# **Dota 2 API Integration – STRIDE Threat Model**

**API**: https://api.opendota.com/api

## **OpenDota API Response Formats**

### **🔹 Player Profile**

GET /players/{account\_id}  
 Example: GET /players/86745912

json

{  
 "profile": {  
 "account\_id": 86745912,  
 "personaname": "Arteezy",  
 "name": "Artour Babaev",  
 "avatarfull": "https://...",  
 "last\_login": "2024-12-05T17:00:00.000Z",  
 "rank\_tier": 80  
 },  
 "mmr\_estimate": {  
 "estimate": 7200  
 }  
}

### **🔹 Recent Matches**

GET /players/{account\_id}/recentMatches

json

[  
 {  
 "match\_id": 7452345612,  
 "player\_slot": 0,  
 "radiant\_win": true,  
 "duration": 2220,  
 "hero\_id": 1,  
 "kills": 8,  
 "deaths": 3,  
 "assists": 14,  
 "start\_time": 1714598311  
 },  
 ...  
]

### **🔹 Win/Loss Record**

GET /players/{account\_id}/wl

json

{  
 "win": 920,  
 "lose": 870  
}

### **🔹 Hero Stats**

GET /players/{account\_id}/heroes

json

[  
 {  
 "hero\_id": 1,  
 "last\_played": 1713985034,  
 "games": 122,  
 "win": 68,  
 "with\_games": 50,  
 "with\_win": 30  
 },  
 ...  
]

### **🔹 Search Player**

GET /search?q={name}  
 Example: GET /search?q=arteezy

json

[  
 {  
 "account\_id": 86745912,  
 "personaname": "Arteezy",  
 "avatarfull": "https://..."  
 },  
 ...  
]

## **🧠**

## **🔐 1. STRIDE Threat Model**

|  |  |  |  |
| --- | --- | --- | --- |
| **STRIDE Category** | **Threat Description** | **Risk** | **Mitigation** |
| **S – Spoofing** | A malicious actor pretends to be a legitimate user or service | **Medium** | - Validate OpenDota player ID responses- Use Firebase Auth or similar for app user authentication- Use HTTPS to prevent interception |
| **T – Tampering** | API responses may be altered in transit or within the app | **High** | - Use HTTPS only (OpenDota supports HTTPS)- Validate data formats before using- Implement response integrity checks |
| **R – Repudiation** | Lack of logging or non-repudiation for user actions | **Low** | - Keep client-side logs for user actions (e.g., viewed profiles)- Display user consent for critical actions |
| **I – Information Disclosure** | Sensitive data (match history, win/loss ratios) exposed without control | **Medium** | - Use OpenDota’s public scope only- Inform users about public nature of data- Avoid caching sensitive data locally |
| **D – Denial of Service (DoS)** | App may overwhelm OpenDota API with requests | **High** | - Add rate-limiting client-side- Debounce search queries- Cache responses when possible |
| **E – Elevation of Privilege** | User performs unauthorized actions (e.g., admin-only data access) | **Low** | - OpenDota has no auth tier in public scope- App should only consume allowed endpoints- Validate endpoint responses and app routing |

## **🛠️ 3. Secure Design Considerations**

### **🔒 Authentication & Session**

* **Use Firebase or Google Sign-In** for app authentication
* **Store only non-sensitive OpenDota data** (e.g., profile names, match stats)

### **Network Communication**

* Use **HTTPS only** for all requests
* Avoid using any third-party proxies that may expose API traffic

### **Data Storage**

* Use encrypted SharedPreferences or Room DB if caching is needed
* Clear sensitive data on logout or timeout