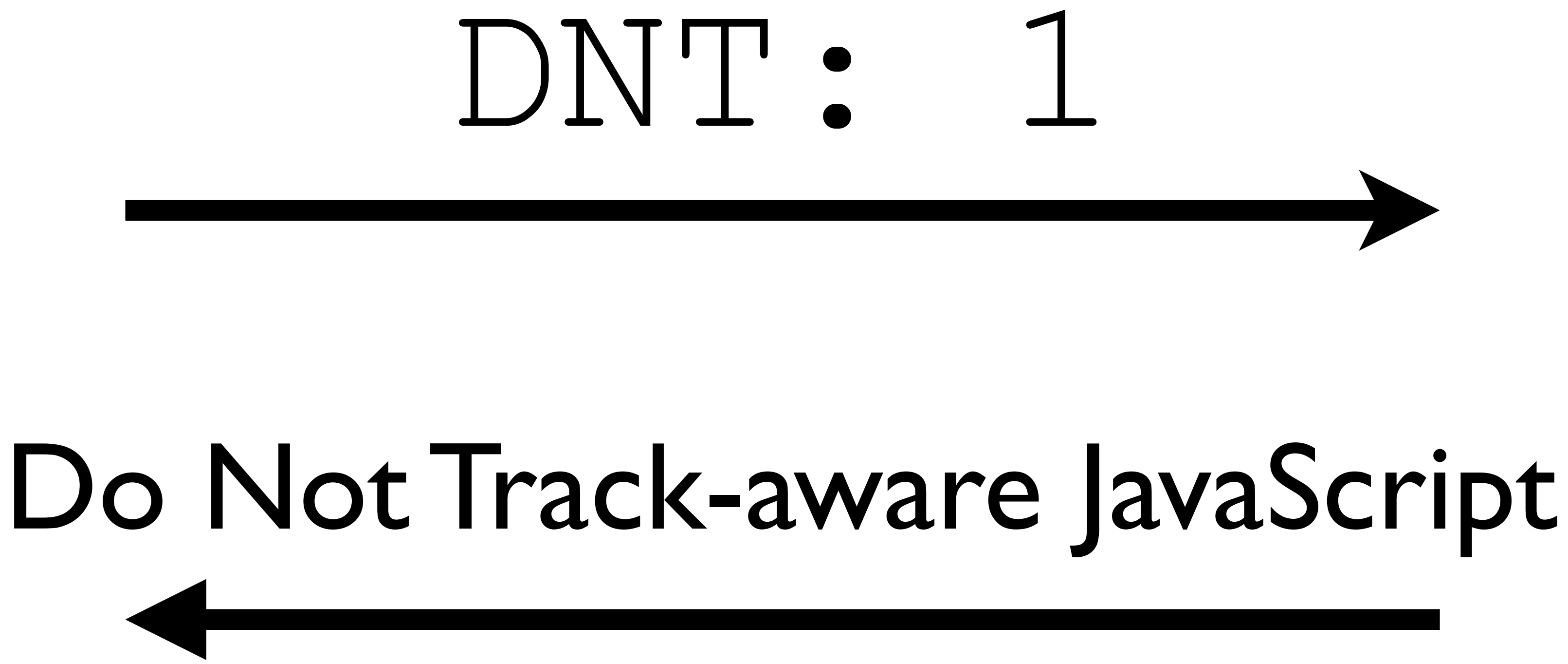
Could include some per-URL or per-domain querying.

Wrinkle #1

No DOM Flag

≠

No Do Not Track-aware JavaScript



DNT: 1

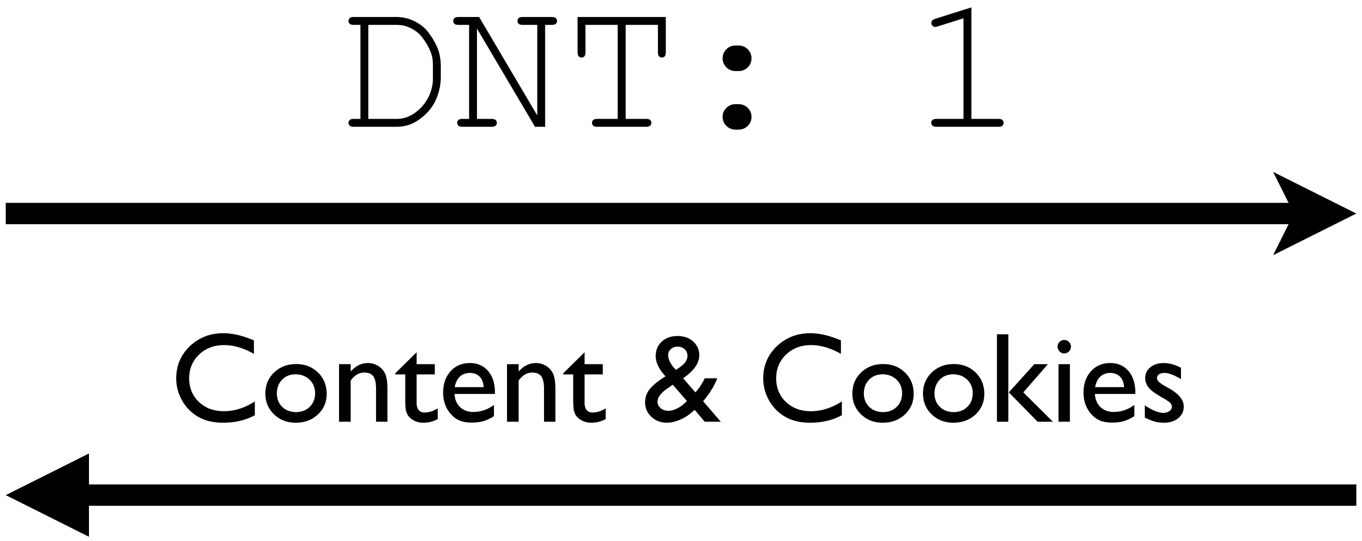
Do Not Track-aware JavaScript

```
<?php
if(array_key_exists("HTTP_DNT", $_SERVER) and $_SERVER["HTTP_DNT"] == "1")
{
      print("var DoNotTrack = 1;\n");
}
?>
```

DoNotTrackScript.js

Wrinkle #2

A third party always has to check for the Do Not Track header.



