Application & Data Security

VocabVersus

Thomas van der Molen

|  |  |
| --- | --- |
| **Project Information** | |
| Project members | Thomas van der Molen |
| Project Name | VocabVersus |

Table of Contents

[Topic 3](#_Toc137390402)

[Goal 3](#_Toc137390403)

[Summary 3](#_Toc137390404)

[Questions 4](#_Toc137390405)

[Server Communication 5](#_Toc137390406)

[Server-Side Authority 6](#_Toc137390407)

[Messaging Security 6](#_Toc137390408)

[Server Load Handling 8](#_Toc137390409)

[Game Instances 8](#_Toc137390410)

[Word Sets 8](#_Toc137390411)

[User Authentication 9](#_Toc137390412)

[Identification 9](#_Toc137390413)

[Cookie Law 9](#_Toc137390414)

[Security 9](#_Toc137390415)

[Word Usage Tracking 11](#_Toc137390416)

[Ethics 11](#_Toc137390417)

[Limitations 11](#_Toc137390418)

[Future Considerations 11](#_Toc137390419)

# Topic

This research document will cover several design patterns, best practices and considerations that have to be made within the VocabVersus context to create a secure and data aware application.

The research is split up into two topics: The application security and data security, within these topics several large concerns as discovered from real world examples [similar to the VocabVersus context](../Project%20Plan.docx#Target_Audience), [OWASP security list](https://owasp.org/) and [GDPR regulations](https://gdpr-info.eu/) will be explored.

# Goal

The goal of this research will be to explore and find the best solutions for the most common and highest severity concerns pertaining to security and data handling.

These solutions will be explored with the goal of allowing for direct implementation into the VocabVersus web application.

# Summary

In this document several possible high-risk security and data privacy concerns have been covered.

Possible attack vectors based on the RPC SignalR communication layer have been explored, alongside possible pre-emptive measures to ensure application integrity such as utilizing full server-side authority.

Other possible application integrity attacks targeting the load handling of the application as well as data storage efficiency has been covered with common mitigation techniques to discourage these kinds of attacks by utilizing API limiting or throttling, as well as creating a sensible maintenance routine to also increase scalability.

Due to the odd authentication requirements of the application, special care had to be taken so that authentication can be done precisely, securely, and lawfully without any user friction. To achieve this goal; best practices in authentication have been explored alongside ethical requirements put in place by the ePrivacy Act and GDPR laws.

Lastly, the ethics of storing and processing user data to improve game balancing has been investigated to find out the limitations and pitfalls to look out for when handling this kind of data, with the goal of finding a way to still collect this data while following all online privacy laws ensuring compliance within any (reasonable) legislations in case of a sudden rise of interest in the application.

# Questions

This research will have one main question, whose answer can be constructed based on several sub questions, these questions are primarily within the [DOT Framework’s](https://ictresearchmethods.nl/The_DOT_Framework) Library, Field, Showroom and Workshop strategies.

|  |  |
| --- | --- |
| **Main Question** | |
| What are common or severe[[1]](#footnote-1) security risks and solutions for the VocabVersus web application? | |
| **Sub Questions** | **Research Methods** |
| How can the communication between the interface and game engine be exploited? | [Document analysis](https://ictresearchmethods.nl/Document_analysis)  [Prototyping](https://ictresearchmethods.nl/Prototyping)  [Literature Study](https://ictresearchmethods.nl/Literature_study) |
| What methods can be used by a bad actor to greatly dimmish the integrity of the VocabVersus application? | [Good & Bad Practices](https://ictresearchmethods.nl/Best_good_and_bad_practices)  [Literature Study](https://ictresearchmethods.nl/Literature_study)  [Interview](https://ictresearchmethods.nl/Interview) |
| What is the best way of authenticating players without requiring a login? | [Interview](https://ictresearchmethods.nl/Interview)  [Literature Study](https://ictresearchmethods.nl/Literature_study) |
| What common methods could be used by a player to gain an unfair advantage? | [Observation](https://ictresearchmethods.nl/Observation)  [Product Review](https://ictresearchmethods.nl/Product_review)  [Useability Testing](https://ictresearchmethods.nl/Usability_testing) |
| What is the most optimal method of storing data used during a game? | [Design Pattern Research](https://ictresearchmethods.nl/Design_pattern_research)  [Good & Bad Practices](https://ictresearchmethods.nl/Best_good_and_bad_practices) |
| How can the game activity of players be ethically stored for use in game balancing? | [Ethical Check](https://ictresearchmethods.nl/Ethical_check)  [Literature Study](https://ictresearchmethods.nl/Literature_study) |

# Server Communication

To ensure low latency communication between the game’s interface (front-end) and server (game-engine); [SignalR](https://learn.microsoft.com/en-us/aspnet/signalr/overview/getting-started/introduction-to-signalr) has been used as explored in the [Project Architecture](../Project%20Architecture.docx).

