Appendix

Appendix A: Evolution Case Study

Appendix B: Methodology #1 - The widget / client ID methodology

Appendix C: Examples of In-Code Evidence

Appendix D: Live Match Coverage of Low-Level Leagues

Appendix E: Methodology 2 - Virtual game client ID

Appendix F: Examples of In Code Evidence - Virtual game client ID

Appendix A

Evolution Case Study

Evolution Case Study

UKGC's Formal Action Against a Major B2B Supplier Sets a Precedent

The UKGC is moving more aggressively toward making supplier accountability an active, proactive obligation rather than a passive reporting one — and has shown willingness to open formal license reviews against major publicly listed B2B companies within weeks of identifying unlicensed access.

The contractual obligation embedded in the LCCP (requiring B2B suppliers to mandate, in their contracts with B2C clients, that UK players are served only via licensed operators) creates a paper trail that the UKGC can test directly through test purchasing — a tool the Commission has explicitly said it will use.

The December 2024 investigation in Evolution (EVO.ST) was a landmark precedent. Under Sections 116 and 117 of the 2005 Gambling Act, the Gambling Commission can commence a license review when it suspects a licensing condition has been breached or the license holder may no longer be suitable.[1] That is precisely the mechanism the Commission triggered against Evolution.

The possible outcomes ranged from no further action being taken, to the imposition of license conditions or financial penalties, or even the suspension or revocation of Evolution's operating license. Evolution responded by cooperating fully with the investigation and implementing corrective measures, including disabling access to its games on websites identified as lacking a UKGC licence.[2]

The market reaction was stark — EVO's share price fell 11.8% on Dec 21, 2024 in response to the news and down approximately one-third since that day with sell-offs coming as additional reports covering the scandal broke.[3]

The Evolution case is significant not only for what it did but what it signaled. It is the first time a supplier was officially investigated by a regulator for its B2B customers' activities.[4]

The UKGC was extending liability upstream from the operator to the content supplier — a materially different posture from most regulators historically.

[1] Vixio: https://www.vixio.com/insights/gc-evolution-facing-uk-licence-review , [2] https://next.io/news/regulation/ukgc-commences-review-evolution-licence/
[3] https://www.bloomberg.com/news/articles/2025-08-13/evolution-games-ran-in-banned-markets-execs-say-in-secret-tapes , [4] https://sbcnews.co.uk/europe/2025/01/10/fight-against-black-jp-sbc/

Appendix B

Methodology 1—Client ID & Widget Loader

Appendix B – Methodology 1: Client ID & Widget Loader

In order to identify and discriminate which operators are and are not SRAD customers with a high degree of confidence, we retained a third-party cyber security expert to analyze the system architecture, code, and online data exchanges. To check our understanding of the code we spoke with two former SRAD IT professionals, including one very senior with firsthand knowledge of the system architecture. He confirmed and explained the use of the Client ID and 32-character encryption code pair as one of SRAD's key authentication methods.

Based solely on data exchanged between SRAD and sports betting websites, it is possible to identify SRAD Client IDs and perform online licensing checks for SRAD services. This Appendix highlights the technical mechanisms that underpin these processes.

To summarize, this method uses the LigaStavok client ID and 32-character code as an example to illustrate a process that an operator would have gone through in order to establish a SRAD-authenticated website.

Overview of the Eight Step Process:

1. Ligastavok requests a Client ID from SRAD and complete the SRAD onboarding process.

2. Implement their assigned SRAD Client ID (in this case: "a53878289ac56ca4694405e80659d93f") on the *ligastavok[.]ru* website to load non-sensitive SRAD data and widgets.

3. Configures its SRAD account to approve the web domain *ligastavok[.]ru* to use the Client ID a53878289ac56ca4694405e80659d93f to load licensed and more sensitive data from SRAD.

4. Ligastavok engineers generate encryption keys using *openssl*, installing one half of the keys on the ligastavok[.]ru website.

5. Ligastavok engineers provide SRAD with the other half of the encryption key, who install it on SRAD servers.

6. The *ligastavok[.]ru* website validates their encryption key by requesting the */licensing* SRAD endpoint and by receiving the *test* value back.

7. The *test* value enables *ligastavok[.]ru* and SRAD to validate the correct encryption key is being used for the correct Client ID and from the authorized web domain *ligastavok[.]ru*.

8. Ligastavok[.]ru now cryptographically 'signs' sensitive messages to SRAD, using their encryption keys and are able to access additional SRAD services such as the Live Match Tracker or Bet Concierge services.

Appendix B – Methodology 1: Client ID

The SRAD Client ID

For a sports betting site to incorporate SRAD data, the betting site must first obtain a SRAD *Client ID,* commonly known as an 'API key' on other platforms such as Twitter. SRAD defines a *Client ID* as:

"*...a 32 characters long id which is automatically created when a client account is created. The Client ID is binded to all of the orders under one account and will work for all of the products enabled under that account. In order for us to create an account, contact our Sales Team.*"[1]

Much like other online services, a prospective SRAD customer needs to go through an onboarding process with SRAD in order to obtain a Client ID. The customer then uses this Client ID on their website or in their mobile application to utilize SRAD data and products.

[1] https://widgets.sir.SRAD.com/docs/tutorial-faq.html#client-id

Appendix B – Methodology 1: Client ID – Cont.

SRAD Infrastructure and Data Exchange

When observing SRAD traffic and performing web socket analysis, our cyber security expert observed the SRAD Client ID being used when betting websites are loading SRAD 'widgets' into their own websites. Our cyber security expert believes the Client ID is present in Akamai CDN URLs, for example, the URL to load a SRAD website widget for *baaji365[.]live* is:

//ws-cdn001.akamaized.net/3d768df46af393a8bb3b68b7b57f4e4a/widgetloader

The SRAD URL can be broken down into sub-components to better understand its structure:

ws-cdn001.akamaized.net - This is a custom Akamai domain, that is owned and managed by SRAD and not shared by other Akamai customers per Akamai documentation.

3d768df46af393a8bb3b68b7b57f4e4a - a 32-character Client ID, that is assigned to a SRAD client after onboarding, in this case baaji365[.] uses this particular Client ID.

/widgetloader - the JSON / HTML code that loads the actual "widget" to display SRAD data on the betting website.

The */licensing* endpoint is a way to authenticate that the domain of the sports betting website, using the appropriate Client ID in the URL, is *authorized by SRAD* to use their service.

Appendix B – Methodology 1: Client ID – Cont.

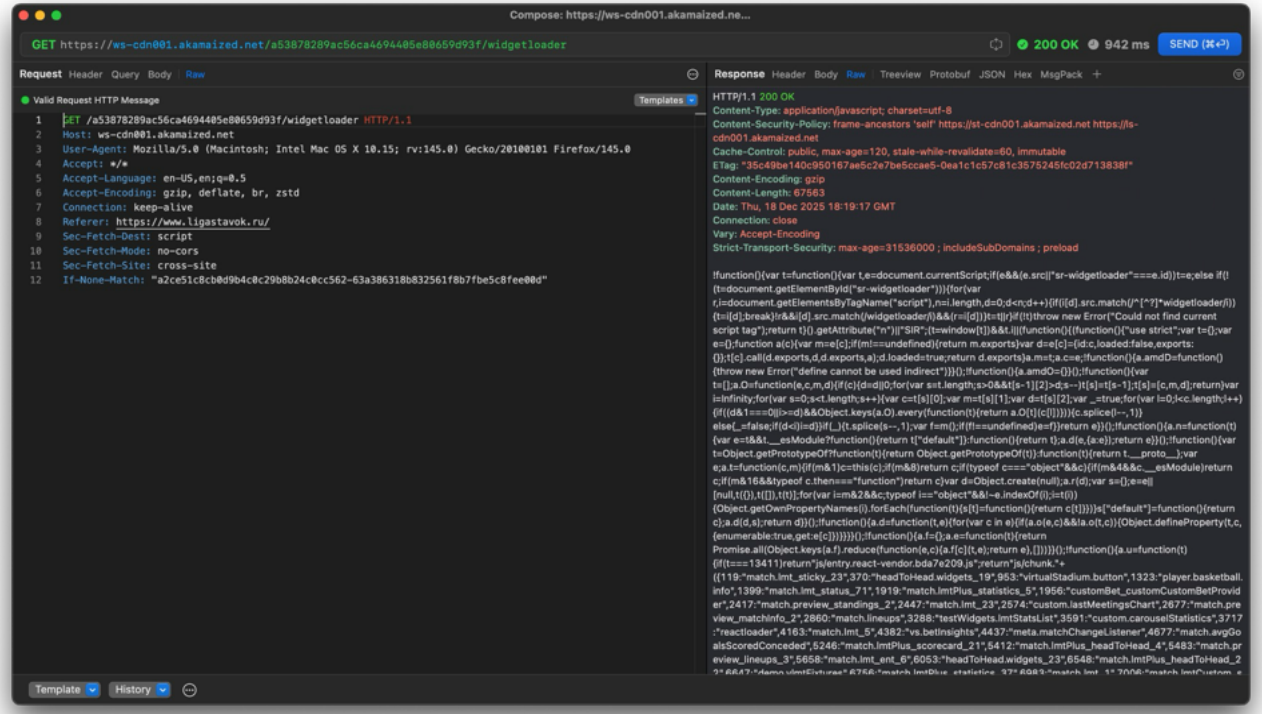
Example: ligastavok[.]ru

To compare, first browse to *ligastavok[.]ru* and examined the web traffic.

See Figure A.

As shown, the */widgetloader* mechanism with a different *Client ID* used in the Akamai URL is used on *ligastavok[.]ru* in the identical fashion as *baaji365[.]live*

Figure A - ligastavok[.]ru loading the SRAD widget via Akamai CDN



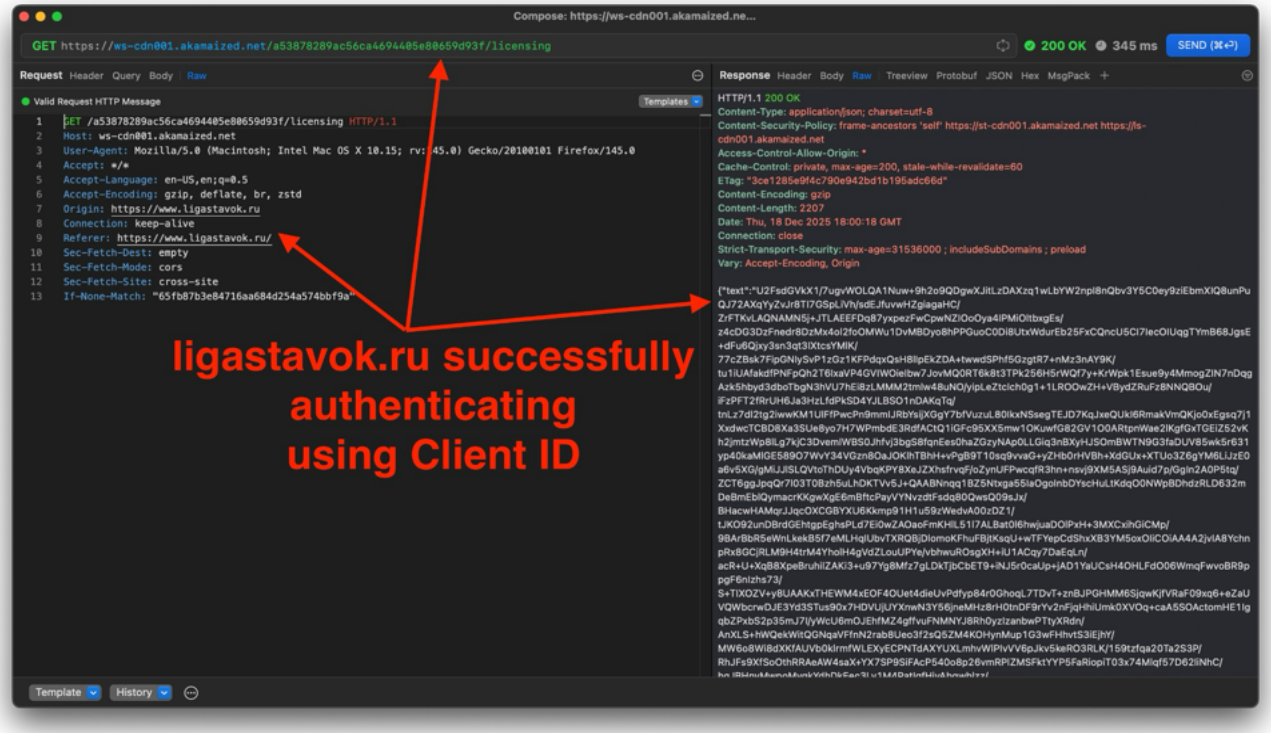
Appendix B – Methodology 1: Client ID – Cont.

Example: ligastavok[.]ru – **Pass**

Continuing the analysis, when the */licensing* endpoint is requested, the SRAD infrastructure responds successfully for *ligastavok[.]ru* – and now presents returned data that indicates there is an active client license active and available.

See Figure B.

Figure B - ligastavok[.]ru successfully authenticating to SRAD licensing via Akamai CDN.



Appendix B – Methodology 1: Client ID – Cont.