DNT: 1

Content & Cookies

What to do about logging the request?

What about clients that don’t support JavaScript?

Have to look at the header at some point.

Wrinkle #3

Script includes make granularity hard.

Third-party scripts often run in the first-party DOM.

OK if Do Not Track is set universally.

```
<script src=”foo.js”>
```

But what if it isn’t?

Naive approach doesn’t work.

Granularity requires non-trivial API.

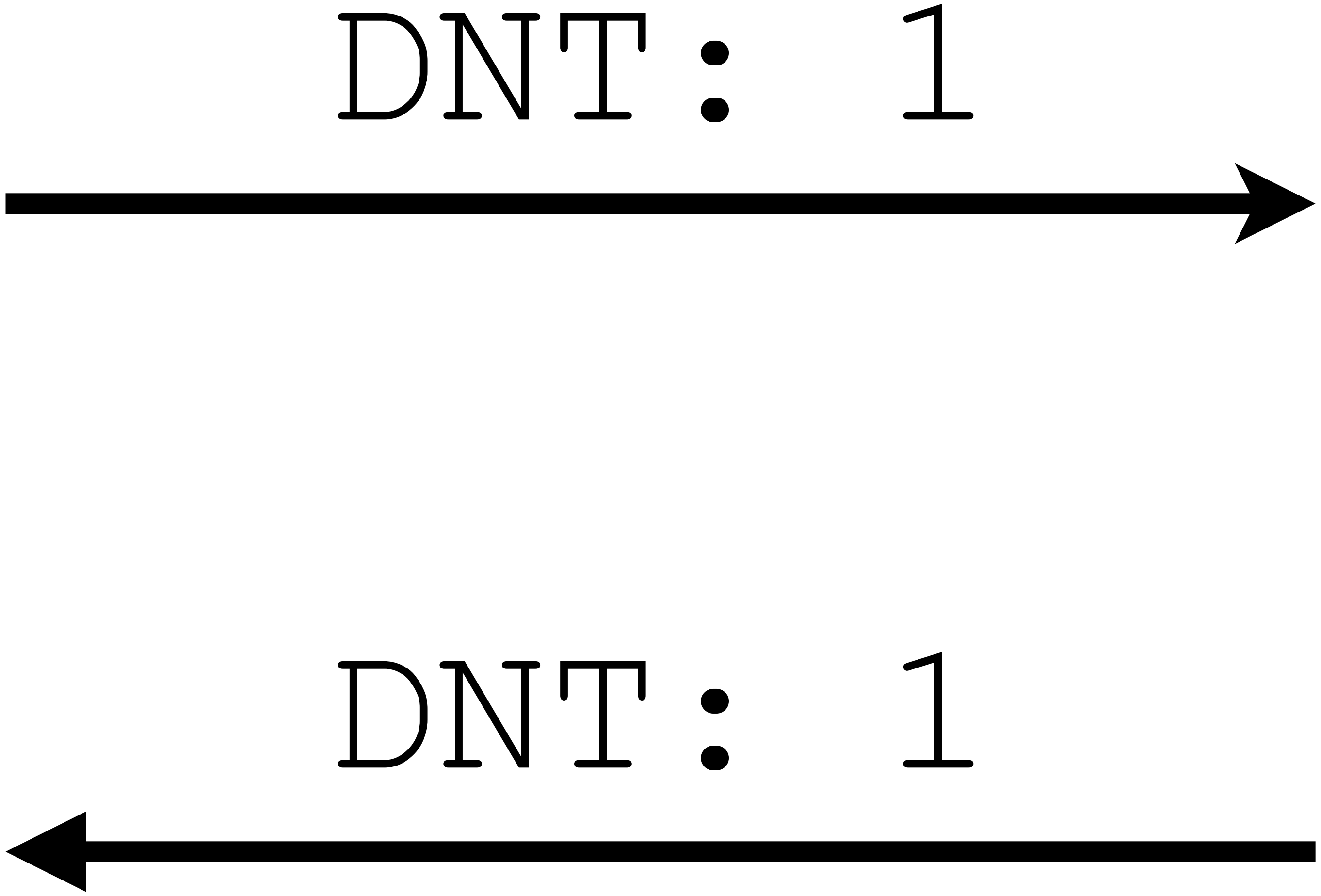
Pro

- Third-party DNT-aware scripts can be hosted from 100% static HTTP servers
  - Open Q: Does this matter?
- Third parties don’t have to implement any server logic to make scripts DNT-aware

Con

- DNT value may be incorrect or not present (depending on implementation)
  - More accurate granularity = fingerprinting risk
- Requires specification/standardization of new JavaScript API
- Browsers have to implement the DOM flag

Response Header



DNT: 1

DNT: 1

Pro

- Easier to get metrics about Do Not Track support
- Allows decentralized quasi-enforcement & nudging mechanisms
  - e.g. third-party (domain) cookie blocking if no response
- Easier technical enforcement
  - response header + tracking = violation
- Makes each Do Not Track response clearly within existing "deceptive business practices" authorities

Con

- Requires third parties to implement a response header
- More traffic on the wire

Thanks

Stanford Security Laboratory

CiS

The Center for
Internet and Society

EFF