SignalR opens a [websocket](https://en.wikipedia.org/wiki/WebSocket#:~:text=WebSocket%20is%20a%20computer%20communications,protocol%20is%20known%20as%20WebSockets.) connection between the client and server and allows for remote procedure calls to be made between them (allowing the client to call for code execution on the server and vice versa).



Figure 1 RPC server-client



Figure 2 RPC client-server

I have previously explored the benefits and functional explanation of the SignalR-websocket messaging implementation as part of a [low latency client-server communication](https://github.com/Thomas-Molen/WebAdventure/blob/main/Documentation/Low-LatencyClient-ServerCommunication.pdf) research.

In short: When the client wants to connect to a server via SignalR (websocket), first a request to initialize the connection is made, which afterwards will create and keep open a websocket connection for bi-directional messaging.

A screenshot of a websocket protocol

Description automatically generated with low confidence

Figure 3 web socket initial request and messaging

## Server-Side Authority

There are [many ways](https://unity.com/how-to/intro-to-network-server-models) multiplayer games can handle networking, VocabVersus uses a dedicated server model, where all clients connect to a single host server separate from any client, which functions as the authoritative single source of truth.

When a player commits any action in the game, it is sent to the game-engine server which handles the given action and will send the necessary response information to all relevant clients, who will only accept game state changes from the single server source.

## Messaging Security

Due to [all game logic being handled by the server](#_Server-side_authority), the client in the client-server connection effectively works exclusively as an interface for the player to view the current game state and perform actions.

Due to the client side only showing a representation of the game-engine’s current game state, there is very little reason to attack communication between the server and client as this communication only contains shallow game state information and does not affect the actual game state stored by the server.

On the other hand, communication between the client to the server can create vulnerabilities, especially due to the RPC protocol used (allowing the client to execute logic on the server). To ensure this communication is not abused, the server will have to validate any requests made by a client to ensure game integrity.

An example of a possible attack made through a client-server RPC message, could be at the username input, here the front-end limits any input to 20 characters.

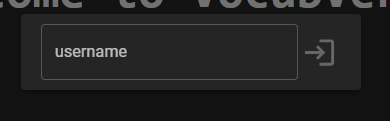


Figure 4 front-end username input field

When a username is committed by the client, a message is sent over the websocket as shown below.



Figure 5 username Thomas was committed

If this message by the client was blindly accepted by the server, a bad actor could intercept this message and edit the username to whatever they want, such as a username exceeding the 20-character length only ensured by the front-end’s design. A screen shot of a computer

Description automatically generated with medium confidence

Figure 6 Minor username edit already breaking front-end for all players

Even a simple edit of the username could have large effects, such as including JavaScript code execution or data bombing clients via massive message sizes (often using characters that have very bad optimization), as has happened in games such as in New World with players blocking the client-server communication [causing players to become invincible](https://www.gamerevolution.com/guides/695037-new-world-how-to-do-invincibility-glitch-bug-god-mode#:~:text=Follow%20these%20steps%20in%20order,freezes%2C%20you%20have%20become%20invincible), or in Minecraft where players would overflow data in certain game areas causing any player entering that area to get [unknowingly data bombed by the server and crashing](https://github.com/PaperMC/Paper/issues/3507).

# Server Load Handling

While there are many ways of attack an online service such as via [DDoS](https://www.fortinet.com/resources/cyberglossary/ddos-attack#:~:text=DDoS%20Attack%20Meaning,connected%20online%20services%20and%20sites.) attacks, I will be focusing on a specific weak point of the application that both effects the performance and data storage of the VocabVersus web application.

The VocabVersus game allows anyone to create their own word sets and start a game, this is done to both increase the possible player base of the game and increase general enjoyment by allowing people to play the game in their own way.

## Game Instances

Whenever a game is created for a group of people to play, short lived data will have to be maintained to keep track of the current game state. While this data has been extremely optimized to lower the amount of overhead caused by a single game instance, players are able to create as many games as they want, which can become a large problem when the player base increases or when bad actors try to exploit this fact.

