

September 28, 2020

CORRELATION STUDY BETWEEN ADS.TXT AND SELLERS.JSON



Abstract	2
Introduction	2
Finding the Issue	3
Verifying a Larger Data Set	3
Reasons for Occurrences	5
Effects on Arbitrage	6
The Supply Chain Object	6
Note on ads.cert	7
Remediation	7
Method to Reproduce	8
About escalated.io	8
References	9

ABSTRACT

The purpose of ads.txt in conjunction with sellers.json is to provide a clear and transparent view into the journey of an inventory unit and the parties of whom touched it. We found that widespread implementation issues whether unintentional or with malintent can lead to problems determining seller types and their relationships. This can cause buyers to pay premiums for resold inventory while believing they are buying direct, have the potential to yield more profitable arbitrage operations, and complicate SPO (supply path optimization). These problems combined make buyers and the advertising systems they purchase through more vulnerable to certain ad fraud schemes.

INTRODUCTION

The goal of our work was to collect general usage statistics and mappings across the ecosystem. However, in pursuit of that goal we observed numerous ads.txt files containing hundreds of lines and sometimes numbers past one thousand. These files also contained an inordinate amount of DIRECT entries.

Given the amount of DIRECT entries that were uncovered, we sought to tie this data back to sellers.json to determine whether the entity type was identified as PUBLISHER, INTERMEDIARY, or BOTH.

It was found that in several cases DIRECT entries were tied to the INTERMEDIARY seller type. The problem is that buyers targeting DIRECT are doing so to purchase directly from the publisher. INTERMEDIARY implies a RESELLER relationship.

The screenshot shows the 'Display & Video 360 Help' page. The main heading is 'Authorized seller targeting in Display & Video 360'. Below the heading, a red box highlights the sentence: 'Display & Video 360 allows buyers to apply ads.txt targeting options at the Partner, Campaign, Insertion Order or Line Item level. Ads.txt targeting options for new entities (e.g. Line item) inherit the settings applied on the parent entity (e.g. the Insertion Order that the line item belongs to)'. Below this, it lists available targeting options:

- **Authorized Direct Sellers and Resellers:** All authorized sellers, including those that directly own or resell the inventory being monetized, as indicated by a **DIRECT** or **RESELLER** declaration in the ads.txt file respectively. Note that this is the default option for new Display & Video 360 campaigns.
- **Authorized Direct Sellers only:** Only authorized sellers that directly own the inventory being monetized, as indicated by a **DIRECT** declaration in the ads.txt file.
- **Authorized and Non-Participating Publishers:** All authorized sellers, including publishers that have not posted an ads.txt file. Display & Video 360 automatically disallows unauthorized sellers.

On the right side of the page, there is a 'Manage' section with a list of items: 'Aut', 'Vie', 'adv', 'Vie', 'Dis', 'Dig', '360'.

Information on how to target DIRECT only inventory in DV360 as declared through ads.txt

FINDING THE ISSUE

```

advertising.com, ██████, DIRECT
appnexus.com, ██████, DIRECT
contextweb.com, ██████, DIRECT, 89ff185a4c4e857c
google.com, ████████████████, DIRECT, f08c47fec0942fa0
google.com, ████████████████, DIRECT, f08c47fec0942fa0
indexexchange.com, ██████, DIRECT, 50b1c356f2c5c8fc
indexexchange.com, ██████, DIRECT, 50b1c356f2c5c8fc
lkqd.net, ██████, DIRECT, 59c49fa9598a0117
openx.com, ██████, DIRECT, 6a698e2ec38604c6
openx.com, ██████, DIRECT, 6a698e2ec38604c6
openx.com, ██████, DIRECT, 6a698e2ec38604c6
pubmatic.com, ██████, DIRECT, 5d62403b186f2ace
pubmatic.com, ██████, DIRECT, 5d62403b186f2ace
rubiconproject.com, ██████, DIRECT, 0bfd66d529a55807
rubiconproject.com, ██████, DIRECT, 0bfd66d529a55807
rhythmone.com, ████████████████, DIRECT, a670c89d4a324e47
rhythmone.com, ████████████████, DIRECT, a670c89d4a324e47
spotx.tv, ██████, DIRECT, 7842df1d2fe2db34

```

The screenshot¹ above is a small snapshot taken from an ads.txt file containing 631 total lines. 235 of those lines declared a DIRECT relationship authorizing 205 unique seats across 51 different advertising systems.

