Creative Scanning in the Protected Audience API

Presented by orrb@google.com at the WICG Protected Audience Call on January 31, 2024

Overview

- https://github.com/WICG/turtledove/issues/792
- Sellers need to enforce publisher restrictions on ads
- Ecosystem generally prefers realtime scanning over pre-registration
- Challenges in Protected Audiences API: auction runs in isolation
- Event-level reporting can’t solve this - it only works for winning ads
- Ads can't generally win an auction before they’ve been scanned

Goals

- Ensure that ads are sent to sellers for creative scanning
- Don't overload sellers' servers with a firehose of ads to scan
- Minimize the privacy impact of sending ads for scanning

What information is being sent?

- *creativeScanningMetadata*, e.g. *adomain* and *seat*
- Intentionally distinct from *metadata*; meant to be small
- Low privacy risk: same data that could have been sent directly from buyer

```
const myGroup = {
  'owner': 'https://www.example-dsp.com',
  'name': 'womens-running-shoes',
  'ads': [
    {
      renderUrl: shoesAd1,
      metadata: { ... },
      creativeScanningMetadata: { ... },
      ...
    },
    ...
  ],
  ...
};
const joinPromise =
navigator.joinAdInterestGroup(myGroup);
```

To what endpoint is it sent?

Each ad's *renderURL*, *creativeScanningMetadata*, and interest group owner is sent by POST to an entrypoint each seller exposes at a .well-known URI:

https://www.example-ssp.com/.well-known/interest-group/creative-scanning

```
{
  "https://www.example-dsp.com": [
    {
      "renderURL": "https://example-adserver.com/...",
      "metadata": "the renderURL's associated creativeScanningMetadata"
    },
    ...
  ]
}
```

Sample body of the POST request

To which sellers are ads sent?

Buyers list sellers to which their ads should be sent at a .well-known URI:

https://www.example-dsp.com/.well-known/interest-group/creative-scanning-buyer-config

```
{
  'sellers': [
    'https://seller1.com',
    'https://seller2.com'
  ]
}
```

Sample list of sellers returned from the buyer config .well-known URI.

If the buyer config is missing or empty, the ad won’t be sent for creative scanning.

Overriding sellers for an interest group

Buyers can override the list of sellers for a given interest group using *seller capabilities*

```
'sellerCapabilities': {
  'https://seller1.com': [ 'creative-scanning' ],
  'https://seller2.com': [ 'latency-stats' ],
  'https://seller3.com': [ 'latency-stats', 'creative-scanning' ],
  '*': [ 'interest-group-counts' ]  // ‘creative-scanning’ is invalid for “*”.
}
```

Sample *seller capabilities* listing to whom ads should be sent for creative scanning, in this case, 'https://seller1.com' and 'https://seller3.com'

If an interest group has *no* seller capabilities with ‘creative-scanning’, the sellers enumerated in the buyer config are used instead.

Limiting traffic to creative scanning entrypoints

Sellers can specify a per-buyer sampling rate at a .well-known URI:

https://www.example-ssp.com/.well-known/interest-group/creative-scanning-seller-config

```
{
  "perBuyerSamplingRates": {

    "https://www.example-dsp.com":
       { "sampling_rate": 1 },  // Send all ads from this buyer
    "https://www.another-dsp.com":
       { "sampling_rate": 0 },  // Don't send any ads from this buyer
    "*":
       { "sampling_rate": 0.1 }  // Send 10% of ads from all other buyers
    // If no default is provided, the default sampling rate is 1.
  }
}
```

Sample per-buyer sampling rates returned from the seller-config .well-known URI

When is it being sent?

- The document enumerates several options
- Most options propose ***interest group join and update time***
- Options provide additional approaches for reducing traffic to creative scanning entrypoints

Option 2b (preferred)

- Send only those ads not previously sent from this device for a given joining site
- Browser maintains a *creativeScanningHistory* data table (*renderURL*, *interest group owner*, *interest group joining site*, and *seller*)
- Ad is sent to seller each time it reaches new devices (join or update)
- Effectively: each ad is sent once, to each configured seller, from each device, everytime the IG is joined from a distinct site

Other options considered

1	Send all ads during interest group join/update	No benefit over Fetch requests
2	Send only those ads not previously sent from this device	Privacy risk from browser ‘memory’
3	Send only those ads as configured by the seller (bloom filter)	Does not scale with number of ads
4	Use k-anonymity as a proxy for ads that a seller has not yet seen	Ads not sent to sellers who join ‘late’
5	Call trusted scoring signals server during join/update	Moves traffic to a more expensive resource
6	Reuse the auction-time call to the trusted scoring signals server	Leaks privacy
7	Reuse the auction-time call, but only send ads that are k-anonymous	No metadata / breaks k-anonymity

More Details in the doc

Protected Audience API
Creative Scanning
Proposed Design

https://docs.google.com/document/d/1s0tTN25AiPwl3ocCFYOLqeKhetZCt_YFIYQEQ7wzHqI