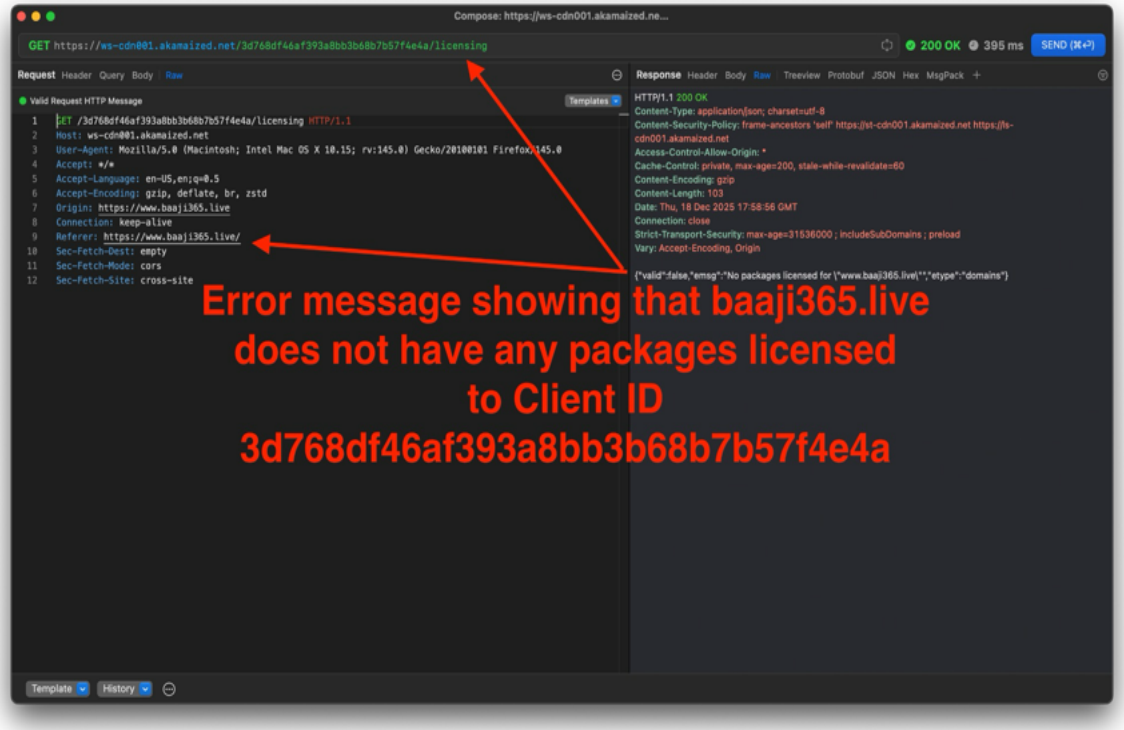
Example: *baaji365[.]live* - *Fail*

The is an example at right is of a failed test.

The key fails likely either because it is an an old or out-of-date *Client ID,* or possibly inappropriately taken a *Client ID* from another website.

See Figure C.

Figure C - *Baaji365[.]live* failing to authenticate to the SRAD services using the referenced Client ID.



Appendix B – Methodology 1: Client ID – Cont.

This three step comparative check shows there is strict domain, and Client ID checking on the */licensing* endpoint. The *Referer* and *Origin* HTTP headers must be set to *ligastavok[.]ru* and use the Client ID shown, otherwise the request will fail or show a 404 not found.

We believe this */licensing* check is a way to disambiguate between websites that may have scraped a SRAD Client ID (i.e. *baaji365[.]live*) and those that appear to have an active contract with SRAD – as is the case with *ligastavok[.]ru*.

There is an additional technical detail that helps to establish that the Client ID is not the *only* mechanism that SRAD uses for authenticating their client's websites – there is also a cryptographic key exchange that occurs for some services that our cyber security expert identified.

Appendix B – Methodology 1: Encryption Key Exchange

Encryption Key Exchange

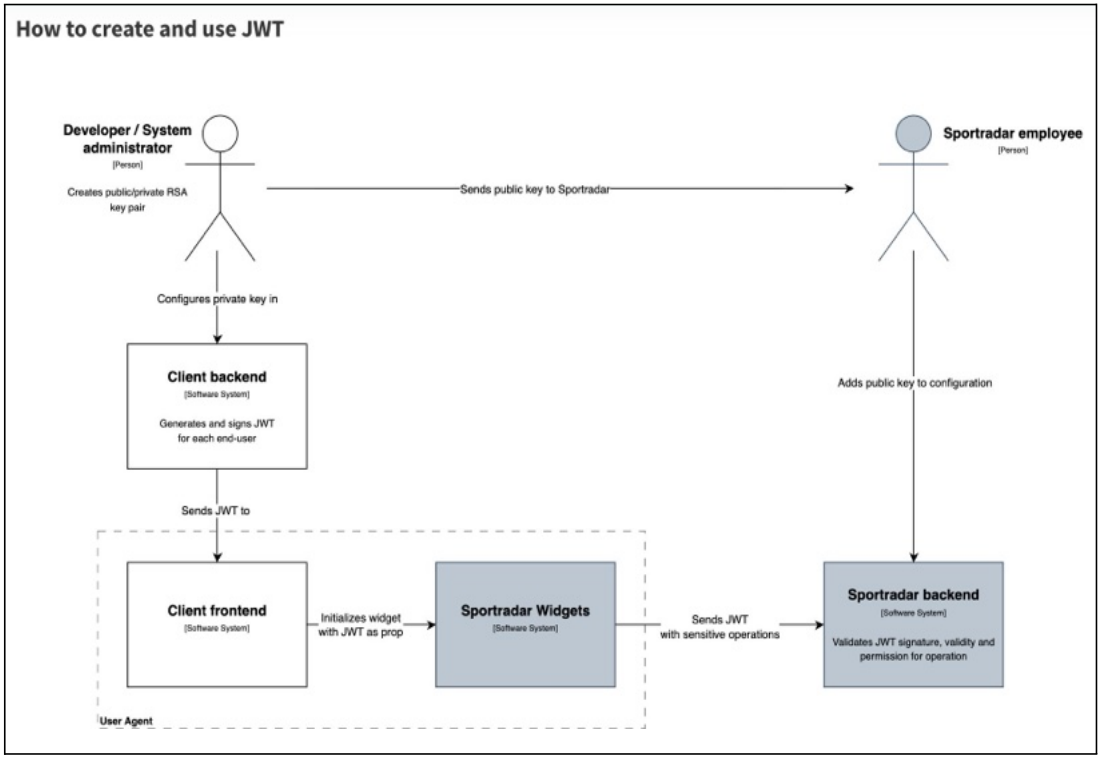
The SRAD documentation explains that there are sensitive data operations that require an additional layer of security to be in place between SRAD and their customer.

The means for protecting these sensitive operations is to use encryption keys, along with specific Json web 'tokens' (JWT) that are signed by the encryption keys. Cryptographic 'signing' is synonymous to signing a document with a pen – a unique 'signature' is applied to a piece of information such as a sports bet, a document, a bank transaction or a stock trade, for example.

For SRAD, there is a *requirement* to set up an encryption key and to authenticate sensitive data requests for SRAD-related products including:

"*Audio/Video integration into Live Match Tracker; Login into Virtual Stadium; Bet Concierge authentication.*"[1]

Figure D - SRAD diagram from SRAD documentation highlighting the role of encryption keys for accessing sensitive SRAD services[1]



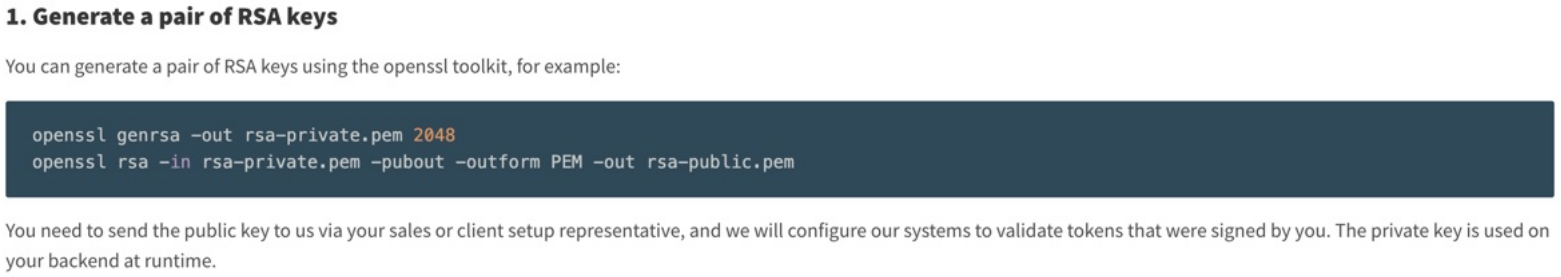
[1] https://widgets.sir.SRAD.com/docs/tutorial-jwt.html

Appendix B – Methodology 1: Encryption Key Exchange Cont.

The SRAD documentation further explains how to generate the necessary encryption keys, how to exchange them with SRAD, and the technical implementation details for using the keys. In the technical notes, one can see that they are using *openssl*[1] to generate the encryption keys.

The instructions explain that the operator needs to send their public key to SRAD to configure its systems and validate tokens. The private key is used on the operator's backend for the programs to run.

Figure E - SRAD documentation detailing how to use *openssl* to generate the necessary encryption keys



1. Generate a pair of RSA keys

You can generate a pair of RSA keys using the openssl toolkit, for example:

```
openssl genrsa -out rsa-private.pem 2048
openssl rsa -in rsa-private.pem -pubout -outform PEM -out rsa-public.pem
```

You need to send the public key to us via your sales or client setup representative, and we will configure our systems to validate tokens that were signed by you. The private key is used on your backend at runtime.

[1] https://widgets.sir.SRAD.com/docs/tutorial-jwt.html

Appendix B – Methodology 1: Encryption Key Exchange Cont.

Our expert determined that data returned from the */licensing* endpoint appears to be part of the SRAD-related encryption and verification process that is detailed in their developer documentation.

For example, the following snippet of data is from *ligastavok[.]ru* requesting the */licensing* endpoint:

Figure F - JSON data being returned from the /licensing endpoint to ligastavok[.]ru.

```
JSON
{

        "text"  :"U2FsdGVkX19PVX+lsBaWKQUV…  <CLIPPED  FOR  BREVITY>...==",
        "valid":true

}
```

Appendix B – Methodology 1: Encryption Key Exchange Cont.

The text key returned (Figure F) from the /licensing endpoint is Base64 encoded in its original form. By decoding the data, our cyber security expertwas able to export it to hexadecimal format. See Figure G.

PRR analyzed the raw data, and determined the test key returned from the /licensing endpoint is openssl-encrypted data. The format of the data matches the format and mechanism that SRAD describes in its documentation for generating encryption keys with openssl.

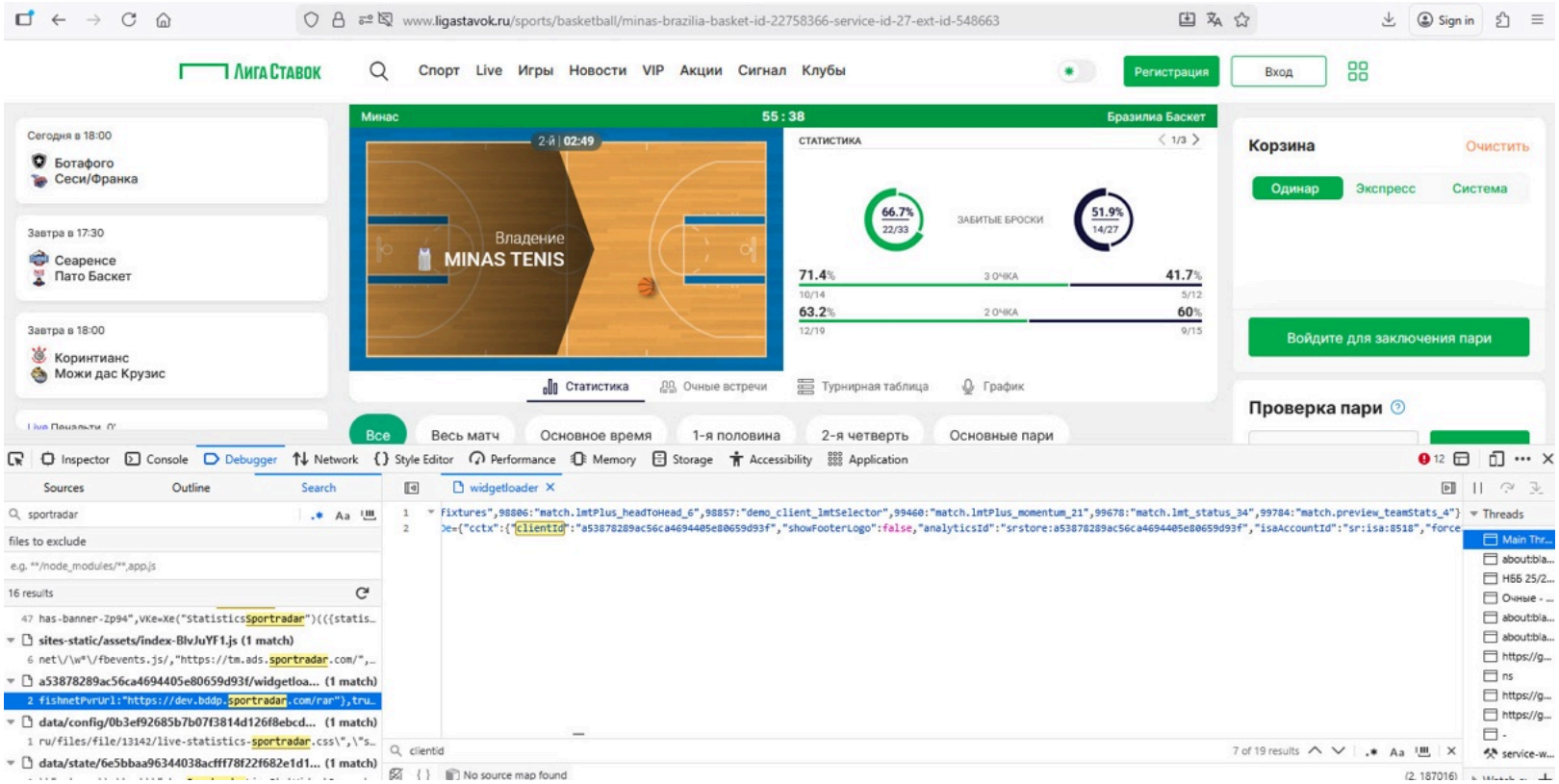
Figure G - Base64 decoded data returned from /licensing endpoint that is detected as *openssl* encrypted data.

