# SiegeAPI Integration & Threat Modeling

# 1. API Overview

Library: siegeapi (https://github.com/CNDRD/siegeapi)  
Purpose: Unofficial Python wrapper for Ubisoft's Rainbow Six Siege player stats API  
Authentication Required: Yes – requires valid Ubisoft credentials (email + password)  
Library Installation:  
 ***pip install siegeapi***  
  
Example Initialization:  
 ***from siegeapi import Auth  
 auth = Auth("email@example.com", "password")  
 player = await auth.get\_player(name="ExamplePlayer")***  
  
- Uses asynchronous I/O with asyncio  
- Wraps multiple API endpoints internally (e.g., for stats, ranking, XP, persona)  
- Works without needing to manually handle JSON

# 2. Request Parameters

|  |  |  |
| --- | --- | --- |
| Parameter | Location | Description |
| Ubisoft Email | auth | Required for login |
| Ubisoft Pass | auth | Required for login |
| name | method | The in-game player name to retrieve details |

# 3. Example JSON Response

While the API wraps JSON responses into Python objects, a typical parsed response includes:

***{  
 "name": "ExamplePlayer",  
 "profile\_pic\_url": "https://...",  
 "level": 143,  
 "rank": "Gold II",  
 "xp": 432500,  
 "ranked\_points": 2950,  
 ...  
}***

# 4. Error Handling

|  |  |  |
| --- | --- | --- |
| Status Code | Reason | Library Behavior |
| 401 | Unauthorized (bad credentials) | Raises login/auth exception |
| 404 | Player not found | Raises lookup exception |
| 429 | Rate limit exceeded | Raises throttling/backoff error |
| 400/422 | Invalid parameters | Raises validation-related exception |

# 5. STRIDE Threat Model

|  |  |  |
| --- | --- | --- |
| Threat Type | Risk Description | Recommended Mitigations |
| Spoofing | Fake account credentials used to scrape data | Enforce secure login, use secure credential storage (e.g., keyring) |
| Tampering | Malformed player names or path injections | Sanitize user input, validate before querying |
| Repudiation | Disputes over account data access | Log all API interactions with timestamps |
| Information Leak | Exposure of player or login info in logs | Mask credentials in logs, avoid storing raw data |
| Denial of Service | Excessive login or data requests | Implement retry/backoff, use a caching layer where possible |
| Privilege Elevation | Querying other users via credential stuffing | Limit login attempts, monitor for unusual behavior |

# 6. Additional Notes

- Maintained by CNDRD (https://github.com/CNDRD)   
- Documentation available at: https://cndrd.github.io/siegeapi  
- Suitable for bots, stat dashboards, or personal tools – not intended for public commercial use due to Ubisoft’s TOS and potential rate limits