DIRECT relationships should be the result of a financial arrangement between the publisher and the advertising system and indicates that the publisher controls the seat ID listed for that advertising system. Having 51 such relationships while not entirely implausible, authorizing 205 seats seemingly warranted additional information.

After spot checking some of these DIRECT entries against the advertising system's sellers.json file, we were able to verify that the seller type was in fact listed as INTERMEDIARY where PUBLISHER would have been expected.

VERIFYING A LARGER DATA SET

Given that the issue had been identified several times through manual audit, we sought out to design a spider capable of correlating ads.txt files with sellers.json files against a list of over one hundred thousand domains.

1. Seat IDs have been obscured. See Reasons For Occurrences section.

SELLER TYPE	SEAT ID	ADVERTISING SYSTEM	DIRECT DOMAINS
INTERMEDIARY		rubiconproject.com	8868
INTERMEDIARY		indexexchange.com	8479
INTERMEDIARY		rubiconproject.com	7960
INTERMEDIARY		pubmatic.com	7144
INTERMEDIARY		rubiconproject.com	6502
INTERMEDIARY		indexexchange.com	6295
INTERMEDIARY		rubiconproject.com	6261
INTERMEDIARY		pubmatic.com	6031
INTERMEDIARY		openx.com	5856
INTERMEDIARY		appnexus.com	5787
INTERMEDIARY		indexexchange.com	5639
INTERMEDIARY		openx.com	5588
INTERMEDIARY		rubiconproject.com	5575
INTERMEDIARY		rubiconproject.com	5572
INTERMEDIARY		indexexchange.com	5572
INTERMEDIARY		indexexchange.com	5571
INTERMEDIARY		openx.com	5571
INTERMEDIARY		openx.com	5571
INTERMEDIARY		pubmatic.com	5571
INTERMEDIARY		rubiconproject.com	4729
INTERMEDIARY		indexexchange.com	4671
INTERMEDIARY		indexexchange.com	4668
INTERMEDIARY		rubiconproject.com	4214
INTERMEDIARY		pubmatic.com	4175
INTERMEDIARY		pubmatic.com	4089
INTERMEDIARY		pubmatic.com	4082
INTERMEDIARY		rubiconproject.com	3979
INTERMEDIARY		indexexchange.com	3480
INTERMEDIARY		rubiconproject.com	3417
INTERMEDIARY		rubiconproject.com	3414

As you can see from the screenshot above, we looked at data resolving to the top five exchanges. The results uncovered a large volume of domains that had declared a DIRECT relationship within ads.txt for seat IDs that, when referenced in sellers.json, were listed as being controlled by an INTERMEDIARY seller type.

REASONS FOR OCCURENCES

Declaring a relationship as being DIRECT when it is not can lead to a host of issues as listed in the ABSTRACT section above. One such issue being premiums paid by buyers who are targeting direct would have their extra spend disbursed among resold inventory instead. Despite these outcomes, we were unable to determine the exact reasoning or intent as to why this type of misrepresentation occurs in each case. Therefore, names and seat IDs have been redacted from all screenshots. However, the following four scenarios have been identified.