Shell

```
00000000  53 61 6c 74 65 64 5f 5f  fb ba 0b d6 38 b4 00 d4  |Salted__....8...|
00000010  db b0 fb d8 76 a3 d4 03  83 05 c9 8a d2 f3 0c 05  |....v...........|
00000020  f3 ab 5c 0b 6d 85 b6 9e  92 3c 9d 06 ef dd 8e 42  |..\.m....<.....B|
00000030  d1 ec bd ce 21 1b 99 72  10 f2 e9 cf b9 02 7b d8  |....!..r......{.|
00000040  05 ea 63 26 6f 26 bf 13  23 b1 92 a4 b8 95 87 fb  |..c&o&..#.......|
00000050  1d 10 97 ee bf 01 d9 82  26 a0 68 70 bf 66 b1 53  |........&.hp.f.S|
00000060  2a f2 c0 40 d0 0c 37 98  fe 25 32 c0 10 41 43 ab  |*..@..7..%2..AC.|
00000070  ce f2 c6 97 b3 17 00 a9  c0 d6 48 3a 83 b2 6b 82  |..........H:..k.|
00000080  0f 32 23 a5 b5 bc 60 12  cf f3 e1 c0 c6 dc 3c c5  |.2#...`.......<.|
00000090  9d e7 6b f0 3c cc c7 8a  25 d9 fa 0e 31 6b b5 0e  |..k.<...%...1k..|
000000a0  f3 01 0f 2a 3c 84 f3 c6  ba 80 b4 0e 2f 14 b7 15  |...*<......./...|
000000b0  9d ba b1 1b db 91 71 09  09 dc 53 90 88 ec 87 9c  |......q...S.....|
000000c0  38 85 2a 81 36 26 07 af  09 82 c1 3e 74 5b ba 42  |8.*.6&.....>t[.B|
000000d0  3c 72 de c9 f7 aa dd c8  5e d7 2c 60 c9 4a ff be  |<r......^.,`.J..|
000000e0  dc 64 1b 24 ec 58 a9 18  d2 32 4a f3 f5 cc 6c f5  |.d.$.X...2J...l.|
000000f0  28 53 dd ab 14 2c 1f c9  48 a4 49 19 0c 0f ad c3  |(S...,..H.I.....|
00000100  07 52 3e 17 f9 1b 38 2d  47 bf a7 33 3d e7 01 8f  |.R>...8-G..3=...|
```

Appendix B – Methodology 1: End Result

LigaStavok.ru Widgetloader Visibly Running on SRAD Datafeeds



Appendix B – Methodology 1: Encryption Key Exchange Conclusion

Conclusion:

The presence of a SRAD customer ID and 32 character is an extremely strong indicator that an operator is a SRAD customer who has been onboarded and directly connected to SRAD for the purpose of transferring data. This methodology is specifically developed to differentiate official, onboarded customers from unauthorized data scrapers.

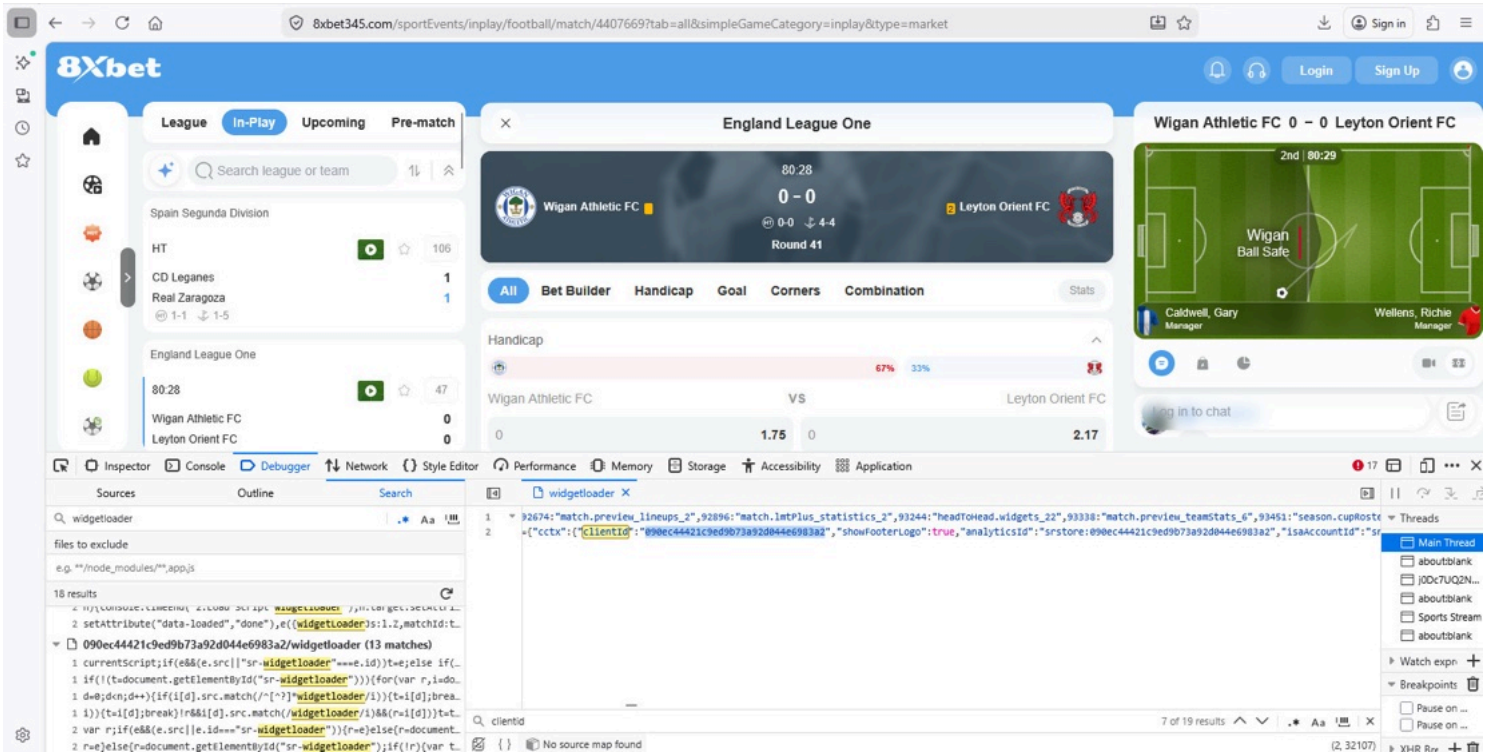
Appendix C shows additional example of other illegal sportsbooks sites found to be powered by SRAD identified using this methodology.

Appendix C:
Methodology 1
Examples of the Widgetloader Client ID

Appendix C: Examples of In-Code Evidence

8xBet345.com Powered by SRAD

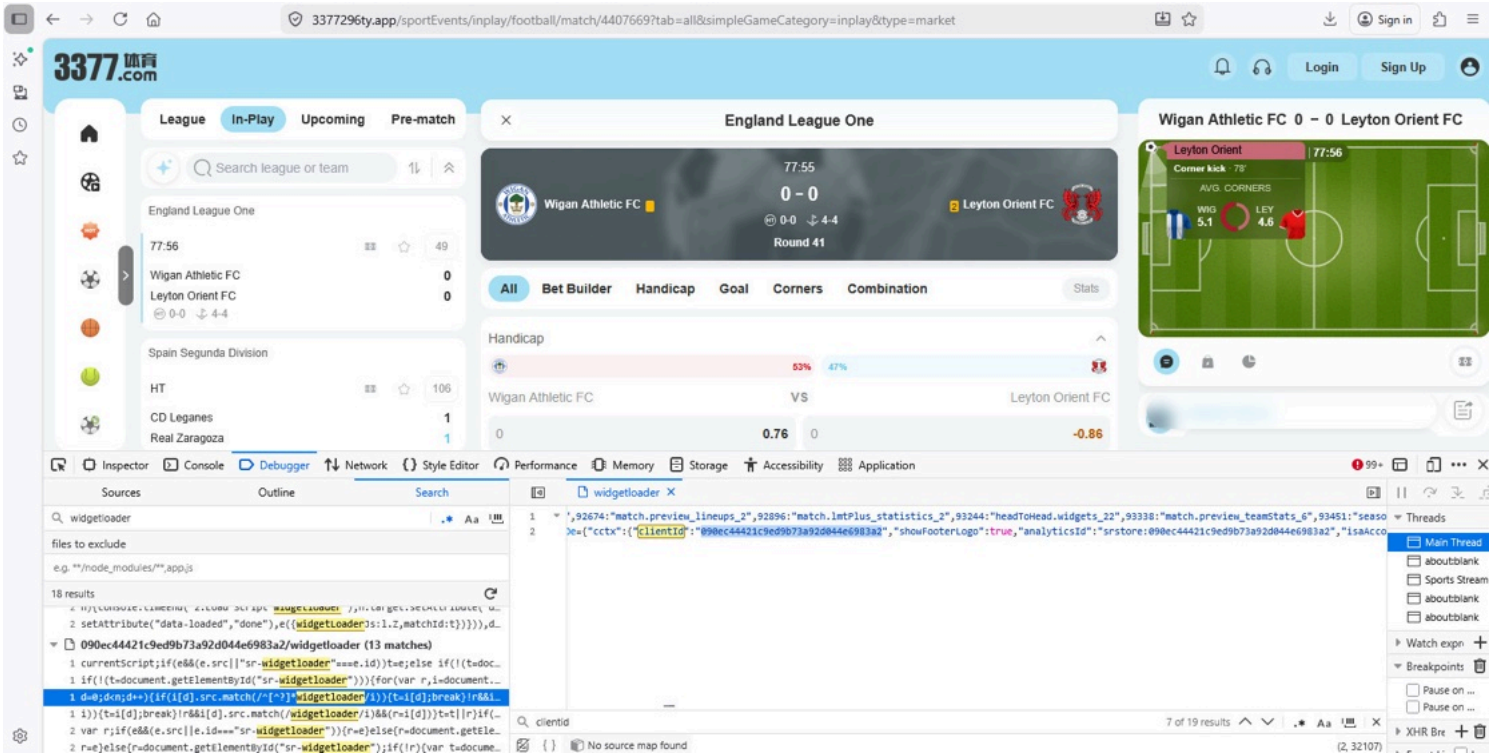
Client ID#: 090ec44421c9ed9b73a92d044e6983a2



Appendix C: Examples of In-Code Evidence

3377.Com (8xBet's China Clone) Powered by SRAD

Client ID#: 090ec44421c9ed9b73a92d044e6983a2

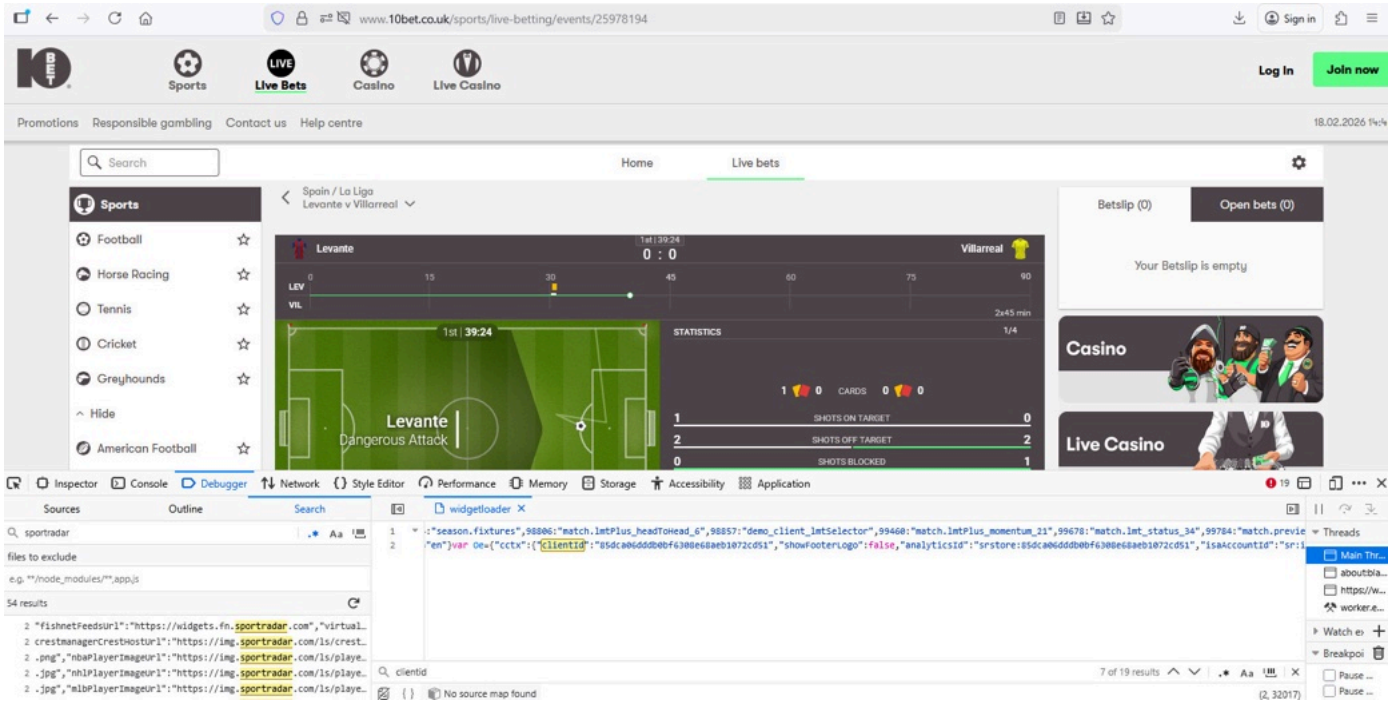


Appendix C: Examples of In-Code Evidence

10bet.co.uk Powered by SRAD

Note - 10bet.co.uk uses a different widgetloader ClientID than the 10Bet.com and 10tenbet.com