While a couple megabytes per game instance of memory doesn’t sound that bad, especially in a cloud-oriented architecture where the game-engine can have [its resources scaled to demand](https://learn.microsoft.com/en-us/azure/well-architected/scalability/design-scale). When bad actors come along and try to create thousands of games, not only could this slow down the server as significant resources will have to be spent maintaining all instances, large cost increases from possibly miss configured cloud scaling systems could [cause financial damages as well](https://www.bitdefender.com/blog/businessinsights/attacks-cloud-services/).

A common method of avoiding these kinds of attacks is by [rate limiting](https://www.imperva.com/learn/application-security/rate-limiting/#:~:text=Rate%20limiting%20is%20a%20technique,Denial%20of%20Service%20(DoS).) these requests, often done on a per client instance however, with [large bot-net attacks becoming more prevalent](https://azure.microsoft.com/en-us/blog/azure-ddos-protection-2021-q3-and-q4-ddos-attack-trends/) [rate throttling](https://nordicapis.com/api-rate-limiting-vs-api-throttling-how-are-they-different/) for the whole service should be considered as well.

## Word Sets

The VocabVersus games stores word sets that can be used during the game via the [Apache Lucene search engine library](https://lucene.apache.org/). While a full research into choosing the [best word evaluator solution](Word%20Evaluator%20Research.docx) has been done for VocabVersus with a major consideration being the wordset storage, this service can still be targeted as a possible data security risk.

As stated in the [introduction](#_Server_Load_Handling), players can create their own word sets, these word sets could theoretically have no limit in size. A bad actor could just as with game instances could create many large word sets, increasing the required storage size of the word evaluator drastically (performance should however not be affected as it is specially designed to be scalable).

To mitigate possible attacks to the word evaluator word set storage system, just as with the [game instances](#GameInstanceRateLimit): A rate limit or throttle should be introduced for the creation of word sets. Furthermore, a sensible limit to the amount of words and length of individual words could be introduced to reduce the possible damage of a single word set insert and lastly, to keep storage size down over time, a regular clean-up of stale (not used in a long time) word sets could be added to remove these word sets and make room for new ones.

# User Authentication

*How does one authenticate a user who does not want to be authenticated.*

To make sharing and playing the VocabVersus game as frictionless as possible, it was decided to not have any kind of login/account system for VocabVersus as looked into early on in the [project plan](../Project%20Plan.docx) and [architecture](../Project%20Architecture.docx).

## Identification

While no identification from users should be required as a user can be identified through their connection ID used by the [SignalR communication layer](#_Server_Communication), this system will break when a player’s connection ID changes, such as when their internet shortly disconnects or they close the game and re-open it later.

To be able to track a player independently of their connection at least for the length of a single game, a unique game instance identifier for that player can be used which is given out when a player first connects to a game and is checked when reconnecting.



Figure 7 Example of a game instance user identifier

## Cookie Law

Due to these player tracking cookies falling under necessary cookies, that are required for the VocabVersus application to function properly it easily falls under the ethical use of cookies without requiring any extra consent by the user as has been required for most tracking cookies by the [ePrivacy Directive](https://edps.europa.eu/data-protection/our-work/subjects/eprivacy-directive_en) and covered under the [GDPR cookies article](https://gdpr.eu/cookies/) (it does however mention, that it should be possible for the user to find a clear explanation of the usage of the cookie).

While using an identifier stored by the client is a very effective way of keeping track of users and is actually similar to how [common account authentication systems keep users logged in](https://deryatanriverdi.medium.com/how-to-user-localstorage-to-keep-user-logged-in-and-out-8fd1e6e46712), these tokens are often targeted by client-side script attacks such as [cross-site scripting attacks](https://www.veracode.com/security/xss#:~:text=Cross%2Dsite%20scripting%20attacks%2C%20also,JavaScript%2C%20to%20an%20end%20user.) that try and steal these tokens to be able to impersonate the person. Attacks targeting these kinds of tokens have gotten more attention in the last couple years as it has been used to [steal many high profile YouTube creator accounts](https://www.wired.com/story/youtube-bitcoin-scam-account-hijacking-google-phishing/).