1. It could be the result of confusion. The standards around ads.txt and sellers.json can be difficult to interpret and put into practice without a greater working knowledge of the wider ecosystem. Even assuming that lack of understanding is not a factor, it is typical that several parties will combine to cohesively create an ads.txt file. Each additional party necessary to complete a file of entries gives way to a greater susceptibility of errors and mishandlings.
2. Such misdeclarations could in fact be malicious. A manipulation deployed to redirect spend towards less opportune businesses designed to feign a more competitive environment. Such a scheme would make it much harder for original content owners and other unique services and platforms to be able to distinguish themselves amongst the crowd. This action would also create additional incentive for those engaging in arbitrage as the practice becomes more profitable being based solely on a DIRECT relationship that does not exist. Further, this intent would also allow for the growth of domain spoofing operations.
3. Another possibility is that the publisher changes the relationship type after the RESELLER lines are delivered by the intermediary. This could be done with the idea that revenues will increase or could simply be the result of a misunderstanding in the interpretation of DIRECT versus RESELLER.
4. A final scenario is that the publisher has outsourced all ad operations and has no direct presence themselves. The intermediary representing the publisher is doing so exclusively. A publisher could even elect several different intermediaries, all representing an exclusive slice of inventory. Since there is currently no mechanism to identify these types of relationships, DIRECT may be selected as an alternative since buyers can only access the inventory directly via an intermediary. Consider that publishers who outsource exclusively and list all ads.txt lines as RESELLER may be competitively disadvantaged by being unable to access buyers who prefer to target DIRECT only.

EFFECTS ON ARBITRAGE

The practice of arbitrage is the idea of acquiring inventory at a lower cost than what advertisers are willing to pay for it. One example would be a publisher sourcing traffic for less than their average page RPM (rate per mille). Another example is an intermediary who purchases a publisher's inventory out of one exchange and then immediately makes it available for sale in another exchange. The goal is to pay less in the first exchange than what the inventory will sell for in the second.

In the case where an intermediary has their exchange seats declared as DIRECT in a publisher's ads.txt file, they can capture premiums meant to go directly to that publisher. The intermediary does not need to have tags on the publisher site, be installed as a bidder, or even have a current business relation. They can simply buy the publisher's inventory from one exchange and sell it as DIRECT into another at the expense of the buyer or brand who is targeting direct.

Further, this opens the door to domain spoofing. If a buyer trusts that a DIRECT relationship is in fact DIRECT when really it's not, then a party with malicious intent has incentive to spoof the targeted domain for the purpose of quickly realizing profits until either the DIRECT lines are removed or the buyer no longer trusts the relationship.

THE SUPPLY CHAIN OBJECT

It was out of our scope in this study to perform an in-depth analysis of the supply chain object and how it is currently being used within programmatic environments. Specifically, whether any platforms currently parse the object as a form of real-time verification, or whether the chain is built for the sole purpose of buyside audits occurring sometime later.

Without checks and balances, it would be trivial to remove yourself from the chain. Consider as an example:

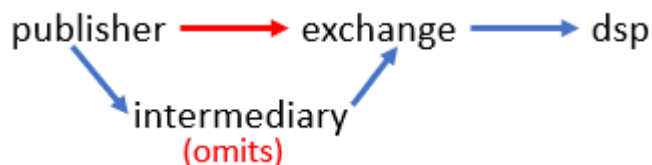
Intermediary.com sells inventory from publisher.com through exchange.com to dsp.com. Publisher.com has a seat ID of 1234 with intermediary.com. Intermediary.com has a seat ID of 5678 with exchange.com.



The serialized supply chain object for this transaction is below:

```
1.0,1!intermediary.com,1234,1,,,!exchange.com,5678,1,,,
```

Now consider that intermediary.com has sent publisher.com ads.txt lines declaring a DIRECT relationship with exchange.com. Intermediary.com can now omit themselves from the supply chain object so that dsp.com believes the DIRECT relationship to be true. Intermediary.com does this by not initiating the supply chain object, rather leaving exchange.com to create one.



The resulting supply chain object in serialized format would be:

```
1.0,1!exchange.com,5678,1,,,
```

The spec explicitly states that sellers who are reselling inventory not containing a supply chain object should create one with the complete flag set to 0 prior to inserting their node. However, exchange.com in this case may believe that it is the initial advertising system responsible for starting a completed chain due to the DIRECT relationship. It is also possible that exchange.com is not reselling the inventory into another exchange but selling directly to a buyer who transacts within exchange.com therefore creating a completed chain may be the default.