www.10bet.co.uk Client ID#: 85dca06ddd0bf6308e68aeb1072cd51

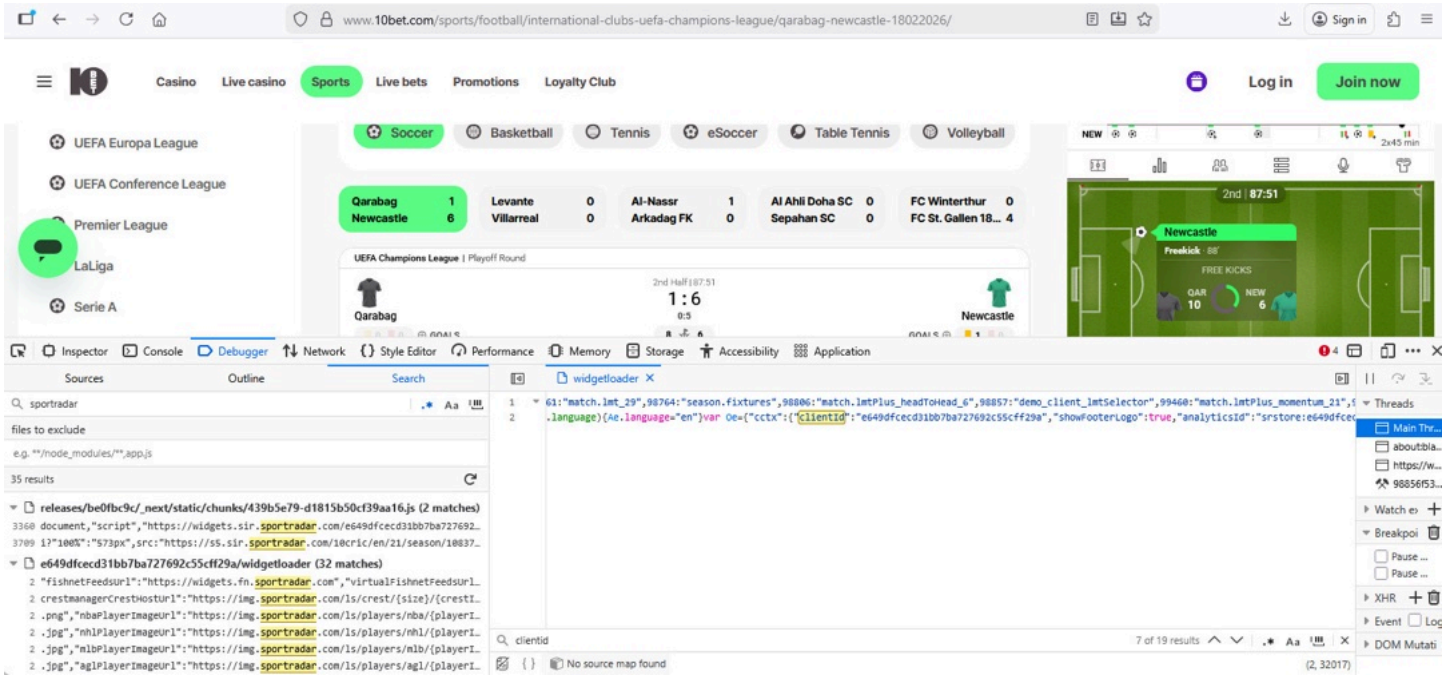


Appendix C: Examples of In-Code Evidence

10bet.com Powered by SRAD

Both 10bet.com and its Korean clone10tenbet.com use the same SRAD client ID# and widget loader.

Client ID#: e649dfcecd31bb7ba727692c55cff29a

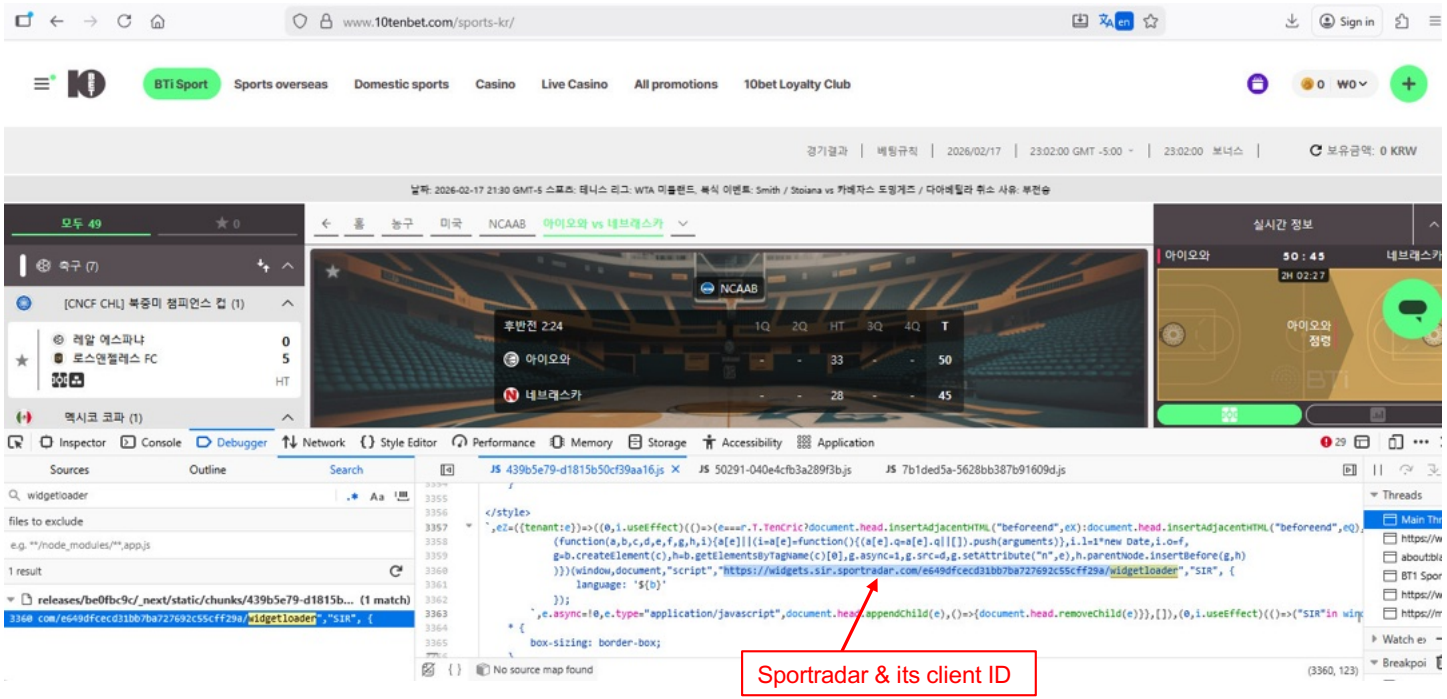


Appendix C: Examples of In-Code Evidence

10tenbet.com Powered by SRAD

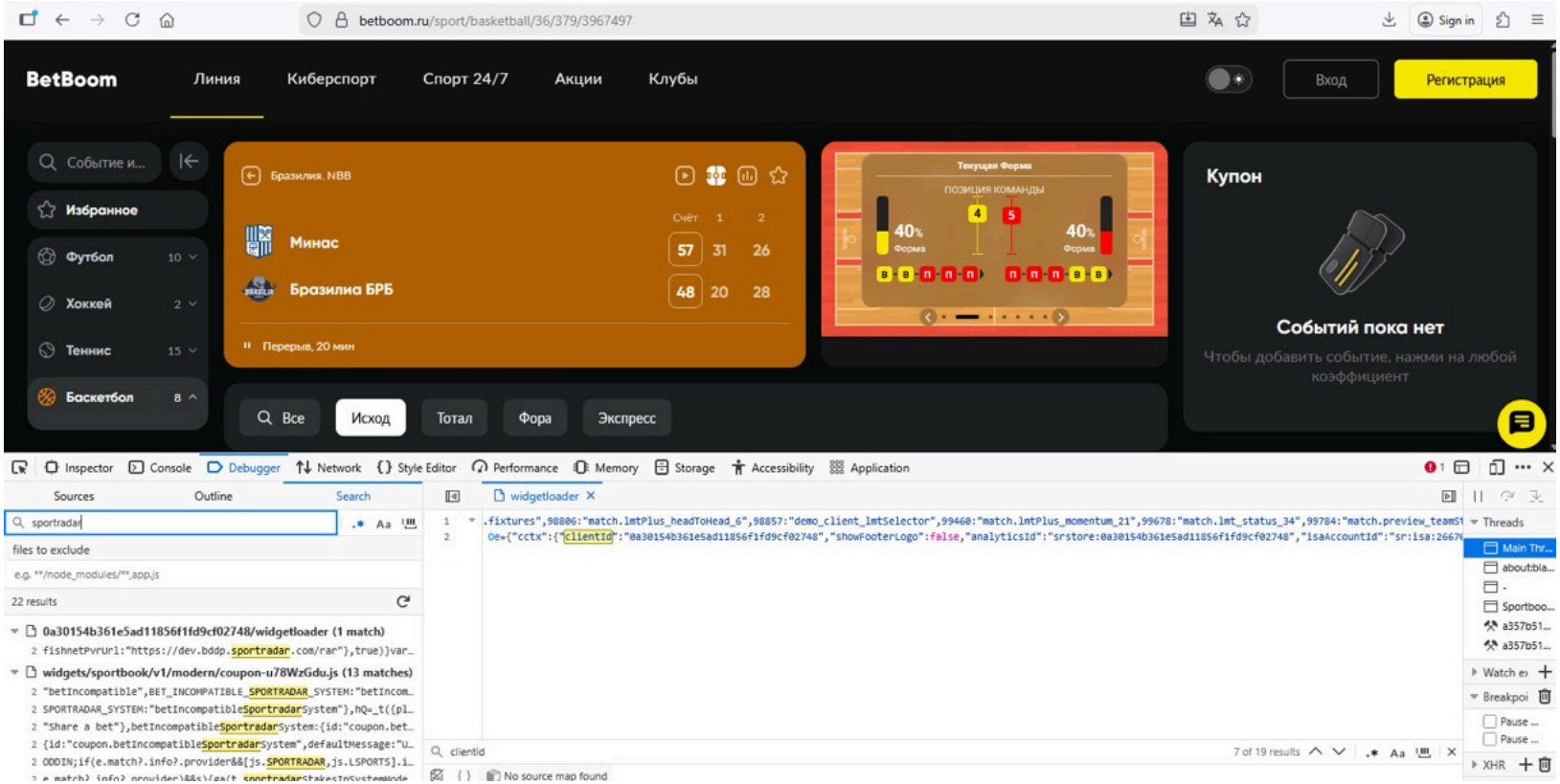
Although the Korean site 10tenbet runs BTi's Sportsbook, the same SRAD Widget Loader Client ID ran on 10bet.com and 10tenbet.com.

