# Stride Threat Model – OSRS Hiscores API

|  |  |  |  |
| --- | --- | --- | --- |
| Threat Type | Description | Example in Context | Mitigation Strategy |
| Spoofing | Attacker pretends to be a legitimate user or server. | A malicious app impersonates a legitimate player and scrapes data under false identity. | Use HTTPS to ensure server authenticity. Validate usernames through client-side input sanitation (e.g., alphanumeric-only). |
| Tampering | Unauthorized modification of data. | A man-in-the-middle modifies the JSON response to inject fake stats. | Use HTTPS to prevent MITM attacks. Validate all data on receipt. Consider checksums/hashes if storing locally. |
| Repudiation | Lack of accountability for actions. | A user denies querying or modifying local stat data. | Implement logging if storing or syncing data. Consider digital signatures on any user-modifiable data files. |
| Information Disclosure | Exposure of sensitive data. | While the API itself is public, cached player data may include unintended leakages when shared. | Ensure cached/stored data is clearly scoped to public profiles. If syncing, encrypt local storage. |
| Denial of Service | Service becomes unavailable. | Spamming the endpoint with automated queries (e.g., for thousands of users per second). | Implement rate limiting in your app. Cache responses where possible. Respect Jagex’s rate policies. |
| Elevation of Privilege | Unauthorized actions beyond intended scope. | A user modifies their local cache to show unrealistic stats or gains access to editing other players’ summaries. | Use role-based access control (RBAC) in any backend component. Never trust client-modified data. |

# Additional Notes

* The OSRS Hiscores API is **read-only** and **public**, which reduces risk around authorization, but increases risk of **abuse**, **caching concerns**, and **integrity of representation** in your app.
* Because it’s public, there’s **no authentication**, so focus should be on ensuring that your **application layer handles spoofing, DoS, and caching integrity** carefully.
* **Storing** or **sync** these stats with user profiles, all STRIDE categories become more relevant due to local trust boundaries.

# Example Request and Response

## Get overall stats

<https://secure.runescape.com/m=hiscore_oldschool/index_lite.json?player={{username>}}

## JSON Response

[Pastebin with JSON response](https://pastebin.com/zqcZ2eNY)