The omission of certain nodes within the supply chain object combined with receiving multiple bid requests across various seat IDs for the same DIRECT inventory may adversely affect buyer SPO (supply path optimization) efforts but can be nullified by referencing the seller type within the appropriate sellers.json file.

NOTE ON ADS.CERT

Ads.cert, once it picks up traction and becomes used routinely in the wild, will not solve for this particular issue. Ads.cert will protect from upstream parties in the chain being able to alter key elements of the original bid request.

REMEDIATION

If a buying entity is currently listening to bid requests that have a DIRECT relationship for the same domain and inventory on two or more different seats per exchange, then all requests can be treated as RESELLER until the true DIRECT seat can be identified.

If the DIRECT relationship is only coming from a single seat, the same action can be taken if the seat holder's seller type is listed as INTERMEDIARY within sellers.json.

Both cases will solve for arbitraging resold inventory masked as being DIRECT.

Concerns around ad fraud can be addressed by working with an ad fraud vendor.

It should be noted that some publishers split their inventory (i.e. by desktop, mobile, video) across multiple seats and this should be taken into consideration.

METHOD TO REPRODUCE

Since ads.txt and sellers.json files are publicly facing, these findings can be reproduced. You will need a data set of URLs, a spider that can be configured to cache files, and a storage medium that allows for sorting, grouping, and joining of the output.

1. Start with a large list of URLs (100k+)
2. Spider the ads.txt file for each of the domains caching only the DIRECT entries.
 - a. Save the domain where the file is being consumed from
 - b. For each DIRECT line, save the advertising system and the seat ID
3. Within the output, group by seat ID along with the corresponding advertising system. Count every domain in which the grouping pair appears on.
 - a. By the end of this step, you should be able to see the number of direct domains assigned to each seat ID

SEAT ID	ADVERTISING SYSTEM	# OF DIRECT DOMAINS
---------	--------------------	---------------------

4. Configure your spider to read sellers.json files.
5. For each unique seat ID and its corresponding advertising system, parse through the advertising system's sellers.json file matching the seat ID to seller ID.
 - a. Save name, domain, and seller type for each seller ID match found
6. Join this output with your previous output to form a unified view of the data. You should now be looking at:
 - a. sellers.json name
 - b. sellers.json domain
 - c. sellers.json seller type
 - d. number of domains declaring a DIRECT relationship to the seller ID
 - e. (optionally) a list of those domains

NAME	DOMAIN	SELLER TYPE	SEAT ID	ADVERTISING SYSTEM	# OF DIRECT DOMAINS
------	--------	-------------	---------	--------------------	---------------------

ABOUT ESCALATED.IO

We are a programmatic ad fraud vendor designed for use within the entire ecosystem from the sell-side to the buy-side. Our range of tools are built for the real-time measurement and blocking of ad fraud to protect ad spend or ad revenue. We work with every client to find a solution within our customizable and flexible platform that addresses the need, delivers the result, and saves companies money while providing actionable insight.

<https://escalated.io>

Kyle Smith



REFERENCES

- [1] https://iabtechlab.com/wp-content/uploads/2019/07/Sellers.json_Final.pdf
- [2] <https://iabtechlab.com/wp-content/uploads/2019/03/IAB-OpenRTB-Ads.txt-Public-Spec-1.0.2.pdf>
- [3] <https://github.com/InteractiveAdvertisingBureau/openrtb/blob/master/supplychainobject.md>
- [4] <https://github.com/InteractiveAdvertisingBureau/openrtb/blob/master/ads.cert:%20Signed%20Bid%20Requests%201.0%20BETA.md>
- [5] https://support.google.com/displayvideo/answer/7509551?hl=en#authorized_seller_targeting
- [6] <https://sellersjsons.com>