Client ID#: e649dfcecd31bb7ba727692c55cff29a



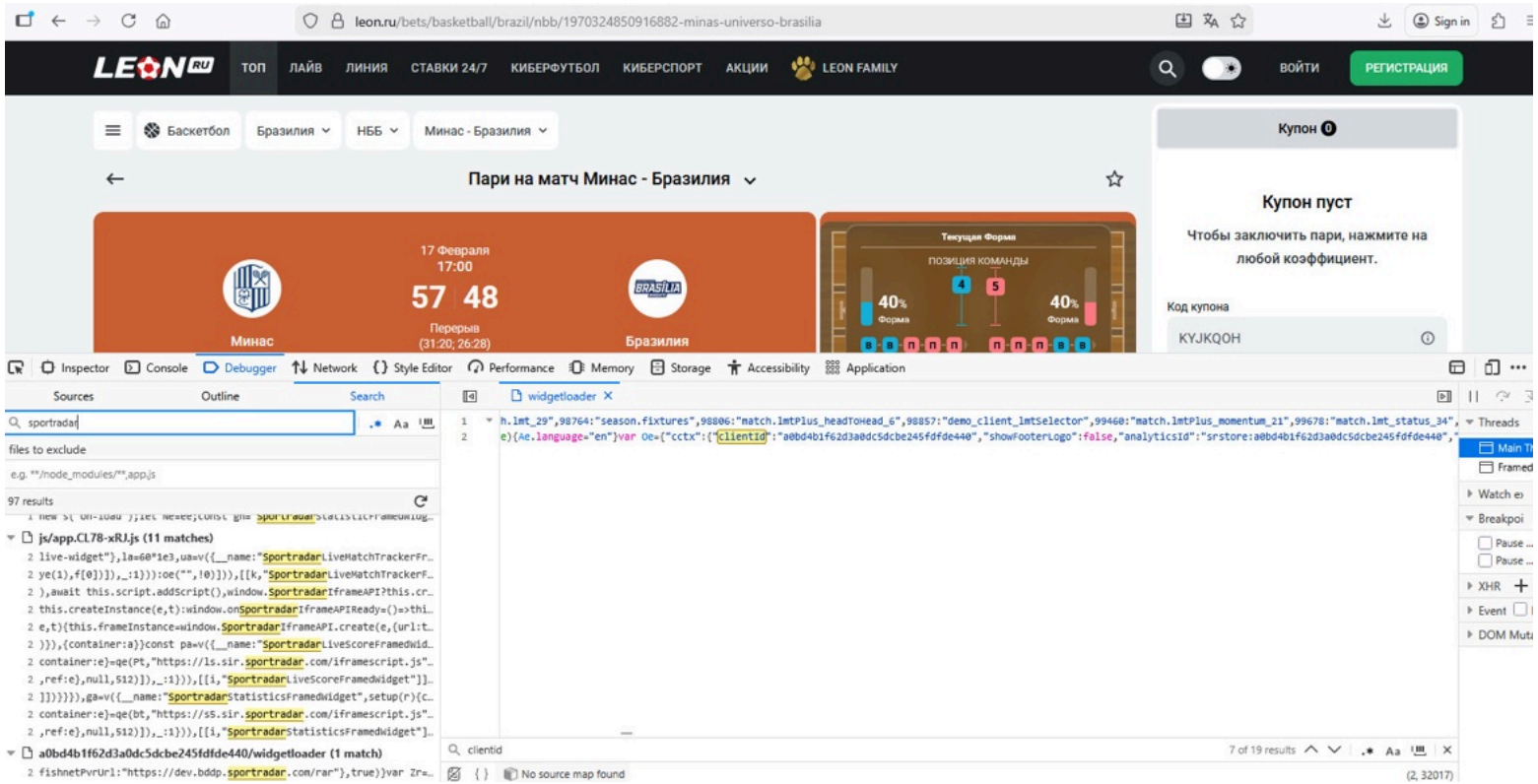
Appendix C: Examples of In-Code Evidence

Betboom.ru – a Russian Sportsbook Powered by SRAD



Appendix C: Examples of In-Code Evidence

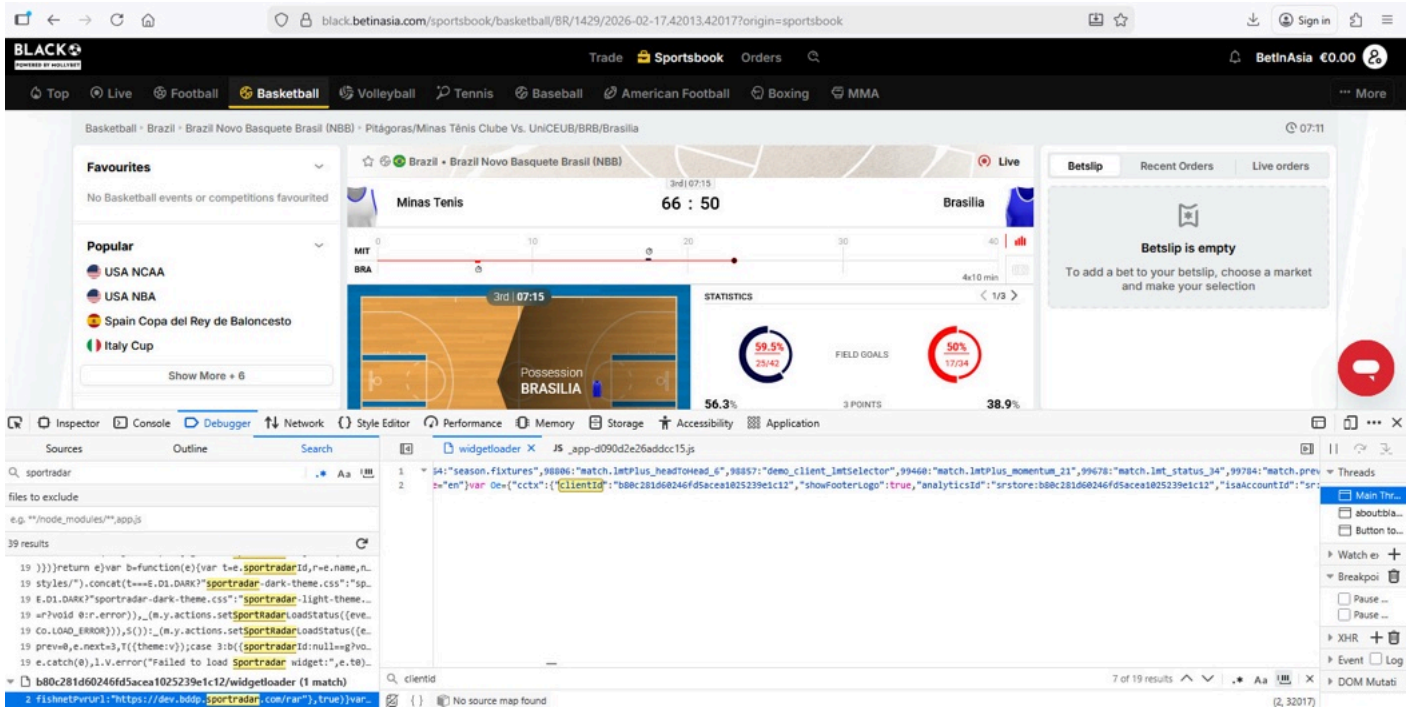
Leonbet.ru – a Russian Sportsbook Powered by SRAD



Appendix C: Examples of In-Code Evidence

Mollybet* via Betinasia – a Sportsbook Powered by SRAD

Mollybet is a sportsbook which accesses SRAD's live match tracker via the 32-character widgetloader client ID: b80c281d60246fd5acea1025239e1c12. It shares this 32-character widgetloader client ID with Cloudbet.

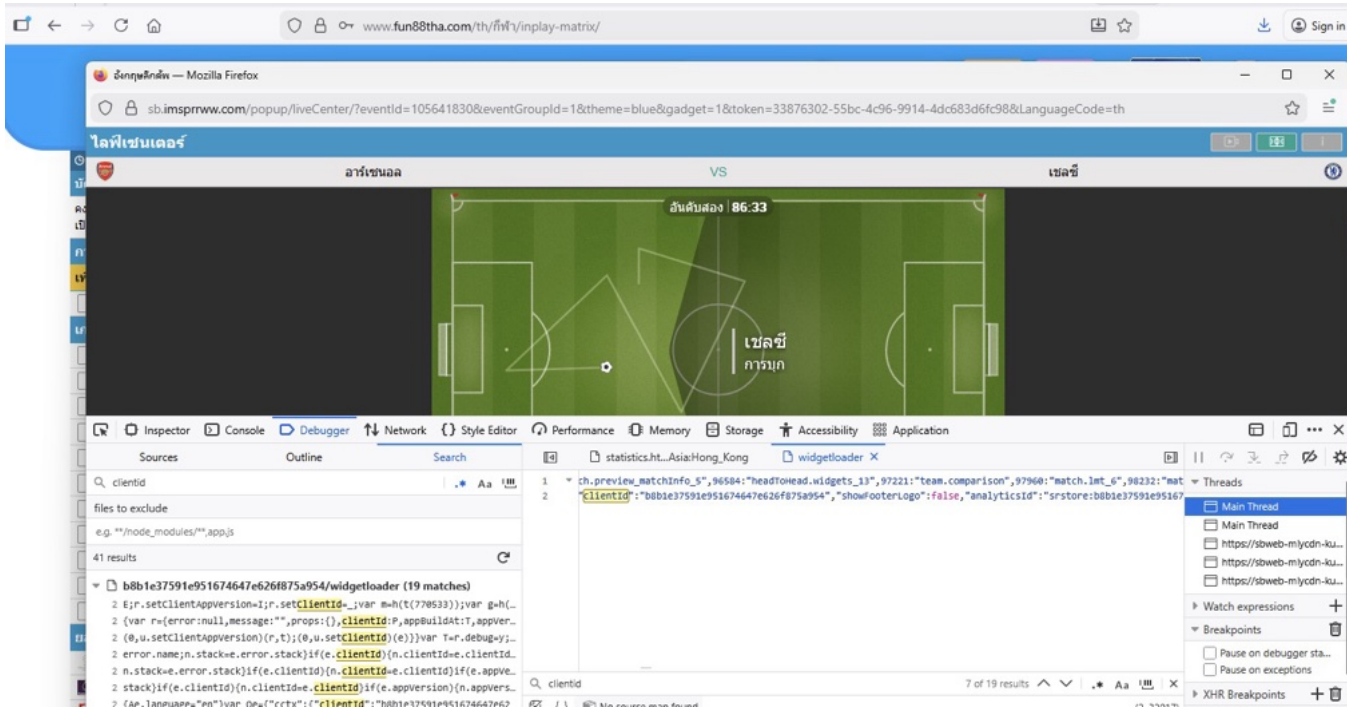


*formerly SingBet

Appendix C: Examples of In-Code Evidence

IM Sports in Thailand Powered by SRAD, Accessed by Fun88tha.com

Client ID#: b8b1e37591e951674647e26f875a954



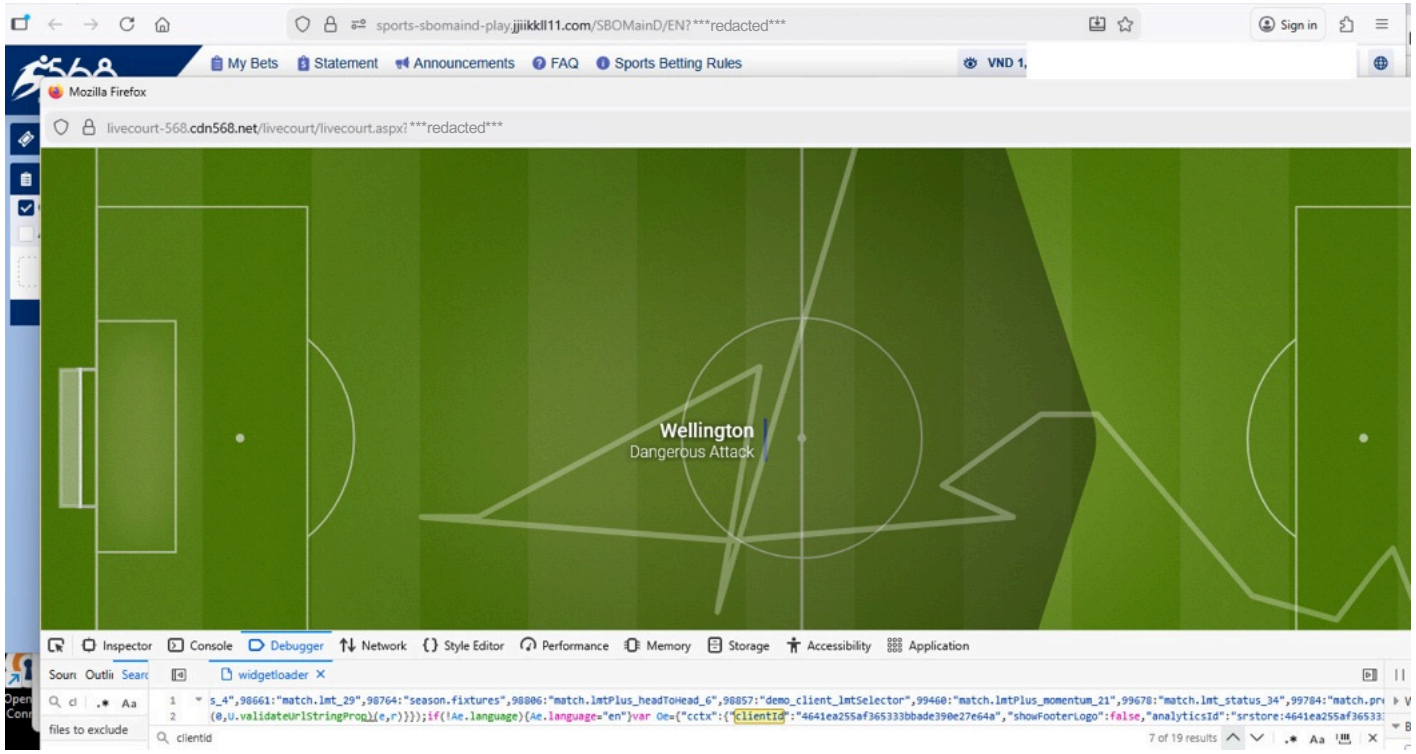
Appendix C: Examples of In-Code Evidence

IM Sports in P.R.China Powered by SRAD Accessed Via Leyu (Yabo)