## Security

While the short-lived accounts used for VocabVersus’ anonymous authentication are far less likely to be targeted by more sophisticated attacks, such as the one mentioned above (due to there not being much to gain from stealing a one-time use account). With these kinds of authentication token attacks often ranking high in the [OWASP top 10](https://owasp.org/www-project-top-ten/), along with the fact that there are some fairly straight forward methods that can be employed to greatly limits these attacks, it still seems worthwhile to explore the most common mitigations to make it futile to even target VocabVersus in this manner.

One of the common recommendations for reducing the possible damage of authentication token attacks, is by [limiting the validity of these tokens](https://stackoverflow.com/questions/7030694/why-do-access-tokens-expire#:~:text=It%20is%20essentially%20a%20security,longer%20than%20the%20access%20token.) which is already built in to the VocabVersus tokens as they are only used for the duration of a single game instance (often only a couple minutes at maximum). Secondly, how the tokens are stored can make a big difference as storing the token for example in local storage is often a bad idea as this data is directly accessible via client-side scripting. While tokens are often stored in cookies via the HTTP headers, this is open for the same kinds of attacks without any extra configuration. When storing authentication tokens in a cookie it is highly recommended to store it as an [HTTP Only cookie](https://www.cookiepro.com/knowledge/httponly-cookie/#:~:text=Using%20the%20HttpOnly%20tag%20when,making%20these%20cookies%20more%20secure.&text=If%20the%20HttpOnly%20flag%20is,through%20the%20client%2Dside%20script.) as these are not accessible via client-side scripts (this is also the most common recommendation by the [OWASP community itself](https://owasp.org/www-community/attacks/xss/)).

# Word Usage Tracking

To keep the VocabVersus game fun and competitive, points given per round should not only be determined by factors such as length of the word and time to answer but could leverage the ability to track user submitted words and encourage the use of uncommon words.

## Ethics

While tracking user input is functionally possible, there are other more ethical concerns to consider when working with what could be considered private user data without their explicit consent. If not only for the ethical integrity of the VocabVersus game, [GDPR data privacy and security laws](https://gdpr.eu/what-is-gdpr/) enforce strict rules to any parties processing personal data of EU citizens with [massive financial fines](https://gdpr-info.eu/issues/fines-penalties/#:~:text=83(4)%20GDPR%20sets%20forth,to%20that%20used%20in%20Art.) attached to not following these regulations.

### Limitations

First of all, to be able to collect/process this user data VocabVersus will have to [clearly state that this data is being collected](https://gdpr-info.eu/issues/right-to-be-informed/) to the users and needs to give the user the ability to [opt-out](https://gdpr-info.eu/issues/consent/), [request](https://gdpr-info.eu/issues/right-of-access/) or [delete](https://gdpr-info.eu/issues/right-to-be-forgotten/) any data related to the individual (among other less common [privacy requirements](https://gdpr-info.eu/issues/privacy-by-design/)).

Due to the general [anonymous nature of VocabVersus players](#_User_Authentication), it becomes fairly straight forward to comply to a lot of the requirements by never linking any data to an individual person ([as long as it can be clearly proven that this is the case](https://gdpr-info.eu/issues/privacy-impact-assessment/)). However, the initial plan was to share the data collected and used for balancing with the general public, this however is not possible under regulations relating to [specific limitations on distributing processed personal data](https://datacompliant.co.uk/data-sharing-under-gdpr-what-you-need-to-know/) even if anonymously collected without [jumped through a lot of extra loops](https://gdpr-info.eu/chapter-5/).

### Future Considerations

Lastly, it is highly recommended that the data collected is made as [minimal as possible](https://gdpr.eu/what-is-gdpr/), for example: While it would be nice to track word usage based on time and location, to possibly create a more advanced algorithm that is able to take into account sudden rises in word usage or different mannerisms of geolocations, it will be harder to [prove this as necessary](https://gdpr-info.eu/art-6-gdpr/) for the game in it’s current scope and (while it can be accounted for in the architecture for the future) might be better to not be implemented in case of a sudden increase of interest in the game, which will most likely also increase the interest by regulatory parties as happened with Roblox after its sudden explosion in popularity with [many investigations being conducted](https://www.lexology.com/library/detail.aspx?g=046cd811-7a91-4a92-9e9e-6cbbf178bca4) which are [not always conducted fairly](https://rtp.fedsoc.org/paper/government-regulation-the-good-the-bad-the-ugly/).

1. referring to concerns that can have a large negative effect on the player experience or application integrity. [↑](#footnote-ref-1)