Client ID#: b8b1e37591e951674647e26f875a954

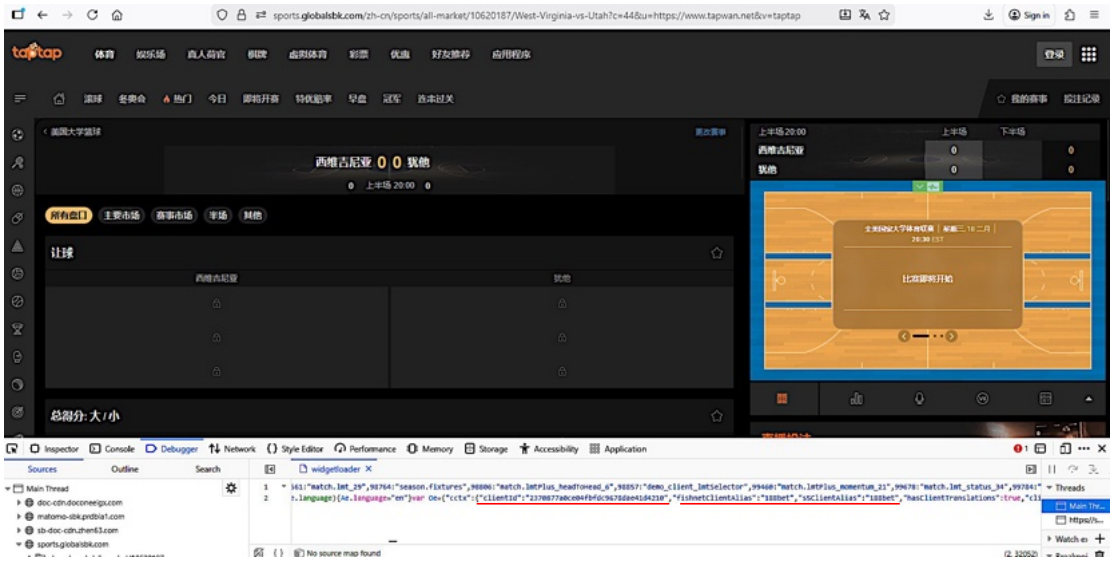
Appendix C: Examples of In-Code Evidence
SBO Powered by SRAD to Run the Bet568 Sportsbook

Below: SBO is running a SRAD Live Match Tracker. The 32 character widgetloader clientID is visible at the bottom of the page using the browser's developer tools.

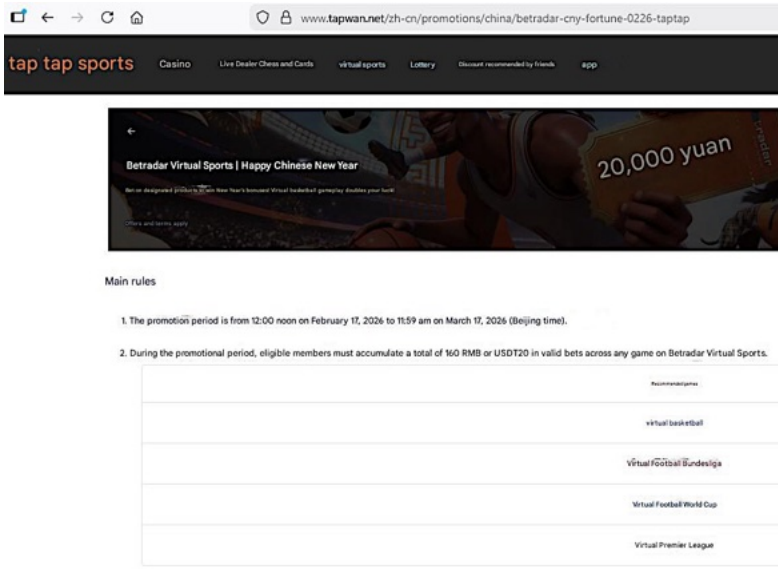


Appendix C: Examples of In-Code Evidence
TapTap Powered by SRAD

Below: TapTap's 32-character widget loader ClientID: 2370877a0ce04fbfdc9678dae41d4210 matches that used by 188bet, XJ Games, and Crown Sports indicating they are all connected as one SRAD customer. Its "client alias" is also listed as "188bet".

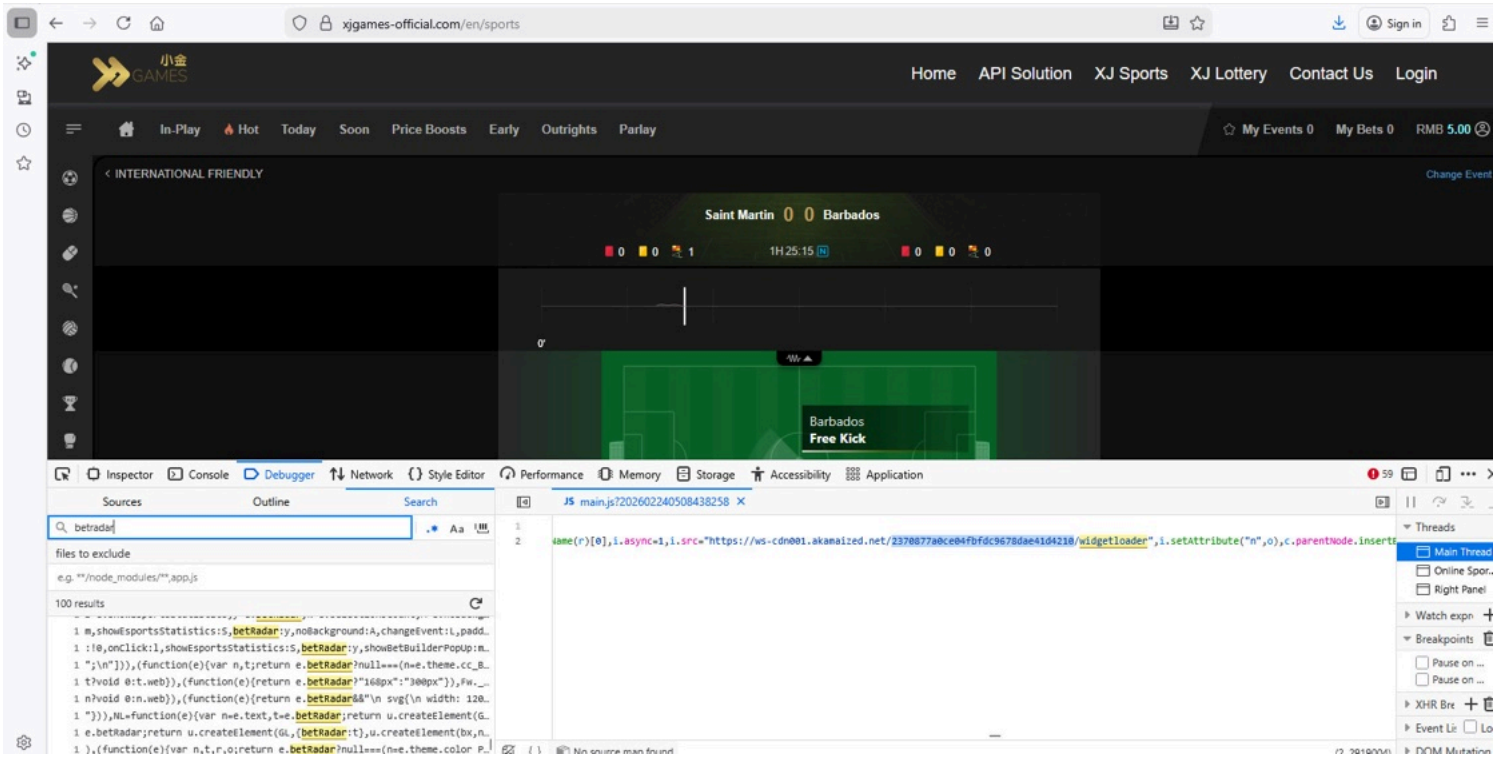


Below: during Chinese new year 2026, Tap Tap Sports offered a special promotion to use BetRadar virtual sports, further confirmation it is a SRAD customer, and using multiple products



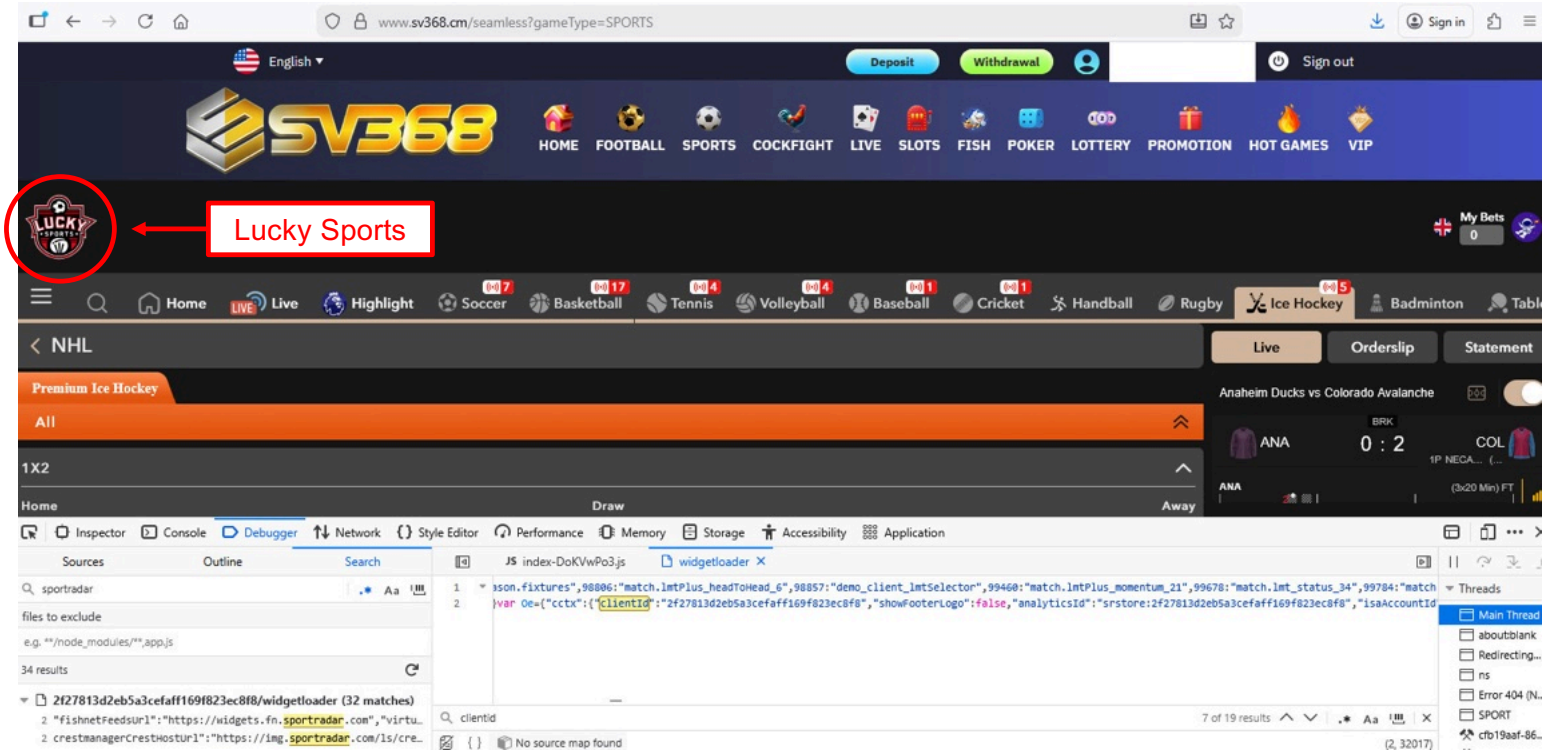
Appendix C: Examples of In-Code Evidence
XJ Games (小金) Powered by SRAD

Below: XJ Games 32-character widget loader ClientID: 2370877a0ce04fbfdc9678dae41d4210 matches that used by 188bet, TapTap, and Crown Sports indicating they are all connected as one SRAD customer running betradar.



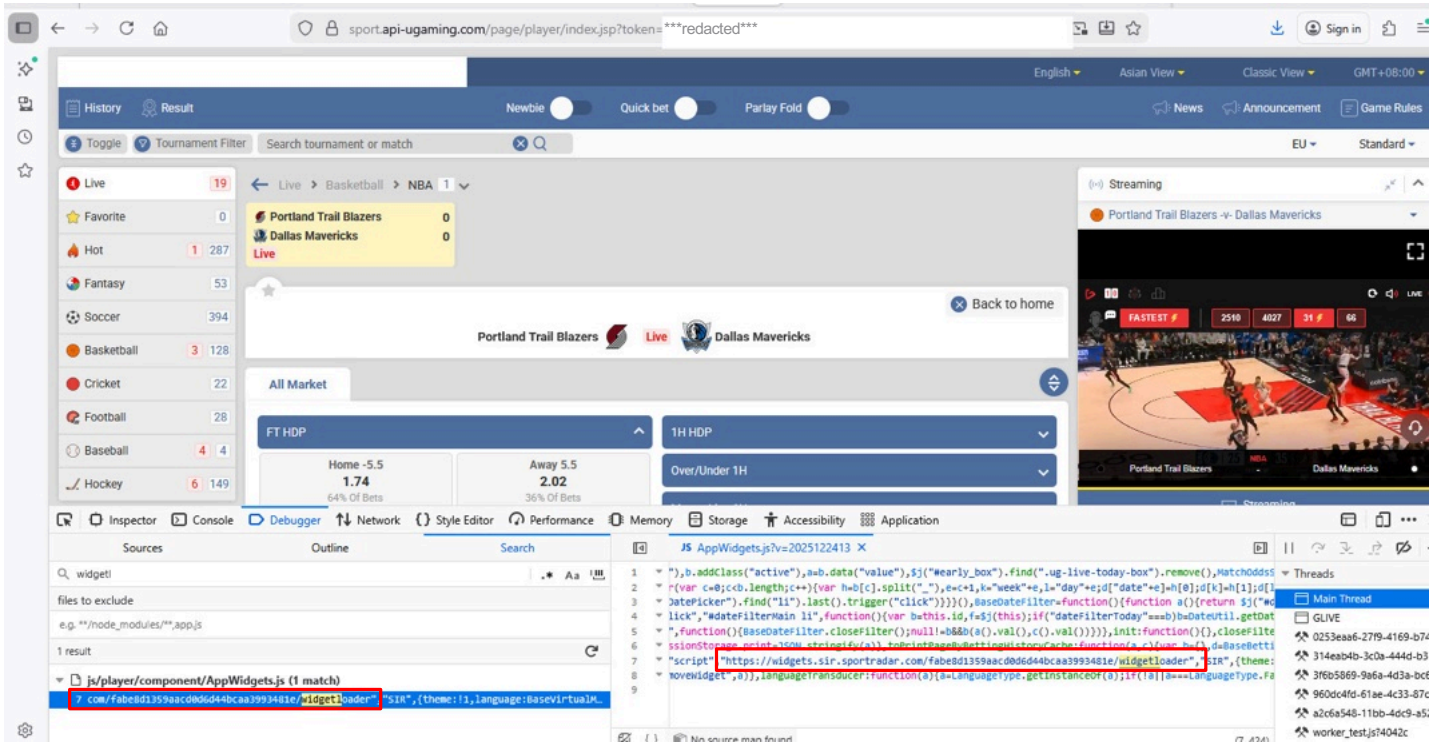
Appendix C: Examples of In-Code Evidence
Lucky Sports on SV368– widget loader

Below: Lucky sport is running a SRAD Live Match Tracker. This site originally appeared on Sv368.cm (Vietnam) with a few other API providers.
The 32-character widgetloader ClientID (2f27813d2eb5a3cefaff169f823ec8f8) is visible at the bottom of the page using the browser's developer tools.



Appendix C: Examples of In-Code Evidence
United Gaming (UG)

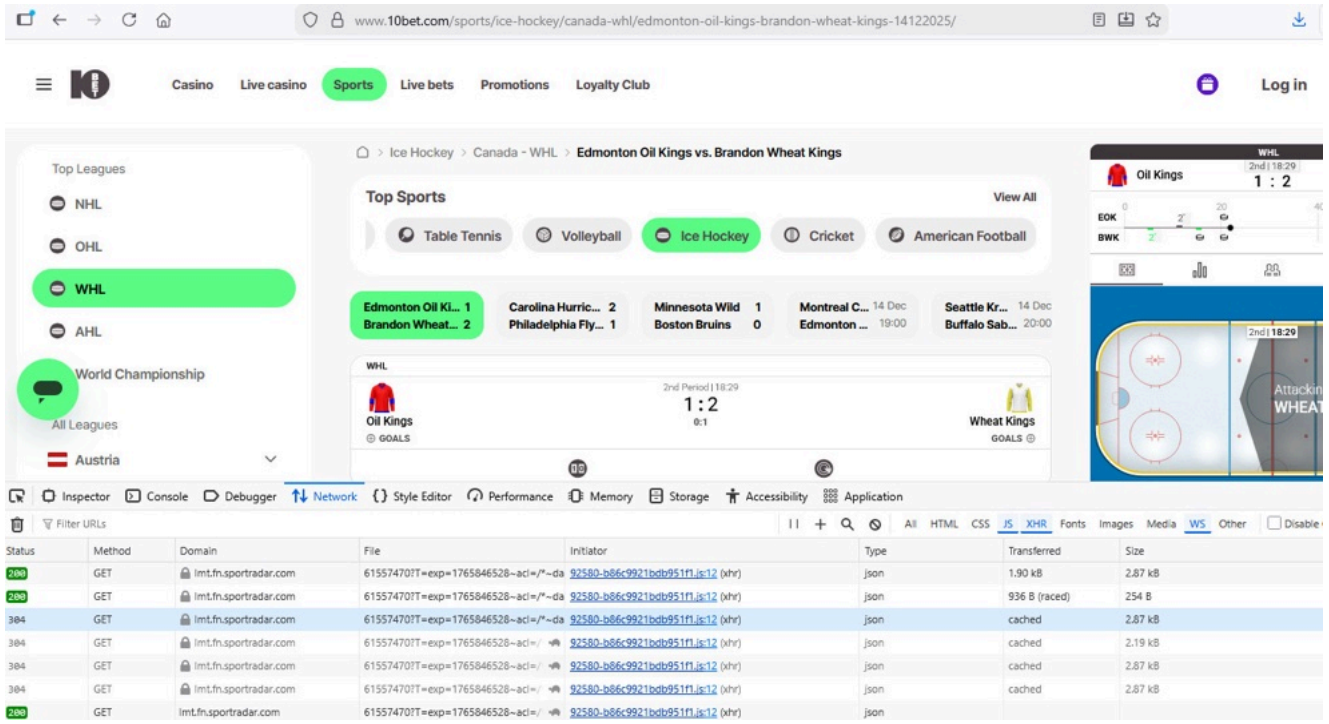
Below: UG Sports (United Gaming) is loading SRAD data via the widget loader. The SRAD 32-character unique ClientID (fabe8d1359aacd0d6d44bcaa3993481e) is visible in the “widgets.sir.SRAD” URL. This site was accessible via both bj88bet & sv368.cm which integrated UG Sports into their platforms’ offerings.



Appendix – C: Live Match Coverage of Low-Level Leagues

SRAD Powering a WHL (Canadian Junior Hockey League) Live Match Tracker on 10bet

Below: 10bet using SRAD's Live Match Tracker of the Edmonton Oil Kings vs. Brandon Wheat Kings, Two Junior CHL Teams



Appendix D
Methodology 2
Virtual Sport Client ID

Methodology 2—Virtual Sports Client ID

SRAD's virtual sports are a product that gives players a chance to watch and bet on a digitally recreated sequence from historic matches whose outcome is based on a randomly generated number (RNG). It is like a custom slot machine featuring star players engaged in competitive matches in various sports.

SRAD issues a unique numerical Client IDs and virtual client alias to each customer which appears in its code. Incrementing each numerical client ID caused SRAD servers to return results with client data and client aliases for each positive match for an active virtual sports customer. A negative condition returns an error message. This interaction with the SRAD servers enabled us to assess whether an operator was an active customer, and not simply an operator scraping or pirating from SRAD.

We found the clientid value was synonymous with the "s4_id" (a 3 or 4 digit number, example in red below) in SRAD virtual sports layout URL and did not match the 32-character widgertloader client ID used by the same client. In other words, the virtual sport client ID or s4_ID it was a different and distinct client ID.

Additionally, we observed that the virtual sports domain was (https://config-vs001.akamaized.net) was indeed another SRAD related domain.

https://config-vs001.akamaized.net/rgsplus?client_id=3111&language=en&platform=desktop&product=vbl&sport=vbl [config-vs001.akamaized.net

Most important, we discovered that by submitting alternative numerical IDs to SRAD's servers we would receive responses indicating whether a number was or was not an active client. An active client ID number would return data including a client alias which was often recognizable as a brand name. The data would include a host of values including the items in the table below:

client_alias	id	s4_id	s5_alias
188betvbl	988	3111	188betvirtuals
leongaming	6464	782	leongamingvirtuals
sportybet	9222	455	sportybet
fonbetvbl	78	2007	fonbetvirtuals

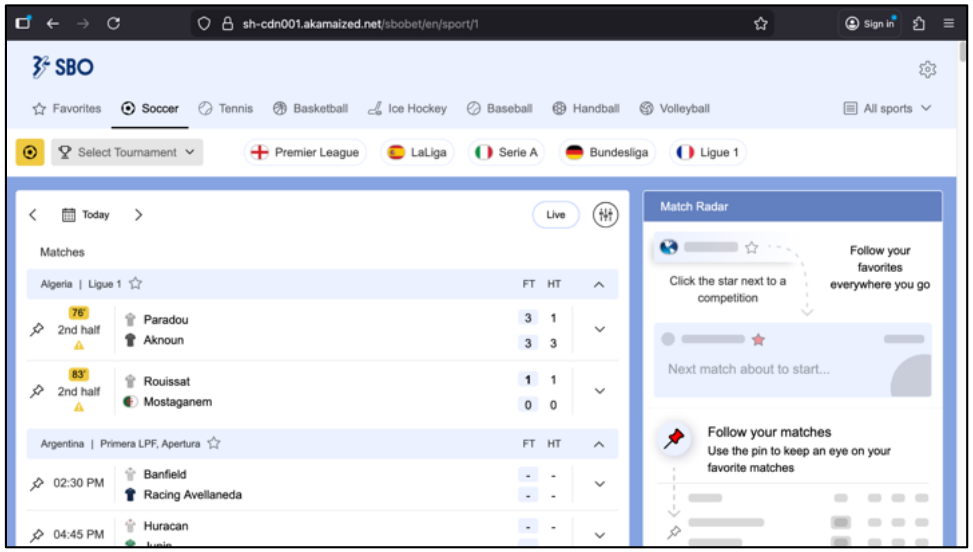
Appendix D - Methodology 2:

Virtual IDs Reside Within SRAD's Integration Ecosystem Within a Web of Client IDs

The use of the *client_alias* is powerful, as there are various data collection endpoints that can be used to affirmatively, and demonstrably show relationships with SRAD clients. For example, the SBOBet example is telling: the *sbobet[.]com* website has clear indicators of a relationship with SRAD, and more strikingly, SRAD hosts a web interface with *sbobet* in the URL that also contains the SBOBet logo visually evident on the screen (Figure below). Seemingly there are no rebuttals from SRAD that can explain this - no web scraping bot or illicit operator is in control of SRAD infrastructure.

Only SRAD is in control of SRAD infrastructure.

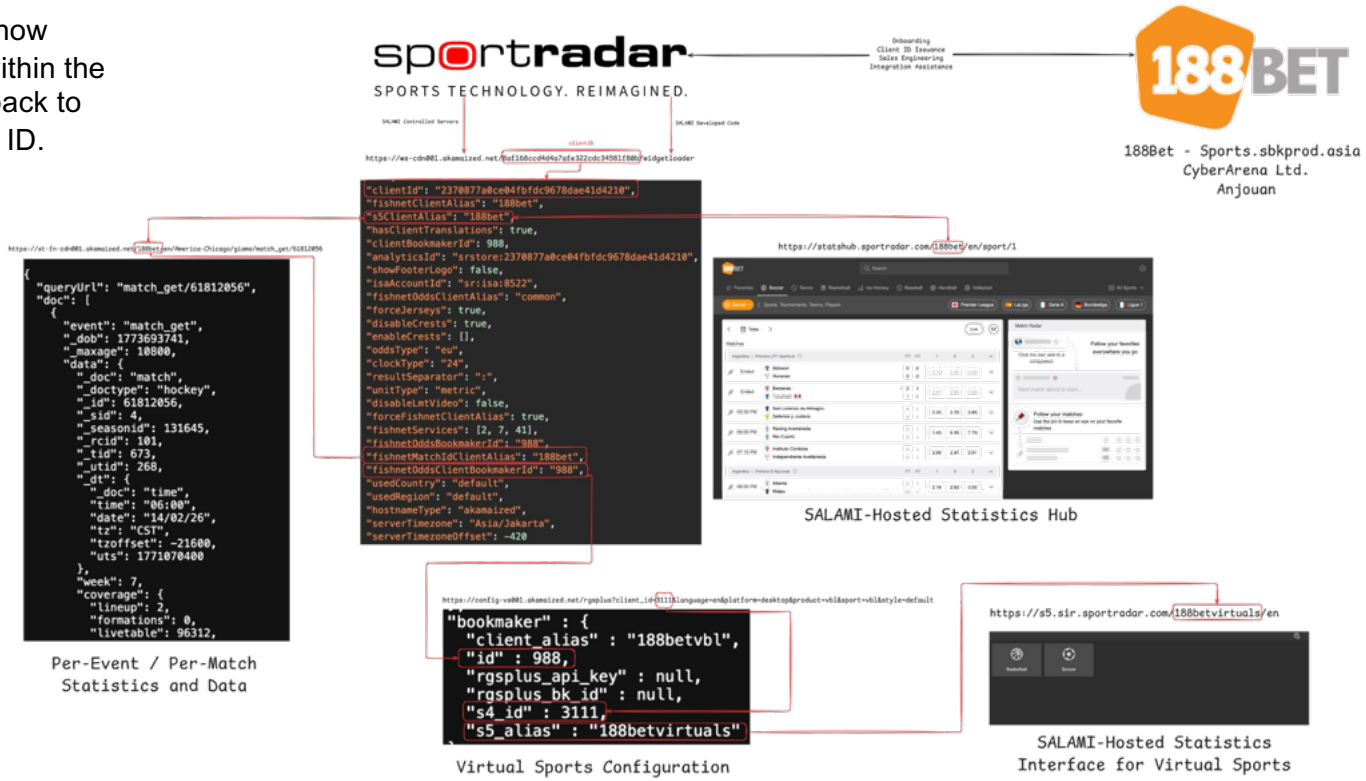
Below: SBOBet web interface hosted on SRAD infrastructure, with SBOBet branding.



Appendix D - Methodology 2:

Virtual IDs Reside Within SRAD's Integration Ecosystem Within a Web of Client IDs

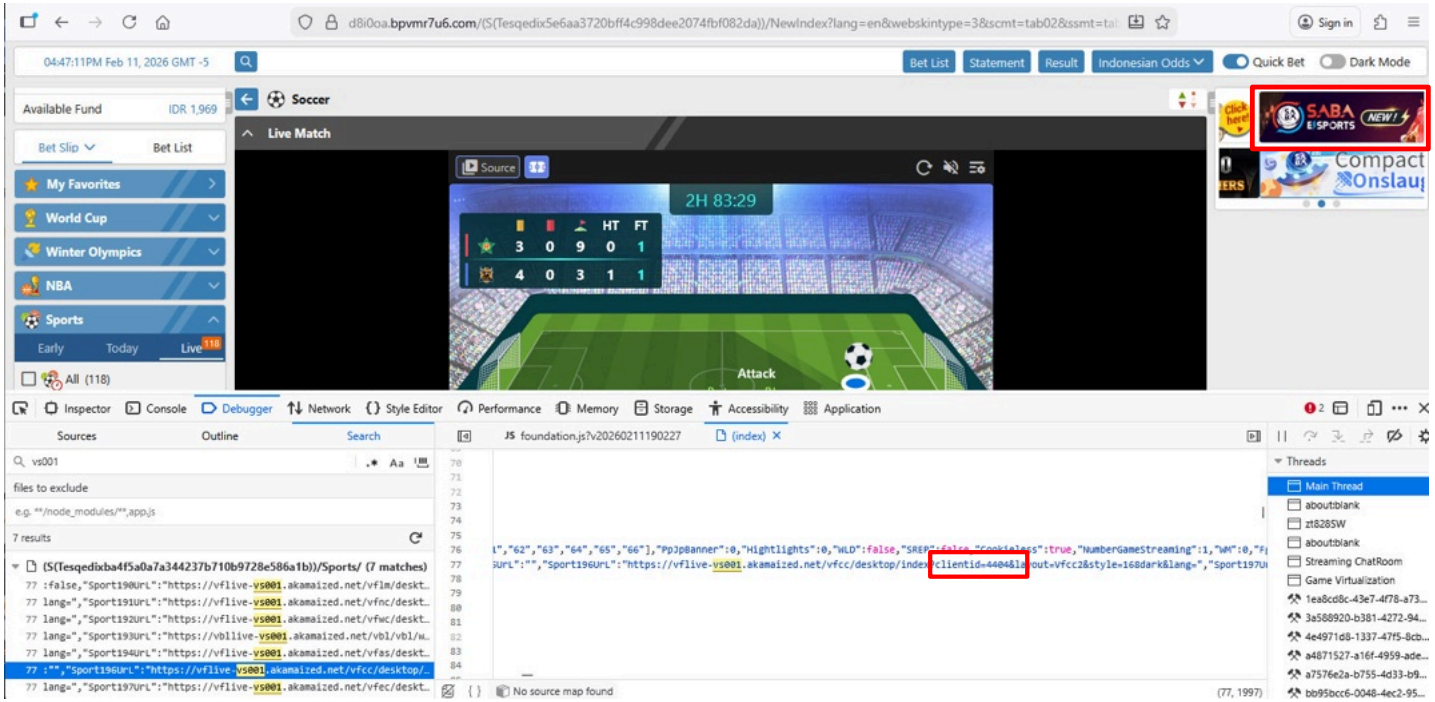
The diagram on the right illustrates how SRAD's virtual sports client ID fits within the ecosystem of integrations and ties back to the 32-character widgetloader client ID.



Appendix E
Methodology 2
Examples of Virtual Sport Client ID

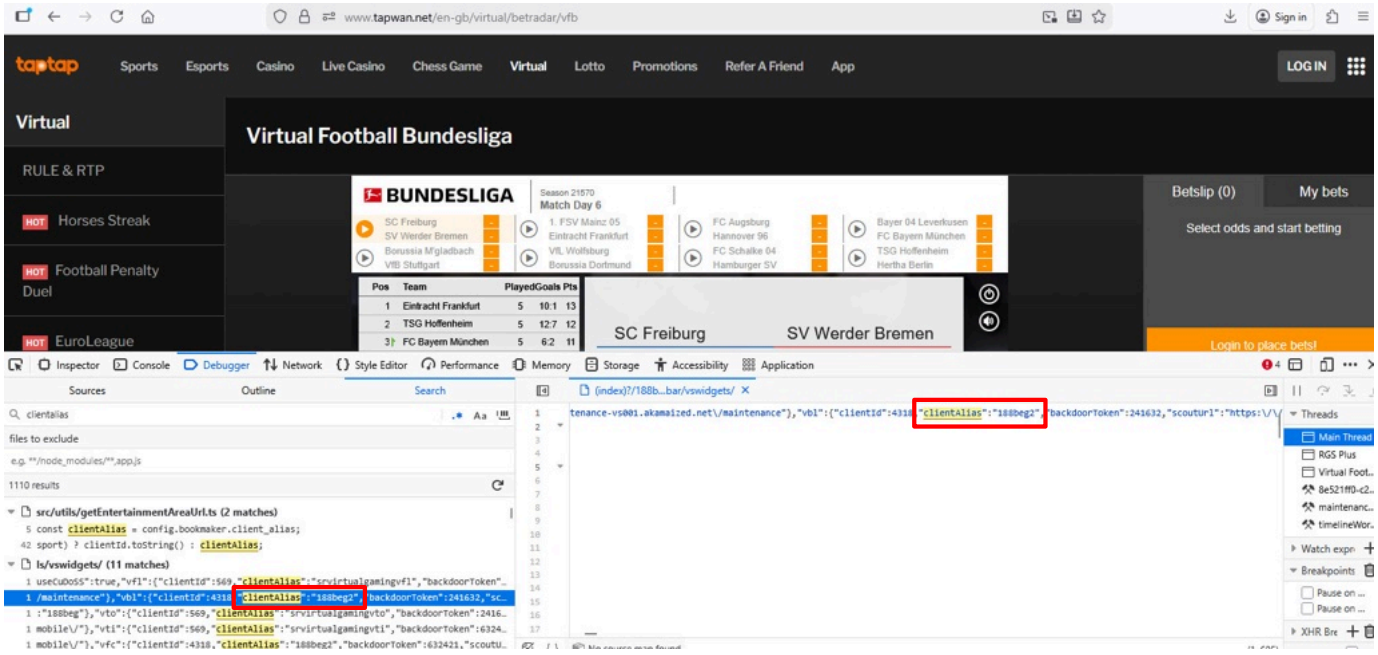
Appendix E: Examples of Virtual Game clientID

Bk8-SABA-02-11-2026-BetRadar-CDN



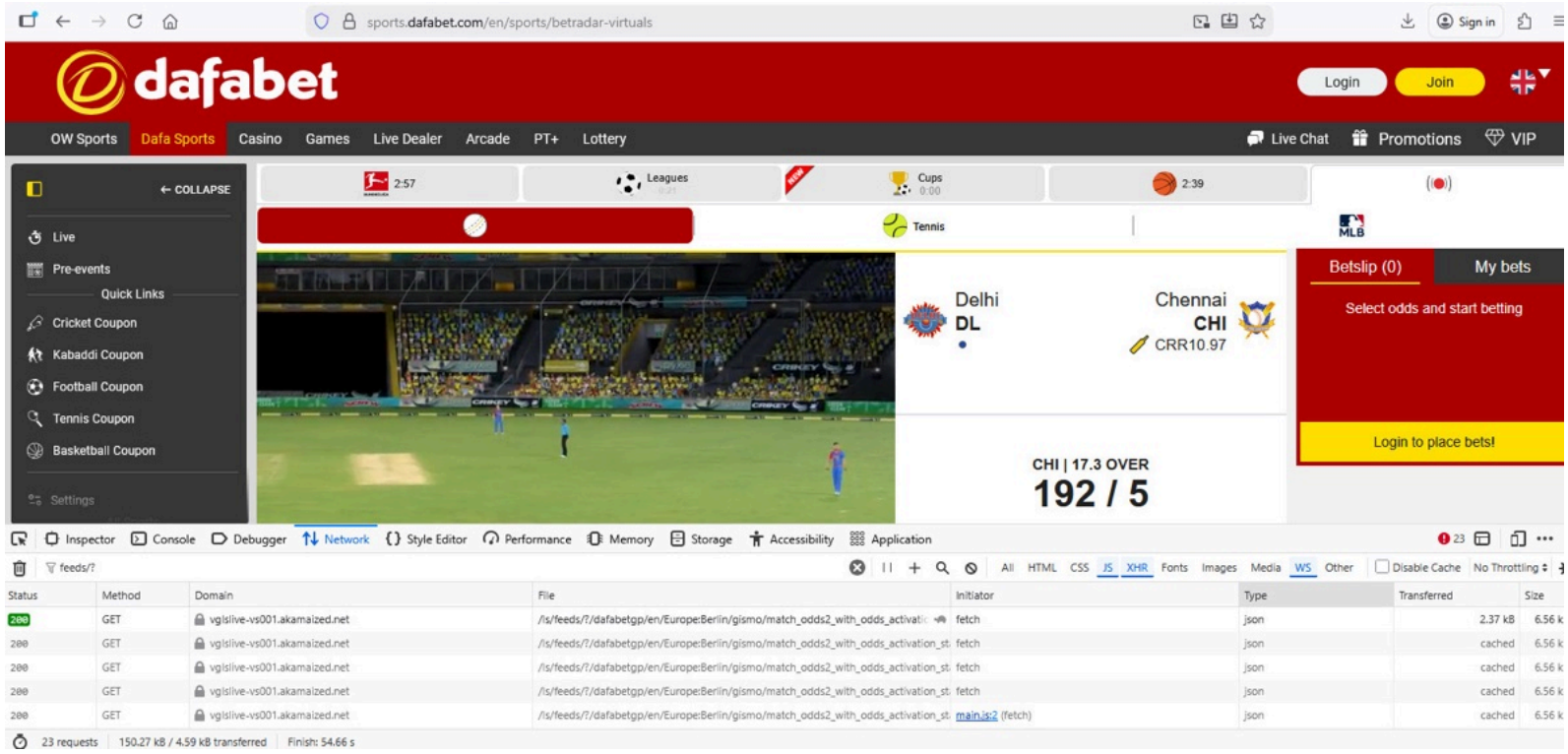
Appendix E: Examples of Virtual Game clientID

188bet / Taptap - virtual sport client ID: 188beg2



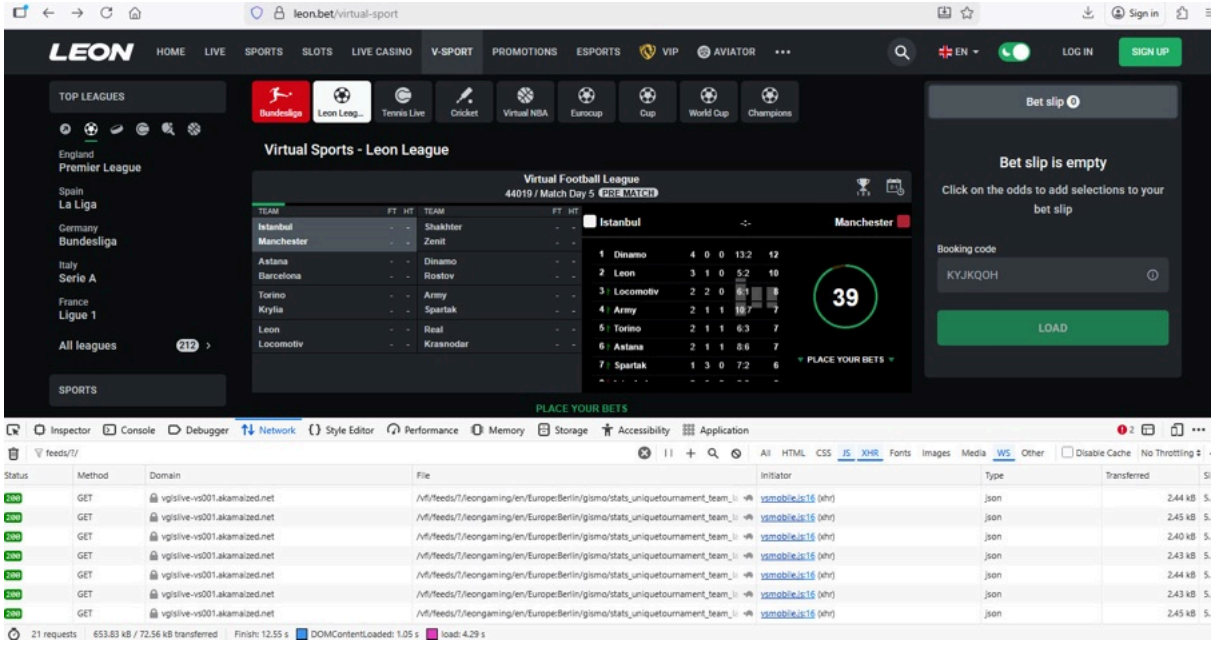
Appendix E: Examples of Virtual Game clientID

Dafabet.com – dafabetgp



Appendix E: Examples of Virtual Game clientID

Leonbet – leongaming



Appendix E: Examples of Virtual Game clientID

SABA – virtual sports client ID: mi001a168darkcdn